SWS Student Projects

All students should publish their implementations as open source projects. Please note that, a production level software is not expected to successfully complete the project assignments; a working prototype will be sufficient. Please produce clean code with comment lines when explanation is necessary. Please pay attention to the restrictions due to licensing of any third-party library you use. For further questions, I will be available via E-mail and in our proseminar hours.

1) JSON-LD Generator

A form-based generator to create JSON-LD blocks in the context of schema.org. Although the focus of the project is touristic services, the generator must be extensible with schema.org concepts from other domains. The form interface should be adaptive to the selected concept. (e.g. If the user wants to create a JSON-LD block about an offer, the form must provide the means of input for the properties whose domain includes http://schema.org/Offer. In other words, the interface should guide the user.)

Input: User input through a form based interface
Conditions: The form interface should be adaptive to the concept for which the user wants to create the data. Domains and ranges of properties should be considered. The generator must be extensible.
Goal: Structured data in JSON-LD format in schema.org context

2) Service Packaging from External Sources

a) **Scraping schema.org annotations from web pages**
Create a software that takes a website with schema.org annotations in RDFa, JSON-LD or Microdata and extract the structured data. The extracted data should be stored and integrated in a triple store. Serve the integrated data as an API. (e.g. Imagine a scenario that different information about an offer from a hotel can be on different websites. Your software should extract the structured data from both pages and integrate them.)

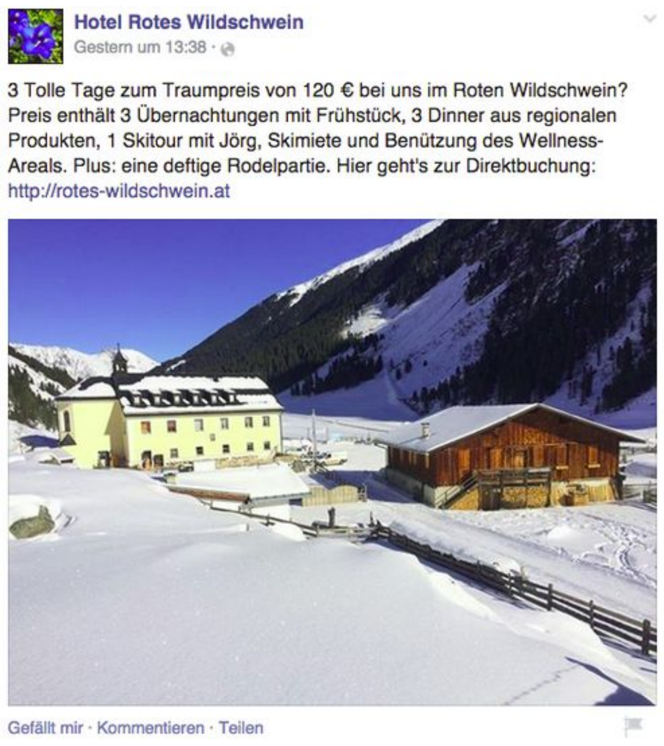
Input: A JSON-LD file or a website that contains structured data.
Conditions: Integrated data should be stored in a triple-store. Recognizing JSON-LD blocks is mandatory. RDFa and Microdata is desirable.
Goal: An API that serves the integrated data as JSON-LD with potential actions on the entities.

b) **Using data from external linked data sources**
Aim of this project is to provide an API that serves the linked data of the touristic offers in Salzburg. The existing data can be enriched with instances from DBPedia, weather data etc.

Input: Data from existing external sources
Conditions: We will provide a data dump that contains touristic offers from Salzburg. Linked data should be stored in a triple store and data from at least one other linked dataset should be integrated.
Goal: An API that serves the integrated data as JSON-LD with potential actions on the entities.

3) Onlim API

Onlim API is a tool that touristic service providers utilize to generate social media posts by injecting data about their offers into some templates. This API should retrieve data from the services provided by the project number 2 and replace some placeholders in a template text to generate posts for Facebook and Twitter.
An example post is shown below.



Hotel Rotes Wildschwein
Gestern um 13:38 ·

3 Tolle Tage zum Traumpreis von 120 € bei uns im Roten Wildschwein? Preis enthält 3 Übernachtungen mit Frühstück, 3 Dinner aus regionalen Produkten, 1 Skitour mit Jörg, Skimiete und Benützung des Wellness-Areals. Plus: eine deftige Rodelpartie. Hier geht's zur Direktbuchung: http://rotes-wildschwein.at

Gefällt mir · Kommentieren · Teilen

Moreover, ONLIM API should serve the offers as JSON-LD in schema.org context with suitable schema.org actions based on the potential actions defined on entities (see the description for the project number 2)

Input: JSON-LD data retrieved from the service packaging API
Conditions: ONLIM API should be able to retrieve offers from the packaging service by date, the organization that makes the offer, location. Students are welcome to add additional filters.
Goal: Creating Facebook and Twitter posts based on templates and JSON-LD data with schema.org actions.

4) CMS Extensions for Schema.org Annotations

An extension/plug-in/add-on for prominent CMS applications in order to annotate content with touristic services related concepts. Following the annotation, the extension should create JSON-LD blocks, RDFa and Microdata snippets. Generated scripts should be either placed in the content page or linked from the content page to a JSON-LD file in a separate location. Annotation can be done via the content editor or inline.

Inline editing example: http://createjs.org/demo/hallo/

Input: Content to be annