

eTourism Platform Survey: Towards a standardized data schema for points of interest

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Business Informatics

by

Maria Kroich

Registration Number 0626579

to the Institute for Software Technology and Interactive Systems
at the Vienna University of Technology

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Hannes Werthner

Assistance: Dr. Rainer Schuster

Vienna, 22.01.2012

Erklärung zur Verfassung der Arbeit

Maria Kroich
Apostelgasse 2-14/5/54, 1030 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)

(Unterschrift Verfasserin)

Abstract

The tourism industry was one of the first sectors utilizing ICT. A multitude of systems, standards and services have been developed. The use and the offered possibilities of the web enable continuous progress in the eTourism sector. eTourism defines the application of e-commerce solutions in the field of tourism. Since there are a big variety of stakeholders, the overlaps and intersections amongst them are very important. In order to ensure interoperability between those business actors data standards are required.

The multitude of different online providers in the tourism industry also leads to a multitude of different goals, policies, standards, etc. Especially standards are highly relevant, as soon as it comes to an integration of content from other providers into the own applications. The survey considers and investigates point of interest (POI) as content. So far, no data standard regarding points of interest has enforced, even though some formats are prevalent.

The integration of external content can become complex, due to the missing data standard. Each data source has to be handled separately, which implies higher efforts in implementing networking processes. The missing standard is complicating the reusability, exchange and comparison of data in the tourism sector. A restraint of innovation may be the consequence.

The approach of the survey is based on the investigation of different online providers in the tourism industry in relation to point of interest data. Thereby, the available platforms are analyzed in regard to their APIs, data availability, trust and security issues, as well as their different policy guidelines for using their data as a third party service. The analytical results are summarized in a comparative matrix.

Based on the analysis of online providers and the evaluation of current dominant formats and protocols a new data standard for points of interest is developed - PXF (Point of Interest eXchange Format). The application of the new PXF format intends to simplify interoperability between different stakeholders. The applicability of the standard is verified via the prototypical implementation.

The PXF standard facilitates cooperation and interaction between different systems and business actors. Thereby the exchange, comparison and reusability of data become more suitable and can lead to new innovations.

Contents

| | | |
|----------|--|----------|
| 1 | Introduction | 1 |
| 1.1 | Problem Statement | 1 |
| 1.2 | Aim of the Work | 2 |
| 1.3 | State of the Art | 2 |
| 1.4 | Methodology | 3 |
| 1.5 | Structure of the Work | 4 |
| 2 | eTourism | 5 |
| 3 | Tourism Organizations and existing IT Standards | 9 |
| 3.1 | ACRISS | 10 |
| 3.2 | ANSI ASC X12 | 10 |
| 3.3 | CENT/TC 329 | 10 |
| 3.4 | DATEX | 10 |
| 3.5 | ECTAA | 10 |
| 3.6 | HEDNA | 11 |
| 3.7 | HITIS | 11 |
| 3.8 | IATA | 11 |
| 3.9 | IATAN | 11 |
| 3.10 | ICAO | 11 |
| 3.11 | IFITT | 12 |
| 3.12 | IHRA | 12 |
| 3.13 | JourneyWeb | 12 |
| 3.14 | KAREN | 12 |
| 3.15 | OTA | 12 |
| 3.16 | TIH | 12 |
| 3.17 | TIN | 13 |
| 3.18 | Transmodel | 13 |
| 3.19 | TransXchange | 13 |
| 3.20 | TRIDENT | 13 |
| 3.21 | TTI | 14 |
| 3.22 | TTRA | 14 |
| 3.23 | UIC | 14 |

| | | |
|----------|---|-----------|
| 3.24 | UN/LOCODE | 14 |
| 3.25 | UN/EDIFACTTT&L | 14 |
| 3.26 | USTOA | 15 |
| 3.27 | WATA | 15 |
| 3.28 | WTO | 15 |
| 3.29 | WTTC | 15 |
| 4 | Points of Interest | 17 |
| 4.1 | Introduction | 17 |
| 4.2 | Existing Formats and Standards | 19 |
| 4.2.1 | ASCII, CSV, TXT | 19 |
| 4.2.2 | BMP | 19 |
| 4.2.3 | GDB | 19 |
| 4.2.4 | GPX | 19 |
| 4.2.5 | KML | 21 |
| 4.2.6 | OV2 | 23 |
| 4.2.7 | WPT | 23 |
| 5 | eTourism Online Providers | 25 |
| 5.1 | Introduction | 25 |
| 5.2 | eTourism Online Provider Selection | 26 |
| 5.3 | Falk Marco Polo Interactive GmbH | 26 |
| 5.4 | Flickr | 29 |
| 5.5 | Foursquare | 33 |
| 5.6 | Garmin | 36 |
| 5.7 | Lonely Planet | 38 |
| 5.8 | Panoramio | 41 |
| 5.9 | Qype | 45 |
| 5.10 | TomTom | 50 |
| 5.11 | Twitter | 52 |
| 5.12 | ViaMichelin | 58 |
| 6 | eTourism Evaluation Results | 63 |
| 6.1 | Evaluation results | 63 |
| 6.1.1 | Comparative Matrix of eTourism evaluation results | 63 |
| 6.1.2 | Fact Sheets of eTourism Online Providers | 65 |
| 7 | Prototypical Implementation | 75 |
| 7.1 | Introduction | 75 |
| 7.2 | Purpose | 75 |
| 7.3 | Scope | 76 |
| 7.4 | Point of Interest eXchange Format - PXF | 76 |
| 7.4.1 | Underlying Formats | 76 |
| 7.4.2 | XML Schema Definition | 76 |

| | | |
|----------|---|------------|
| 7.5 | Prototype Architecture | 87 |
| 7.6 | Qype API Requirements | 89 |
| 7.6.1 | Main concepts | 89 |
| 7.6.2 | Authentication | 89 |
| 7.6.3 | Resource <i>Place</i> | 89 |
| 7.6.4 | Resource <i>Review</i> | 91 |
| 7.6.5 | Resource <i>Assets</i> | 91 |
| 7.6.6 | Resource <i>Category</i> | 92 |
| 7.6.7 | Resource <i>Tags</i> | 92 |
| 7.6.8 | Resource <i>Attributes</i> | 92 |
| 7.7 | Flickr API Requirements | 92 |
| 7.7.1 | Main concepts | 92 |
| 7.7.2 | Authentication | 92 |
| 7.7.3 | Resource Photo | 92 |
| 7.8 | Database Requirements | 95 |
| 7.8.1 | System evaluation | 95 |
| 7.8.2 | MongoDB | 96 |
| 7.8.3 | Database Specification | 97 |
| 7.9 | Implementation | 98 |
| 7.9.1 | Sequence Flow | 98 |
| 7.9.2 | API Request | 100 |
| 7.9.3 | Parsing | 100 |
| 7.9.4 | MongoDB Storage | 102 |
| 7.9.5 | PXF Output | 104 |
| 7.10 | Prototype Verification | 105 |
| 7.10.1 | POI 1: Karlskirche | 106 |
| 7.10.2 | POI 2: Happel Stadion | 108 |
| 7.10.3 | POI 3: Hotel Sacher | 110 |
| 7.10.4 | POI 4: Albertina | 112 |
| 7.10.5 | POI 5: Das Rochus | 114 |
| 7.10.6 | Summary | 115 |
| 8 | Summary | 117 |
| 8.1 | Results | 117 |
| 8.2 | Conclusion & Outlook | 118 |
| 9 | Glossar | 119 |
| A | GPX format | 121 |
| B | Qype Resource Representation Reference | 123 |
| B.1 | Resource <i>Place</i> | 123 |
| B.2 | Resource <i>Reviews</i> | 124 |
| B.3 | Resource <i>Checkins</i> | 124 |

| | | |
|----------|----------------------------------|------------|
| B.4 | Resource <i>Assets</i> | 125 |
| B.5 | Resource <i>Locators</i> | 125 |
| B.6 | Resource <i>Categories</i> | 125 |
| B.7 | Resource <i>Coupon</i> | 126 |
| C | ViaMichelin POI structure | 127 |
| D | Prototype Verification | 131 |
| D.1 | XML Karlskirche | 131 |
| D.2 | XML Happel Stadium | 133 |
| D.3 | XML Hotel Sacher | 135 |
| D.4 | XML Albertina | 137 |
| D.5 | XML Das Rochus | 139 |
| | Bibliography | 141 |

Introduction

The combination of increasing mobility and the enormous amount of information available is creating and innovating services. In the course of time people gradually turned more mobile and the use of the World Wide Web is familiar for most of them. At any time and place users want to be supplied with required information.

Regarding to the increasing mobility and the booming tourism industry users will be supplied with new services to increase sales figures, tourism inflow, performance of mobility applications etc. They want to receive in-time information for a certain location. Such a location can be defined as point of interest, which is an interesting place or location. Point of interest data can include further additional information and enriches diverse applications.

The use of points of interest have its origin in mobile navigation. Starting in 2002 mobile navigation devices provided location based information for users. This has been achieved on the one hand with the increasing performance of portable devices, as well as higher data availability. The topic of this master thesis is about investigating different online providers in the tourism industry in relation to point of interest data. Thereby, the available platforms should be analyzed in regard to their APIs and interfaces, their data availability, trust and security issues, as well as their different policy guidelines for using their data as a third party service. The survey considers technical aspects dealing with the requirements for interacting with the different platforms, as well as analytical aspects in regard to the platforms' data availability. Another part of this thesis is the evaluation of existing tourism standards as well as existing standards for point of interest data.

1.1 Problem Statement

Currently there a lot of different online providers in the tourism industry. Most of them are focused on own goals, policies, countries, unions and agencies, what entails, that no data standard regarding to point of interest data enforced until now. The providers differ in the sense of usability, availability, reliability and topicality of their provided data.

The missing standard is complicating the reusability, exchange and comparison of tourism data. An interaction between miscellaneous systems becomes challenging and also innovation can be restrained. Regarding these facts, the problem statement of the master thesis can be defined as follows:

How a standardized data schema for points of interest can look like to facilitate collaboration, reusability and innovation for providers in eTourism and finally encourage the arising of beneficial effects for the end-consumer?

1.2 Aim of the Work

The aim of the work comprises different issues. First an introduction in the topic of eTourism is given. The current situation and state of the art regarding to tourism data standards is observed. The establishment of the standardized data schema for points of interest is based on the survey covering eTourism online providers and their analysis. The survey of online providers focuses on their

- Application Programming Interfaces: availability, free use, cost policy etc.
- Data availability: format, accuracy, completeness etc.
- Trust and security issues: access, reuse of content etc.
- Policy guidelines: copyrights, registration etc.

The main objective is to create a matrix showing all aspects like APIs, data availability, security issues, fees, policy guidelines etc. of the various eTourism online providers to get an overview of the current situation and answering the research question.

1.3 State of the Art

There is a big variety of online providers in the area of eTourism - Lonely Planet, Falk, Garmin, Google, TomTom, MarcoPolo just to name some of them. Currently many different data formats are used to store point of interest data. However, most of them are based on the WGS84 (World Geodetic System) format. WGS84 [27] is a geodetic reference system to enable an uniform positioning, therefore it provides a standard coordinate frame for the Earth.

Gradually some data formats for points of interest have established:

- GPS eXchange Format (GPX)
- Keyhole Markup Language (KML)
- Comma-Separated Values Format (CSV)
- etc.

eTourism online providers have different focuses depending on their application area (e.g. navigation systems, travel guides, GPS devices etc.). Most data exists in the field of navigation, routing and GPS devices. Generally vendors provide optimized data for the usage of their devices, software or services. An interesting aspect is the integration of user generated content (UGC). This type of content (locations, pictures, ratings, recommendations, etc.) can enrich the data basis of the providers and make it more topical and interactive.

1.4 Methodology

The thesis is based on an analytical and constructive methodological approach. Therefore the proceeding is divided into separate work packages. The sequence of these work packages is highly relevant for the methodical steps because of the interdependencies among each other.

Work package 1: Evaluation of existing tourism standards

This work package comprises the identification of relevant online providers, institutions and agencies. This will be carried out via the internet, literature research, contacting institutions etc. The main focus through out the thesis will be on standards, schemes, information etc. regarding to points of interest.

Work package 2: Investigation of eTourism online providers

The second work package includes the identification and evaluation of eTourism online providers, who are dealing with point of interest data. The analysis of available APIs (Application Programming Interface) and data is part of this package. Furthermore, the policy guidelines as well as trust and security issues are examined. The requirements of WP2 (work package) are obtained mostly via the internet. Direct contact to the providers can also lead to results, which are relevant for this work package.

Work package 3: Realization of a data schema for points of interest

The application-oriented part of the thesis is included in WP3 and WP4. The development of a standardized data schema for point of interest data is the essential content of this part. Existing and established data formats are considered and taken into account. A leading purpose is the development of an extendable and reusable data schema for point of interest and a the realization of the data structure for this schema. Result of this step is a database for collected point of interest data.

Work package 4: Implementation of eTourism online provider APIs

The concrete implementation of several available APIs is part of the fourth work package. In the course of this implementation phase data reliability, topicality and accuracy are investigated and determined.

Work package 5: Result presentation, summary and outlook

The final work package comprises the presentation of the survey results. This is done via a matrix, showing all investigated aspects of the individual eTourism online provider. In this manner comparability in regard to the analysis is facilitated. The evaluation of the implemented prototype is also part of this final work package. Furthermore, it includes the answering of the research question and gives an outlook with respect to the study area.

1.5 Structure of the Work

The structure of the work is strongly related to the methodological approach and the contained work packages. The introduction gives a general prospect of the topic and defines the research question. The aim of the work as well as the state of the art are discussed subsequently.

An overview of eTourism, tourism organizations and existing IT standards in tourism is pointed out afterwards. The current situation as well as future trends are taken into account. After an introduction into the eTourism topic, point of interest data is evaluated. Therefore existing formats are determined, evaluated and described.

One of the main parts of the thesis is the evaluation of eTourism online providers. The selection method as well as the established results are described in detail. Therefore a comparative matrix of the selected eTourism online providers is established. The prototypical implementation is based on this analysis.

The constructive methodological approach finds application in the prototypical implementation. After an introduction and definition of purpose and scope of the prototype, the new, potential POI schema PXF is introduced. The prototype architecture, API requirements and database requirements are pointed out subsequently. The documentation as well as the verification of the prototype completes the chapter. Concluding a summary of the master thesis is given.

eTourism

The tourism industry has great importance for the global economy. In Austria 5.4% of the gross domestic product (GDP) was generated directly by the tourism and leisure industry [6] in 2009. The direct value-added effects in tourism industry increased continuously until the end of the last decade. Now after the economic crisis a rise is predicted again.

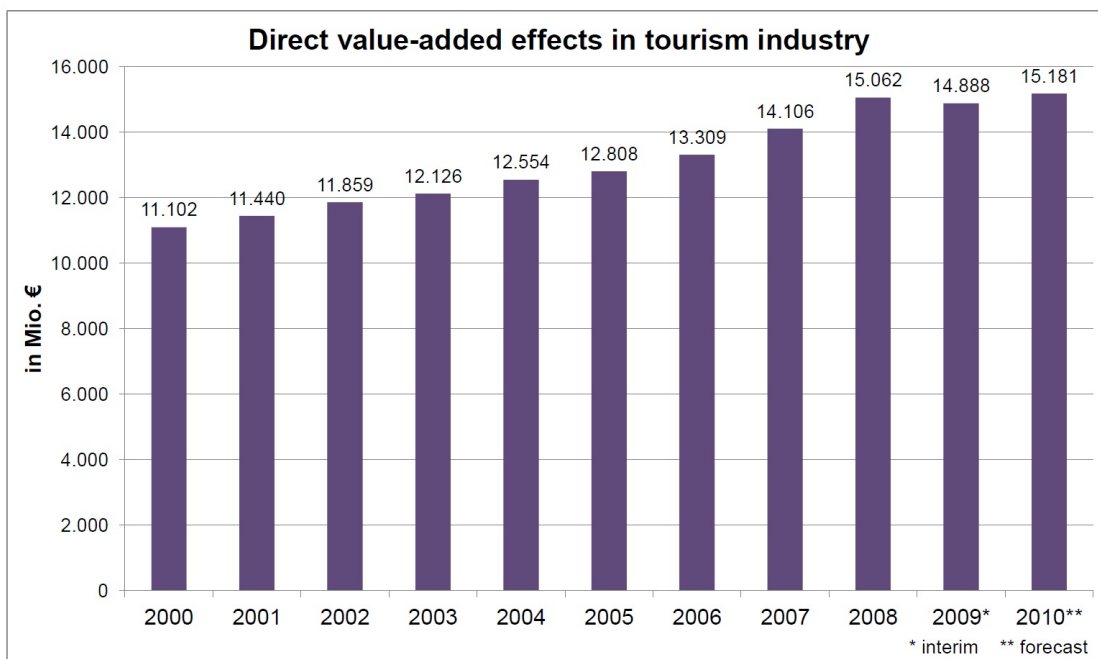


Figure 2.1: Direct value-added effects [6]

The tourism industry was one of the first sectors utilizing ICT. This development started in the 1960s [14]. The tourism industry comprises a great variety of stakeholders, which are cooperating to offer best services [91] and optimize their business processes. The networking between the different stakeholders is highly important and distinctive. ICT and the web are offering numerous opportunities for this industry. Business processes, operations, communication etc. can be redesigned and optimized. Especially the web provides an enormous amount of information, which is usable and applicable in the tourism sector.

Buying products or services via the web has become a familiar process. In 2011 more than 2.8 million people in Austria bought goods or services via the web during the last 12 months [6].

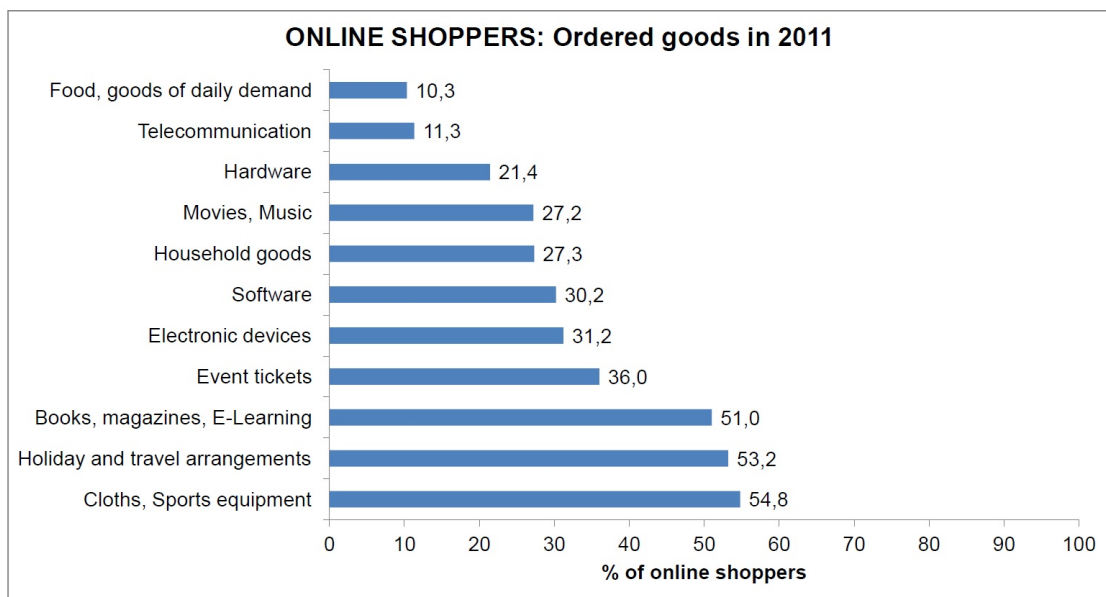


Figure 2.2: Online ordered goods in 2011

Figure 2.2 shows the most popular products ordered online. Holiday accommodations and travel arrangements are the second most popular product group shopped via the web.

The use of ICT in tourism leads up to the so called eTourism. In principle eTourism can be defined as the analysis, design, implementation and application of e-commerce solutions in the tourism industry as well as the investigation of associated economic processes and market structures [92]. Furthermore, it can be seen as the intersection [10] between ICT, business and tourism (Figure 2.3).

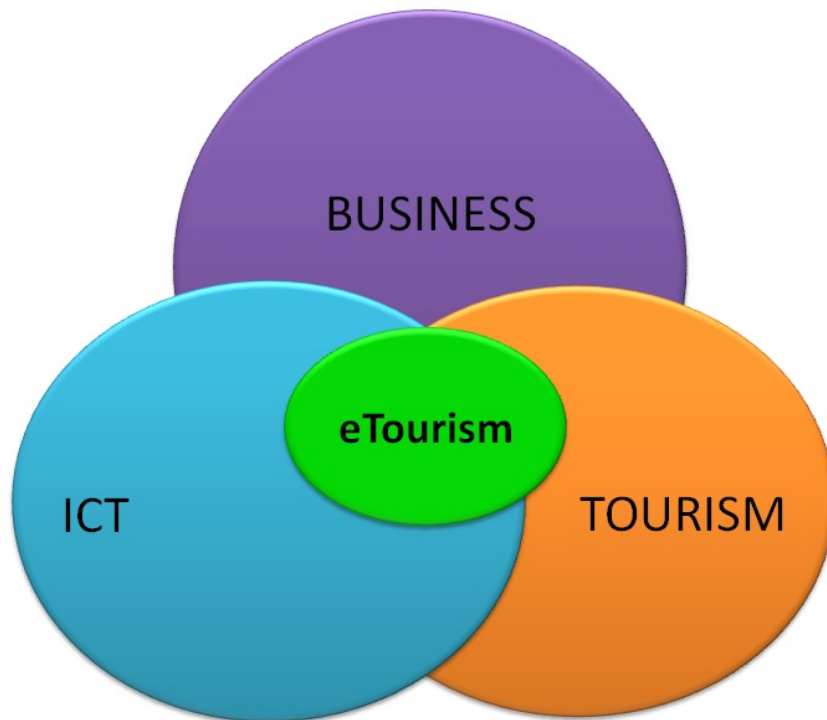


Figure 2.3: eTourism

Due to the multitude of stakeholders and application areas many providers in the eTourism are operating on the market, for example tour operators, travel agencies, hotels, booking portals or rating website operators. For the master thesis eTourism online providers are investigated, which are offering interfaces to access their data, especial focus is given to point of interest data.

Tourism Organizations and existing IT Standards

Tourism organizations encourage the cooperation between various parties. They develop standards, norms and constitute a source of know-how in the field of tourism. Due to the multitude of organizations in this area the focus is laid on the main international leaders.

Secure, efficient and uniform interaction in the tourism network is becoming increasingly important. To enable collaboration between systems, services and users consistent policies are needed. In the following chapter tourism organizations are introduced and existing IT standards, which are applied directly and indirectly in the tourism domain, are discussed [22]. Since some of these organizations are developing data standards and ensure interoperability between various institutions, they are relevant for the further survey.



Figure 3.1: Tourism IT Standards

3.1 ACRISS

ACRISS (Association of Car Rental Industry Systems Standards) was founded in 1989 by the car rental companies Avis, Budget, Europcar and Hertz [2]. The need of a standard in this area leads to that foundation. The focus of the association lies on the area of Europe, Middle East and Africa. Their mission is to provide a standard for the car rental industry, to improve the interoperability, comparability and profitability. Therefore a car classification code was defined, which is standardizing the classification of different vehicles. The use of this code is reserved for members.

The code consists of 4 characters each one represents a different attribute of a vehicle. Example: CCMN = Compact Car - 2/4 Door - Manual Transmission - Non air-conditioned

3.2 ANSI ASC X12

The interoperability between business partners is one goal of the ANSI ASC X12 [4] standard, which defines a data interchange format for EDI messages. With the X12 standard the ANSI (American National Standards Institute) developed a standard for business transactions in various branches, which is mainly used in North America. X12 is represented via a hierarchical structure and based on XML syntax. The first release of the ANSI ASC X12 standard was in 1982. There are multiple subcommittees, which are focusing on different areas of application. The subcommittee X12I TG08 is in charge of travel, tourism and leisure [26].

3.3 CENT/TC 329

The service standards committee NA 159 is responsible for standardization work in the service area [17]. The committee belongs to the German Institute for Standardization (DIN) and develops standards in the domain of tourism services, hotels and other types of tourism accommodation, travel agencies and tour operators, recreational diving services, language study tours and tourist guide services.

3.4 DATEX

DATEX (Data Exchange) is defining a standard for the interoperability in the domain of traffic [8]. It provides a set of specifications for the data exchange between traffic control systems and other service systems. The DATEX standard is applied in Europe.

3.5 ECTAA

The ECTAA is an association of Travel Agents & Tour Operators in Europe. ECTAA was founded in 1961 and is a non profit organization. The association gathers and determines EU policies, which may have an impact on the tourism sector [18]. ECTAA has 25 EU member states as well as some non-EU member states.

3.6 HEDNA

HEDNA is the Hotel Electronic Distribution Network Association which is a non profit organization and was founded in 1991 [36]. The focus of HEDNA lies on the identification of distribution opportunities (fn) and the effective usage of newest information technology in the domain of lodging industry. The association is operating worldwide.

3.7 HITIS

HITIS stands for Hospitality Industry Technology Integration Standards [3]. This project was initiated by the American Hotel Motel Association with the goal build up a standard for hospitality services and functions. In the course of the HITIS Project also a common data dictionary for hospitality data was established. The HITIS standards are based on XML.

3.8 IATA

The International Air Transport Association was founded more than 60 years ago and is the global trade organization of airlines [38]. The number of members comprises around 230 airlines. IATA is developing standards for the air transport industry to build a global industry. The identifiability of airports, airlines, railway stations are enabled via the IATA codes.

3.9 IATAN

IATAN stands for International Airlines Travel Agent network and is a division of the IATA [43]. Their focus is on the US tourism industry with developing business standards to make the services more efficient, economical and professional.

3.10 ICAO

The International Civil Aviation Organization was founded in 1944 by the United Nations [44]. ICAO is defining techniques and principles for the international civil aviation and comprises around 190 member states. Some responsibilities of the ICAO are:

- Definition of standards for the international aviation
- Regulation of traffic rights
- Elaboration of guidelines and recommendations
- Implementation of infrastructure
- Allocation of ICAO codes
- Development of standards for machine readable travel documents
- etc.

3.11 IFITT

IFITT is the International Federation for Information Technologies in Travel and Tourism [40]. The IFITT Reference Model Special Interest Group (RMSIG) developed a reference model to harmonize electronic tourism markets. Therefore they provide a framework to model tourism markets. The modeling is based on the Unified Modeling Language (UML).

3.12 IHRA

The International Hotel & Restaurant Association (IHRA) was founded in 1947 and is a non-profit organization. National hotels and restaurants are members of the association and also hotel and restaurant chains. The representation of the collective industry outwards as well as the promotion of the hospitality industry are goals of IHRA.

3.13 JourneyWeb

The project JourneyWeb developed a XML (eXtensible Markup Language) communication protocol for journey planning computer systems to support multimodal journeys in different regions [45]. JourneyWeb is a research project funded by the UK Department for Transport.

3.14 KAREN

The Keystone Architecture Required for European Networks project started in 1998 and is providing a framework architecture for intelligent transport systems (ITS). KAREN is dealing with different architectural issues like user needs, framework architecture and supporting documents [19].

3.15 OTA

The Open Travel Alliance focuses on the electronic exchange of business information among all sectors of the travel industry [55]. Members of the alliance are among airlines, hotel companies, car rental firms, cruise lines, railroads, tour operators, travel agencies, software provider etc. The exchange of data, including vocabulary and grammar, is based on XML (eXtensible Markup Language).

3.16 TIH

The Travel Information Highway (TIH) is an open community, which operates in the traffic and travel sector [71]. Their aim is the efficient, economical exchange of data and information in this area. The membership is free. They defined TIH principles, which are best practice approaches in the field of travel data exchange, to facilitate a framework to save time, money and effort when it comes to system interconnections.

3.17 TIN

The Tourism Information Norm (TIN) is a framework of standards, guidelines and definitions of the German Tourism Association [16]. The definitions are used for classification of rooms, hotels, facilities, etc. The aim of TIN is the more consistent description of tourist services and thereby provides an easy access to those services.

3.18 Transmodel

The Transmodel is a reference data model for public transport, it is conceptual and platform independent [76]. This model supports public transport operators and other providers to implement improved information services in this field of application. Transmodel focuses on the information needs in the following areas:

- Tactical planning
- Personnel disposition
- Operations monitoring and control
- Passenger information
- Fare collection
- Management information and statistics

A further aim of Transmodel is the opening of the market for collaborations between different software systems, which leads to a more efficient handling of public transport processes.

3.19 TransXchange

TransXChange is a XML standard for exchanging bus schedules and related data in the UK. The standard was initiated by the UK Department of Transport [85]. The latest version is 2.4, which was released in February this year. TransXChange includes two different XML schemas, one for the registration of bus timetables and one for the exchange of bus timetables between several information systems.

3.20 TRIDENT

TRIDENT stands for Transport Intermodality Data sharing and Exchange Networks. This project was realized within the Information Society Technologies (IST) programme [77]. The focus of TRIDENT lies on the European wide multimodal transport systems. The goal is to facilitate the exchange of data between different vendors. Results of TRIDENT included new standards as well as recommendations for the implementation of multimodal transport information systems.

3.21 TTI

The Travel Technology Initiative (TTI) was founded in 1989 [78]. It is an European wide institution and works on the development of standards for the areas traveling, tourism and leisure. TTI publishes the travel industry specific EDI messages known as Unicorn. Currently there are more than 130 Unicorn EDI messages, which are used in this business area.

3.22 TTRA

The Travel and Tourism Research Association (TTRA) provides networking and educational opportunities in the field of travel and tourism for it's members [80]. Through publications, conferences and their website they encourage research work.

3.23 UIC

The UIC is the international railway organization and comprises 197 members across all continents [86]. Their mission is to improve transport processes and meet the challenges of mobility. UIC develops standards for railways, which should enable world wide interoperability between systems and also other transport modes.

3.24 UN/LOCODE

The United Nations Code for Trade and Transport Locations is defined by the Economic Commission for Europe (UN/ECE), which is a sub organization of the United Nations [87]. The code specifies geographic locations, which are relevant for trade and transport. Typically UN/LOCODEs consists of five characters. The first two letters code the country and the other three the location within the country.

| | |
|--------|---------------|
| DE BER | Berlin |
| CH BRN | Bern |
| US NYC | New York City |
| AT VIE | Wien |
| NL AMS | Amsterdam |

Figure 3.2: UN/LOCODE Examples

3.25 UN/EDIFACTTT&L

The United Nations rules for Electronic Data Interchange for Administration, Commerce and Transport Travel Tourism & Leisure is defining a framework for cross-industry standards in

the area of business data interchange. TT&L is a subgroup and focuses on tourism standards to define messages, when it comes to data exchange between services and systems.

3.26 USTOA

The United States Tour Operators Association (USTOA) is representing the tour operator industry in the United States [88]. The Association was founded in 1972. Their aim is to develop improved industry standards for tour operators.

3.27 WATA

The World Association of Travel Agencies (WATA) is a worldwide organization [95]. Their members include travel agencies, which are applying the highest standards to improve services and interoperability between systems, operators and agents.

3.28 WTO

The World Tourism Organization (WTO) was founded in 1974 and is a United Nations organization. Focus of this global organization is the collection and reporting statistical information in the tourism sector. A further aim is to reduce obstacles to international trade, to encourage economic growth and development [96].

3.29 WTTC

The World Travel & Tourism Council (WTTC) was founded in 1990. It's members are international business leaders in travel and tourism companies [97]. WTTC builds a forum for issues concerning tourism and supports partnerships between the public and private sectors parties.

Points of Interest

4.1 Introduction

A point of interest is a specific point location that someone may find useful or interesting. In terms of navigation, routing, but also in tourism, such points are beneficial and used in different contexts. A point of interest could describe various things like:

- Gas stations
- Pharmacies
- Cash dispensers
- Police stations
- Hospitals
- Sights
- Parks
- Museums
- etc.

Such locations can be of interest in daily life as well as during vacation. About 10 years ago mobile navigation systems started to support our routing behavior. The first devices were PDA's with additional GPS receivers. In the year 2005 the first devices, which were exclusive and specialized for navigation, were introduced to the market and very successful.

In the last years the situation has changed. Modern smartphones captured the market and are competing strongly with mobile navigation systems. Smart phones are multifunctional and equipped with GPS receivers, which opened up the possibility to do also routing via the phone.

A lot of research and development is done in this area to optimize the use of a smart phone as a mobile navigation system. There are costly and also free applications available. In terms of accuracy, traffic information and usability a continuous improvement takes place.

Navigation systems are using maps to route the user to the desired destination. Points of interest are often represented as icons on the map (Figure 4.1). Those pins can have different appearances to display the different categories of point of interests.



Figure 4.1: POI icons

Point of interests are typically described by GPS coordinates (latitude and longitude), a name and optional additional information (Figure 4.2). This additional information can be a detailed description, opening hours, phone number, address etc. For point of interests the most common geodic reference system is the World Geodetic System 1984 (WGS84), which is a standard for determining exact positions on earth.

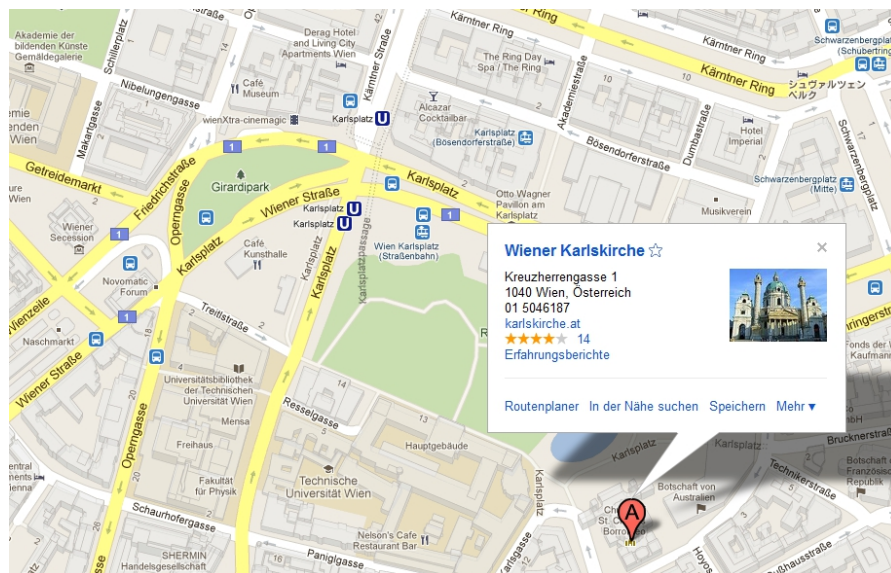


Figure 4.2: Karlskirche in Vienna [33]

4.2 Existing Formats and Standards

Points of interest currently have no uniform standard, which leads to the fact, that there are a lot of different file formats for such information. In the following several common file formats in this area will be discussed.

4.2.1 ASCII, CSV, TXT

The ASCII (American Standard Code for Information Interchange) standard could be seen as the most generic format for points of interest, which is supported by most providers that use such data.

```
16.221889, 48.115343, "Karlskirche Wien"  
16.379895, 48.190821, "Schloss Belvedere Wien"  
16.311844, 48.184531, "Schoenbrunn Wien"
```

Listing 4.1: POI example in ASCII format

The individual entries are separated by commas in the file. Data files in the ASCII standard typically have the file extensions:

- .asc
- .txt
- .csv

4.2.2 BMP

POI data vendors often provide an additional icon for a POI. These icons are mostly in the Windows Bitmap (.bmp) format and represent the POI on the map. The size of the icon depends on the type of application (navigation, web, mobile phone, etc.) it is used for.

4.2.3 GDB

Garmin Mapsource (GDB) is a data format used by Garmin. Files in the GDB format can be imported to Garmin devices and programs. Garmin MapSource (Figure 4.3) is the software to manage maps, tours and points of interest. The program enables a synchronization with the Garmin device. GDB is the internal format for data exchange between MapSource and GPS devices. There is no explicit documentation available for the GDB format.

4.2.4 GPX

GPX (GPS Exchange Format) is a general format for GPS data. The format was developed by the US company TopoGrafix and first released in 2002. The GPX format is based on XML and supports the storage of POI data, routes and tracks. Due to the fact that the format is based on XML, which is an open standard, GPX is used by many vendors. A variety of GPS devices and

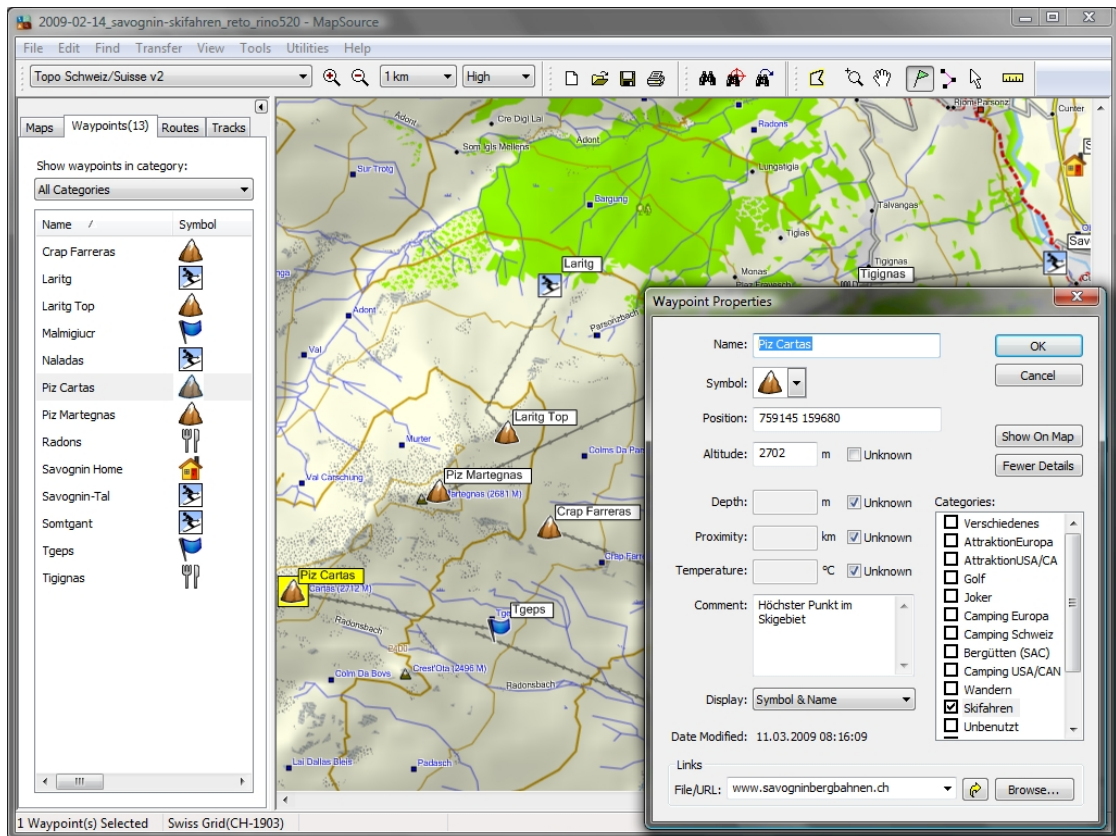


Figure 4.3: Garmin MapSource [75]

software can deal with this format, whereby an import, export and editing of GPX data is easily possible.

The current schema version is GPX 1.1 and was released in August 2004. A detailed documentation of the format is provided by TopoGrafix [74]. The schema provides several components.

Element Types

- gpx

Complex Types

- gpxType
- metadataType
- wptType
- rteType

- `trkType`
- `extensionsType`
- `trksegType`
- `copyrightType`
- `linkType`
- `emailType`
- `personType`
- `ptType`
- `ptsegType`
- `boundsType`

Simple Types

- `latitudeType`
- `longitudeType`
- `degreesType`
- `fixType`
- `dgpsStationType`

Points of interest are represented by the complex type *wptType*. The documentation includes a general part, a XML Instance Representation (Appendix A.1) and a Schema Component Representation (Appendix A.2).

4.2.5 KML

KML (Keyhole Markup Language) is a format used among others by Google Earth and Google Maps. The format is based on XML and was initially developed by Keyhole Corp. Since 2004 Google is developing the standard by the company's takeover of Keyhole Corp. The current version of KML is 2.2 [94] and was released in April 2008. The standard is maintained by the OGC (Open Geospatial Consortium).

The complete specification of the KML standard is provided by OGC, a detailed reference is supplied by Google. Figure 4.4 shows a class tree, which includes all schema elements based on KML Version 2.2. Points of interest can be represented by a combination of several KML elements, like placemark and point (Listing 4.2).

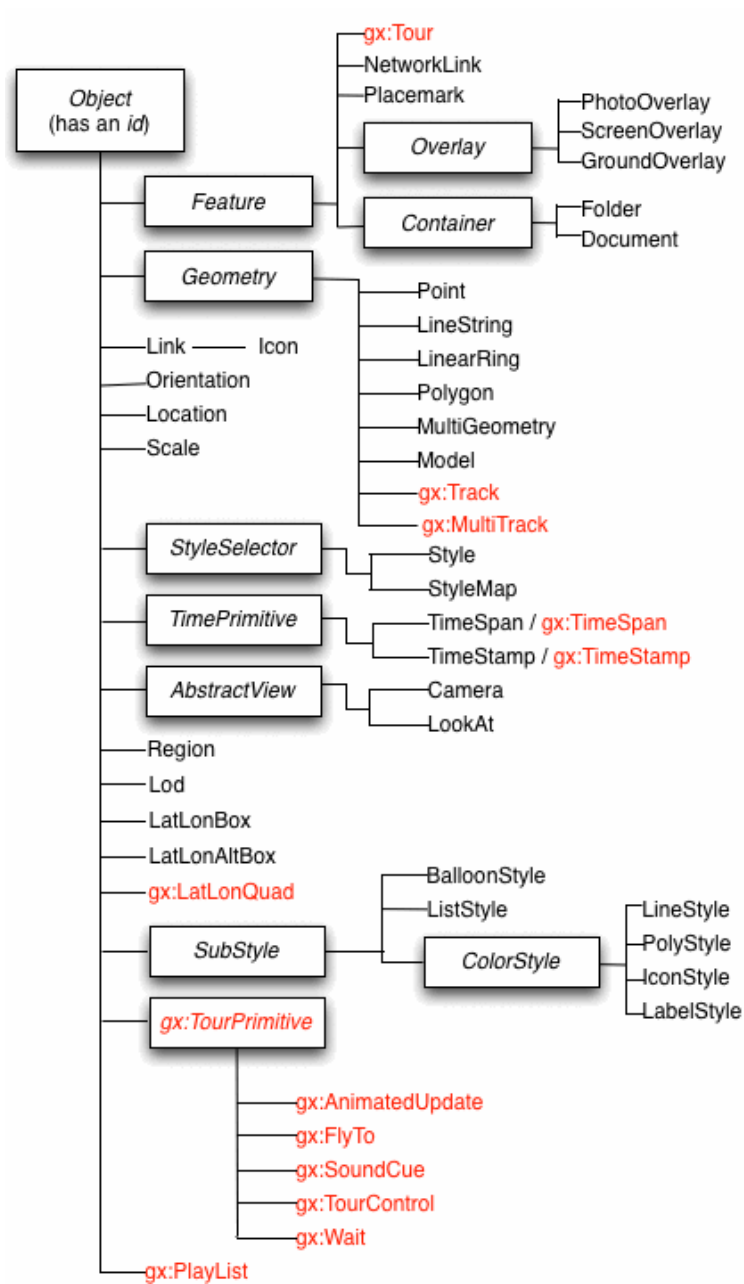


Figure 4.4: KML Schema


```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<Placemark>
  <name>New York City</name>
  <description>New York City</description>
  <Point>
    <coordinates>-74.006393,40.714172,0</coordinates>
  </Point>
</Placemark>
</Document>
</kml>
```

Listing 4.2: KML placemark

KML enables the storage of place marks, lines, polygons, images, 3D models as well as descriptions and camera view positions. KMZ files (.kmz) often occur in this context, which are compressed KML files in the ZIP format.

4.2.6 OV2

OV2 is a file format used by TomTom to store point of interest data and can be in binary or ASCII format. Usually an OV2 file is representing one point of interest with the attributes longitude, latitude and name. The OV2 file possesses a corresponding BMP (Windows Bitmap) file, which is an icon belonging to the POI. The icon can visualize the POI on the map.

4.2.7 WPT

Another format used to store POI data is WPT. This format is used by OziExplorer [15], which is a GPS mapping software developed by Des Newman. The software enables to work with scanned and digital maps. The import, export, creation and editing of waypoints, routes and tracks is supported, above that OziExplorer is compatible with most GPS receivers of various brands.

The WPT (waypoint) format is derived from the CSV (Comma-Separated Values) format. Points of interest (waypoints) are described in a line, where the individual attributes are separated by commas.

eTourism Online Providers

5.1 Introduction

The web is becoming enormously important in the are of tourism. Therefore a great variety of data online providers and portals have established in this field.

The first step with regard to the evaluation of eTourism online providers, who are operating with point of interest data, were the identification of relevant sources. After the accomplished selection of providers, available APIs (Application Programming Interface) and data were analyzed. Furthermore, the policy guidelines as well as trust and security issues were examined.



Figure 5.1: eTourism online providers

5.2 eTourism Online Provider Selection

The selection of the online providers in the field of eTourism is based on a web inquiry. Companies were identified, which have impact on eTourism. Those companies operate in different sectors, like travel guidebooks, travel magazines, rating platforms, navigation systems, etc.

The focus was set on the market leaders in the respective area. Online communities, services, blogs, applications, etc. were relevant for the selection and increased the chance for a provider to be selected and evaluated afterwards.

Furthermore, a local reference to Austria and accordingly Europe was considered. This is because that data can be checked more easily. The evaluation was performed for data sources that were available in German or English. Sources in other languages were not considered.

5.3 Falk Marco Polo Interactive GmbH

Falk is a publishing company for maps and was founded 1945 in Germany [20]. In 1998 Falk was bought by MairDumont, another publishing company in the field of cartography. Their focus is on maps, but they also offer software tools for navigation.

On the website an interactive map, route planner, hotel guide, business directory, etc. are available. On the map points of interest can be fade in and out. Each of these points can be clicked on the map and the user receives additional information such as address, telephone number, opening times, rating etc.

Application Programming Interface

Falk offers an application programming interface, the Falk Map API, which supports the implementation of interactive map applications [21]. The API enables the integration of Falk maps in other applications. Maps can be edited and enriched with own content. Furthermore, route planning can be done via the Falk Map API.

Currently there are two versions of the API available, although the versions only differ in the coordinate format they use. Version 1 uses the Mercator format, whereas Version 2 operates with the WGS84 (World Geodetic System 1984) format. To use the Falk Map API a user interface has to be implemented, this is not part of the API. The only graphical interface supported, is the map widget (see Javascript 5.2 and the corresponding map result 5.3). A detailed documentation with various examples of the API is available on the Web [21].

Data availability

Point of interest data is available, but only inside of Falk Services, like the route planner on their website or within other business solutions. The Falk Content & Internet Solutions (Falk CIS), which also belongs to the MairDumont group, develops solutions to improve mobility processes, location based services and also acts as data provider for third parties. Furthermore, Falk CIS runs falk.de and gofald.com, which are leading web portals in the field of mobility.

```
view plain copy to clipboard print ?
01. registerNodeListener(window, "load", main);
02.
03. // html loaded
04. function main(){
05.     // create a new map
06.     var map = new FalkMap("mymap");
07.     map.changeMapLayer("MSMAP");
08.     var geocode = new FalkPoint(9.223379437790086, 48.72223860481109);
09.
10.     // get the overlay
11.     var overlay = map.getMapOverlay();
12.
13.     // create a poi
14.     var fmo = new FalkMapObject();
15.     fmo.setGeocode(geocode);
16.
17.     // create a icon
18.     fmo.icon = new FalkIcon();
19.     fmo.icon.url = "/img/icons/sonnig.gif";
20.
21.     // add the falk map object to the overlay
22.     overlay.addFalkMapObject (fmo);
23.
24.     map.centerMapToGeocode(geocode, 13);
25. }
```

Figure 5.2: Javascript: Show POI on map

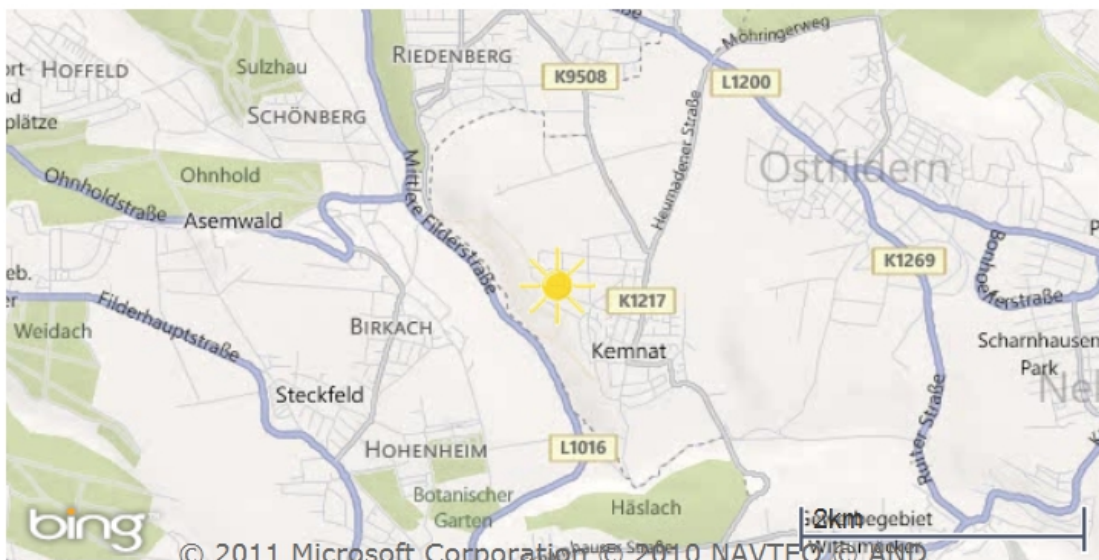


Figure 5.3: Result of Javascript: Show POI on map

The company is integrating POI data in B2B solutions, for example in navigation systems and other mobility solutions. Different travel guide solutions are provided by Falk CIS:

- TravelGuide
- CityGuide
- EventGuide
- RegionGuide
- Truck POIs
- Science Routes
- etc.

Those digital guides can be acquired from other companies and integrated in their products and services. The guides are using point of interest data to serve the users with interesting location based information. Falk CIS offers this information in different languages and updates are released once a year. Data is processed in the XML format and coded in WGS84.

Summing up, Falk is focused on own mobility services and B2B solutions. Point of interest data is not freely available.

Trust and security issues

For the usage of the Falk Map API a registration is necessary. The approval is done for a specific domain name or an IP address. Via the granted API key the Falk Map API can be integrated in the own application.

Policy guidelines

The Falk CIS owns all rights related to the Falk Map API, which is hosted under the domain <http://maps.falk.de>. Since Falk is not providing content (e.g. point of interest data) via the Falk Map API, the dissemination of content to third parties or the commercial use are no relevant subjects.

Third party examples

Tripwolf is an online platform offering a *social travel guide* [79]. Tripwolf is combining information from Falk Marco Polo with user generated recommendations and reviews. Travel guides can be adapted individually and sent to the smartphone application.

BMW provides Falk on-board navigation systems in various vehicles. In addition to street information, Falk travel guides are integrated and offer further travel information for the driver [9].

5.4 Flickr

Flickr is a photo and video sharing platform. Flickr emerged 2004 by Ludicorp. 2005 Yahoo bought the company and the product Flickr. The platform enables the upload, sharing, tagging, etc. of photos and videos and further community activities. Content can be published in an arbitrary license, including the Creative Commons license.

Application Programming Interface

The Flickr API provides nearly all functionality the website does and is free of use [98]. In general to access the API a Flickr user account and a registration of the application is required. After a successful registration an API key is provided. Via this key the OAuth 1.0 authentication has to be done, but not all methods require an authentication (e.g. flickr.photos.getInfo).

API requests can be carried out via REST, XML-RPC or SOAP. Depending on the selection, the appropriate target service URL has to be used. Response formats supported by the API are XML, JSON or SOAP. The API provides methods for various resources:

- activity
- auth
- auth.oauth
- blogs
- collections
- commons
- contacts
- favorites
- galleries
- groups
- interestingness
- machinetags
- panda
- people
- photos
- photosets
- places

- prefs
- push
- reflection
- stats
- tags
- test
- url

All methods including parameters, error codes, etc. regarding the resources are documented in detail on the website. An online explorer enables the testing of API requests (see Figure 5.4).

The screenshot displays the Flickr API Explorer interface for the `flickr.photos.getInfo` method. It includes a table for arguments, a section for useful values, and a text area showing the XML response.

| Name | Required | Send | Value |
|----------|----------|-------------------------------------|-----------|
| photo_id | required | <input checked="" type="checkbox"/> | 201153546 |
| secret | optional | <input type="checkbox"/> | |

Useful Values

- Your user ID:
- Your recent photo IDs:
- Your recent photoseid IDs:
- Your recent group IDs:
- Your contact IDs:

Output: XML (REST)

Sign call as [redacted] with full permissions?
 Sign call with no user token?
 Do not sign call?

Call Method...

[Back to the flickr.photos.getInfo documentation](#)

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
  <photo id="201153546" secret="417458f7be" server="72" farm="1" dateuploaded="1154198776" isfavorited="0"
  <owner nsid="87742066@N00" username="let" realname="" location="w, d" iconserver="6120" iconfarm="1"
  <title>Wien 2006 - Rathaus</title>
  <description>Wien 2006, Austria - Rathaus</description>
  <visibility ispublic="1" isfriend="0" isfamily="0" />
  <dates posted="1154198776" taken="2006-06-28 12:38:56" takengranularity="0" lastupdate="1311111111"
  <editability cancomment="0" canaddmeta="0" />
  <publiceditability cancomment="1" canaddmeta="0" />
  <usage candownload="0" canblog="0" canprint="0" canshare="0" />
  <comments>1</comments>
```

Figure 5.4: Flickr API Explorer

Particular focus will be set on resources, which have relevance for points of interest. Those resources are described in the following.

Photos

The main resource of Flickr is photo, since it is an online photo sharing platform. Using the different API methods, the photo resource can be accessed, searched, edited, updated or deleted.

The photo resource can comprise several sub-resources like comments, geo, licenses, note, people or upload. All calls must be effected through the subsequent URL:

```
http://api.flickr.com/services/rest/?method=flickr.photos.name_of_method
```

Listing 5.1: Flickr URL Resource *photo*

Tags

Flickr assimilates two types of tags, standard tags and machine tags. A tag is an arbitrary label assigned to a picture or video with the purpose to describe the resource and respectively what is shown there more accurately. With tags finding photos, which have something in common, is easier. The owner of the picture or video can specify, who is allowed to add tags to the resource. Flickr limits the number of tags per picture or video to 75. Figure 5.5 shows the most popular tags of flickr.

All time most popular tags

animals architecture **art** asia australia autumn baby band barcelona **beach** berlin bike bird
birds birthday black blackandwhite blue bw **california** canada **canon** car cat
chicago china christmas church **city** clouds color **concert** dance day de dog
england europe fall **family** fashion festival film florida flower flowers food
football **france** friends fun garden geotagged germany girl graffiti green halloween
hawaii holiday house india **instagramapp** iphone **iphoneography** island italia **italy**
japan kids la lake landscape light live london love macro me mexico model museum
music nature new newyork newyorkcity night **nikon** nyc ocean old paris
park party people photo photography photos portrait raw red river rock
san sanfrancisco scotland sea seattle show sky snow spain spring square
squareformat street summer sun sunset taiwan texas thailand tokyo toronto tour
travel tree trees trip uk urban usa vacation vintage washington water
wedding white winter woman yellow ZOO

Figure 5.5: Flickr tags

Machine tags

Tags help to describe, find and allocate pictures and content. Throughout the reason, that several users use different words to describe the same thing [7], a further classification can be beneficial. Machine tags have the same purpose as standard tags, with the difference that they have a specific syntax to define extra information. Although the name suggests something different, also machine tags are user generated.

Machine tags are also known as triple tags [99]. Throughout their special syntax the representation of additional semantic information is possible. The interpretation of such tags by a software becomes feasible. A machine tag consists of 3 parts:

namespace:predicate=value

The namespace, predicate and value can be selected arbitrarily. However, namespaces and predicates must begin with a character between a - z and they are case insensitive.

This syntax corresponds to the Resource Description Framework (RDF) model, which is used to describe information on the web [89]. The structure of any expression in RDF is based on a subject, a predicate and an object (Figure 5.6).

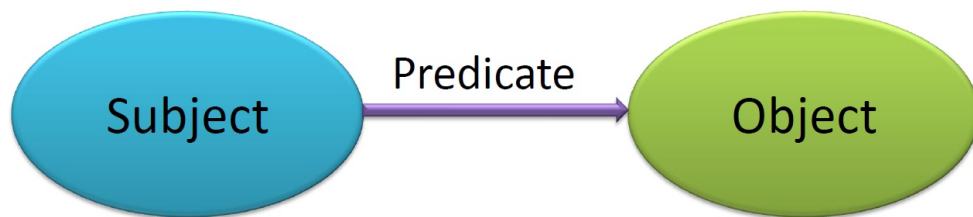


Figure 5.6: RDF Graph Data Model

In the case of Flickr machine tags can be used among other things to describe the geographical location of a picture. By entering the following two machine tags the location information can be assigned to the picture:

geo:lat=12.345678
geo:lon=42.123454

Data availability

Through the focus on photos and videos Flickr can enrich point of interest data with this kind of media. Via the provided API a comprehensive collection of methods can be used to access the resources. In August 2011 Flickr crossed the number of 6 billion pictures uploaded [59]. The number of registered users in June 2011 was about 51 million.

Trust and security issues

To use the full range of API methods a registration of the own application is required. Via the provided API key the authentication has to be carried out by OAuth 1.0. To prevent the API from misuse, Flickr limits the number of requests to the API to 3600 per hour. A caching of data is allowed up to 24 hours.

Policy guidelines

In order to use the Flickr API the Terms of Use stated on the website have to be accepted. Flickr provides access to various resources. However, Flickr is not owner of the pictures. The license under which an image was uploaded, must be respected at any time. To use pictures for commercial purposes, they have to be under Creative Commons license.

Applications using the Flickr API have to embed a standard header, which is shown in Figure 5.7.



Figure 5.7: Flickr Standard Header

Flickr prohibits to display more than 30 pictures on one page within the own application. Caching and storage of photos is not allowed longer than the period of service provisioning of the own application. Intellectual property rights, rights of privacy or rights of personality have to be respected at any time.

In general it is not allowed to generate revenue from the use or provision of applications using the Flickr APIs. For commercial usage a separate commercial API key has to be requested, in this case it is possible that Flickr charges fees. By default the usage of the Flickr API is free of charge.

Third party examples

Flickr offers the *App Garden* on their website, where users can provide their own applications to the community. All applications in the App Garden were implemented with the Flickr API [25] and can be made available via this platform.

Since 2009 Flickr has a partnership with Getty Images, which is a stock photo agency [24]. Flickr images can be labeled as suitable for the stock by the users themselves.

5.5 Foursquare

Foursquare is a platform, where users can make their current geographical location public, stay in contact with friends and gain deals for various locations. Foursquare was founded in 2009 and currently there are about 15 million users around the world [28]. The platform is mainly operated with applications for smart phones, which have GPS positioning. A major part of the mobile devices and platforms are supported. A user can *checkin* at a certain geographical position via the website, a SMS or the application on the smart phone. The *checkin* process is designated as the publication of the own location at a specified place. Furthermore, Foursquare offers a Merchant and a Venue Platform for venue managers to promote their location, organize events or offer special deals.

Application Programming Interface

Foursquare provides two APIs for developers to interact with their platform. The Core API enables to do user interaction, similar to the website or application on the smart phone. The Real-time API provides functionality for vendor and application operators.

Core API

The Core API provides various methods to access resources to interact with Foursquare [29]. The following resources are used within the Core API:

- users
- venues
- checkins
- tips
- lists
- updates
- photos
- settings
- specials
- campaigns
- events

To operate with the API a registration is required. With the application key provided the authentication has to be done via OAuth 2.0. The website offers an online explorer (see Figure 5.8, where API calls can be tested directly. By default the number of API calls is limited to 500 requests per hour per authenticated user. The response format of Foursquare APIs is JSON.

Real-time API

The Real-time API is experimental. It pushes information to operators, if users check in at their location on Foursquare. The API distinguishes between User Push API and Venue Push API. The User Push API traces user checkins and provides the information to the developer of the application. The Venue Push API makes information about current checkins available. Thereby the traffic at the several venues can be monitored. This API focuses on vendor managers, who want to record current user appearance.

API Explorer

https://api.foursquare.com/v2/venues/4b058890f964a520

OAuth token automatically added. https://api.foursquare.com/v2/venues/4b058890f964a520aacd22e3?oauth_token=JXVZRTCihYIY0MU0BF1TADXAQVU2QHVA3UMH2ABWA1DRQFEN&v=201111212

```

{
  meta: {
    code: 200
  },
  notifications: [
    {
      type: "notificationTray",
      item: {
        unreadCount: 0
      }
    }
  ],
  response: {
    venue: {
      id: "4b058890f964a520aacd22e3",
      name: "Hotel Sacher Wien",
      contact: {},
      location: {
        address: "Philharmonikerstrasse 4",
        lat: 48.204074,
        lng: 16.369703,
      }
    }
  }
}

```

Figure 5.8: Foursquare API Explorer

Data availability

Regarding point of interest data a link to a possible Foursquare representation could be useful. Via this reference the mainly static point of interest data could be enriched with topical information about reside users, events or special offers. The Foursquare data can be accessed via the provided APIs.

Trust and security issues

The access to the APIs requires a registration of the own application on Foursquare. Subsequently an authentication key is provided. With this key the use of the APIs via OAuth 2.0 authentication is possible. Foursquare limits the access rate to their APIs. By default 500 requests per hour per authenticated user are possible.

Policy guidelines

To operate with the APIs the policy of Foursquare has to be obeyed. Private information of users are only available, if the user has granted access to the own application. Without this authentication no user data is accessible. With a privacy policy of the own application the user must be informed about the data, which will be used, shown, reported, etc.

The attribution of Foursquare as the source of content has to be done, e.g via buttons, logo, etc. The resources for attribution are available on the website. Since Foursquare is based on topical information and actions a frequent refresh of data is regulated by the policy [29]. Following update guidelines have to be obeyed:

- Checkins must be refreshed every 24 hours
- No data can be kept for more than 30 days without being refreshed
- The data of users who havent authenticated the application may only be stored for up to three hours

User privacy and the intellectual property rights must be respected at any time. The copying or extracting of the Foursquare venue database is strictly not permitted. The use of the APIs is free.

Third party examples

The Foursquare Brand Platform supports the establishment of brand presence. Brands like Red-Bull, Nasa, New York Times, Louis Vuitton, etc. are using Foursquare to connect to their followers and community [30].

Wanderfly is a travel recommendation engine. Based on budget and interests a user can search for appropriate trips [90]. Users can plan, share and book complete trips via the platform. Wanderfly integrates content from various parties, including Foursquare.

5.6 Garmin

The Garmin International Inc was founded 1989 in the US [31]. Garmin is a manufacturer of GPS receivers and one of the leading companies in the field of navigation. Eponyms for the company were the founders Gary Burrell and Min Kao. In the beginning Garmin was focused solely on the production of GPS receivers. Meanwhile, Garmin is also operating successfully in other areas, such as road navigation, aviation, sport devices and navy.

The product range includes devices for street and outdoor navigation, mobile phones, sports, etc. Garmin devices work in principle with commercial maps, but also offer the possibility to load non-commercial maps on the units, like maps from the project Openstreetmap [54].

Application Programming Interface

Garmin provides several APIs with respect to various applications areas and use cases. However, all APIs require the interaction with a Garmin device.

The Garmin Device Interface SDK (Software Development Kit) is addressing advanced users, who want to develop own applications or services, which communicate with Garmin products. The SDK is documented, but there is no user support from Garmin side. The communication between application and Garmin device is done via RS-232 (Recommended Standard 232) or USB. The description of the protocols is included in the SDK documentation.

Furthermore, Garmin provides a Content API, which enables the creation of content for Garmin devices. This can be point of interest data (GPI Creator) as well as customizable voices (Voice Creator). The Garmin format for POI data is GPI. The Content API provides the conversion of XML and GPX data into this format. Developers can offer their content to others via download. Different business models are possible here, free or paid download.

Another service they offer is a plugin (Garmin Communicator Plugin) to synchronize, update and create for example points of interest, routes, tracks and map data between the Garmin device and a personalized website. The API of the plugin enables the transfer of GPS data. Files in the XML format (GPX and TCX) can be processed.

The Garmin Communicator API is a framework based on JavaScript and consists of three layers shown in Figure 5.9. The provision of UI components enables a more user friendly and simple integration of the plugin on an own website. Each layer can be accessed, depending on whether the user just wants to display information (DeviceDisplay), handle information via events (DeviceControl) or have a direct low-level access (DevicePlugin) to the device.

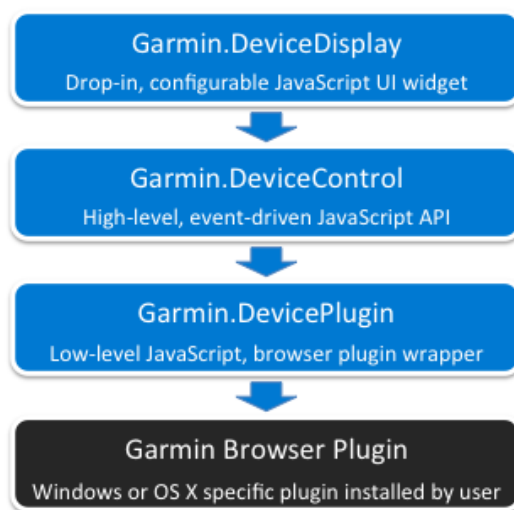


Figure 5.9: Garmin Communicator API architecture

Data availability

Especially in street navigation points of interest have a supporting role. Garmin supports the illustration of POIs on the maps. The point of interest data is provided by Garmin itself and can only be downloaded onto Garmin devices.

The data format used by Garmin is .gpi (Garmin Point of Interest). The format is not documented and can only be read using Garmin devices. The APIs enable the conversion of other formats (XML and GPX) into the GPI format, which leads to more interoperability. Point of interest data files are available at the company website and can be downloaded onto proprietary devices or used for integration in own Garmin API based applications. There are free and paid downloads.

Trust and security issues

Garmin grants a limited, non-exclusive, non-transferable and non-sublicensable license for the usage of their APIs and underlying content. To access a Garmin API a registration is necessary. After registration and agreeing to the Terms of Use a unique key for the own application respectively website is provided. The Garmin APIs are imported via the HTML Tag *script*, which includes the registration key. Requests to the API are only possible with a valid key.

Another requirement for the use of the API is the mandatory notification on the own website, which includes a highly visible explanation text and an authorized Garmin logo. These two components will advise the end user of the agreed and allowed communication between the application and the Garmin device or service.

Policy guidelines

The Intellectual Property Rights, which include all rights existing under patent law, copyright law, moral rights law, trade secret law, trademark law, etc. have to be respected at any time. It is not allowed to remove copyright or trademark notices contained in the API or any other Garmin content. Intellectual property rights of third party data providers must also be respected.

The full responsibility of content and conduct of the own application is assumed by the user of the API. A further condition is the reservation of all rights by Garmin to discontinue the API access and the underlying content at any time.

Third party examples

Since all Garmin APIs require the interaction with a Garmin device the applicability is restricted.

5.7 Lonely Planet

Lonely Planet was founded in 1972 by Maureen and Tony Wheeler. Today it is the world's largest guide book publisher. Since February 2011 Lonely Planet is entirely owned by BBC Worldwide. The product portfolio of Lonely Planet comprises travel guidebooks, digital applications, online community, monthly travel magazines, TV productions, etc. The website of Lonely Planet is leading in the area of online traveling information provision.

Application Programming Interface

Lonely Planet offers a content API with several functions for accessing places and points of interest. The API supports the following operations:

- search for places by name
- retrieve a list of POIs in a place
- retrieve a list of POIs within a geographic bounding box
- get detailed information about a POI

- filter a list of POIs by POI type

The services of the Lonely Planet API are dealing with two resources, places and points of interest. The resources and functions for accessing them, are described in detail in the following section.

Place

Places can be searched by name (see Example 5.2). The request returns a list of places, where the search attribute *name* is matching exactly or partly (substring).

```

http://apigateway.lonelyplanet.com/api/places?name={name-substring}

<places type="array">
  <place>
    <id>358656</id>
    <full-name>Europe –&gt; Austria –&gt; Vienna</full-name>
    <short-name>Vienna</short-name>
    <north-latitude>48.3440352760002</north-latitude>
    <south-latitude>48.1049878440002</south-latitude>
    <east-longitude>16.5510853190001</east-longitude>
    <west-longitude>16.1675806370002</west-longitude>
  </place>
</places>

```

Listing 5.2: Lonely Planet *Find Place by name*

Point of interest

The API facilitates three ways of POI search. Points of interest can be searched by ID (see Listing 5.3), by PlaceID (see Listing 5.4) or via a bounding box (see Listing 5.5). The search by POI ID will return one single point of interest, since the ID is unique within the API data pool.

```

http://apigateway.lonelyplanet.com/api/pois/{poi-id}

<places type="array">
  <place>
    <id>1322674</id>
    <full-name>Europe –&gt; France –&gt; Paris –&gt; Eiffel Tower, 16e &amp; La D&#233;fense
    </full-name>
    <short-name>Eiffel Tower, 16e &amp; La D&#233;fense</short-name>
    <north-latitude></north-latitude>
    <south-latitude></south-latitude>
    <east-longitude></east-longitude>
    <west-longitude></west-longitude>
  </place>
</places>

```

Listing 5.3: Lonely Planet *Find POI by id*

The search of points of interest via place ID enables a request of POIs, which are dedicated to the respective place.

```
http://apigateway.lonelyplanet.com/api/places/{placeID}/pois[?poi_type={type-name}]
```

```
<pois>
  <poi>
    <id>367907</id>
    <poi-type>Night</poi-type>
    <name>Coogee Bay Hotel</name>
    <digital-latitude>-33.920940998916</digital-latitude>
    <digital-longitude>151.256557703018</digital-longitude>
  </poi>
  <poi>
    <id>367711</id>
    <poi-type>Night</poi-type>
    <name>New Theatre</name>
    <digital-latitude>-33.9031756797962</digital-latitude>
    <digital-longitude>151.179843842983</digital-longitude>
  </poi>
</pois>
```

Listing 5.4: Lonely Planet *Find POI by place id*

The search parameter n, sh, e, w define a bounding box. Only points of interest will be returned, which are geographically referenced in this area. Additionally the search can be constrained by POI type. Possible POI type names are:

- eat
- sleep
- see
- shop
- night
- do

```
http://apigateway.lonelyplanet.com/api/bounding_boxes/{n},{sh},{e},{w}/pois?poi_type={typename}
```

Listing 5.5: Lonely Planet *Find POI by bounding box*

Data availability

Lonelyplanet.com enfolds over 20.000 pages of specific and detailed information regarding to travel destinations. The website has more than 4 million unique visitors per month and 85% of their travellers use the website frequently [5]. Through the availability of the content API, places and points of interest can be accessed. The access is limited to 5000 calls per day per API key and maximum 2 calls per second.

Trust and security issues

There are two types of accessing the content API, anonymously over HTTP or authenticated using OAuth or HTTP basic. The anonymous access is intended for testing the API and has a limit of 30 request per minute for all users.

For a more applicable access, the own application has to be registered. An application key will be provided for doing the authentication via HTTP basic or OAuth. Subsequently the number of possible requests per minute will be 120 per individual application. First time using the Lonely Planet content API the Developer Terms have to be signed.

Policy guidelines

To reproduce content of Lonely Planet a permission is needed and Lonely Planet has to be mentioned as owner of that content [61].

Third party examples

Lonely Planet content is integrated in various platforms, websites and services [60]. For example, the portal VisitBritain is using the content to inspire users and drive bookings. Travel companies like Air France, Air New Zealand, Qantas, Singapore Air, etc. also integrate content from Lonely Planet.

5.8 Panoramio

Panoramio is a platform for sharing pictures and was released in 2005 by a Spanish company. Since 2007 Google is the owner of Panoramio [34]. The platform enables the geographic referencing of photos, what is done via *geo tags*. Photos are shared and visible at the platform website, but also integrated in other Google services, like Google Earth.

The terms and conditions stipulate that only the user can upload pictures, that were taken from him. The tagging of an image with a geotag can be done either manually or by recording already geocoded pictures. For adding additional meta information, like keywords, EXIF files can be used. EXIF (Exchangeable Image File Format) is a standard format used by image recording devices for storing meta data regarding to a picture.

Application Programming Interface

Panoramio provides two APIs, the Panoramio Widget API and the Panoramio Data API. Both APIs are well documented on the Panoramio website.

Panoramio Widget API

This API enables the user to display pictures from Panoramio within own websites to enrich maps or other information provided. This can be done by two manners. The first possibility is to use the JavaScript library and second one is to use HTML templates, which can be integrated

with iframe tags into the own website. Figure 5.10 shows a comparison of the two possible types of application [41].

| | JavaScript library | HTML templates |
|--|-------------------------------------|-------------------------------------|
| Ease of deployment | Requires some programming | Just cut and paste HTML |
| Required knowledge of programming languages | JavaScript, HTML, CSS optional | HTML |
| Flexibility to combine widgets | Full power of the Widget API | Limited configurations available |
| User interaction | Page behavior is fully customizable | Some predefined behaviors available |
| Interaction between widgets and with rest of the page | Fully customizable | Generally not possible |
| Programmatic control over widgets | Yes | No |
| UI customization | Possible with CSS | Size and base color |
| Hosting web site | Must allow JavaScript | Must allow iframes |

Figure 5.10: Panoramio Widget API

The Panoramio Widget API is based on two components, the widget and the request. Through the request the user defines the pictures he wants to display. The widget is a frame element, which shows the selected pictures in the web browser. The API provides a single picture widget and a list widget showing several pictures at one time. The API request can be extended with specific search attributes, which a picture has to fulfill to be shown. Figure 5.11 is an example of a list widget showing pictures of mountains in an order of 2 rows and 6 columns. The picture navigation within the widget is a standard feature of the API.

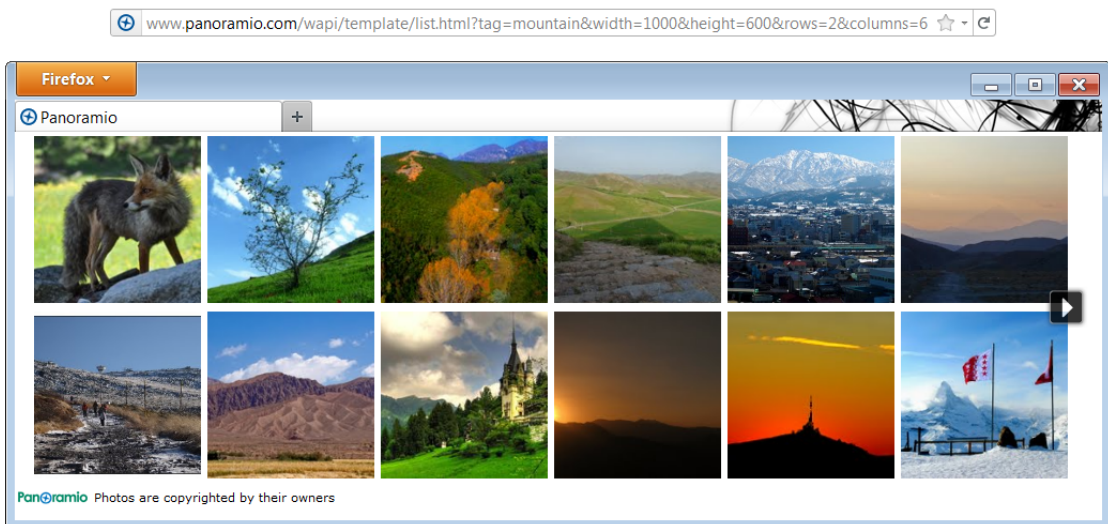


Figure 5.11: Panoramio Photo Widget

Panoramio Data API

The Panoramio Data API is based on the Representational State Transfer (REST) architecture for web applications. To access pictures a GET request has to be executed. The result data is provided in the JavaScript Object Notation (JSON) format. A GET request (see Listing 5.6) can be specified by several attribute settings, which will be explained in the following. The example shows the request of pictures, which are geographically referenced at the area of the Karlskirche in Vienna.

```
http://www.panoramio.com/map/get_panoramas.php?set=public&from=0&to=2&minx=16.37102&miny=48.19775&maxx=16.37268&maxy=48.19879&size=medium&mapfilter=true
```

Listing 5.6: Panoramio Data API request

Attribute *set*:

- public (popular photos)
- full (all photos)
- user (ID number)

Attribute *size*:

- original
- medium (default value)
- small
- thumbnail
- square
- mini_square

The attributes *minx*, *miny*, *maxx*, *maxy* define a bounding box. Only photos will be shown, which are geographically referenced in this area (minimum longitude, latitude, maximum longitude and latitude). With the attributes *from* and *to* the number of photos can be defined. 0 means the latest photo uploaded to Panoramio meeting all other selection criteria. For example, 'from=0 to=20' will extract the last 20 photos uploaded to Panoramio. One GET request is limited to the maximum of 100 returned photos. The attribute *mapfilter* can be true or false. True optimizes the representation of photos on a map, which means that pictures of the same location are avoided.

The response regarding to the GET request above is shown in Listing 5.7. The two last uploaded pictures to panoramio located at the area of the Karlskirche in Vienna are retrieved.

```

{
  "count": 348,
  "photos": [
    {
      "upload_date": "05 February 2007",
      "owner_name": " Kojak",
      "photo_id": 702975,
      "longitude": 16.371592,
      "height": 500,
      "width": 375,
      "photo_title": "Karlskirche",
      "latitude": 48.198602,
      "owner_url": "http://www.panoramio.com/user/123698",
      "owner_id": 123698,
      "photo_file_url": "http://mw2.google.com/mw-panoramio/photos/medium/702975.jpg",
      "photo_url": "http://www.panoramio.com/photo/702975"
    },
    {
      "upload_date": "22 January 2007",
      "owner_name": "Andre Bonacin",
      "photo_id": 524397,
      "longitude": 16.371517,
      "height": 364,
      "width": 500,
      "photo_title": "St.Charles Church, Vienna (late 1990s), Austria.",
      "latitude": 48.198277,
      "owner_url": "http://www.panoramio.com/user/44319",
      "owner_id": 44319,
      "photo_file_url": "http://mw2.google.com/mw-panoramio/photos/medium/524397.jpg",
      "photo_url": "http://www.panoramio.com/photo/524397"
    }
  ],
  "has_more": true
}

```

Listing 5.7: Panoramio Data API response

Data availability

Panoramio comprises more than 50 million pictures and the user community is growing continuously. Pictures are an important content to enrich point of interest data. By the fact that the content is user generated and the community is very active, a rich variety of pictures is available for various places, locations, points of interest etc. The two APIs provided by Panoramio are offering a simple way to integrate picture content into own websites or applications.

Trust and security issues

The Panoramio APIs (Panoramio Widget API and Panoramio Data API) are free for commercial and non-commercial purposes. For some kind of commercial applications and certain amount of accesses Panoramio reserves the right to charge fees. The use of Panoramio pictures is limited to displaying it to end users. The form, format and appearance provided by Panoramio must not be changed. The maximum amount of queries per day is 100.000, exceeding this value the service may stop working temporarily. Generally the APIs can be used without a registration.

Policy guidelines

Every photo displayed must include Panoramio's name and logo. Each mini square-, square-, thumbnail- or small-size version of a photo must link to a medium or original sized version of the same photo. Each medium or original sized version must link to the photo page on the Panoramio.com and also the author has to be mentioned. The text *'Photos provided by Panoramio are under the copyright of their owners'* has to be put under Panoramio photos. It is not allowed to use CSS or other techniques to hide the Panoramio logo.

Third party examples

Tripbase is a travel platform providing users a personalized, comprehensive and easy-to-use online search [42]. Tripbase integrates images from Panoramio to enrich their services. Also Tripwolf uses Panoramio pictures to complete their portfolio [79]. Users can link to Panoramio pictures in their personal trips on Tripwolf.

5.9 Qype

Qype was founded in 2006 by Stephan Uhrenbacher in Hamburg. It is an online rating portal where users can rate and review local businesses [65].

Adding POI data is possible by users as well as companies. The Qype content is mainly user-generated. Users have several possibilities within the portal, they can add places and corresponding photos, post reviews, tag other comments, share reviews, etc. Currently Qype is operating in the countries Austria, Brazil, France, Germany, Ireland, Italy, Poland, Spain, Switzerland and UK.

Application Programming Interface

Qype offers an application programming interface, which supports the integration of Qype data into own applications. The API uses basic web concepts like REST [23], HTTP, XML and JSON. To work with the API a user and application registration on the website is required. After successful registration an application key can be requested. First, an application key for a Standard Account is assigned, which allows up to 200 queries per day. For commercial or larger projects this limit may be too low, wherefore a Pro Account has to be requested.

***QYPE** Vienna [Change](#)

Business owner? [Write a review](#) [Sign up/Login](#) [English](#)

Search for: (pizza, pub, the ivy) Where? (town, suburb, postcode...) (13)

Home Eating & Drinking Nightlife Shopping **Arts & Entertainment** Services Health & Beauty All

You are in: Austria » Vienna » Arts & Entertainment » Zoos » Tiergarten Schönbrunn [EVENTS](#) | [USER GUIDES](#) | [FORUMS](#) | [FIND YOUR FRIENDS](#)

This is a Qype Austria place. View the Qype Austria page here: [Tiergarten Schönbrunn - Vienna](#)

Tiergarten Schönbrunn, Vienna

★★★★★ 58 reviews [Rating details](#)

[Write a review](#) [Fotos](#)

Star distribution
58 reviews

| | |
|-------|----|
| ★★★★★ | 42 |
| ★★★★☆ | 14 |
| ★★★☆☆ | 1 |

These are other people's reviews. How would you rate this place? [Write a review](#)

Category: Zoos Vienna

Contact us: [zoovienna](#)

Address: Maxingstr. +43-1-87...

Website: [www.zoo...](#)

Opening hours:
 Jänner 9.00-16.30
 Februar 9.00-17.00
 März 9.00-17.30
 April 9.00-18.30
 Mai 9.00-18.30
 Juni 9.00-18.30
 Juli 9.00-18.30
 August 9.00-18.30
 September 9.00-18.30
 Oktober bis Ende der Sommerzeit 9.00-17.30
 November 9.00-16.30
 Dezember 9.00-

Know an upcoming event? [Add it](#)

More details: [Show](#)

[All photos](#) [Upload photo](#)

[View larger map](#) [Print a map](#)

Places nearby Tiergarten Schönbrunn

Cafe Gloriette
 Schloßpark Schönbrunn (Gloriette), Zugang über die Hohenbergstraße oder über das Meiereitor Zufahrt Elisabethallee, 1130 Wien
 ★★★★★

Figure 5.12: Qype POI example

General concepts

The Qype API is based on the Representational State Transfer (REST) architecture for web applications. REST relies on the following paradigms:

- The communication is based on a client-server-architecture, where clients send requests to servers and servers respond according to the request.
- Every object and resource has a unique identifier. Through this Uniform Resource Identifier (URI) an object or resource can be addressed.
- The client can request the representation of a resource at the server. Such a representation is typically a document, which can be in the format of HTML, XML or JSON.
- The server holds no states of the clients. The requests have to include all the necessary information to perform the request.
- Responses can be defined as cacheable or non cacheable, which leads to an improved performance between client and server.

The Qype API enables to send HTTP request. Responses are received in the format of XML or JSON (Java Script Object Notation) [13], which format type will be returned, can be specified. To perform server requests different operations are facilitated by the Hypertext Transfer Protocol (HTTP):

- POST: add a new entry to a resource
- GET: request a resource at the server
- PUT: create or edit a resource
- DELETE: delete a resource

Authentication

To access the Qype API via an own application an authentication is necessary. There are two types of authentication, read-only and read-write. The read-only authentication is done via the consumer key, which is sent as parameter in every request.

```
http://api.qype.com/v1/places/67370?consumer_key=mrOwJVwplDKzDxnmqLp6g
```

Listing 5.8: Authentication via consumer key

To get a read-write access to the Qype API, the authentication has to be done using the OAuth Core 1.0a protocol. The read-write authentication process is shown in Figure 5.13.

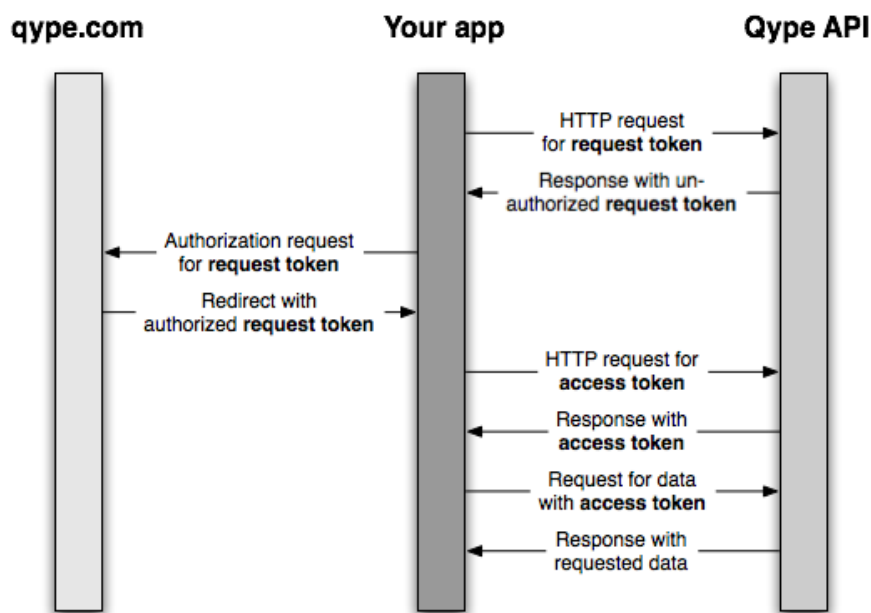


Figure 5.13: OAuth authentication process [62]

Resources

Inside the Qype API every object is modeled as resource with an URI. Throughout the unique identifier the object can be addressed and modified via the HTTP operations. There are two different types of resources:

- member resources: correspond to a single object
- collection resources: contain a list of resources within the collection.

| |
|---|
| http://api.qype.com/v1/places – Returns a list of places (collection) |
| http://api.qype.com/v1/places/42 – Returns a single place (member) |

Listing 5.9: Types of resources

In the following table all defined resources of the Qype API are listed:

| Resource | Description | Relevance |
|----------------|---|-----------|
| Places | Main resource type of Qype | X |
| Reviews | Reviews generated by users | X |
| Checkins | Show recent visitors of the place | X |
| Assets | Photos belonging to a place added by users | X |
| Users | People creating content | |
| Locators | Physical locations | X |
| Categories | Category of places | X |
| Tags | Tagged reviews and places by users | |
| Coupons | Special offers of a place | X |
| Positions | Positions defined by latitude / longitude coordinates | |
| BoundingBoxes | Rectangular area defined by two pairs of coordinates | |
| ContactRequest | Incoming and outgoing contact requests for a user | |
| Contacts | Friends of a user | |
| Badges | Awards of a user | |
| Events | Upcoming events | |

Table 5.1: Defined resources within Qype API

Resource representation

In the following section the focus will be on those resource types, that are relevant to the elaboration of a unified POI schema. Within the Qype API every resource has a specific format, which is a custom XML format. The Qype XML format is based on the following paradigms:

- All resources have a unique ID
- All resources containing datetime information are formatted in style of the Atom Syndication Format [46]

- Relationships between resources are represented by link elements, the type of interconnection is specified via the *rel* attribute
- All resources have a self-referencing link element that points to the API URL for this resource
- All resources have a link element that points to the corresponding URL on www.qype.com for this resource

Resources are represented in the format of XML or JSON. Which format is used can be defined by the URI ending .xml or .json.

Place The main and most important resource used by Qype is *Place*. For detailed description of the elements of the resource see Appendix B.1.

Reviews User ratings and reviews of places are represented by the resource *Reviews*. For detailed description of the elements of the resource see Appendix B.2.

Checkins The resource *Checkins* is representing a timestamp for users, who are visiting a place. For detailed description of the elements of the resource see Appendix B.3.

Assets The resource *Assets* represents pictures, which are uploaded by users. Every picture refers to a place. For detailed description of the elements of the resource see Appendix B.4.

Locators The resource *Locators* represents physical locations in the world. Each place refers to a locator. For detailed description of the elements of the resource see Appendix B.5.

Categories Places can be related to categories, which are represented by the resource *Categories*. For detailed description of the elements of the resource see Appendix B.6.

Coupon Special offers provided by places are represented by the resource *Coupon*. For detailed description of the elements of the resource see Appendix B.7.

Data availability

Qype is a growing online portal focused on service providers in Europe. The portal has more than 22 million visits per month and 1 million active members rating and writing critics. Qype.com enfolds more than 13 million places and has more than 2 million user generated recommendation and review entries. It is the largest online reviewing portal in Europe.

Trust and security issues

To use the Qype API a registration has to be done. Qype will then provide an API key. For a read-only access, the API key has to be sent with all queries. A read-write access requires an authentication via the OAuth Core 1.0a protocol. The Qype API key has to be kept confidential and not shared with others.

Policy guidelines

Qype content can be integrated in own applications. The applications must include a text link to Qype.com and the logo of Qype. By using the Qype API respectively Qype content the following obligations must be met [66]:

- regular update of all information (at least once per week)
- referencing the source of the content (Qype and also the name of the member, who published the content)
- linking the content with Qype.com

Third party examples

BMW ConnectedDrive enables access to Qype content during the drive [64]. The driver can read or hear user generated reviews of points of interest.

GoMio is an online booking platform for hostels [32] all over the world. With local restaurant and attraction reviews from Qype, GoMio supports the user when it comes to planning what to do.

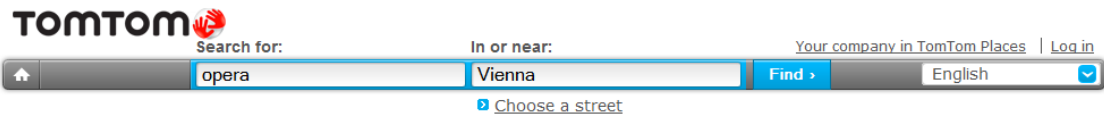
5.10 TomTom

TomTom was founded in 1991 under the name of Palmtop by Peter-Frans Pauwels and Pieter Geelen in Amsterdam [73]. Since 2001 the company operates under the name of TomTom and is focused on the development of applications for mobile navigation. In Europe TomTom is market leader and internationally they hold a very strong position on the market. TomTom is manufacturer of devices and producer of applications and services in the field of navigation.

TomTom enhances their solutions with point of interest data. They are using the world biggest location database provided by Google via Google Maps. With the TomTom LocalSearch the user can look for POIs on the navigation device, smart phone or via the website. The search is performed directly via the Google Maps database, what ensures a high topicality of data. With TomTom Places (see Figure 5.14) they provide an online search service for points of interest.

Application Programming Interface

TomTom is not providing an API to interact with their applications or point of interest data as a third party. They only provide an API (TomTom WEBFLEET) for a business application in the area of fleet management [72] and a SDK (software developers kit) for TomTomNavigator.



opera in Vienna

Categories

- Hotels (2)
- Music venues (1)
- Gifts (1)
- Opticians (1)
- Restaurants (1)

Distance From center Vienna:

- within 1 km (5)
- within 2 km (6)

Street

- Kärntner Straße (2)
- Opernring (2)
- Augustinerstraße (1)
- Schikanedergasse (1)

All results [Relevance](#) | [Distance](#) | [Alphabet](#)

- 1 **Opera**
Augustinerstraße 12, 1010 1. Bezirk, Innere Stadt, Österreich
Category: Restaurants
- 2 **Arcadia Opera Shop**
Kärntner Straße 41, 1010 1. Bezirk, Innere Stadt, Österreich
Category: Gifts
- 3 **Opera Suites**
Kärntner Straße 47, 1010 1. Bezirk, Innere Stadt, Österreich
Category: Hotels
- 4 **Optic Opera**
Opernring 4/2, 1010 1. Bezirk, Innere Stadt, Österreich
Category: Opticians
- 5 **Hotel Carlton Opera**
Schikanedergasse 4, 1040 Wien, Österreich
Category: Hotels
- 6 **Wiener Staatsoper**
Opernring 2, 1010 Wien, Österreich
Category: Music venues
Specialism: Opera

Showing 1 to 6 of 6 results

Map showing the locations of the search results in Vienna. The map includes street names like Opernring, Kärntner Straße, and Naschmarkt. A 'Show' button with checkboxes for 'ATM's' and 'Parking' is visible at the bottom of the map area.

Figure 5.14: TomTom Places

The SDK enables access to a TomTom navigation device through a dynamic link library or an ActiveX control, by means of these functions users can add their own point of interest data. The use of the SDK is applicable only in combination with a TomTom navigation device.

Data availability

Point of interest data is only available via the TomTom devices and services. TomTom uses two types of POI sets, depending on the device the user has. Newer navigation devices can access the Google Maps database for searching points of interest online, whereas older devices access the TomTom internal POI set. The set provided by Google is much more comprehensive than the internal set and also extended with user ratings and additional information such as opening hours, etc.

Trust and security issues

Due to the fact that no content API is available for third parties, the trust and security aspects can not be evaluated.

Policy guidelines

The terms and conditions [11] of TomTom grant the user access to the TomTom content for personal and private use only. It is not allowed to use the TomTom content with more than one computer and navigation system, what implies prohibition of dissemination to third parties. The TomTom license is non-exclusive and non-transferable. The prohibition of renting, lending, public presentation, performance or broadcasting of TomTom content is also defined in the terms and conditions. TomTom does not claim any ownership on the user generated content.

Third party examples

TomTom content is only available for TomTom devices and services, therefore the applicability is restricted. The provided API for fleet management is a business application used by business clients.

5.11 Twitter

Twitter was founded in 2006 and is a platform for microblogging [84]. Short text messages (tweets) with a maximum length of 140 characters can be published in real time. All users following the stream will see the text message. Twitter provides an easy possibility to spread real time information, opinions, news, etc. Furthermore, a discussion regarding a tweet among users is possible.

Various eTourism online providers are using Twitter to promote their region, product, country, etc. They use the communication platform as an easy and unconstrained connection to the user and potential customer. For points of interest a dedicated Twitter channel enriches the data basis. Static point of interest data can be extended with dynamic data out of Twitter, which increases data topicality. Figure 5.15 shows a list of various eTourism users, which are returned by the search request '*tourism*'.

Application Programming Interface

Twitter provides three different types of APIs, a Search API, a Streaming API and a RESTful API [84]. Via those APIs various data like tweets, timelines, users, etc. can be requested and integrated in own applications. The three APIs will be discussed in the following.

Search API

The Search API enables the search of recent tweets with some limitations. The API returns only tweets, which were generated in the last week. It is not possible to search for older tweets. The Search API focuses on relevance, whereas the Streaming API concentrates on completeness of the tweets.

The number of requests addressing the Search API is not limited. Limitations exist only in respect to complexity and frequency of search requests. The determination of these limits is carried over the IP address, because the search requests are performed without authentication.



Figure 5.15: Twitter Tourism streams

The exact limitation numbers are not made public, but they are higher than the RESTful API limitation rate, which is 150 requests per hour.

The resource URL for a request is <http://search.twitter.com/search.format>. The response can be in JSON format or ATOM Syndication Format. The ATOM Syndication Format is a standard used for web feeds and based on XML. Such web feeds include content which is updated frequently.

Different query parameters can be attached to the URL, whereas only the parameter *q* is required, any further parameters are optional.

- *q*: Search parameter e.g. @austriatourism, @TourismMalaysia etc.
- *callback*: Only available for JSON. If supplied, the response will use the JSONP format with a callback of the given name.
- *geocode*: Returns tweets created by users located at a given latitude and longitude within a defined radius. The units for the radius have to be defined as *mi* (miles) or *km* (kilometers).

- lang: Only tweets in the defined language will be returned.
- local: Specifies the language of the sent request.
- page: The number of the page to be returned (maximum results are approximately 1500).
- result_type: Defines the type of search results.
 - mixed: popular and real time results
 - recent: only most recent results
 - popular: only most popular results
- rpp: The number of tweets to return per page, up to a max of 100.
- show_user: Defines, if the user of the tweets is shown.
- until: Returns tweets generated before the given date.
- since_id: Returns results with an ID greater than the specified ID.
- include_entities: Defines, if metadata about the tweet in a discreet structure is included in the result.

Listing 5.10 shows an example of a Search API request, where tweets from Kitzbühel Tourismus by the user *@kitzbuehel* are requested. An extract of the returned tweets in JSON format is shown in Listing 5.11.

```
http://search.twitter.com/search.json?q=from:kitzbuehel&include_entities=true;result_type=recent
```

Listing 5.10: Twitter Search API request

```
{
  "completed_in": 0.188,
  "max_id": 145809711080554500,
  "max_id_str": "145809711080554496",
  "page": 1,
  "query": "from%3Akitzbuehel",
  "refresh_url": "?since_id=145809711080554496&q=from%3Akitzbuehel",
  "results": [
    {
      "created_at": "Sun, 11 Dec 2011 10:18:39 +0000",
      "from_user": "kitzbuehel",
      "from_user_id": 17337939,
      "from_user_id_str": "17337939",
      "from_user_name": "Kitzbhel Tourismus",
      "geo": null,
      "id": 145809711080554500,
      "id_str": "145809711080554496",
      "iso_language_code": "de",
      "metadata": {
```



```

        "result_type": "recent"
    },
    "profile_image_url": "http://a3.twimg.com/profile_images/481452823/gams_normal.jpg",
    "source": "&lt;a href=&quot;http://twitter.com/&quot;&gt;web&lt;/a&gt;",
    "text": "Advent, Advent die 3. Kerze brennt! http://t.co/k5eO2tIR http://t.co/iRbj64LE",
    "to_user": null,
    "to_user_id": null,
    "to_user_id_str": null,
    "to_user_name": null
},
{
    "created_at": "Sat, 10 Dec 2011 14:01:19 +0000",
    "from_user": "kitzbuehel",
    "from_user_id": 17337939,
    "from_user_id_str": "17337939",
    "from_user_name": "Kitzbhel Tourismus",
    ...
}

```

Listing 5.11: Twitter Search API response

Streaming API

The Streaming API enables access to various subsets of public and protected Twitter data. The concept of the API consists of three parts, the Streaming API itself, a User Stream and a Site Stream. In contrast to the Search API the Streaming API does not filter or rank tweets. All tweets matching the query parameter will be returned. The Streaming API can be seen as a super set of the Search API. The authentication to use the Streaming API is done via HTTP Basic Authentication or OAuth. The response format is JSON.

The User Stream is used to create tweets for a specific user, therefore an authentication via OAuth is required. The Site Stream is available in a beta version and enables the multiplexing of multiple user streams.

RESTful API

The RESTful API provides access to most twitter resources. Several methods require an authentication via OAuth, some can be accessed without authentication. There are various supported response formats: XML, JSON, RSS and the Atom Syndication format. The access is limited depending on, if the request is authenticated or not. The limit of unauthenticated requests is 150 per hour. OAuth requests are limited to the number of 350 per hour.

The API documentation [81] comprises all resources, which can be accessed via the RESTful API. There, all methods, parameters and requirements are described in detail. Following resources can be invoked by various methods:

- Timelines: collection of tweets
- Tweets: 140 character messages with additional associated metadata

- Search: find relevant tweets
- Direct Messages: messages between two users
- Friends & Followers: one-way and mutual following relationships
- Users: author of tweets
- Suggested Users: categorized users that others may be interested to follow
- Favorites: favorite tweets
- Lists: collections of tweets from a list of users
- Accounts: configuration settings for users
- Notification: SMS-based notifications for a user
- Saved Searches: save references to search criteria for reuse
- Local Trends: content specific to certain geographic area
- Places & Geo: attach location data to tweets
- Trends: explore trends on Twitter
- Block: block and unblock other users
- Spam Reporting: report user accounts as spam accounts
- OAuth: authentication
- Help: assistance in working and debugging with the API
- Legal: address legal issues

Data availability

Regarding point of interest data, the resources Tweets, Timelines, Local Trends and Places & Geo have the most relevance. A certain POI could be enriched with single tweets, streams or information published for a geographic area, which includes the POI. If the POI represents a shop, restaurant, another service center or company, it is feasible, that this POI has its own twitter channel. Available data can be accessed via the different Twitter APIs.

Trust and security issues

The secure access to the APIs is achieved by different security concepts. The Search API can be unauthenticated, whereas the RESTful API and the Streaming API require a registration and an authentication via HTTP Basic Authentication respectively OAuth. An access key can be requested on the Twitter website, whereat a Twitter account is postulated.

Throughout the limitation rates for requests Twitter reduces the risk of overload and abuse. Accounts, which exceed the limitation rates, will receive an error message as consequence of a further request. After the time limit, the API is available again.

Policy guidelines

Via the developer rules and the display guidelines Twitter regulates the use of the APIs as well as the content in detail. In general, the access to the APIs is free. It is prohibited to edit or revise user-generated content retrieved through the Twitter API. The user of a tweet has to be identify and published. If an application violates the Twitter rules it can be suspended or terminated by Twitter. User privacy and the intellectual property rights must be respected at any time.

By integrating Twitter content in an own application the display guidelines must be obeyed. The website therefore provides a detailed description [82]. Figure 5.16 shows a graphical template for the content presentation.

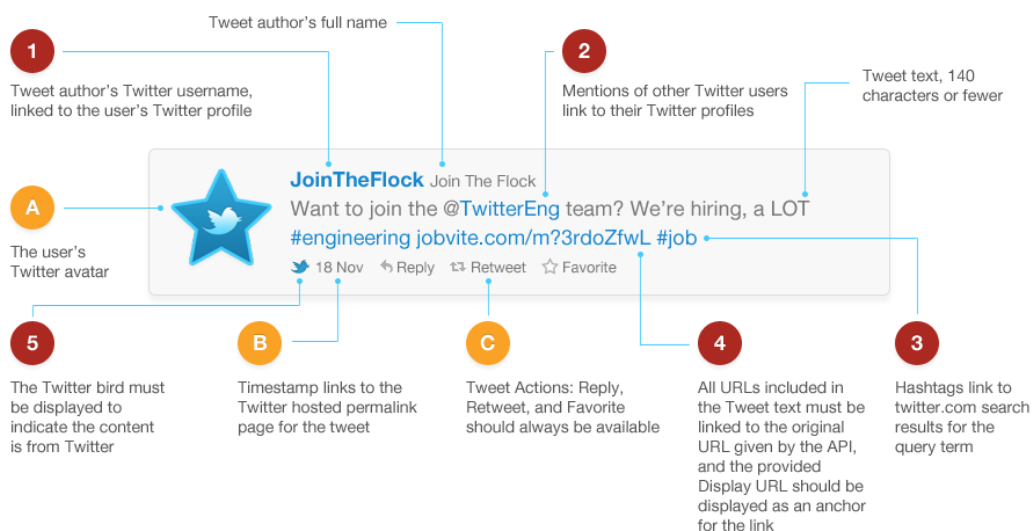


Figure 5.16: Twitter Display Guidelines

Third party examples

Twitter content is integrated via the API in a multitude of web services and applications. TweetDeck by Twitter is an application supporting power users with more flexibility and insights [83]. Another example integrating Twitter channels is Technorati [70]. Technorati is a real time search engine for weblogs. HootSuite is a social media dashboard to manage social networks, where also the twitter account can be managed [37].

Furthermore, Twitter finds application in customer relationship management (CRM) systems, like Salesforce [68]. There Twitter is integrated as direct communication channel to the customers.

5.12 ViaMichelin

ViaMichelin is an affiliate of the Michelin group and was founded in 2001. The head office is located in Frankfurt am Main, Germany. The company offers online services in the area of traveling, mobility, booking, navigation, etc., which are provided via the web, smart phones or navigation devices. The website is available in eight languages [49].

Application Programming Interface

ViaMichelin offers three APIs, which have different scopes of application (web, mobile, business applications). All APIs can be evaluated and tested via a trial account. This account can be requested on the website. Subsequently the API authentication can be done via the submitted API key. Trial accounts have a total limit of 1000 accesses.

ViaMichelin REST API

The ViaMichelin REST API is applicable for web and mobile platforms. It comprises geocoding services, map integration, route calculation, address verification and point of interest search. Compatibility with common web protocols (HTTP, XML, JSON, etc.) and JavaScript frameworks is given. The REST API is especially suited to mobile applications and can be integrated easily.

To evaluate the REST API a trial account can be opened. This account will be valid for 45 days, after this test period the provided evaluation key expires. The trial account is limited to 1000 accesses. The number of requests for a conventional account is limited to:

- 10000 requests to view maps and hotels
- 1000 requests to calculate routes and geocoding

The usage fees of the REST API will be calculated based on consumption. A local customer support is available in five European countries.

POI structure With the proximity search the REST API offers the possibility to operate with point of interest data. The response of a POI search request will be in XML or JSON format. A detailed description of the response structure is included in Appendix C.1.

Proximity search The API supports several request types for points of interest. POI data is hosted at ViaMichelin in a database. To access the proximity search the API key must be the value of authkeys parameter in all API request. An overview of the proximity search potential is given in the table 5.2.

| Request | Description | Response |
|--------------------|--|--------------------------------|
| FindPOI | closest POI from a specific location (crow flies based calculation) | sorted POI list (asc distance) |
| FindPOIByRoad | closest POI from a specific location (road-based calculation) | sorted POI list (asc distance) |
| FindInputPOI | closest dynamic POI from a specific place (crow flies based calculation); POIs are given as parameters | sorted POI list (asc distance) |
| FindInputPOIByRoad | closest dynamic POI in relation to a specific place (crow flies based calculation); POIs are given as parameters | sorted POI list (asc distance) |
| FindPOIByCriteria | POI matching criteria given as parameters | sorted POI list (asc distance) |
| findPOINearRoute | POI along a route matching criteria given as parameters | sorted POI list (asc distance) |

Table 5.2: ViaMichelin REST API *Proximity Search*

ViaMichelin JavaScript API

The JavaScript API is enabling the integration of ViaMichelin content and services on own websites. It comprises geocoding, address verification, search for hotels, GPS format conversions and route calculations. Interactive and navigable maps can be integrated, which are optimized for the specific usage (web, smart phone, etc.). The maps can be enriched with point of interest data (see Figure 5.17).

The JavaScript API is supporting the concept of AJAX (Asynchronous JavaScript and XML) and is compatible with JavaScript frameworks. To test the JavaScript API a trial account can be opened. This account will be valid for 45 days, after this test period the provided evaluation key expires. The trial account is limited to 1000 accesses. The number of requests for a conventional account is limited to:

- 10000 requests to view maps and hotels
- 1000 requests to calculate routes and geocoding

Input address (search center)

Address :

Postal code :

City :

Country :

GPS Export format :

Results

| | | | | |
|---|-----------------------------|------|-------|--|
| 1 | Garage Peugeot Dacourt | 0m | ND257 | |
| 2 | Garage Volkswagen Desailly | 0m | ND258 | |
| 3 | Garage Opel Sagnol | 197m | ND253 | |
| 4 | Garage Fiat Coupet | 202m | ND251 | |
| 5 | Garage Opel Vandebeukelaere | 501m | ND252 | |

Figure 5.17: ViaMichelin JavaScript API example [48]

The usage fees of the JavaScript API will be calculated per request. A local customer support is available in five European countries.

ViaMichelin SOAP API

The SOAP API is focused on business management applications. Geocoding services, address verification, route calculations and map integration, these are some of the services offered by the ViaMichelin SOAP API.

A SDK (Software Development Kit) is available and can be downloaded on the website. To test the integration of the SOAP API a trial account can be opened. This account will be valid for 45 days, after this test period the provided evaluation key expires. The trial account is limited to 1000 accesses. The usage fees of the ViaMichelin SOAP API will be calculated based on consumption. A local customer support is available in five European countries.

Data availability

ViaMichelin records more than 27.400 points of interest and more than 50.000 hotels or restaurants [49]. The database includes more than 10 million kilometers of European roads, including detailed description of streets. The website has more than 35 million hits per month. Through the provided APIs' point of interest data, maps and other services can be accessed.

Trust and security issues

Access to the APIs is only possible with an API key. Such a key can be request on the website. Every API (REST, JavaScript, SOAP) requires a separate API key. The access to the API has to be on HTTPS protocol. There are three different authentication modes [49]:

- server mode, if requests to the ViaMichelin REST API are issued by a web server
- client mode, if requests are issued by a client station or a mobile phone
- hybrid mode, if the trade is initiated by a web server and then managed directly by the client station or mobile phone

Policy guidelines

ViaMichelin prohibits the extraction, storage, copying, archiving or printing of ViaMichelin content. Geocoding functions and their results should only be used temporarily for the duration of the authorized User session to locate the corresponding address on a ViaMichelin map. Copyright notices and statements of ViaMichelin have to be displayed and not infringed.

Third party examples

Nespresso illustrates the distribution network via web services of ViaMichelin, including a map [51]. MMA, a French insurance company, developed a web application based on the ViaMichelin API that allows customers to calculate the cost of a car insurance based on various criteria [50].

eTourism Evaluation Results

6.1 Evaluation results

6.1.1 Comparative Matrix of eTourism evaluation results

The matrix shown in Figure 6.1 compares the evaluated eTourism online providers based on various attributes. Each column represents one provider. The value of each attribute was added in the respective row. The detailed description of each eTourism online provider, in form of a fact sheet, is provided after the comparative matrix.

| Evaluation attribute | Falk | Flickr | Foursquare | Garmin | Lonely Planet | Panoramio | Qype | TomTom | Twitter | ViaMichelin |
|----------------------|---------------------|--------|------------|--------|---------------|-----------|----------|--------|----------|-------------|
| API | Available | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 3 | 3 |
| | Open access | Yes | Yes | No | Yes | Yes | Yes | No | Yes | Yes |
| | Account Required | Yes | Yes | Yes | Yes | No | Yes | No | Yes | Yes |
| | API Key Required | Yes | Yes | Yes | Yes | No | Yes | No | Yes | Yes |
| Usage | Evaluation period | No | No | Yes | No | No | No | No | No | Yes |
| | Fees non-commercial | No | No | No | No | No | No | No | No | No |
| | Fees commercial | Yes | possible | Yes | No | possible | possible | No | possible | Yes |
| | Limits | No | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes |
| Protocols | REST | X | X | X | X | X | X | X | X | X |
| | JavaScript | X | | X | | X | | | | X |
| | SOAP | X | | | | | | | | X |
| Data Formats | XML | | | X | X | | X | | | X |
| | ATOM | | | | | | | | X | |
| | JSON | | X | | | X | | | X | |
| | HTML | X | | | | X | | | | X |
| Security | Anonymous | | | | X | X | | X | X | |
| | HTTP/HTTPS | X | | X | X | | X | | X | X |
| | OAuth 1.0 | | | | X | | X | | X | |
| | OAuth 2.0 | | | X | | | | | | |
| Support | Manual | X | X | X | X | X | X | X | X | X |
| | Api Reference | X | X | X | X | X | X | X | X | X |
| | API Forum | | X | X | X | X | X | | X | |
| | API Blog | | X | X | X | X | X | | X | |
| | Developer Support | | | X | X | | X | | X | X |

Figure 6.1: Results

6.1.2 Fact Sheets of eTourism Online Providers

Following, each evaluated eTourism online provider is described in detail, via a fact sheet.


| FACT SHEET | |  |
|---------------------|-------------------------|---|
| API | Name | Falk Map API |
| | Access | Open |
| | Account Required | Yes |
| | API Key Required | Yes |
| Usage | Evaluation period | No |
| | Fees (non-commercial) | No |
| | Fees (commercial) | No |
| | Limits | No |
| Protocols | REST | |
| | JavaScript | x |
| | SOAP | |
| Data Formats | XML | |
| | JSON | |
| | HTML | x |
| Security | Authentication Model | API Key |
| | Authentication Protocol | HTTP |
| Content | Maps | x |
| Support | Manual | http://maps.falk.de/doku/index.htm |
| | API Reference | http://maps.falk.de/doku/js_api/index.html |
| | API Forum | |
| | API Blog | |
| | Developer Support | |

Figure 6.2: Fact Sheet Falk


| FACT SHEET | |  |
|---------------------|-------------------------|---|
| API | Name | Flickr API |
| | Access | Open |
| | Account Required | Yes |
| | API Key Required | Yes |
| Usage | Evaluation period | No |
| | Fees (non-commercial) | No |
| | Fees (commercial) | individual agreement |
| | Limits | 3600 requests per hour |
| Protocols | REST | x |
| | JavaScript | |
| | SOAP | x |
| Data Formats | XML | x |
| | JSON | x |
| | HTML | |
| Security | Authentication Model | Anonymous, API Key |
| | Authentication Protocol | OAuth 1.0 |
| Content | activity | x |
| | auth | x |
| | auth.oauth | x |
| | blogs | x |
| | collections | x |
| | commons | x |
| | contacts | x |
| | favorites | x |
| | galleries | x |
| | groups | x |
| | interestingness | x |
| | machinetags | x |
| | people | x |
| | photos | x |
| | photosets | x |
| | places | x |
| | prefs | x |
| | push | x |
| reflection | x | |
| stats | x | |
| tags | x | |
| test | x | |
| url | x | |
| Support | Manual | http://www.flickr.com/services/api/ |
| | API Reference | http://www.flickr.com/services/api/ |
| | API Forum | http://tech.groups.yahoo.com/group/yws-flickr/ |
| | API Blog | http://code.flickr.com/ |
| | Developer Support | |

Figure 6.3: Fact Sheet Flickr


| FACT SHEET | |  |
|---------------------|-------------------------|---|
| API | Name | Core API Real-time API |
| | Access | Open |
| | Account Required | Yes |
| | API Key Required | Yes |
| Usage | Evaluation period | No |
| | Fees (non-commercial) | No |
| | Fees (commercial) | individual agreement |
| | Limits | 500 requests per hour |
| Protocols | REST | x |
| | JavaScript | |
| | SOAP | |
| Data Formats | XML | |
| | JSON | x |
| | HTML | |
| Security | Authentication Model | API Key |
| | Authentication Protocol | OAuth 2.0 |
| Content | users | x |
| | venues | x |
| | checkins | x |
| | tips | x |
| | lists | x |
| | updates | x |
| | photos | x |
| | settings | x |
| | specials | x |
| | campaigns | x |
| events | x | |
| Support | Manual | https://developer.foursquare.com/ |
| | API Reference | https://developer.foursquare.com/ |
| | API Forum | http://stackoverflow.com/questions/tagged/foursquare |
| | API Blog | http://blog.foursquare.com/ |
| | Developer Support | api@foursquare.com |

Figure 6.4: Fact Sheet Foursquare


| FACT SHEET | |  | |
|---------------------|-------------------------|---|--|
| API | Name | Device Interface SDK Content API | |
| | Access | with Garmin devices | |
| | Account Required | Yes | |
| | API Key Required | Yes | |
| Usage | Evaluation period | 90 days | |
| | Fees (non-commercial) | after 90 days: Content Toolkit for \$500 | |
| | Fees (commercial) | after 90 days: Content Toolkit for \$500 | |
| | Limits | 500 requests per hour | |
| Protocols | REST | | |
| | JavaScript | x | |
| | SOAP | | |
| Data Formats | XML | x | |
| | JSON | | |
| | HTML | | |
| Security | Authentication Model | API Key | |
| | Authentication Protocol | HTTP | |
| Content | maps | x | |
| | points of interest | x | |
| Support | Manual | https://developer.foursquare.com/ | |
| | API Reference | https://developer.foursquare.com/ | |
| | API Forum | http://stackoverflow.com/questions/tagged/foursquare | |
| | API Blog | http://blog.foursquare.com/ | |
| | Developer Support | api@foursquare.com | |

Figure 6.5: Fact Sheet Garmin


| FACT SHEET | |  |
|---------------------|-------------------------|---|
| API | Name | Content API |
| | Access | open |
| | Account Required | Yes |
| | API Key Required | Yes |
| Usage | Evaluation period | No |
| | Fees (non-commercial) | No |
| | Fees (commercial) | No |
| | Limits | 5000 calls per day per API key and maximum 2 calls per second |
| Protocols | REST | x |
| | JavaScript | |
| | SOAP | |
| Data Formats | XML | x |
| | JSON | |
| | HTML | |
| Security | Authentication Model | Anonymous, API Key |
| | Authentication Protocol | HTTP, OAuth 1.0 |
| Content | categories | x |
| | points of interest | x |
| Support | Manual | http://developer.lplabs.com/index.php?title=Main_Page |
| | API Reference | http://developer.lplabs.com/index.php?title=Main_Page |
| | API Forum | http://groups.google.com/group/lonely-planet-developers |
| | API Blog | |
| | Developer Support | |

Figure 6.6: Fact Sheet Lonely Planet

| FACT SHEET | | Panoramio | |
|---------------------|-------------------------|---|--|
| API | Name | Widget API Content API | |
| | Access | open | |
| | Account Required | No | |
| | API Key Required | No | |
| Usage | Evaluation period | No | |
| | Fees (non-commercial) | No | |
| | Fees (commercial) | individual agreement | |
| | Limits | 100,000 queries per day | |
| Protocols | REST | x | |
| | JavaScript | x | |
| | SOAP | | |
| Data Formats | XML | x | |
| | JSON | x | |
| | HTML | | |
| Security | Authentication Model | Anonymus | |
| | Authentication Protocol | | |
| Content | pictures | x | |
| | tags | x | |
| Support | Manual | http://www.panoramio.com/api/data/api.html | |
| | API Reference | http://www.panoramio.com/api/data/api.html | |
| | API Forum | http://www.panoramio.com/forum | |
| | API Blog | | |
| | Developer Support | api@panoramio.com | |

Figure 6.7: Fact Sheet Panoramio


| FACT SHEET | |  |
|---------------------|-------------------------|---|
| API | Name | Content API |
| | Access | Open |
| | Account Required | Yes |
| | API Key Required | Yes |
| Usage | Evaluation period | No |
| | Fees (non-commercial) | No |
| | Fees (commercial) | individual agreement |
| | Limits | Standard Account: 200 requests per day Pro Account: unlimited |
| Protocols | REST | x |
| | JavaScript | |
| | SOAP | |
| Data Formats | XML | x |
| | JSON | |
| | HTML | |
| Security | Authentication Model | API Key |
| | Authentication Protocol | HTTP, OAuth 1.0 |
| Content | categories | x |
| | checkins | x |
| | coupons | x |
| | events | x |
| | pictures | x |
| | points of interest | x |
| | reviews | x |
| | tags | x |
| Support | Manual | http://apidocs.qype.com/ |
| | API Reference | http://apidocs.qype.com/resource_reference |
| | API Forum | http://groups.google.com/group/qype-api/topics |
| | API Blog | http://de.blog.qype.com/category/api/ |
| | Developer Support | |

Figure 6.8: Fact Sheet Qype


| FACT SHEET | | TomTom®  | |
|---------------------|-------------------------|---|--|
| API | Name | Navigator SDK | |
| | Access | with TomTom devices | |
| | Account Required | No | |
| | API Key Required | No | |
| Usage | Evaluation period | No | |
| | Fees (non-commercial) | No | |
| | Fees (commercial) | No | |
| | Limits | No | |
| Protocols | REST | | |
| | JavaScript | | |
| | SOAP | | |
| Data Formats | XML | | |
| | JSON | | |
| | HTML | | |
| Security | Authentication Model | | |
| | Authentication Protocol | | |
| Content | maps | x | |
| | | | |
| Support | Manual | http://www.tomtom.com/lib/doc/ttnavsdk3_manual.pdf | |
| | API Reference | | |
| | API Forum | | |
| | API Blog | | |
| | Developer Support | | |

Figure 6.9: Fact Sheet TomTom

| FACT SHEET | | twitter |
|--------------|-------------------------|---|
| API | Name | Search API Streaming API RESTful API |
| | Access | Open |
| | Account Required | Yes |
| | API Key Required | Yes |
| Usage | Evaluation period | No |
| | Fees (non-commercial) | No |
| | Fees (commercial) | individual agreement |
| | Limits | Search API: more than REST API requests per hour RESTful API: unauthenticated 150 requests per hour authenticated 350 per hour |
| Protocols | REST | x |
| | JavaScript | |
| | SOAP | |
| Data Formats | XML | |
| | JSON | x |
| | RSS | x |
| | ATOM | x |
| | HTML | |
| Security | Authentication Model | Anonymous, API Key |
| | Authentication Protocol | HTTP, OAuth 1.0 |
| Content | timelines | x |
| | tweets | x |
| | search | x |
| | direct messages | x |
| | friends & followers | x |
| | users | x |
| | favorites | x |
| | lists | x |
| | accounts | x |
| | notifications | x |
| | local trends | x |
| | places & geo | x |
| | trends | x |
| | block | x |
| | spam | x |
| Oauth | x | |
| Help | x | |
| Legal | x | |
| Support | Manual | https://dev.twitter.com/docs |
| | API Reference | https://dev.twitter.com/docs |
| | API Forum | https://dev.twitter.com/discussions |
| | API Blog | https://dev.twitter.com/blog |

Figure 6.10: Fact Sheet Twitter


| FACT SHEET | |  |
|---------------------|-------------------------|---|
| API | Name | REST API JavaScript API SOAP API |
| | Access | Open |
| | Account Required | Yes |
| | API Key Required | Yes |
| Usage | Evaluation period | 45 days |
| | Fees (non-commercial) | No |
| | Fees (commercial) | usage fees calculated per request |
| | Limits | Trial Account: 1000 accesses Conventional Account: 10000 requests (maps and hotels); 1000 requests (routes and geocoding) |
| Protocols | REST | x |
| | JavaScript | x |
| | SOAP | x |
| Data Formats | XML | x |
| | JSON | |
| | HTML | x |
| Security | Authentication Model | API Key |
| | Authentication Protocol | HTTPS |
| Content | maps | x |
| | points of interest | x |
| Support | Manual | http://dev.viamichelin.com/ |
| | API Reference | http://dev.viamichelin.com/ |
| | API Forum | |
| | API Blog | |
| | Developer Support | http://business.viamichelin.com/contact |

Figure 6.11: Fact Sheet ViaMichelin

Prototypical Implementation

7.1 Introduction

The constructive methodological approach finds application in the implementation of the prototype. This application-oriented part of the thesis is matter of this chapter. The development of a standardized data schema for point of interest data is an essential part. Existing and established data formats are considered and taken into account for creating the extendable and reusable data schema. The concrete implementation of several available APIs is done via the prototype. During the implementation data reliability, topicality and accuracy are investigated and determined.

7.2 Purpose

The purpose of the prototype is the implementation of available data interfaces of eTourism online providers. Points of interest, including general information, ratings, pictures, etc., are used in miscellaneous areas. Navigation devices were one of the first fields of application for POI data, but today various other areas like eTourism, online booking, route planning, mobility applications etc. are operating with such data.

Currently there is a multitude of online providers in the tourism industry, who are mainly focused on own goals, policies, countries, unions and agencies. This entails, that no data standard regarding to point of interest data has enforced until now. GPX, a XML based format, is a standard supported by plenty of vendors. However, this format has not enforced completely. The missing standard is impeding reusability, exchange and comparison of tourism data.

The purpose of the thesis, which is achieved mainly by the prototype, is the development of a potential standardized format for point of interest data. This new format PXF (Point of Interest eXchange Format) is based on the supported formats and standards of the implemented APIs. Imported point of interest data from Qype and Flickr is transformed into the PXF format and stored in a database.

7.3 Scope

The prototype is implementing the APIs of Qype and Flickr. The main point of interest data source will be Qype, whereas Flickr constitutes the source of tags related to imported POIs. Therefore the provided data, functions and formats have been evaluated in detail. The prototype merges the two data sources, which means that information, pictures, tags, etc. regarding one point of interest is assembled to one entity.

7.4 Point of Interest eXchange Format - PXF

The new developed standardized format for points of interest is PXF, which is first introduced in the master thesis.

7.4.1 Underlying Formats

GPX (GPS Exchange Format) is based on XML. Due the fact that XML is an open standard, GPX is used by many providers. Even though GPX could not prevail as a standard format for point of interest data. The degree of diffusion of the format was a criteria for selecting it as an underlying format for the new developed schema. Also the XML response format of the Qype API provided a basis. The evaluation of the current dominant formats and used protocols and technologies leads to the potential standardized format for points of interest - PXF.

7.4.2 XML Schema Definition

The developed standardized PXF (Point of Interest eXchange Format) format is based on a XML structure. The XSD (XML Schema Definition) of the PXF format is explained in this section, using the XSD (Listing 7.1) and a fictive example (Listing 7.2) to illustrate the application of the XSD in a concrete application. Figure 7.1 gives an overview of the structure and the components of the PXF schema. The individual parts of the schema are exposed in the following.

Header

The first line in a PXF-File has to be the XML declaration. This declaration is induced by an angle bracket followed by *?xml*. The attribute *version* is set to 1.0, which is the common XML standard, supported by many parsers. The attribute *encoding* defines the character encoding used for the file. UTF-8 is the most common international encoding for Unicode characters, specified in the ISO 10646 norm. The XML declaration is completed again by an angle bracket (see Listing 7.2).

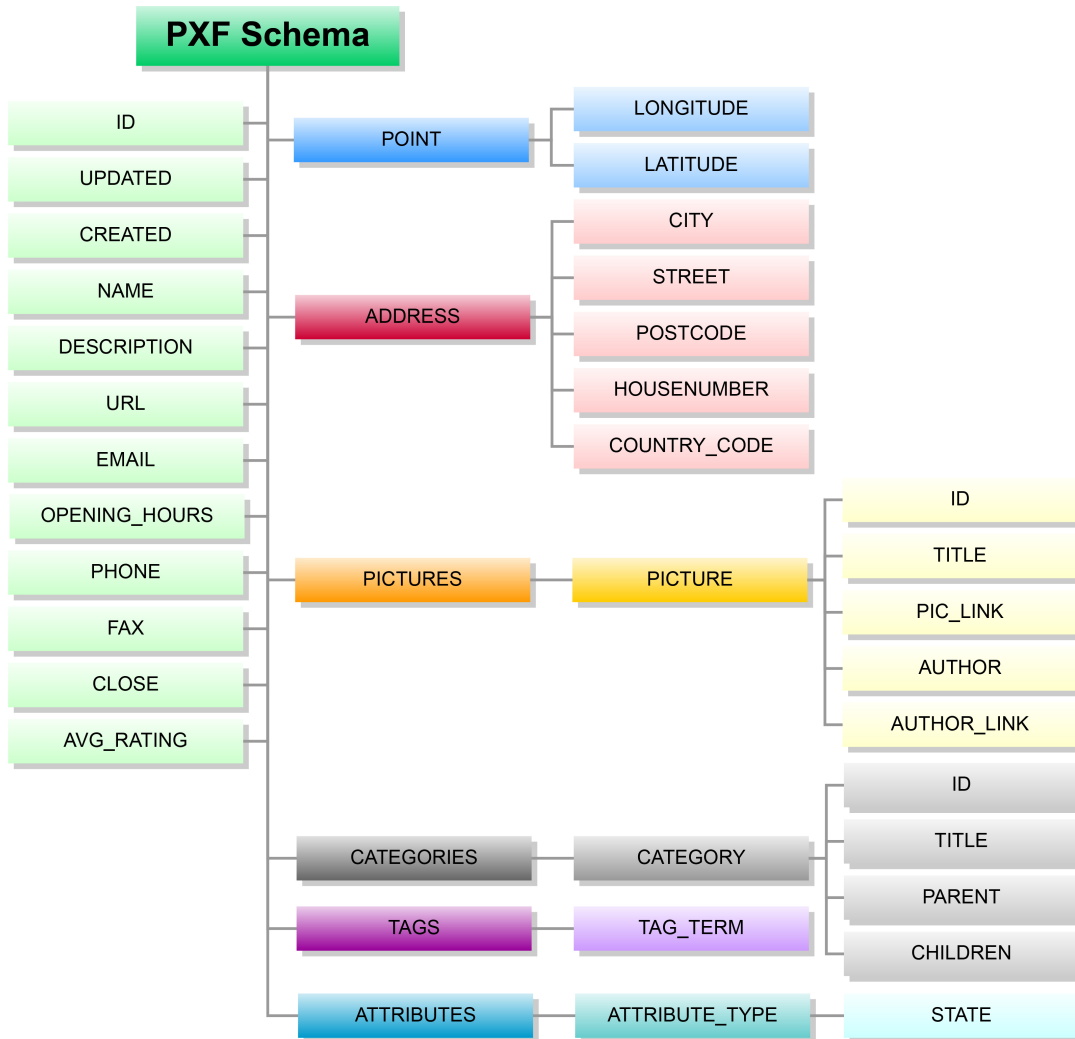


Figure 7.1: PXF Overview

POI Tag

The description of a point of interest is nested in the tag **<poi>**. All information concerning the POI is contained within this tag. The PXF schema includes single elements and nested elements. Nested elements contain other elements and are explained in separate subsections.

- **<id>** unique identifier for the POI
- **<updated>** date and time this POI was last updated
- **<created>** date and time this POI was created
- **<name>** name of the POI
- **<description>** longer description of the POI
- **<url>** web address of the POI
- **<email>** email address of the POI
- **<phone>** phone number of the POI
- **<fax>** fax number of the POI
- **<closed>** true, if the place is closed (independent of the general opening hours)
- **<avg_ranking>** average user ranking of the POI (Five-Star rating)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- W3C Schema erstellt mit XMLSpy v2012 sp1 (http://www.altova.com) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="id" type="xs:int"/>
  <xs:element name="poi">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="id"/>
        <xs:element ref="updated"/>
        <xs:element ref="created"/>
        <xs:element ref="name"/>
        <xs:element ref="description"/>
        <xs:element ref="url"/>
        <xs:element ref="email"/>
        <xs:element ref="phone"/>
        <xs:element ref="fax"/>
        <xs:element ref="closed"/>
        <xs:element ref="avg_rating"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="name" type="xs:string"/>
</xs:schema>
```



```

<xs:element name="description" type="xs:string"/>
<xs:element name="url" type="xs:string"/>
<xs:element name="created" type="xs:dateTime"/>
<xs:element name="updated" type="xs:dateTime"/>
<xs:element name="phone" type="xs:int"/>
<xs:element name="fax" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
<xs:element name="closed" type="xs:boolean"/>
<xs:simpleType name="rating">
  <xs:restriction base="xs:string">
    <xs:enumeration value="*"/>
    <xs:enumeration value="**"/>
    <xs:enumeration value="***/>
    <xs:enumeration value="****"/>
    <xs:enumeration value="*****"/>
  </xs:restriction>
</xs:simpleType>
<xs:element name="avg_rating" type="rating"/>
</xs:schema>

```

Listing 7.1: PXF POI XSD

```

<?xml version="1.0" encoding="UTF-8"?>
<poi>
  <id>123456</id>
  <updated>2011-02-01T12:00:00+01:00</updated>
  <created>2007-01-01T13:00:00+01:00</created>
  <name>My POI 1</name>
  <description>This is the longer description of my POI 1.</description>
  <url>www.mypoi1.com</name>
  <email>office@mypoi1.com</email>
  <phone>012342392</phone>
  <fax>01234120-2</fax>
  <closed>>false</closed>
  <avg_rating>****</avg_rating>
  <point>
    <address>
    <opening_hours>
    <pictures>
    <categories>
    <tags>
    <attributes>
  </point>
</poi>

```

Listing 7.2: PXF POI Example

Point

The Point tag contains the geographical position of the POI, which is defined by longitude and latitude.

```
<xs:element name="point">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="longitude"/>
      <xs:element ref="latitude"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="longitude" type="xs:decimal"/>
<xs:element name="latitude" type="xs:decimal"/>
```

Listing 7.3: PXF Point XSD

```
<point>
  <longitude>16.36915</longitude>
  <latitude>48.19927</latitude>
</point>
```

Listing 7.4: PXF Point Example

Address

The Address tag comprehends several other tags (city, postcode, street, housenumber, country code) for defining the address of the POI.

```
<xs:element name="address">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="city"/>
      <xs:element ref="postcode"/>
      <xs:element ref="street"/>
      <xs:element ref="housenumber"/>
      <xs:element ref="country_code"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="city" type="xs:string"/>
<xs:element name="postcode" type="xs:short"/>
<xs:element name="street" type="xs:string"/>
<xs:element name="housenumber" type="xs:short"/>
<xs:element name="country_code" type="xs:string"/>
```

Listing 7.5: PXF Address XSD

```

<address>
  <city>Vienna</city>
  <postcode>1010</postcode>
  <street>Rotenturmstrasse</street>
  <housenumber>12</housenumber>
  <country_code>AT</country_code>
</address>

```

Listing 7.6: PXF Address Example

Opening hours

Within this tag the opening hours, related to each weekday, are set.

```

<xs:element name="opening_hours">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="open_monday"/>
      <xs:element ref="open_tuesday"/>
      <xs:element ref="open_wednesday"/>
      <xs:element ref="open_thursday"/>
      <xs:element ref="open_friday"/>
      <xs:element ref="open_saturday"/>
      <xs:element ref="open_sunday"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="open_wednesday" type="xs:string"/>
<xs:element name="open_tuesday" type="xs:string"/>
<xs:element name="open_thursday" type="xs:string"/>
<xs:element name="open_sunday" type="xs:string"/>
<xs:element name="open_saturday" type="xs:string"/>
<xs:element name="open_monday" type="xs:string"/>
<xs:element name="open_friday" type="xs:string"/>

```

Listing 7.7: PXF Opening hours XSD

```

<opening_hours>
  <open_monday>08.00 – 20.00</open_monday>
  <open_tuesday>closed</open_tuesday>
  <open_wednesday>08.00 – 20.00</open_wednesday>
  <open_thursday>08.00 – 20.00</open_thursday>
  <open_friday>08.00 – 24.00</open_friday>
  <open_saturday>08.00 – 24.00</open_saturday>
  <open_sunday>12.00 – 20.00</open_sunday>
</opening_hours>

```

Listing 7.8: PXF Opening Hours Example

Pictures

The Pictures tag can include a list of various pictures. Each picture is nested within a picture tag. A picture has an ID, title, URL, an author and a reference to the author.

```
<xs:element name="pictures">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="picture" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="picture">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="title"/>
      <xs:element ref="pic_link"/>
      <xs:element ref="author"/>
      <xs:element ref="author_link" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="pic_link" type="xs:anyURI"/>
<xs:element name="author" type="xs:string"/>
<xs:element name="author_link" type="xs:anyURI"/>
```

Listing 7.9: PXF Pictures XSD

```
<pictures>
  <picture>
    <id>123</id>
    <title>Picture1</title>
    <pic_link>http://www.mypoi1.com/pictures/AT123456</pic_link>
    <author>User1</author>
    <author_link>http://www.mypoi1.com/User1</author_link>
  </picture>
  <picture>
    <id>456</id>
    <title>Picture 2</title>
    <pic_link>http://www.mypoi1.com/pictures/AT23456</pic_link>
    <author>User2</author>
    <author_link>http://www.mypoi1.com/User2</author_link>
  </picture>
</pictures>
```

Listing 7.10: PXF Pictures Example

Categories

To categorize points of interest a hierarchical category tree was implemented. A point of interest can belong to various categories. Each category is described within the category tag.

The id tag assigns the unique identifier of the category and the title tag includes the name of the category. The tree structure of the categories is represented via the parent and children tag, which are relating to the parent category and the potential successors.

```
<xs:element name="categories">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="category" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="category">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="id"/>
      <xs:element ref="title"/>
      <xs:element ref="parent"/>
      <xs:element ref="children"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="id" type="xs:int"/>
<xs:element name="title" type="xs:string"/>
<xs:element name="parent" type="xs:int"/>
<xs:element name="children" type="xs:string"/>
```

Listing 7.11: PXF Categories XSD

```
<categories>
  <category>
    <id>4002000</id>
    <title>Cafe</title>
    <parent>4</parent>
    <children>4002001,4002002,4002003,4002004,4002005,4002006,4002007</children>
  </category>
  <category>
    <id>4004007</id>
    <title>Crepe</title>
    <parent>4004000</parent>
    <children></children>
  </category>
</categories>
```

Listing 7.12: PXF Categories Example

The structure of the hierarchical category tree is shown in Figure 7.2. All categories have a common root element. Based on the previous survey 13 main categories are introduced. Regarding to point of interest data and the prototypical implementation some branches of the tree were specified in more detail. Consequently, the category tree should be supplemented and updated, based on the different fields of application of the schema.

Tags

A point of interest can be described by a multitude of words, named tags. Principally tags are user generated content and can be obtained by point of interest tags themselves or tags of a picture related to the POI. Tags are aggregated within the Tags tag. Each tag entry is set via the term attribute.

```
<xs:element name="tags">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="tag" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="tag">
  <xs:complexType>
    <xs:attribute name="term" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string"/>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Listing 7.13: PXF Tags XSD

```
<tags>
  <tag term="cafe"></tag>
  <tag term="cocktail"></tag>
  <tag term="cocktails"></tag>
  <tag term="breakfast"></tag>
  <tag term="nice"></tag>
  <tag term="restaurant"></tag>
  <tag term="good service"></tag>
</tags>
```

Listing 7.14: PXF Tags Example

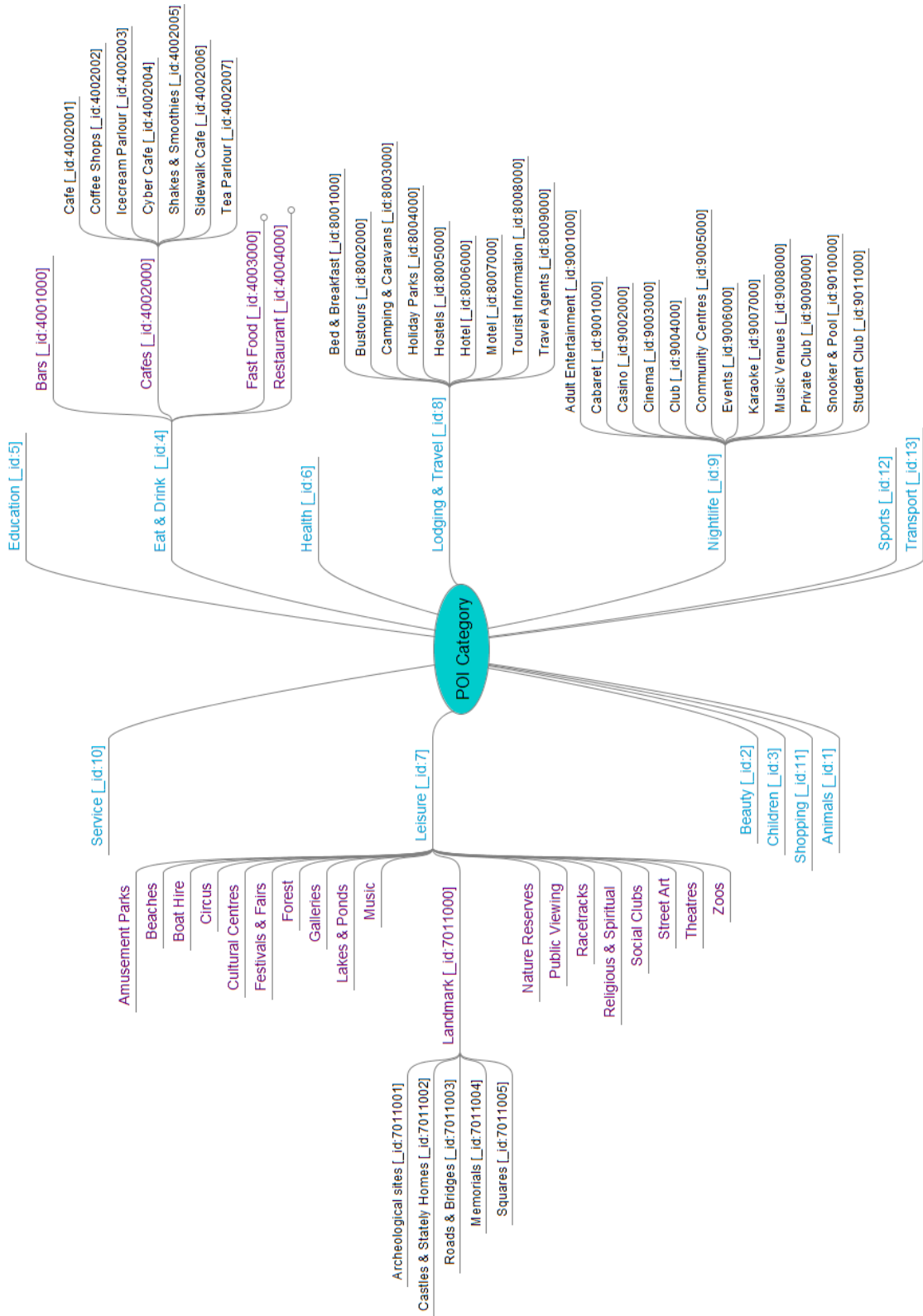


Figure 7.2: POI hierarchical category tree

Attributes

Based on the evaluation of various eTourism providers a list of attributes for points of interest was determined. The attribute list can be understood as a first draft for the new PXF schema. Consequently, this list should be supplemented and updated, based on the different areas of application of the schema.

Each attribute is contained in an attribute tag, where the type attribute defines the attribute name. The state tag determines the presence respectively the value (e.g. dresscode could be 'casual', 'business' etc.) of the stated attribute. For the prototypical implementation the list of attributes shown in Listing 7.15 is used.

```
<xs:element name="attributes">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="attribute" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="attribute">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="state"/>
    </xs:sequence>
    <xs:attribute name="type" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="air_condition"/>
          <xs:enumeration value="child_friendly"/>
          <xs:enumeration value="credit_cards"/>
          <xs:enumeration value="dresscode"/>
          <xs:enumeration value="happy_hour"/>
          <xs:enumeration value="official_star_rating"/>
          <xs:enumeration value="outdoor_seating"/>
          <xs:enumeration value="parking"/>
          <xs:enumeration value="pets_allowed"/>
          <xs:enumeration value="price_range"/>
          <xs:enumeration value="reservations"/>
          <xs:enumeration value="room_service"/>
          <xs:enumeration value="smoking"/>
          <xs:enumeration value="take_away"/>
          <xs:enumeration value="toilet"/>
          <xs:enumeration value="tv"/>
          <xs:enumeration value="vegan"/>
          <xs:enumeration value="vegetarian"/>
          <xs:enumeration value="wheelchair"/>
          <xs:enumeration value="wifi"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```



```
</xs:complexType>
</xs:element>
<xs:element name="state" type="xs:string"/>
```

Listing 7.15: PXF Attributes XSD

```
<attributes>
  <attribute type="wheelchair">
    <state>true</state>
  </attribute>
  <attribute type="air_condition">
    <state>true</state>
  </attribute>
  <attribute type="child_friendly">
    <state>>false</state>
  </attribute>
  <attribute type="smoking">
    <state>only outside</state>
  </attribute>
  <attribute type="outdoor_seating">
    <state>true</state>
  </attribute>
  ...
  ...
</attributes>
```

Listing 7.16: PXF Attributes Example

7.5 Prototype Architecture

The prototype implements a first application of the developed PXF schema. As a result of the previous research the prototype implements the APIs of Qype and Flickr. This selection was based on high data availability, locality and the elaborated data diversity regarding to points of interest. Qype offers a multitude of data related to Austrian points of interest. Therefore the verification of the data could be done easier and more purposeful. As supplementation Flickr was chosen, since the image database is comprehensive and an extensive API is offered.

Figure 7.3 shows the architecture of the prototype. As the evaluation has shown, there are various eTourism online providers on the web. The prototype implements two of them, Qype and Flickr. Via their provided API functions point of interest data can be accessed. Since the prototype is using the data in a read-only mode, requests are done through HTTP. Qype claims a registration of the application, which is interacting with the API. Also a Qype account is required. After the successful registration an application key is provided, which has to be sent within every HTTP request. For the master thesis a PRO Account was provided by Qype, by default Qype awards only Simple Accounts. Also for accessing the Flickr API a registration of the own application is required. With the provided API key the methods of the API can be accessed.

Both APIs are accessed via an adapter. The adapters are the controlling part for the interaction with the APIs. The format of the response is XML in the case of Qype. A Flickr API request can return the formats XML, JSON or SOAP. For the prototype the responses of Flickr are processed as XML. The adapters are passing the retrieved data to the mapping and parsing layer. The data is parsed and mapped to the new point of interest format (PXF). Output of this layer is a PXF-File and a JSON object for each point of interest. The merged JSON object is used for the subsequent storage in the document-based database (MongoDB).

The persistence layer implements the MongoDB manipulation functions for the interaction with the database. The merged JSON object is transformed to the storage format of MongoDB and then stored in the database.

The visualization of the retrieved data is carried out by a GUI. For the prototype an existing tool was selected. The Database Master 4.0 visualizes document-oriented databases, like MongoDB.

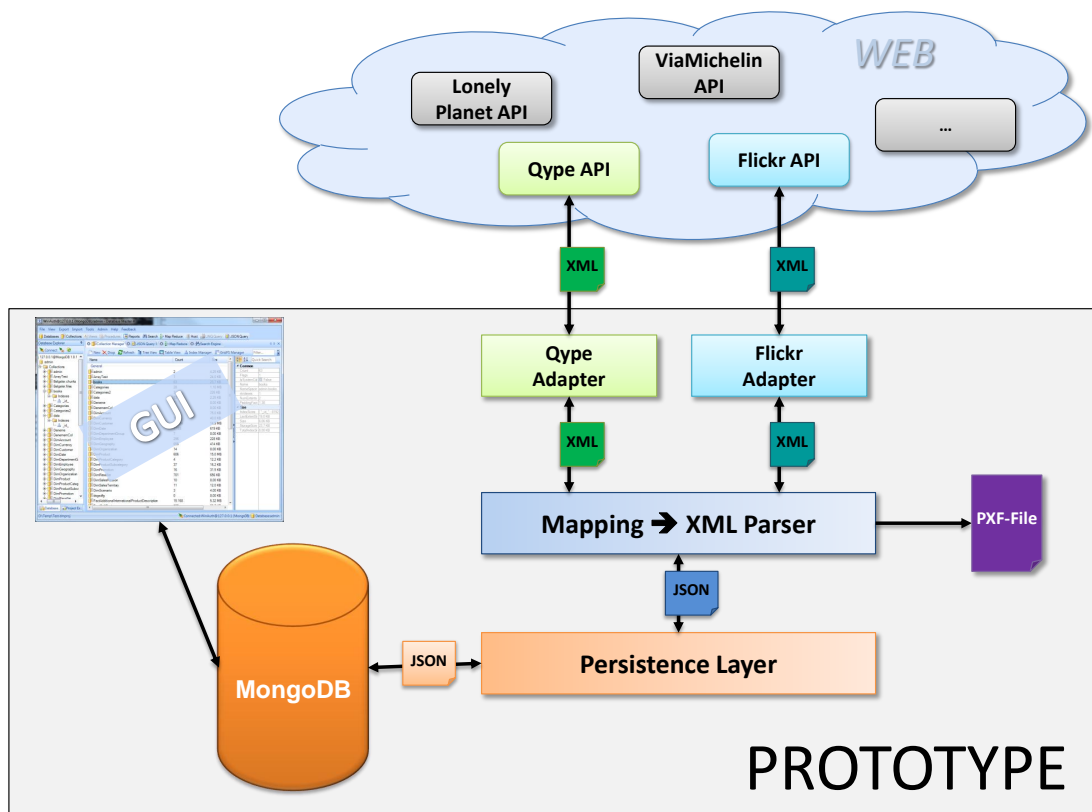


Figure 7.3: Prototype architecture

7.6 Qype API Requirements

7.6.1 Main concepts

The Qype API is a RESTful [67] service. By using this concept a simplified access to the web services and their resources is provided. REST (Representational State Transfer) is based on principles described in section 5.9. The API enables the sending of HTTP requests. Responses can be received in the format of XML or JSON. For the prototypical implementation the response format is XML.

7.6.2 Authentication

To operate with the Qype API a Qype account and an application registration is necessary. For each registered application a single API key will be provided to access the resources. By default a standard account will be provided, which enables access to the API with a request limit of 200 per day and limited API functionality. PRO accounts, which have unlimited, free access to all resources, can be requested at the website. After a successful application review by Qype a PRO account can be provided.

In principle there are two types of authentication, depending on the purpose of the application, which is accessing the API:

- read-only authentication by forwarding the API key in every HTTP request
- read-write authentication by using the OAuth Core 1.0a protocol [35]

The prototype implements a read-only authentication, because data will only be read, not manipulated. Every HTTP request includes the API key for authentication (Figure 7.17). For the master thesis a PRO account was requested and approved, what grants a unlimited and free access to all provided resources.

```
http://api.qype.com/v1/places?show=Rochus&consumer_key=mrOwJVwVpDKzDxnmqLp6g
```

Listing 7.17: read-only authentication via API key

7.6.3 Resource *Place*

The main resource related to point of interest data is the Qype resource place. A place can be searched by different attributes:

- ID
- name
- keyword
- near a position
- in a city

- in a rectangular area
- category

The prototype is considering points of interest for a rectangular area, the center of Vienna. The geographical section of the area is shown in Figure 7.4.



Figure 7.4: Center of Vienna [54]

The rectangular area is defined by longitude and latitude:

| Variable | Value | Description |
|----------|----------|-------------------|
| minx | 16.35826 | minimum longitude |
| miny | 48.20168 | minimum latitude |
| maxx | 16.38347 | maximum longitude |
| maxy | 48.21612 | maximum latitude |

Table 7.1: Bounding box for the center of Vienna

The request for such a query has to be in the form of Listing 7.18. The API key has to be forwarded in the variable `consumer_key`. By default the response is in XML format. To specify the response format the HTTP Accept header field can be used (e.g. 'Accept: application/json').

```
http://api.qype.com/v1/bounding_boxes/{sw_lat},{sw_lng},{ne_lat},{ne_lng}/places
```

Listing 7.18: Places in a rectangular area

The Qype API uses a custom XML format for the representation of resources. The format is based on Atom and JSON and pursues the following principles [63]:

- All elements include an unique id element.
- All elements are containing date and time information (created, updated).
- Related resources are interconnected with link elements. The attribute *rel* is defining the type of relation.
- All elements have a link element with *rel*=self that points to the permanent, unique API URL for this element.
- All elements have a link element with *rel*=alternate that contains the corresponding URL on www.qype.com for this element.

7.6.4 Resource Review

A further resource that is considered in the prototype are reviews. Reviews are user evaluations of a place. The interconnection between a place and a review is represented by the link element with the type *rel*='http://schemas.qype.com/reviews'. The variable *href* includes the link to the review of the place.

7.6.5 Resource Assets

Pictures related to a place are represented via the resource assets. The interconnection between a place and various assets is represented by a link element with the relation attribute *rel*='http://schemas.qype.com/assets'. The variable *href* includes the link to the picture of the place.

7.6.6 Resource Category

A place can have one or more categories, which classify the places in more detail. Categories can be Restaurants, Pubs, Services, Shops, etc. Categories are organized as hierarchical tree structure, which facilitates a detailed search.

7.6.7 Resource Tags

A place can be described via multiple words, named tags. Mainly tags are user generated. The interconnection between a place and tags is represented by a link element with the relation attribute `rel='http://schemas.qype.com/tags'`. The variable `href` includes the link to the tags of the place.

7.6.8 Resource Attributes

Multiple attributes can define a place in more detail. Attributes are predefined and not user generated. The connection between a place and attributes is represented by a link element with the relation attribute `rel='http://schemas.qype.com/attributes'`. The variable `href` includes the link to the attributes of the place. There the attributes and the values set are located.

7.7 Flickr API Requirements

7.7.1 Main concepts

The Flickr API provides access after a registration of the own application. With the provided API key the authentication can be carried out via HTTP or OAuth 1.0. API requests can be done via REST, XML-RPC or SOAP. Response formats supported by the API are XML, JSON or SOAP. The prototype will address the REST interface and the response format will be defined as XML.

7.7.2 Authentication

In general an authentication is required to use the API methods of Flickr, this has to be done via HTTP or the OAuth 1.0 protocol. However, several functions are accessible without authentication.

7.7.3 Resource Photo

The Flickr API operates amongst other with a photo resource. The retrieved photo resource includes different elements. To access the resource photo all calls must be effected through the subsequent URL:

```
http://api.flickr.com/services/rest/?method=flickr.photos.name_of_method
```

Listing 7.19: Flickr URL Resource *photo*

The prototypical implementation is focusing on tags, which are assigned to a picture referencing a point of interest. The tags of a picture can also be assigned indirectly to the point of interest. By this means, conclusions regarding the point of interest can be deduced.

Listing 7.20 shows the RESTful API request, to retrieve all information related to a certain photo. Listing 7.21 shows the corresponding response of the previously described API call.

```
http://api.flickr.com/services/rest/?method=flickr.photos.getInfo&api_key=123456&photo_id=6089902211&format=rest
```

Listing 7.20: Flickr request

```

<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
  <photo id="6089902211" secret="09c086e706" server="6206" farm="7" dateuploaded="1314563611
    " isfavorite="0" license="2" safety_level="0" rotation="0" originalsecret="f2a264669b"
    originalformat="jpg" views="2" media="photo">
    <owner nsid="80327421@N00" username="stormgrass" realname="Richard" location="UK"
      iconserver="2" iconfarm="1" />
    <title>Lunch at Cobenzl</title>
    <description />
    <visibility ispublic="1" isfriend="0" isfamily="0" />
    <dates posted="1314563611" taken="2011-08-28 22:28:06" takengrularity="0" lastupdate="
      1314564599" />
    <editability cancomment="1" canaddmeta="0" />
    <publiceditability cancomment="1" canaddmeta="0" />
    <usage candownload="1" canblog="0" canprint="0" canshare="1" />
    <comments>0</comments>
    <notes />
    <people haspeople="0" />
    <tags>
      <tag id="41056955" author="80327421@N00" raw="exif:height=768" machine_tag="1">exif:
        height=768</tag>
      <tag id="75079795" author="80327421@N00" raw="exif:lat=48.262691666667" machine_tag="
        1">exif:lat=48262691666667</tag>
      <tag id="75079797" author="80327421@N00" raw="exif:long=16.322330555556" machine_tag
        ="1">exif:long=16322330555556</tag>
      <tag id="75079799" author="80327421@N00" raw="hidden:city=Vienna" machine_tag="1">
        hidden:city=vienna</tag>
      <tag id="74064415" author="80327421@N00" raw="hidden:filter=Normal" machine_tag="1">
        hidden:filter=normal</tag>
      <tag id="75079801" author="80327421@N00" raw="hidden:venue=4
        b3e4321f964a520529a25e3" machine_tag="1">hidden:venue=4b3e4321f964a520529a25e3
        </tag>
      <tag id="1763" author="80327421@N00" raw="lunch" machine_tag="0">lunch</tag>
    </tags>
    <urls>
      <url type="photopage">http://www.flickr.com/photos/gibarian/6089902211/</url>
    </urls>
  </photo>
</rsp>

```

Listing 7.21: Flickr response

Within the XML response of the requested picture there is a section *tags*. Each tag is included within an XML tag with the identifier *tag*. Every tag entry has the attributes *ID*, author reference, *raw* and *machine_tag*. *ID* is the unique identifier of the tag. *Author* is the reference to the user, who added the tag. *Raw* is the entered tag by the user, which can contain spaces and punctuation. The attribute *machine_tag* specifies, if it is a standard tag or a machine tag. The tag body contains the tag as it is processed by Flickr. This version of the tag is used for constructing URLs.

7.8 Database Requirements

7.8.1 System evaluation

The prototype stores the extracted point of interest data from the external APIs in a database. Here the new standardized schema PXF is adopted. For the prototype different types of database systems [58] were evaluated:

- **Hierarchical Systems**

The information model can be represented as a tree (Figure 7.5). Nodes of the tree store data and can relate to a parent node, respectively various child nodes. The XML format is also based on a hierarchical tree structure, therefore this rather antiquated database system architecture was observed. However, today hierarchical data base systems have been replaced by other approaches.

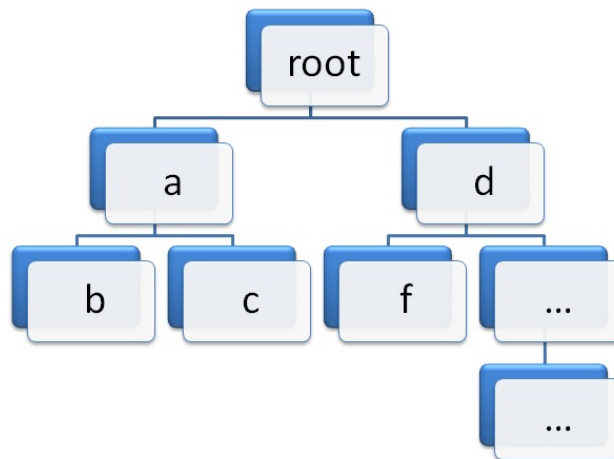


Figure 7.5: Hierarchical DBS

- **Relational Systems**

The relational approach was introduced by E.F. Codd [12] in 1970. Today it is an established standard in the field of database systems. Data is stored in collections of tables (relations). Rows of the table represent the individual data record, the columns represent the attributes. A connection between tables can be modeled as reference. Figure 7.6 shows a basic example of an RDMS (relational database system).

- **Document-oriented Systems**

In this approach the storage units for information are documents. In contrast to relational database systems, there is no predefined structure (like in tables) for the data, it is schema-less [69]. The documents can be encoded in binary formats like XML, JSON, BSON, PDF, etc. and accessed via a unique identifier within the database.

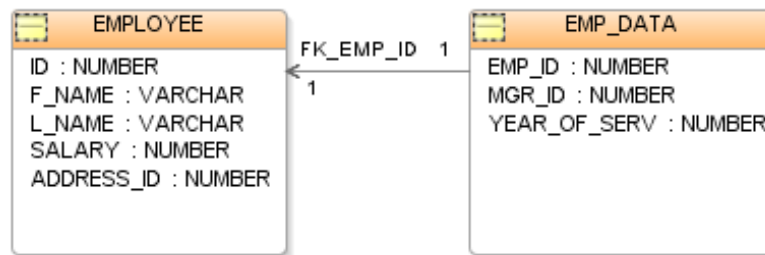


Figure 7.6: Relational DBS

For document processing applications the use of a document-oriented database system can be very efficient. Relational database systems are designed to have a fixed data structure and perform efficient, dynamic queries [69]. A mapping of divergent documents into a relational database system can be difficult, therefore document-oriented database systems were designed.

For web applications, document-based systems are well suited. There are various systems, under different license types, available. CouchDB, MongoDB, BaseX, Lotus Notes are just some of them [93]. Also XML databases are document-based database systems. The prototypical implementation will process the data with MongoDB. The database is introduced and analyzed in the next section, regarding to the requirements of the prototype.

7.8.2 MongoDB

MongoDB (from the word „humongous“) is a document-oriented database under GNU AGPL v3.0 license [1]. The first release was in 2009 by 10gen. It is a cross-platform database and the last stable release is 2.0.1 (October 2011). A mapping between SQL and MongoDB [1] is shown in Table 7.2.

| SQL term | Mongo term |
|-------------|-----------------------|
| Database | Database |
| Table | Collection |
| Index | Index |
| Row | BSON document |
| Column | BSON field |
| Join | embedding and linking |
| Primary key | _id field |

Table 7.2: SQL to Mongo Mapping

Basic Concepts

The schematic MongoDB database model [47] is shown in Figure 7.7. MongoDB can hold a set of databases. A database has a namespace limit of 24,000, whereupon a collection accounts for at least two namespaces. A collection can be seen as a container for documents. Each database can have up to 12,000 containers by default, based on the limited namespaces. A document is a set of fields, whereat a field consists of a key-value pair. A value can hold the following types:

- basic type like string, integer, float, timestamp, binary, etc.
- document, or
- array of values

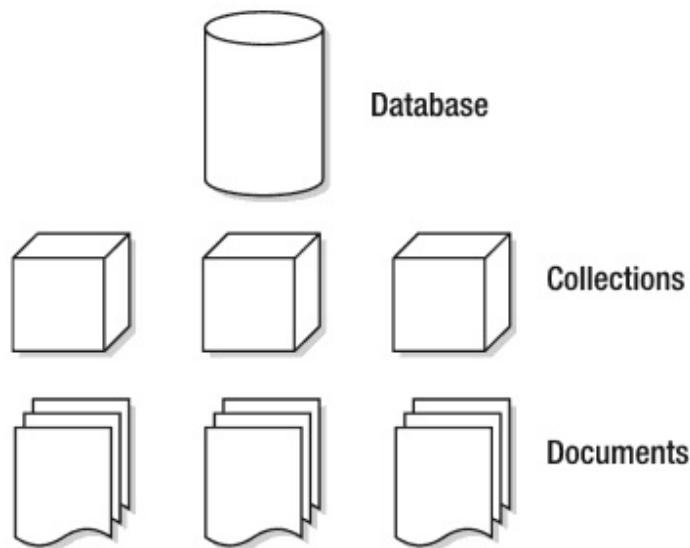


Figure 7.7: MongoDB Database Model

MongoDB stores documents in the BSON (Binary JSON) format. Currently many common programming languages like C, C++, Erlang, Java, Javascript, .NET, Ruby, etc. are supported. Furthermore, there is a long list of drivers supported by the community. Drivers for the platforms can be downloaded on the MongoDB website [1].

7.8.3 Database Specification

The prototype stores point of interest data in a MongoDB. Within this database a collection for POIs is disposed. The collection is the container for the individual points of interest. Each POI is stored in a document, which has to be in a binary JSON format (BSON). Figure 7.8 shows the concept of the MongoDB design for the prototype.

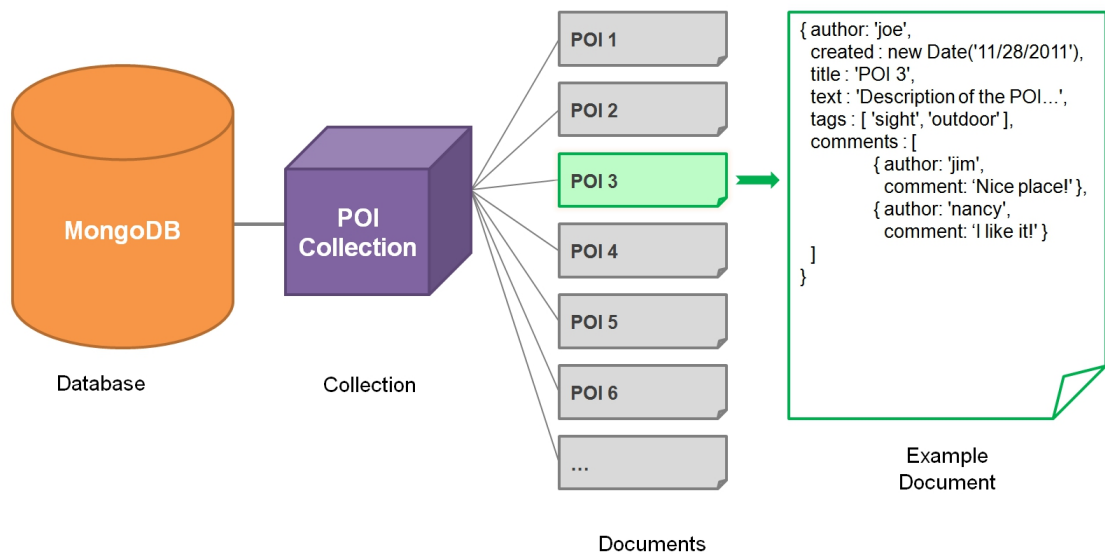


Figure 7.8: MongoDB Database Design

The name of MongoDB database used for the prototypical implementation is **POI_DB**. Within this database the collection of POI's will be stored, the collection is name **POI_Collection**. Each imported POI is represented by a single document, which is nested in the POI_Collection. For the prototypal one collection is sufficient.

7.9 Implementation

The prototype is implemented in the programming language Java using Eclipse. The implementation of the RESTful web services is done via the open source API Jersey [57]. Jersey is a reference implementation of JAX-RS (Java API for RESTful Web Services).

The processing and parsing of XML responses were implemented with SAX (Simple API for XML). SAX [52] is widely used for processing XML documents. Another existing widespread standard is DOM (Document Object Model). The SAX API implements various methods to process XML data. SAX has the advantage of sequentially loading of XML tags, what increases memory efficiency. Through the event-driven handling a faster processing of large documents is ensured.

7.9.1 Sequence Flow

The sequence flow of the prototype is shown in the chart 7.9. Initially the APIs of Qype and Flickr are accessed as RESTful webservices. Point of interest data for the defined area is requested. The APIs return data, which is parsed afterwards. In both cases, Qype and Flickr, the response is in XML format. Via the SAX API the XML tags of the Qype response are parsed and brought into a format for storage and file output. The Flickr API provides tag information

regarding to points of interest. After the parsing process the points of interest are stored in the MongoDB in consideration of the new developed PXF schema. As final step, each POI is stored in a separate .pxf file.

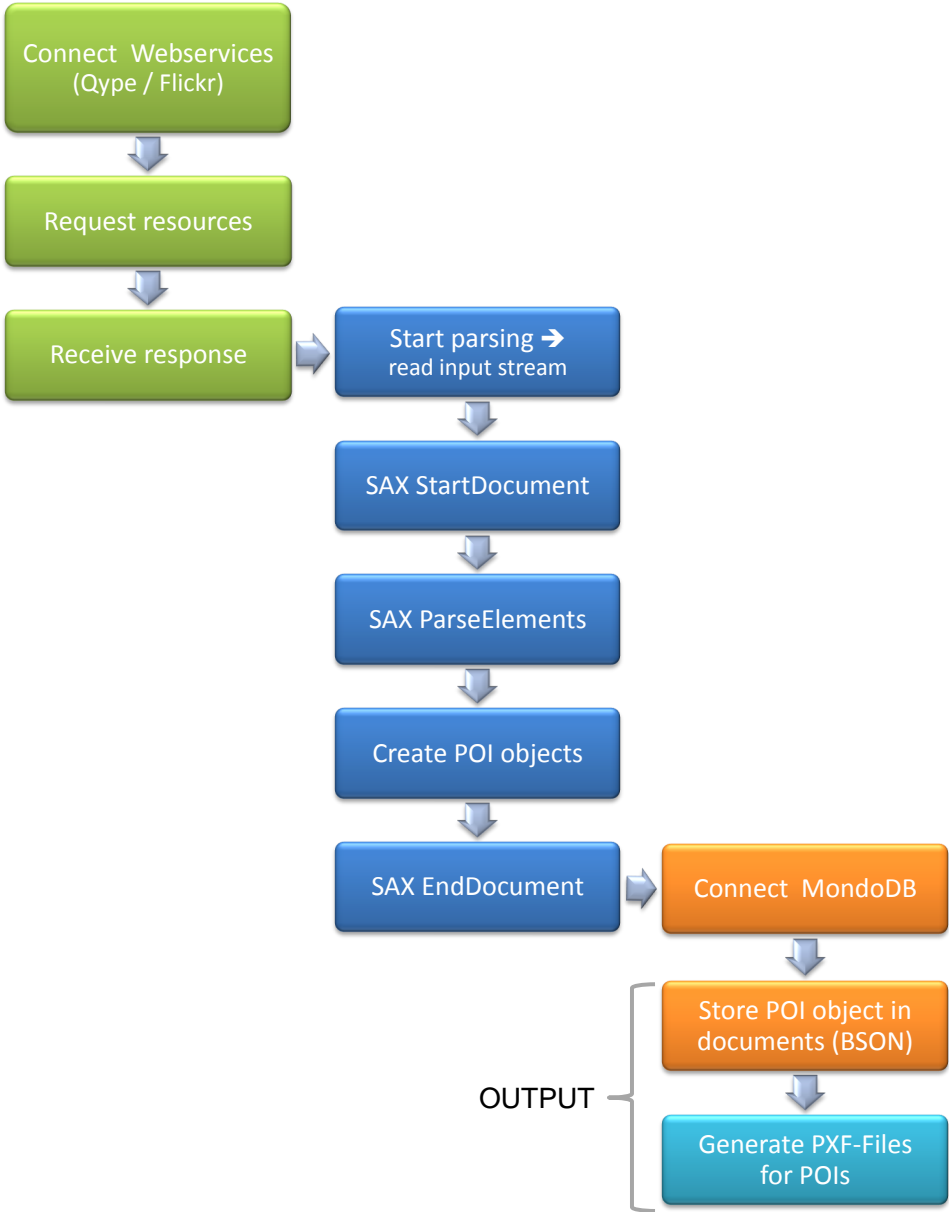


Figure 7.9: Flow Chart

7.9.2 API Request

Listing 7.22 shows the access to the APIs of Qype and Flickr via the Jersey API. The area for considered points of interest is defined by a bounding box. The two pairs of coordinates set the border of this area.

To access the Qype API an API key is required. For the master thesis a PRO Account was provided, which ensures an unlimited access to the webservices of Qype. The API key is forwarded with each request within the variable *consumer_key*. The format of the response is defined via the variable *MediaType*. The returned result of the query is a string, including a list of POIs, which is handed over to the parsing process.

```
private static URI getBaseURI() {
    return UriBuilder.fromUri("http://api.qype.com/v1/positions
        /48.19775,16.37102,48.19879,16.37268/places").build();
}
...

String response = service.queryParam("consumer_key", s.getQype_api_key()).
    accept(MediaType.TEXT_XML).get(String.class);
QypePlaceParser parser = new QypePlaceParser();
ArrayList<Place> places = parser.parse(response);
```

Listing 7.22: Qype API request

7.9.3 Parsing

For parsing the places retrieved from the Qype API a *QypePlaceParser* was implemented (Listing 7.23). The input parameter for the parser is XML file returned. The return value of the procedure is an array of places.

```
public class QypePlaceParser
{
    private XMLReader reader;
    private PlaceHandler h;

    public ArrayList<Place> parse(String xml) {
        h = new PlaceHandler(); //reader for XML
        reader = XMLReaderFactory.createXMLReader(); //register contentHandler
            to catch all content events reported by the SAX parser
        reader.setContentHandler(h);
        if(xml!="") {
            try{
                InputStream is = new ByteArrayInputStre(xml.getBytes("UTF-8"));
                InputSource isource = new InputSource();
                isource.setByteStream(is);
                isource.setEncoding("UTF-8");
            }
            ...
        }
    }
}
```

Listing 7.23: QypePlaceParser

For the parsing process the ContentHandler interface of SAX was implemented [56]. Basic parsing events and document-related events, like the start and end of elements and character data are reported via this interface.

Listing 7.24 shows the implementation of the SAX ContentHandler method *startElement*. For each XML element this method is executed. Depending on the XML tag the information contained is parsed and stored in a corresponding data object.

```
public void startElement(String uri, String localName, String qName,
    Attributes attributes) throws SAXException{
    Settings s = new Settings();

    //temporary place object
    if(qName.equalsIgnoreCase("place")){
        tmpPlace = new Place();
    }

    //place found ->create new place variable
    if(qName.equalsIgnoreCase("places")){
        places = new ArrayList<Place>();
    }

    //address found -> create new address variable
    if(qName.equalsIgnoreCase("address")){
        tmpAd = new Address();
    }

    //attributes found -> extract link (href) to retrieve related attributes
    if(qName.equalsIgnoreCase("link") && attributes.getValue(attributes.
        getIndex("rel")).contains("attributes"))
    { attributes.getValue("href").toString();
      ClientConfig config = new DefaultClientConfig();
      Client client = Client.create(config);
      WebResource service = client.resource(attributes.getValue(attributes.
        getIndex("href")));

      try{
          String att_response = service.queryParam("consumer_key", s.
              getQype_api_key()).accept(MediaType.TEXT_XML).get(String.class);
          QypeAttributeParser parser = new QypeAttributeParser();
          att = parser.parse(att_response);
          tmpPlace.setAttributes(att);
      }
      catch (Exception ex)
      { if (ex.toString().contains("404"))
        { System.out.println("NO ATTRIBUTES!");
          tmpPlace.setAttributes("");
        }
      }
    }
    ...
}
```

Listing 7.24: SAX ContentHandler method *startElement*

7.9.4 MongoDB Storage

After the parsing process the points of interest are stored in an array of place objects. With this array as parameter the MongoDB storage method is called (Listing 7.25). The MongoDB used for the prototype is named POI_DB. Within this database the information is stored in a collection POI_Collection considering the new developed PXF schema. Since the storage of points of interest is document-based, each POI object is mapped to a document. With the *document.put* method, the individual attributes of the POI document can be set. If the POI document is complete, it can be inserted into the collection via the *coll.insert* method. Subsequently, completing all POIs this way, the collection is stored in the database.

```
public static void save(ArrayList<Place> myplaces)
{
    Mongo m;
    try {
        m = new Mongo();
        //get handle to MongoDB
        DB db = m.getDB("POI_DB");

        //get handle to Collection
        DBCollection coll = db.getCollection("POI_Collection");

        //drop all data in the collection
        coll.drop();

        //create new document
        BasicDBObject doc = new BasicDBObject();

        //store each POI as a document
        for (int i=0; i<myplaces.size(); i++)
        { doc.put("id", myplaces.get(i).getId());
          doc.put("name", myplaces.get(i).getTitle());
          ...
          ...
          //insert POI document into collection
          coll.insert(new BasicDBObject().append("poi", doc));
        }
    }
    ...
}
```

Listing 7.25: Mongo Insert

The evaluated Flickr tags are also stored in the database. Therefore an update operation is performed (Listing 7.26). The tag attribute is updated with die evaluated picture tags from Flickr.


```

public static void update(String poi, ArrayList<String> tags)
{
    Mongo m;
    String tmpTags;
    try {
        m = new Mongo();
        tmpTags = "";

        //get handle to MongoDB
        DB db = m.getDB("POI_DB");

        //get handle to Collection
        DBCollection coll = db.getCollection("POI_Collection");

        //build tag string for POI
        for (String tag : tags) {
            if (tmpTags.isEmpty()){
                tmpTags = tag;
            }
            else{
                tmpTags = tmpTags + "," + tag;
            }
        }

        //find POI by name
        BasicDBObject updateQuery = new BasicDBObject();
        updateQuery.put("poi.name", poi);

        //set tag list
        BasicDBObject update = new BasicDBObject();
        update.put("$set", new BasicDBObject("poi.tags", tmpTags));

        //update the POI document
        coll.update(updateQuery, update, true, false);
    }
    catch (UnknownHostException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    catch (MongoException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

```

Listing 7.26: Mongo Update

After the successful storage of POI documents, they are available in the MongoDB. For the prototype the program Database Master 4 was used for browsing the data in the MongoDB. Figure 7.10 shows the GUI of Database Master.

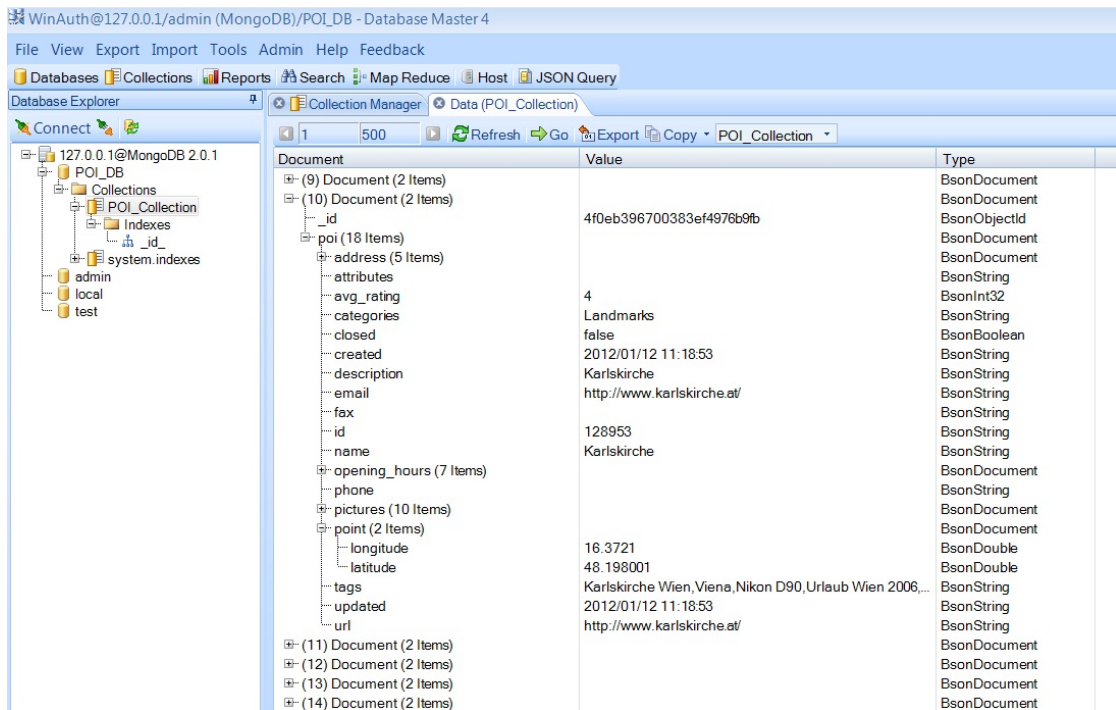


Figure 7.10: Database Master 4

7.9.5 PXF Output

As final step, each POI is stored in a separate PXF file (Listing 7.27). The XML version used is 1.0 and the encoding is UTF-8. The generated PXF file is validated via a XSD (XML Schema Definition) file. By this means it can be assured that the produced file corresponds to the defined PXF schema.

```
public static void toPXF(ArrayList<Place> myplaces) {

    //create .pxf file for each POI
    for (int i=0; i<myplaces.size(); i++)
    {
        String out = "";
        out += "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n";
        out += "<poi xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" xsi
            :noNamespaceSchemaLocation=\"file:///Mongo01/PXF.xsd\">\n";

        out += "<id>"+ myplaces.get(i).getId() + "</id>\n";
        out += "<name>"+ myplaces.get(i).getTitle() + "</name>\n";
        ...
    }
    file = new File(myplaces.get(i).getId() + ".pxf");
    ...
}
```

Listing 7.27: PXF

7.10 Prototype Verification

In order to verify the prototypical implementation a selection of different POI types was evaluated in detail. Five different types of points of interest in Vienna were analyzed. The applicability of the newly developed schema was examined. Furthermore, data availability, reliability and topicality were evaluated on the basis of the two implemented APIs, Qype and Flickr. The investigated points of interest in Vienna were:

- Karlskirche (sight)
- Happel Stadion (venue)
- Hotel Sacher (lodging)
- Albertina (museum)
- Das Rochus (bar/restaurant)

7.10.1 POI 1: Karlskirche

The Karlskirche in Vienna is a famous sight. Both, Qype and Flickr, offer information regarding this point of interest. Figure 7.11 shows the information about the Karlskirche how it is presented on the Qype website. The search request on Flickr returns 850 results for “Karlskirche Vienna“ and 257 results for “Karlskirche Wien“. Hence the conclusion can be drawn, that the language of the query is highly relevant.

Karlskirche, Vienna Is this your business?
★★★★☆ 8 reviews
[Write a review](#) [Follow us](#)

***QYPE deals** Find the best places in your town...
Save up to 80% with QypeDeals! [» More Information](#)

Category: [Landmarks Vienna](#)
Address: Kreuzherrengasse 1, 1040 Wien
Website: www.karlskirche.at/
Opening hours: Öffnungszeiten:
Montag bis Samstag:
9 Uhr bis 12.30 Uhr und 13 Uhr bis 18 Uhr
Sonn- und Feiertag:
12 Uhr bis 17.45 Uhr
Know an upcoming event? [Add it](#)

[Edit this information](#)
Something wrong with this information?

by Rommeck
[All photos](#) [Upload photo](#)

Figure 7.11: Qype website *Karlskirche*

The actual Qype XML file of the Karlskirche in Vienna is attached in the Appendix D.1. The description of the POI is extensive and includes basic information as well as pictures, attributes, reviews, etc. Due to the great amount of Flickr results the verification was done for the 5 most relevant images of the Karlskirche in terms of their tags.

| Tag | Count | Tag | Count |
|--------------|-------|-------------|-------|
| 2000 | 1 | iPhoto | 1 |
| Architecture | 1 | Karlskirche | 2 |
| art | 1 | light | 1 |
| Austria | 4 | terreich | 1 |
| ceiling | 1 | pictophile | 1 |
| Church | 3 | stucco | 1 |
| City | 1 | Summer | 1 |
| europe | 3 | Travel | 1 |
| flickr | 1 | Wien | 6 |
| fresco | 1 | zx81 | 1 |
| geotagged | 1 | | |

Table 7.3: Tags *Karlskirche*

The evaluated tags show (Table 7.3), that most of them are referring to the name of the point of interest. Also the geographical location like city or country is a common assigned tag. Since it is a sight, tags are existing in various languages. Several tags of the pictures, showing the point of interest, can not be related to the POI itself, for example type of the camera taken this picture, date, personal comments, etc. are not relevant for the description of the POI.

7.10.2 POI 2: Happel Stadion

The Ernst Happel Stadium is a famous sports facility in Vienna. Relevant information can be found on Qype and Flickr. The Flickr query returns 691 results for “Happel Stadium“ and 2818 results for “Happel Stadion“. Figure 7.12 shows a snapshot of the retrieved results on the the Flickr website.

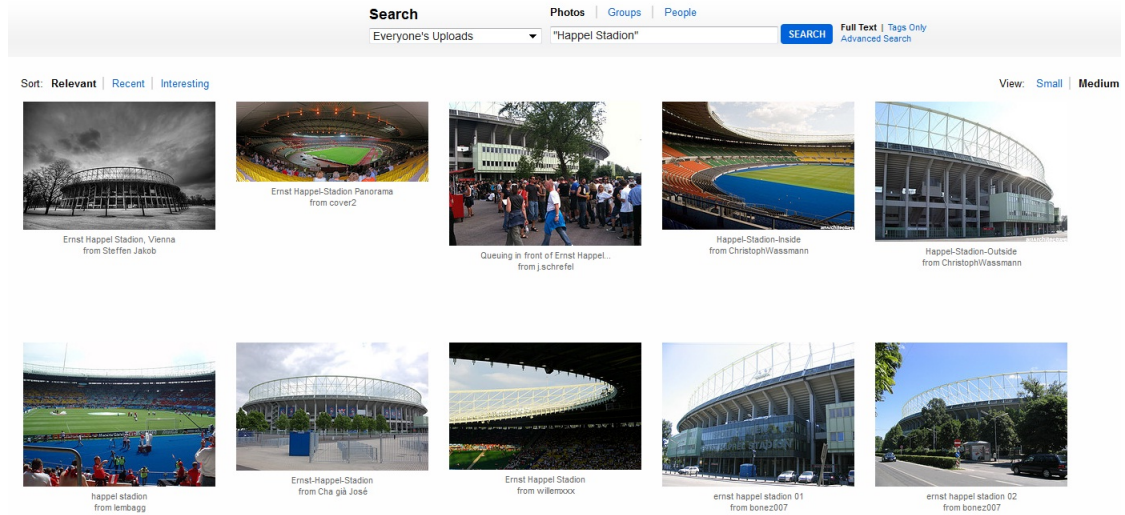


Figure 7.12: Flickr website *Happel Stadium*

The actual Qype XML file of the Happel Stadium in Vienna is attached in the Appendix D.2. The description of the POI is extensive and includes basic information as well as pictures, checkins, reviews, etc. Qype assigns several categories to this point of interest: sports, stadium and event location. Similarly to the previous POI the verification of the Flickr results was carried out for the 5 most relevant images of the Happel Stadium in terms of their assigned tags.

| Tag | Count | Tag | Count |
|----------------------------|-------|-----------------------|-------|
| 1. euro-spiel | 1 | FK Austria Wien | 1 |
| aline | 1 | football | 1 |
| architecture | 1 | Fussball | 1 |
| Austria | 2 | getty-artist-pick | 1 |
| Austria Vienna | 1 | happel stadion | 1 |
| Austria Wien | 1 | Leopoldstadt | 1 |
| austria-croatia | 1 | nationalmannschaft | 1 |
| Autostitch | 1 | nationalstadion | 1 |
| Benfica | 1 | nationalteam | 1 |
| blogged | 1 | fb | 1 |
| blue | 1 | Olympiaplatz | 1 |
| bw | 1 | terreich | 3 |
| Champions League | 1 | prater | 1 |
| Champions League Qualifier | 1 | Praterstadion | 3 |
| em | 1 | rot-weirotd | 1 |
| ernst-happel-stadion | 5 | soccer | 2 |
| erst happel | 1 | Sports | 1 |
| euro | 1 | Springsteen | 1 |
| euro 08 | 1 | stadion | 1 |
| euro 2008 | 1 | stadium | 1 |
| euro2008 | 1 | UEFA Champions League | 1 |
| Europa | 1 | vienne | 5 |
| Europacup | 1 | wien | 4 |
| europameisterschaft | 1 | Machine Tag | |
| Europe | 1 | geo:lat=48.209167 | |
| FAK | 1 | geo:lon=16.42161 | |
| fans | 1 | | |

Table 7.4: Tags *Happel Stadium*

What can be investigated in the case of the Happel Stadium is, that a lot of tags differ only in their notation, like the usage of upper and lower case and space characters (Table 7.4). Since the Happel Stadium is a venue for sports events a lot of tags are related to events, which have taken place there. Those tags are not highly relevant for the POI itself. This POI has also two machine tags, which are describing the geographical location of the stadium with longitude and latitude.

7.10.3 POI 3: Hotel Sacher

The Hotel Sacher is a prominent lodging in Vienna. Figure 7.13 shows the information about the Hotel Sacher, how it is present on the Qype website. There was only one Flickr request performed, because of the same notation of the point of interest name in German and in English. The Flickr query „Hotel Sacher“ returns 1757 items.

Hotel Sacher Wien, Vienna
★★★★☆ 67 reviews [Rating details](#)

[Write a review](#) [Follow us](#) [2 follow this place](#)

***QYPE deals** Find the best places in your town...
Save up to 80% with QypeDeals! [More Information](#)

[Check rates and availability to book a room:](#)
Check-in: Check-out: [Book Now!](#)

Categories: [Hotels Vienna](#)
[Restaurants Vienna](#)
[Cafés Vienna](#)

Contact us: [hotel_sacher_wien](#)

Address: Philharmoniker-Straße 4, 1010 Wien
+43 (0)1 - 51 456 0

Website: [www.booking.com/hotel/at/sache...](#)

Opening hours: täglich geöffnet von 08:00 bis 24:00 Uhr
warme Küche bis 23:30 Uhr
Know an upcoming event? [Add it](#)

More details: [Show](#)

Credit cards accepted: Yes **Outdoor seating:** No

Price range: High end

[All photos](#) [Upload photo](#)

Figure 7.13: Qype website *Hotel Sacher*

Appendix D.3 shows the Qype XML file of the Hotel Sacher. The point of interest is assigned to several categories: hotel, restaurant and cafe. Furthermore, a remarkable number of reviews is present. The evaluation of the Flickr tags is again carried out for the five most relevant Flickr images referring this point of interest.

| Tag | Count | Tag | Count |
|------------------|-------|---------------------|-------|
| ambiente | 1 | Gastronomie | 1 |
| architecture | 3 | historisch | 1 |
| Austria | 2 | hotel | 2 |
| balcony | 3 | Hotel Sacher | 3 |
| bekannt | 1 | innenstadt | 1 |
| beruehmt | 1 | kaiserlich | 1 |
| bruestung | 1 | luxushotel | 1 |
| building | 1 | May 2008 | 1 |
| Casino Wien sign | 1 | metropole | 1 |
| eingang | 1 | mondaen | 1 |
| entrance | 1 | nobel | 1 |
| eu-flagge | 1 | pforte | 1 |
| Europe | 1 | promonentenabsteige | 1 |
| expensive | 1 | statue | 1 |
| fahnen | 1 | teuer | 1 |
| famous | 1 | Vienna | 3 |
| fassade | 1 | wahrzeichen | 1 |
| fenster | 1 | well-known | 1 |
| flaggen | 1 | weltstaedtisch | 1 |
| flair | 1 | Wien | 2 |

Table 7.5: Tags *Hotel Sacher*

Pictures of the Hotel Sacher have many tags assigned (Table 7.5). Many of the analyzed tags are descriptive attributes and adjectives. The most commonly used tags are referring to the POI name and location.

7.10.4 POI 4: Albertina

The Albertina is a famous museum in Vienna. Both, Qype and Flickr, offer information regarding this point of interest. The Flickr query returns 4221 results for “Albertina Vienna“ and 3632 results for “Albertina Wien“. A snapshot of the retrieved results on the the Flickr website is shown in Figure 7.14.

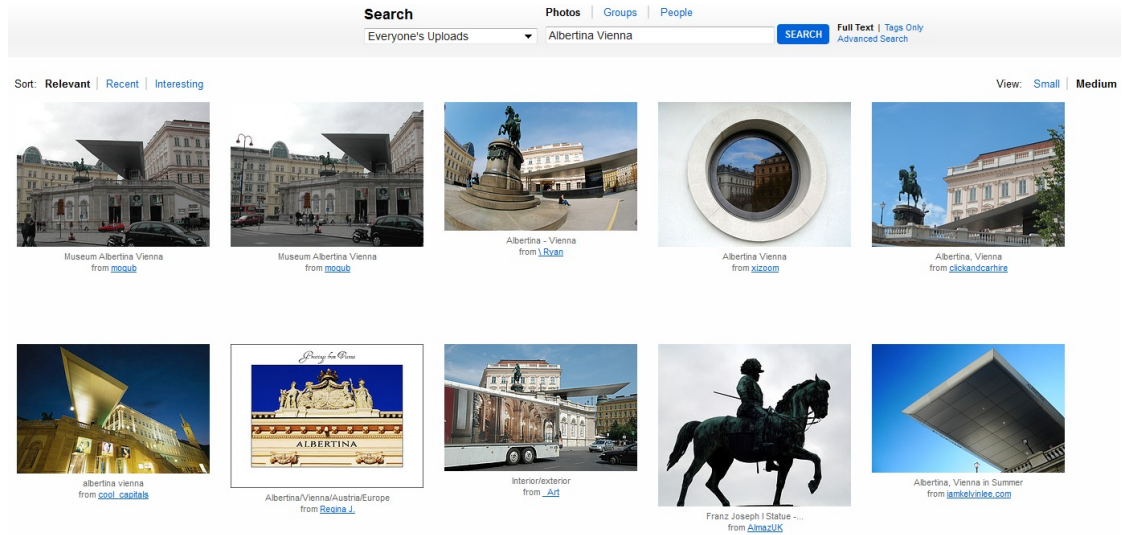


Figure 7.14: Flickr website *Albertina*

The actual Qype XML file of the Albertina in Vienna is attached in the Appendix D.4. The description of the POI is extensive and includes basic information as well as pictures, checkins, reviews, etc. Three categories are assigned to this point of interest: museum, sight and gallery. Again the verification of the Flickr results was carried out for the 5 most relevant images of the Albertina in terms of their assigned tags.

| Tag | Count | Tag | Count |
|-------------|-------|-------------|-------|
| Albertina | 3 | Great | 1 |
| Art | 1 | Hall | 1 |
| Austria | 1 | Hire | 1 |
| Burgtheater | 1 | Imperial | 1 |
| Car | 1 | Leica | 1 |
| Cathedral | 1 | museum | 2 |
| City | 1 | terreich | 1 |
| Court | 1 | Palace | 1 |
| Danube | 1 | Riesenrad | 1 |
| Eingang | 1 | Schnbrunn | 1 |
| entrance | 1 | Stephansdom | 1 |
| Europe | 1 | Theatre | 1 |
| Festival | 1 | Viena | 1 |
| Film | 1 | Vienna | 5 |
| Gloriette | 1 | Wien | 1 |

Table 7.6: Tags *Albertina*

The case of the museum Albertina shows, that a sight or museum is often visited by tourists (Table 7.6). Foreigners and tourists have a higher probability to assign tags to a picture, which are not directly related to the POI shown on it.

7.10.5 POI 5: Das Rochus

„Das Rochus“ is a popular bar and restaurant in Vienna. Qype provides more information on this point of interest than Flickr does. Figure 7.15 shows the representation of the POI on the Qype website. The Flickr query „Das Rochus“ returns only 40 images. However, it can be seen that not all of them are relevant for the POI „Das Rochus“. Further Flickr searches with „Rochus Vienna“ retrieves 41 entries and with „Cafe Rochus“ 3 images were found.

Das Rochus, Vienna
★★★★☆ 37 reviews Rating details ▾

[Write a review](#) [Follow us](#)

***QYPE deals** Find the best places in your town... Save up to 80% with QypeDeals! [» More Information](#)

Category: [Bars Vienna](#)
Contact us: [das_rochus](#)
Address: Landstraßer Hauptstraße 55-57, 1030 Wien 017101060
Website: [www.rochus.at](#)
Opening hours: Mo-So: 08.00 - 24.00
Know an upcoming event? [Add it](#)
More details: [Show ▾](#)
Credit cards accepted: Yes **Price range:** Average

[Edit this information](#)

Something wrong with this information?

by Stefan P.

[All photos](#) [Upload photo](#)

Figure 7.15: Qype website *Das Rochus*

Appendix D.5 shows the Qype XML file of Das Rochus. A remarkable number of reviews and checkins are present. The evaluation of the Flickr tags is again carried out for the five most relevant Flickr images referring this point of interest.

| Tag | Count |
|-------------|-------|
| austria | 1 |
| breakfast | 1 |
| caf | 1 |
| cameraphone | 1 |
| coffee | 2 |
| espresso | 1 |
| essen | 1 |
| food | 1 |
| lokal | 1 |
| Restaurant | 1 |
| Rochus | 3 |
| Vienna | 1 |
| Wien | 1 |

Table 7.7: Tags *Das Rochus*

Since Flickr is not providing a lot of pictures related to this point of interest, also the tag list is short (Table 7.7). Most tags are describing drinks and food served in this restaurant. Due the fact that it is not a tourist hot spot, the low number of pictures can be explained. Natives are not so likely to make pictures from a familiar environment.

7.10.6 Summary

The verification of the prototype showed that the newly developed schema is applicable for various types of points of interest. The aggregated data could be successfully mapped to the data schema. Basic information is available for most POI's. Temporary information like ratings, reviews and checkins were not considered in the PXF schema.

The evaluation of the Flickr tags showed, that tags assigned to pictures can not be adopted directly to the point of interest itself. Although many tags have high relevance for the point of interest.

Summary

The evaluation of the eTourism market shows that there is a great variety of online providers. Providers in the field of navigation, tour operators, rating portals etc. operate on the market. The thesis focused on well known providers. A local reference, availability of POI related data and the provision of an Application Programming Interface (API) were essential criteria in the selection process of the online providers.

8.1 Results

The survey shows that some point of interest formats are very similar. However, no uniform standard regarding point of interest data has established until now. The usage of a web compliant markup language like XML or JSON as the basis for a point of interest data standard seems reasonable, since the data is mainly used on the web and accessed via web services.

Many of the investigated eTourism online providers are offering API's to access their data and resources. As the comparative result matrix in chapter 6.1.1 shows, a major part of the vendors require a registration in order to use the their interfaces. For authenticated access also an API key for the own application is necessary. It is remarkable that no provider demands fees for the non-commercial use of the interface. For commercial use fees can be collected in some cases. Many of the online providers support the common web protocols (REST, JavaScript, SOAP) and data formats (XML, JSON etc.). The API support is very extensive in many cases.

The newly developed data schema for points of interest is primarily based on the former evaluation. The elaborated schema PXF (Point of interest eXchange Format) has an XML structure, which is defined via XSD (XML Schema Definition) in chapter 7.4. Figure 7.1 gives an overview of the structure and the individual components of the PXF schema.

The implemented prototype verifies the applicability of the newly developed schema PXF. By implementation of the Qype API and the Flickr API various aspects regarding the usage of the interface and data integration could be determined.

8.2 Conclusion & Outlook

The PXF schema is applicable for point of interest data. However, it could be analyzed that not every eTourism online provider obtains all the data for the complete PXF schema. Some providers offer only parts of the data schema. An aggregation of different data sources could resolve this problem. The evaluation also shows that there is static and dynamic data regarding points of interest. The thesis focused on the static information. Dynamic data like ratings, checkins, discount promotion etc. were not taken into account.

The analysis of the Flickr tags shows that not all tags of pictures can directly be adopted to the point of interest itself. This may be because tourists and foreigners, who are not familiar with the area or environment, are assigning tags, which are not directly connected with the point of interest. However, many tags have a high relevance for the point of interest.

The research in the field of eTourism is ongoing. For example, the annual International Conference on Information Technology and Travel & Tourism, called ENTER, is a forum for state of the art research and industrial case studies on the application of information and communication technologies to travel and tourism [39]. At the ENTER conference in 2011 the focus was set among other things on open standards and harmonizing data interchange in the travel and tourism industry.

Furthermore, point of interest data finds its application in various projects. The platform PixMe-Away, for example, is substituting a text and attribute based search for points of interest with an image based search [53]. By this means search criteria do not have to be determined previously by the user. By selecting pictures the user defines the search criteria indirectly. The platform PixMe-Away demonstrates the integration of point of interest data from external sources.

CHAPTER 9

Glossar

| | |
|--------|--|
| ACRISS | Association of Car Rental Industry Systems Standards |
| AJAX | Asynchronous JavaScript and XML |
| ANSI | American National Standards Institute |
| API | Application Programming Interface |
| ASCII | American Standard Code for Information Interchange |
| BSON | Binary JSON |
| BMP | Windows Bitmap |
| CSV | Comma-Separated Format |
| GPS | Global Positioning System |
| GPX | GPS eXchange Format |
| GUI | Graphical User Interface |
| HTTP | Hypertext Transfer Protocol |
| JAX-RS | Java API for RESTful Web Services |
| JSON | JavaScript Object Notation |
| KML | Keyhole Markup Language |
| OGC | Open Geospatial Consortium |
| OSM | Open Street Map |
| PDF | Portable Document Format |
| POI | Point of interest |
| PXF | Point of Interest eXchange Format |
| RDBS | Relational Database System |
| REST | Representational State Transfer |
| SAX | Simple API for XML |
| SDK | Software Development Kit |
| UCS | Universal Character Set |
| UGC | User generated content |
| URI | Uniform Resource Identifier |
| UTF-8 | 8-bit UCS Transformation Format |

| | |
|-------|----------------------------|
| WGS84 | World Geodetic System 1984 |
| XML | Extensible Markup Language |
| XSD | XML Schema Definition |

GPX format

```
<...  
  lat="latitudeType [1] ?"  
  lon="longitudeType [1] ?">  
  <ele> xsd:decimal </ele> [0..1] ?  
  <time> xsd:dateTime </time> [0..1] ?  
  <magvar> degreesType </magvar> [0..1] ?  
  <geoidheight> xsd:decimal </geoidheight> [0..1] ?  
  <name> xsd:string </name> [0..1] ?  
  <cmt> xsd:string </cmt> [0..1] ?  
  <desc> xsd:string </desc> [0..1] ?  
  <src> xsd:string </src> [0..1] ?  
  <link> linkType </link> [0..*] ?  
  <sym> xsd:string </sym> [0..1] ?  
  <type> xsd:string </type> [0..1] ?  
  <fix> fixType </fix> [0..1] ?  
  <sat> xsd:nonNegativeInteger </sat> [0..1] ?  
  <hdop> xsd:decimal </hdop> [0..1] ?  
  <vdop> xsd:decimal </vdop> [0..1] ?  
  <pdop> xsd:decimal </pdop> [0..1] ?  
  <ageofdgpsdata> xsd:decimal </ageofdgpsdata> [0..1] ?  
  <dgpsid> dgpsStationType </dgpsid> [0..1] ?  
  <extensions> extensionsType </extensions> [0..1] ?  
</...>
```

Listing A.1: XML Instance Representation of a waypoint

```

<xsd:complexType name="wptType">
  <xsd:sequence>
    <!-- elements must appear in this order -->
    <!-- Position info -->
    <xsd:element name="ele" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="time" type="xsd:dateTime" minOccurs="0"/>
    <xsd:element name="magvar" type="degreesType" minOccurs="0"/>
    <xsd:element name="geoidheight" type="xsd:decimal" minOccurs="0"/>
    <!-- Description info -->
    <xsd:element name="name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="cmt" type="xsd:string" minOccurs="0"/>
    <xsd:element name="desc" type="xsd:string" minOccurs="0"/>
    <xsd:element name="src" type="xsd:string" minOccurs="0"/>
    <xsd:element name="link" type="linkType" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="sym" type="xsd:string" minOccurs="0"/>
    <xsd:element name="type" type="xsd:string" minOccurs="0"/>
    <!-- Accuracy info -->
    <xsd:element name="fix" type="fixType" minOccurs="0"/>
    <xsd:element name="sat" type="xsd:nonNegativeInteger" minOccurs="0"/>
    <xsd:element name="hdop" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="vdop" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="pdop" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="ageofdgpsdata" type="xsd:decimal" minOccurs="0"/>
    <xsd:element name="dgpsid" type="dgpsStationType" minOccurs="0"/>
    <xsd:element name="extensions" type="extensionsType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="lat" type="latitudeType" use="required"/>
  <xsd:attribute name="lon" type="longitudeType" use="required"/>
</xsd:complexType>

```

Listing A.2: Schema Component Representation

Qype Resource Representation Reference

B.1 Resource *Place*

| Element | Description | Mandatory |
|--|--|-----------|
| title | name of this place | X |
| average_rating | rating based on all reviews and a secret formula | X |
| point | latitude and longitude of this place | X |
| url | website of the place | |
| phone | phone number, as entered by the user | |
| image | link to the main photo of this place | |
| id | unique string identifying this place | X |
| created | time this place was created | X |
| updated | time this place was last updated | X |
| closed | whether the place has been shut down (true/false) | X |
| liked | shows current authorized user attitude to this place | |
| The place resource embeds an address element: | | |
| city | name of this place | X |
| street | rating based on all reviews and a secret formula | |
| houenumber | number of the place | |
| postcode | postcode of the place | |

Table B.1: Resource *Place*

B.2 Resource Reviews

| Element | Description | Mandatory |
|-------------|---|-----------|
| language | language the review is written in (e.g. 'en') | X |
| rating | user rating for the place (one of 1,2,3,4,5) | X |
| point | latitude and longitude of this place | X |
| content | the actual text of the review (can contain sanitized HTML markup) | X |
| summary | short, truncated version of the review, without any HTML | X |
| tag | a tag describing the place. the term attribute contains the name of the tag. There can be multiple tag elements | X |
| id | unique string identifying this review | X |
| created | time this review was created | X |
| updated | time this review was last updated | X |
| liked | shows current authorized user attitude to this review | |
| locale_code | locale code of the review (e.g. 'en_GB') | |

Table B.2: Resource Reviews

B.3 Resource Checkins

| Element | Description | Mandatory |
|-----------------|--|-----------|
| active | true if this is the last checkin a user did | X |
| shout | optional comment associated with this checkin (max 140 chars) | |
| became_champion | true if this checkin resulted in the user becoming champion of the place | X |
| ousted_champion | name of the user this checkin just overtook in order to become champion if appropriate | |
| id | unique string identifying this checkin | X |
| created | time this checkin was created | X |
| liked | shows current authorized user attitude to this checkin | |
| image | link to the place photo loaded by checked in user | |

Table B.3: Resource Checkins

B.4 Resource Assets

| Element | Description | Mandatory |
|---------|---|-----------|
| image | link to the image in three thumbnail sizes: small, medium and large | |
| id | unique string identifying this asset | X |
| created | time this asset was created | X |

Table B.4: Resource Assets

B.5 Resource Locators

| Element | Description | Mandatory |
|-----------|---|-----------|
| title | name of this Locator, in the currently selected user interface language | X |
| point | latitude and longitude of this Locator (centered) | X |
| id | unique string identifying this locator | X |
| domain_id | string unambiguously identifying the location represented by this locator | X |
| admin1 | administrative area 1: country | X |
| admin2 | administrative area 2: city or state | |
| admin3 | administrative area 3: city or region (coarse) | |
| admin4 | administrative area 4: city or region (fine) | |

Table B.5: Resource Locators

B.6 Resource Categories

| Element | Description | Mandatory |
|------------|---|-----------|
| title | name of this category, in the currently selected user interface language | X |
| long_title | a more expressive name for this category, in the currently selected user interface language | X |
| id | unique string identifying this category | X |
| created | time this category was created | X |
| updated | time this category was last updated | X |

Table B.6: Resource Categories

B.7 Resource *Coupon*

| Element | Description | Mandatory |
|----------------|--|------------------|
| title | title of the coupon | X |
| description | text describing the details of the coupon | X |
| restriction | text describing possible restrictions which apply to this coupon | |
| end_date | time this coupon expires | |
| id | unique string identifying this coupon | X |
| created | time this coupon was created | X |
| updated | time this coupon was last updated | X |

Table B.7: Resource *Coupon*

ViaMichelin POI structure

```

<response>
  <poi>
    <id>/* POI unique identifier in the database (POIID) */</id>
    <db>/* Database unique identifier */</db>
    <lg>/* Language of returned informations */</lg>
    <name>/* name of the POI */</name>
    <encodedName>/* encoded name of the POI to be used in a query part of a URL */</encodedName>
  >
  <location>
    <coords>
      <lon>/* Longitude in decimal degrees (between -180.0 and 180.0) */</lon>
      <lat>/* latitude in decimal degrees (between -90.0 and 90.0) */</lat>
    </coords>
    <formattedAddressLine>/* Correctly formatted value of the street address line */</formattedAddressLine>
    <formattedCityLine>/* Correctly formatted value of the city address line */</formattedCityLine>
    <streetLabel>/* Street name (can include or not the street number)*/</streetLabel>
    <city>/* City name */</city>
    <postalCode>/* Postal code (country dependant format) */</postalCode>
    <countryISO>/* Country ISO 3166-1 alpha-3 code */</countryISO>
    <countryOfficial>/* licence plate country code */</countryOfficial>
    <countryLabel>/* Country name */</countryLabel>
  </location>
  <datasheet>
    <iconId>/* Icon identifier linked to the POI */</iconId>
    <geolvl>/* Level of geocoding:
    *1*: city level
    *2*: street level
    *3*: street number level
    *4*: manual */</geolvl>
    <tel><number>/* Phone number */</number></tel>
  </datasheet>
</response>

```

```

<fax><number>/* Phone number */</number></fax>
<email>/* email address associated with the POI */</email>
<web>/* url associated with the POI (usually a generic website) */</web>
<url>/* url associated with the POI (usually a direct access to the POI page on the client website)
*/</url>
<dataPath>/* Client data path (private storage path for the client on ViaMichelin domain) */</
dataPath>
<imgPath>/* Image path (private storage path for the client on ViaMichelin domain) */</imgPath
>
<mapIconPath>/* Icon path to display on map (private storage path for the client on ViaMichelin
domain) */</mapIconPath>
<metanumList help="/* Criteria (also called 'Numeric metadata') associated with the POI in the
form of a list of key–value pairs. This list stores up to 50 pairs. The meaning of each value
is defined by the client and depends on the metadata identifier and on POI type. Creteria
can be used to filter POI searches."/>
<metanum>
  <idx>/* Metanum key: integer which defines the numeric metadata identifier */</idx>
  <value>/* Metanum value as integer. Its semantics is defined by the client */</value>
</metanum>
  <metanum>/* Another metanum */</metanum>
</metanumList>
<metastringList help="/* Translated strings associated with the POI in the form of a list of key–
value pairs. This list stores up to 30 pairs. The meaning of each value is defined by the
client."/>
  <metastring>
    <idx>/* Metastring index as integer */</idx>
    <value>/* Metastring value as string */</value>
  </metastring>
    <metastring>/* Another metastring */</metastring>
</metastringList>
<descList help="/* Data (also called 'Descriptions') associated with the POI in the form of a list
of key–value pairs. This list stores up to 30 pairs. The meaning of each value is defined by
the client and depends on the metadata identifier and on POI type. Creteria can be used to
filter POI searches. Data cannot be used to filter POI searches."/>
  <desc>
    <idx>/* Data key: integer which defines the data identifier */</idx>
    <visible>/* Boolean to tell if current description is to be displayed to customer */</visible>
    <value>/* Data value as string */</value>
  </desc>
    <desc>/* Another description */
  </desc>
</descList>
<photoList>
  <photo>
    <idx>/* Photo index as integer */</idx>
    <url>/* url to access the photo */</url>
  </photo>
    <photo>/* Another photo */
  </photo>

```

```
</photoList>
<categoryList>
  <category>
    <value>/* Category label */</value>
  </category>
  <category>/* Another category */
  </category>
</categoryList>
</datasheet>
</poi>
</response>
```

Listing C.1: ViaMichelin POI XML structure

Prototype Verification

D.1 XML Karlskirche

```
<?xml version="1.0" encoding="UTF-8"?>
<place>
  <average_rating>4</average_rating>
  <title>Karlskirche</title>
  <closed>>false</closed>
  <url>http://www.karlskirche.at/</url>
  <id>tag.api.qype.com,2008-04-04:/places/128953</id>
  <updated>2011-12-28T10:52:45+01:00</updated>
  <created>2008-04-04T20:08:03+02:00</created>
  <point>48.198,16.3721</point>
  <image user_link="http://api.qype.com/v1/users/iNord" medium="http://assets2.qype.com/uploads/
    photos/0332/7892/karlskirche_653_thumb.jpg" large="http://assets2.qype.com/uploads/photos/
    /0332/7892/karlskirche_653_gallery.jpg" user_name="iNord" small="http://assets1.qype.com/
    uploads/photos/0332/7892/karlskirche_653_mini.jpg" medium2x="http://assets1.qype.com/
    uploads/photos/0332/7892/karlskirche_653_large.jpg"/>
  <opening_hours>fnungszeiten:
  Montag bis Samstag: 9 Uhr bis 12.30 Uhr und 13 Uhr bis 18 Uhr
  Sonn- und Feiertag: 12 Uhr bis 17.45 Uhr
</opening_hours>
  <explicit_content>>false</explicit_content>
  <address>
    <city>Wien</city>
    <houenumber>1</houenumber>
    <country_code>AT</country_code>
    <postcode>1040</postcode>
    <street>Kreuzherrengasse</street>
  </address>
</category>
```

```

<id>tag:api.qype.com,2007-08-23:/places/categories/30</id>
<updated>2009-12-09T10:21:12+01:00</updated>
<created>2007-08-23T14:19:57+02:00</created>
<title lang="de">Sehenswrdigkeiten</title>
<full_title lang="de">Sehenswrdigkeiten</full_title>
<link rel="self" href="http://api.qype.com/v1/place_categories/30"/>
<link rel="http://schemas.qype.com/place_categories.children" href="http://api.qype.com/v1/
place_categories/30/children"/>
<link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
place_categories/4"/>
</category>
<link rel="self" href="http://api.qype.com/v1/places/128953"/>
<link rel="alternate" href="http://www.qype.com/place/128953-Karlskirche-Wien?utm_source=api
&utm_medium=referral&utm_campaign=1743&utm_term=place" hreflang="de"/>
<link count="5" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/128953/
reviews/de" hreflang="de"/>
<link count="1" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/128953/
reviews/en" hreflang="en"/>
<link count="1" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/128953/
reviews/fr" hreflang="fr"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/128953/
reviews/es" hreflang="es"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/128953/
reviews/pt" hreflang="pt"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/128953/
reviews/pl" hreflang="pl"/>
<link count="1" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/128953/
reviews/it" hreflang="it"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/128953/
reviews/nl" hreflang="nl"/>
<link title="Wien" rel="http://schemas.qype.com/locator" href="http://api.qype.com/v1/locators/
at130-wien"/>
<link count="17" rel="http://schemas.qype.com/assets" href="http://api.qype.com/v1/places/128953/
assets"/>
<link rel="http://schemas.qype.com/places.nearby" href="http://api.qype.com/v1/positions
/48.198,16.3721/places?without=128953"/>
<link rel="http://schemas.qype.com/attributes" href="http://api.qype.com/v1/places/128953/attributes
"/>
<link count="0" rel="http://schemas.qype.com/checkins" href="http://api.qype.com/v1/places
/128953/checkins"/>
<link rel="http://schemas.qype.com/checkins_rankings" href="http://api.qype.com/v1/places/128953/
leaderboard"/>
<link count="0" rel="http://schemas.qype.com/likes" href="http://api.qype.com/v1/likes?likeable_id
=128953&likeable_type=place"/>
</place>

```

Listing D.1: Qype XML Karlskirche

D.2 XML Happel Stadium

```
<?xml version="1.0" encoding="UTF-8"?>
<place>
  <average_rating>4</average_rating>
  <title>Ernst Happel Stadion</title>
  <closed>>false</closed>
  <url>http://www.wien.gv.at/spezial/euro2008/ernst-happel-stadion/</url>
  <id>tag:api.qype.com,2008-03-04:/places/113541</id>
  <updated>2011-08-08T15:30:07+02:00</updated>
  <created>2008-03-04T14:02:28+01:00</created>
  <point>48.2077,16.4197</point>
  <image medium="http://assets0.qype.com/uploads/photos/0288/7033/Wien_014_thumb.jpg" large="
    http://assets0.qype.com/uploads/photos/0288/7033/Wien_014_gallery.jpg" medium2x="http://
    assets3.qype.com/uploads/photos/0288/7033/Wien_014_large.jpg" user_link="http://api.qype.
    com/v1/users/pequenito" user_name="pequenito" small="http://assets3.qype.com/uploads/
    photos/0288/7033/Wien_014_mini.jpg"/>
  <opening_hours>je nach Veranstaltung</opening_hours>
  <phone>+43 1 728 08 54</phone>
  <explicit_content>>false</explicit_content>
  <address>
    <city>Wien</city>
    <house_number>7</house_number>
    <country_code>AT</country_code>
    <postcode>1020</postcode>
    <street>Meiereistra&#223;e</street>
  </address>
  <category>
    <id>tag:api.qype.com,2007-11-06:/places/categories/275</id>
    <updated>2009-12-09T10:22:04+01:00</updated>
    <created>2007-11-06T15:27:07+01:00</created>
    <title lang="de">Stadien & Sporthallen</title>
    <full_title lang="de">Stadien & Sporthallen</full_title>
    <link_rel="self" href="http://api.qype.com/v1/place_categories/275"/>
    <link_rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
    place_categories/7"/>
  </category>
  <category>
    <id>tag:api.qype.com,2007-08-24:/places/categories/7</id>
    <updated>2009-12-09T10:22:03+01:00</updated>
    <created>2007-08-24T12:13:08+02:00</created>
    <title lang="de">Sport</title>
    <full_title lang="de">Sport</full_title>
    <link_rel="self" href="http://api.qype.com/v1/place_categories/7"/>
    <link_rel="http://schemas.qype.com/place_categories.children" href="http://api.qype.com/v1/
    place_categories/7/children"/>
    <link_rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
    place_categories/220"/>
  </category>
</place>
```

```

<category>
  <id>tag:api.qype.com,2007-11-07:/places/categories/303</id>
  <updated>2010-11-11T16:58:48+01:00</updated>
  <created>2007-11-07T12:09:40+01:00</created>
  <title lang="de">Event Location</title>
  <full_title lang="de">Event Location</full_title>
  <link rel="self" href="http://api.qype.com/v1/place_categories/303"/>
  <link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
    place_categories/2"/>
</category>
<link rel="self" href="http://api.qype.com/v1/places/113541"/>
<link rel="alternate" href="http://www.qype.com/place/113541-Ernst-Happel-Stadion-Wien?
  utm_source=api&utm_medium=referral&utm_campaign=1743&utm_term=place
  " hreflang="de"/>
<link count="10" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places
  /113541/reviews/de" hreflang="de"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/113541/
  reviews/en" hreflang="en"/>
<link count="1" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/113541/
  reviews/fr" hreflang="fr"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/113541/
  reviews/es" hreflang="es"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/113541/
  reviews/pt" hreflang="pt"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/113541/
  reviews/pl" hreflang="pl"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/113541/
  reviews/it" hreflang="it"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/113541/
  reviews/nl" hreflang="nl"/>
<link title="Wien" rel="http://schemas.qype.com/locator" href="http://api.qype.com/v1/locators/
  at130-wien"/>
<link count="7" rel="http://schemas.qype.com/assets" href="http://api.qype.com/v1/places/113541/
  assets"/>
<link rel="http://schemas.qype.com/places.nearby" href="http://api.qype.com/v1/positions
  /48.2077,16.4197/places?without=113541"/>
<link rel="http://schemas.qype.com/attributes" href="http://api.qype.com/v1/places/113541/attributes
  "/>
<link count="4" rel="http://schemas.qype.com/checkins" href="http://api.qype.com/v1/places
  /113541/checkins"/>
<link rel="http://schemas.qype.com/checkins_rankings" href="http://api.qype.com/v1/places/113541/
  leaderboard"/>
<link count="0" rel="http://schemas.qype.com/likes" href="http://api.qype.com/v1/likes?likeable_id
  =113541&likeable_type=place"/>
</place>

```

Listing D.2: Qype XML Happel Stadium

D.3 XML Hotel Sacher

```
<?xml version="1.0" encoding="UTF-8"?>
<place>
  <average_rating>4</average_rating>
  <title>Hotel Sacher Wien</title>
  <closed>false</closed>
  <url>http://www.booking.com/hotel/at/sacher-wien.html?aid=337191</url>
  <id>tag:api.qype.com,2009-10-08:/places/984450</id>
  <updated>2011-12-06T10:09:59+01:00</updated>
  <created>2009-10-08T01:19:05+02:00</created>
  <point>48.2039,16.3699</point>
  <image medium="http://assets1.qype.com/uploads/photos/0217/5468/IMG1180_thumb.JPG" large=
    "http://assets1.qype.com/uploads/photos/0217/5468/IMG1180_gallery.JPG" medium2x="http
    ://assets0.qype.com/uploads/photos/0217/5468/IMG1180_large.JPG" user_link="http://api.
    qype.com/v1/users/kyselak" user_name="kyselak" small="http://assets3.qype.com/uploads/
    photos/0217/5468/IMG1180_mini.JPG"/>
  <opening_hours>t&#228;glich ge&#246;ffnet von 08:00 bis 24:00 Uhr
  warme K&#252;che bis 23:30 Uhr</opening_hours>
  <phone>+43 (0)1 - 51 456 0</phone>
  <explicit_content>false</explicit_content>
  <address>
    <city>Wien</city>
    <housenumber>4</housenumber>
    <country_code>AT</country_code>
    <postcode>1010</postcode>
    <street>Philharmoniker-Stra&#223;e</street>
  </address>
  <category>
    <id>tag:api.qype.com,2007-08-23:/places/categories/69</id>
    <updated>2009-12-09T10:21:36+01:00</updated>
    <created>2007-08-23T15:26:04+02:00</created>
    <title lang="de">Hotel</title>
    <full_title lang="de">Hotel und &#220;bernachtung</full_title>
    <link rel="self" href="http://api.qype.com/v1/place_categories/69"/>
    <link rel="http://schemas.qype.com/place_categories.children" href="http://api.qype.com/v1/
    place_categories/69/children"/>
    <link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
    place_categories/5"/>
  </category>
  <category>
    <id>tag:api.qype.com,2007-08-20:/places/categories/1</id>
    <updated>2010-12-14T13:57:02+01:00</updated>
    <created>2007-08-20T18:11:47+02:00</created>
    <title lang="de">Restaurant</title>
    <full_title lang="de">Restaurant</full_title>
    <link rel="self" href="http://api.qype.com/v1/place_categories/1"/>
    <link rel="http://schemas.qype.com/place_categories.children" href="http://api.qype.com/v1/
    place_categories/1/children"/>
  </category>
</place>
```

```

    <link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
      place_categories/883"/>
  </category>
  <category>
    <id>tag:api.qype.com,2007-08-23:/places/categories/73</id>
    <updated>2009-12-09T10:21:14+01:00</updated>
    <created>2007-08-23T15:30:42+02:00</created>
    <title lang="de">Caf&#233;s</title>
    <full_title lang="de">Caf&#233;s</full_title>
    <link rel="self" href="http://api.qype.com/v1/place_categories/73"/>
    <link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
      place_categories/332"/>
  </category>
  <link rel="self" href="http://api.qype.com/v1/places/984450"/>
  <link rel="alternate" href="http://www.qype.com/place/984450-Hotel-Sacher-Wien-Wien?
    utm_source=api&utm_medium=referral&utm_campaign=1743&utm_term=place
    " hreflang="de"/>
  <link count="31" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places
    /984450/reviews/de" hreflang="de"/>
  <link count="21" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places
    /984450/reviews/en" hreflang="en"/>
  <link count="5" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/984450/
    reviews/fr" hreflang="fr"/>
  <link count="5" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/984450/
    reviews/es" hreflang="es"/>
  <link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/984450/
    reviews/pt" hreflang="pt"/>
  <link count="5" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/984450/
    reviews/it" hreflang="it"/>
  <link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/984450/
    reviews/nl" hreflang="nl"/>
  <link title="Wien" rel="http://schemas.qype.com/locator" href="http://api.qype.com/v1/locators/
    at130-wien"/>
  <link count="3" rel="http://schemas.qype.com/assets" href="http://api.qype.com/v1/places/984450/
    assets"/>
  <link rel="http://schemas.qype.com/places.nearby" href="http://api.qype.com/v1/positions
    /48.2039,16.3699/places?without=984450"/>
  <link rel="http://schemas.qype.com/attributes" href="http://api.qype.com/v1/places/984450/attributes
    "/>
  <link count="4" rel="http://schemas.qype.com/checkins" href="http://api.qype.com/v1/places
    /984450/checkins"/>
  <link rel="http://schemas.qype.com/checkins_rankings" href="http://api.qype.com/v1/places/984450/
    leaderboard"/>
  <link count="2" rel="http://schemas.qype.com/likes" href="http://api.qype.com/v1/likes?likeable_id
    =984450&likeable_type=place"/>
</place>

```

Listing D.3: Qype XML Hotel Sacher

D.4 XML Albertina

```
<?xml version="1.0" encoding="UTF-8"?>
<place>
  <average_rating>4</average_rating>
  <title>Albertina, Graphische Sammlung</title>
  <closed>>false</closed>
  <url>http://www.albertina.at</url>
  <id>tag:api.qype.com,2008-06-28:/places/169748</id>
  <updated>2012-01-01T19:44:10+01:00</updated>
  <created>2008-06-28T16:46:44+02:00</created>
  <point>48.2047,16.3683</point>
  <image user_link="http://api.qype.com/v1/users/fiercedragon" medium="http://assets2.qype.com/uploads/photos/0234/7650/-1780008700_thumb.jpg" large="http://assets2.qype.com/uploads/photos/0234/7650/-1780008700_gallery.jpg" user_name="fiercedragon" small="http://assets3.qype.com/uploads/photos/0234/7650/-1780008700_mini.jpg" medium2x="http://assets1.qype.com/uploads/photos/0234/7650/-1780008700_large.jpg"/>
  <opening_hours>Montag – Dienstag, Donnerstag – Sonntag: 10:00 – 18:00
Mittwoch: 10:00 – 21:00</opening_hours>
  <phone>01-534830</phone>
  <explicit_content>>false</explicit_content>
  <address>
    <city>Wien</city>
    <housenumber>1</housenumber>
    <street>Augustinerstra&#223;e</street>
    <country_code>AT</country_code>
    <postcode>1010</postcode>
  </address>
  <category>
    <id>tag:api.qype.com,2007-08-23:/places/categories/64</id>
    <updated>2009-12-09T10:21:12+01:00</updated>
    <created>2007-08-23T15:00:55+02:00</created>
    <title lang="de">Galerien</title>
    <full_title lang="de">Galerie und Kunst</full_title>
    <link rel="self" href="http://api.qype.com/v1/place_categories/64"/>
    <link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/place_categories/4"/>
  </category>
  <category>
    <id>tag:api.qype.com,2007-08-23:/places/categories/27</id>
    <updated>2009-12-09T10:21:13+01:00</updated>
    <created>2007-08-23T14:18:35+02:00</created>
    <title lang="de">Museum und Ausstellung</title>
    <full_title lang="de">Museum und Ausstellung</full_title>
    <link rel="self" href="http://api.qype.com/v1/place_categories/27"/>
    <link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/place_categories/4"/>
  </category>
</category>
```

```

<id>tag:api.qype.com,2007-08-23:/places/categories/30</id>
<updated>2009-12-09T10:21:12+01:00</updated>
<created>2007-08-23T14:19:57+02:00</created>
<title lang="de">Sehensw&#252;rdigkeiten</title>
<full_title lang="de">Sehensw&#252;rdigkeiten</full_title>
<link rel="self" href="http://api.qype.com/v1/place_categories/30"/>
<link rel="http://schemas.qype.com/place_categories.children" href="http://api.qype.com/v1/
place_categories/30/children"/>
<link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
place_categories/4"/>
</category>
<link rel="self" href="http://api.qype.com/v1/places/169748"/>
<link rel="alternate" href="http://www.qype.com/place/169748-Albertina-Graphische-Sammlung
-Wien?utm_source=api&utm_medium=referral&utm_campaign=1743&utm_term=place" hreflang="de"/>
<link count="13" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places
/169748/reviews/de" hreflang="de"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/169748/
reviews/en" hreflang="en"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/169748/
reviews/fr" hreflang="fr"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/169748/
reviews/es" hreflang="es"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/169748/
reviews/pt" hreflang="pt"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/169748/
reviews/pl" hreflang="pl"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/169748/
reviews/it" hreflang="it"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/169748/
reviews/nl" hreflang="nl"/>
<link title="Wien" rel="http://schemas.qype.com/locator" href="http://api.qype.com/v1/locators/
at130-wien"/>
<link count="13" rel="http://schemas.qype.com/assets" href="http://api.qype.com/v1/places/169748/
assets"/>
<link rel="http://schemas.qype.com/places.nearby" href="http://api.qype.com/v1/positions
/48.2047,16.3683/places?without=169748"/>
<link rel="http://schemas.qype.com/attributes" href="http://api.qype.com/v1/places/169748/attributes
"/>
<link count="9" rel="http://schemas.qype.com/checkins" href="http://api.qype.com/v1/places
/169748/checkins"/>
<link rel="http://schemas.qype.com/checkins_rankings" href="http://api.qype.com/v1/places/169748/
leaderboard"/>
<link count="0" rel="http://schemas.qype.com/likes" href="http://api.qype.com/v1/likes?likeable_id
=169748&likeable_type=place"/>
</place>

```

Listing D.4: Qype XML Albertina

D.5 XML Das Rochus

```
<?xml version="1.0" encoding="UTF-8"?>
<place>
  <average_rating>4</average_rating>
  <title>Das Rochus</title>
  <closed>>false</closed>
  <url>http://www.rochus.at</url>
  <id>tag:api.qype.com,2007-03-08:/places/31930</id>
  <updated>2011-11-29T18:18:26+01:00</updated>
  <created>2007-03-08T12:11:48+01:00</created>
  <point>48.2023,16.3914</point>
  <image medium2x="http://assets3.qype.com/uploads/photos/0111/5327/-421595793_large.jpg"
    user_link="http://api.qype.com/v1/users/DJmomo" medium="http://assets0.qype.com/uploads/
    photos/0111/5327/-421595793_thumb.jpg" large="http://assets0.qype.com/uploads/photos
    /0111/5327/-421595793_gallery.jpg" user_name="DJmomo" small="http://assets3.qype.com/
    uploads/photos/0111/5327/-421595793_mini.jpg"/>
  <opening_hours>Mo-Sa: 08.00 - 24.00
</opening_hours>
  <phone>017101060</phone>
  <explicit_content>>false</explicit_content>
  <address>
    <city>Wien</city>
    <housetnumber>55-57</housetnumber>
    <country_code>AT</country_code>
    <postcode>1030</postcode>
    <street>Landstra&#223;er Hauptstra&#223;e </street>
  </address>
  <category>
    <id>tag:api.qype.com,2007-08-23:/places/categories/22</id>
    <updated>2009-12-09T10:21:15+01:00</updated>
    <created>2007-08-23T14:16:31+02:00</created>
    <title lang="de">Bars</title>
    <full_title lang="de">Bars</full_title>
    <link rel="self" href="http://api.qype.com/v1/place_categories/22"/>
    <link rel="http://schemas.qype.com/place_categories.children" href="http://api.qype.com/v1/
    place_categories/22/children"/>
    <link rel="http://schemas.qype.com/place_category.parent" href="http://api.qype.com/v1/
    place_categories/609"/>
  </category>
  <link rel="self" href="http://api.qype.com/v1/places/31930"/>
  <link rel="alternate" href="http://www.qype.com/place/31930-Das-Rochus-Wien?utm_source=api
  &utm_medium=referral&utm_campaign=1743&utm_term=place" hreflang="de"/
  >
  <link count="34" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/31930/
  reviews/de" hreflang="de"/>
  <link count="3" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/31930/
  reviews/en" hreflang="en"/>
</place>
```

```

<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/31930/
reviews/fr" hreflang="fr"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/31930/
reviews/es" hreflang="es"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/31930/
reviews/pt" hreflang="pt"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/31930/
reviews/pl" hreflang="pl"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/31930/
reviews/it" hreflang="it"/>
<link count="0" rel="http://schemas.qype.com/reviews" href="http://api.qype.com/v1/places/31930/
reviews/nl" hreflang="nl"/>
<link title="Wien" rel="http://schemas.qype.com/locator" href="http://api.qype.com/v1/locators/
at130-wien"/>
<link count="4" rel="http://schemas.qype.com/assets" href="http://api.qype.com/v1/places/31930/
assets"/>
<link rel="http://schemas.qype.com/places.nearby" href="http://api.qype.com/v1/positions
/48.2023,16.3914/places?without=31930"/>
<link rel="http://schemas.qype.com/attributes" href="http://api.qype.com/v1/places/31930/attributes"/
>
<link count="18" rel="http://schemas.qype.com/checkins" href="http://api.qype.com/v1/places
/31930/checkins"/>
<link rel="http://schemas.qype.com/checkins_rankings" href="http://api.qype.com/v1/places/31930/
leaderboard"/>
<link count="1" rel="http://schemas.qype.com/likes" href="http://api.qype.com/v1/likes?likeable_id
=31930&likeable_type=place"/>
</place>

```

Listing D.5: Qype XML Das Rochus

Bibliography

- [1] 10gen Inc. <http://www.mongodb.org/>. Accessed: 2011-11-28.
- [2] ACRISS. <http://www.acriss.org>. Accessed: 2011-09-23.
- [3] American Hotel & Motel Association. <http://xml.coverpages.org/hitis.html>. Accessed: 2011-09-23.
- [4] ASC X12. <http://www.x12.org>. Accessed: 2011-09-23.
- [5] Comscore US Audience and Lonely Planet Travellers. Lonely Planet Travellers Pulse Survey. Comscore, 2009.
- [6] Statistics Austria. http://www.statistik.at/web_en/statistics/tourism/. Accessed: 2011-12-28.
- [7] G. Begelman, P. Keller, and Smadja F. Automated tag clustering: Improving search and exploration in the tag space. WWW, 2006.
- [8] Blachere, Booth, and al. Introductory User Guide to DATEX-Net Specifications for Interoperability - version 1.1. Commission of the European Communities, RTD Transport Telematics, Application Programme, 1996.
- [9] BMW. <http://www.bmw.at/de/neufahrzeuge/m/x5-m/2009/connectivity.html>. Accessed: 2012-01-12.
- [10] D. Buhalis. *eTourism: Information technology for strategic tourism management*. Longman, 2003.
- [11] TomTom International B.V. http://www.tomtom.com/en_gb/legal/ts-content-services/. Accessed: 2011-11-16.
- [12] E. F. Codd. A relational model of data for large shared data banks. pages 377–387. ACM, 1970.
- [13] D. Crockford. JSON: The fat-free alternative to XML. XML 2006 Boston, 2006.
- [14] M. Dell’erba, O. Fodor, W. Hopken, and H. Werthner. Exploiting semantic web technologies for harmonizing e-markets. pages 201–219. *Journal of Information Technology and Tourism* 7, 2005.

- [15] Des Newman. <http://www.ozieplorer3.com/>. Accessed: 2011-10-05.
- [16] Deutscher Tourismusverband e.V. <http://www.deutschertourismusverband.de/service/touristische-informationsnorm-tin/vorwort.html>. Accessed: 2011-09-23.
- [17] DIN Deutsches Institut für Normung e. V. <http://www.nadl.din.de>. Accessed: 2011-09-23.
- [18] ECTAA. <http://www.ectaa.org/>. Accessed: 2011-12-02.
- [19] ERTICO. <http://www.cvisproject.org/en/links/karen.htm>. Accessed: 2011-09-23.
- [20] Falk. <http://www.falk.de>. Accessed: 2011-10-06.
- [21] Falk Content & Internet Solutions. <http://maps.falk.de/doku/>. Accessed: 2011-09-09.
- [22] D. Faveur, M. Hackl, M. W. Kuester, and C. Lamsfus. Harmonization of data interchange in tourism. pages 40–50. CEN, 2009.
- [23] R. T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, 2000.
- [24] Flickr. <http://www.flickr.com/help/gettyimages/>. Accessed: 2012-01-12.
- [25] Flickr. <http://www.flickr.com/services/>. Accessed: 2012-01-12.
- [26] O. Fodor and H. Werthner. Harmonise: A Step Towards an Interoperable e-Tourism Marketplace. pages 11–39. International Journal of Electronic Commerce, 2005.
- [27] European Organization for the Safety of Air Navigation. WGS 84 Implementation Manual. EUROCONTROL, 1998.
- [28] Foursquare. <https://de.foursquare.com/about/>. Accessed: 2011-12-02.
- [29] Foursquare. <https://developer.foursquare.com/>. Accessed: 2011-12-03.
- [30] Foursquare. <https://foursquare.com/business/brands>. Accessed: 2012-01-12.
- [31] Garmin International Inc. <http://www.garmin.com>. Accessed: 2011-09-09.
- [32] GoMio. <http://www.gomio.com>. Accessed: 2012-01-13.
- [33] Google. <http://maps.google.com/>. Accessed: 2011-12-30.
- [34] Google. <http://www.panoramio.com>. Accessed: 2011-10-11.
- [35] E. Hammer-Lahav and D. Recordon. The OAuth 1.0 Protocol. <http://tools.ietf.org/html/drafthammeroauth10>, 2010.
- [36] Hedna. <http://www.hedna.org>. Accessed: 2011-09-23.
- [37] HootSuite. <http://hootsuite.com/>. Accessed: 2012-01-13.

- [38] IATA. <http://www.iata.org>. Accessed: 2011-09-23.
- [39] IFITT. <http://www.ifitt.org/home/view/enter-resources>. Accessed: 2012-01-07.
- [40] IFITT Reference Model Special Interest Group. <http://www.rmsig.de>. Accessed: 2011-09-23.
- [41] Google Inc. <http://www.panoramio.com/api/widget/api.html>. Accessed: 2011-11-16.
- [42] Trip Technologies Inc. <http://www.tripbase.com>. Accessed: 2012-01-13.
- [43] International Air Transport Association. <http://www.iatan.org>. Accessed: 2011-09-23.
- [44] International Civil Aviation Organization. www.icao.int. Accessed: 2011-09-23.
- [45] JourneyWeb. <http://www.journeyweb.org/>. Accessed: 2011-09-23.
- [46] Nottingham M. and R. Sayre. The atom syndication format. RFC 4287. Internet Engineering Task Force, 2005.
- [47] P. Membrey, E. Plugge, W. Thielen, and T. Hawkins. *The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing*. Apress, 2010.
- [48] Michelin. <http://dev.viamichelin.com/data/api-jsv2/poi1.htm>. Accessed: 2011-11-22.
- [49] Michelin. <http://www.viamichelin.de>. Accessed: 2011-11-22.
- [50] MMA. <http://www.mma-insurance.com/>. Accessed: 2012-01-13.
- [51] Nespresso. <http://www.nespresso.com>. Accessed: 2012-01-13.
- [52] St. Niedermeier and M. Scholz. *Java und XML: Grundlagen, Einsatz, Referenz*. Galileo Computing, 2006.
- [53] Pixtri OG. <http://www.pixtri.com/wp/>. Accessed: 2012-01-11.
- [54] OpenStreetMaps. <http://www.openstreetmap.org/>. Accessed: 2011-11-24.
- [55] OpenTravel. <http://www.opentravel.org/>. Accessed: 2011-09-23.
- [56] Oracle. <http://docs.oracle.com/javase/1.4.2/docs/api/org/xml/sax/ContentHandler.html>. Accessed: 2011-12-29.
- [57] Oracle. <http://jersey.java.net/>. Accessed: 2011-12-03.
- [58] Rob P. and Coronel C. *Database systems: design, implementation, and management*. Course Technology, 2009.
- [59] Lucian Parfeni. <http://news.softpedia.com/news/Flickr-Boasts-6-Billion-Photo-Uploads-215380.shtml>. Accessed: 2011-12-12.

- [60] Lonely Planet. <http://www.lonelyplanet.biz/product01.htm>. Accessed: 2012-01-12.
- [61] Lonely Planet. <http://www.lonelyplanet.com/legal/intellectual-property>. Accessed: 2011-11-21.
- [62] Qype. http://apidocs.qype.com/_detail/oauth_diagram.pngid=readwriteaccess. Accessed: 2011-09-09.
- [63] Qype. http://apidocs.qype.com/representation_reference. Accessed: 2011-11-24.
- [64] Qype. <http://en.business.qype.com/news/qype-partners-with-bmw/>. Accessed: 2012-01-12.
- [65] Qype. <http://www.qype.com>. Accessed: 2011-09-09.
- [66] Qype. http://www.qype.com/developers/api_terms. Accessed: 2011-11-23.
- [67] L. Richardson and S. Ruby. *RESTful Web Services*. O'Reilly, 2007.
- [68] Salesforce. <http://www.salesforce.com>. Accessed: 2012-01-13.
- [69] M. Scheliga. *CouchDB - kurz & gut*. O'Reilly, 2010.
- [70] Technorati. <http://technorati.com/>. Accessed: 2012-01-13.
- [71] TIH. <http://www.tih.org.uk>. Accessed: 2011-09-23.
- [72] TomTom. http://business.tomtom.com/en_gb/products/webfleet. Accessed: 2011-09-09.
- [73] TomTom. <http://www.tomtom.com>. Accessed: 2011-09-09.
- [74] TopoGrafix. <http://www.topografix.com/gpx/1/1/>. Accessed: 2011-10-04.
- [75] TRAMsoft GmbH. http://www.tramsoft.ch/gps/screenshots/garmin/mapsourc/allgemein/garmin_mapsourcewaypoint_big.jpg. Accessed: 2011-10-04.
- [76] Transmodel. <http://www.transmodel.org/>. Accessed: 2011-09-23.
- [77] Transport Intermodality Data sharing and Exchange Networks. <http://www.transport-research.info>. Accessed: 2011-09-23.
- [78] Travel Technology Initiative Ltd. <http://www.tti.org/>. Accessed: 2011-09-23.
- [79] Tripwolf. <http://www.tripwolf.com/de/page/about>. Accessed: 2012-01-12.
- [80] TTRA. <http://www.ttra.com/>. Accessed: 2011-12-02.
- [81] Twitter. <https://dev.twitter.com/docs>. Accessed: 2011-12-02.
- [82] Twitter. <https://dev.twitter.com/terms/display-guidelines>. Accessed: 2011-12-02.

- [83] Twitter. <http://www.tweetdeck.com/>. Accessed: 2012-01-13.
- [84] Twitter. <http://www.twitter.com>. Accessed: 2011-12-03.
- [85] UK Department of Transport. <http://www.dft.gov.uk/transxchange/>. Accessed: 2011-09-23.
- [86] Union internationale des chemins de fer. <http://www.uic.org/>. Accessed: 2011-09-23.
- [87] United Nations Economic Commission for Europe. <http://www.unece.org/>. Accessed: 2011-09-23.
- [88] United States Tour Operators Association. <http://www.ustoa.com>. Accessed: 2011-09-23.
- [89] W3C. <http://www.w3.org/TR/rdfschema/>. Accessed: 2011-12-29.
- [90] Wanderfly. <http://www.wanderfly.com/>. Accessed: 2012-01-12.
- [91] H. Werthner. Intelligent systems in travel and tourism. pages 1620–1628. Proceedings of the 18th International Joint Conference on Artificial Intelligence. Acapulco, Mexico, 2003.
- [92] H. Werthner and S. Klein. *Information Technology and Tourism - A Challenging Relationship*. Springer Verlag, 1999.
- [93] Wikipedia. http://en.wikipedia.org/wiki/Document-oriented_database. Accessed: 2011-11-28.
- [94] T. Wilson, D. Burggraf, R. Lake, S. Patch, B. McClendon, M. Jones, M. Ashbridge, B. Hagemark, J. Wernecke, and C. Reed. *KML 2.2 An OGC Best Practice: Open Geospatial Consortium*, 2007.
- [95] World Association of Travel Agencies. <http://www.wata.net>. Accessed: 2011-09-23.
- [96] WTO. www.wto.org/. Accessed: 2011-12-02.
- [97] WTTC. <http://www.wttc.org/>. Accessed: 2011-12-02.
- [98] Yahoo. <http://www.flickr.com/services/api/>. Accessed: 2011-12-04.
- [99] J. Yang, Y. Matsuo, and M. Ishizuka. Triple tagging: Toward bridging folksonomy and semantic web. ISWC07, 2007.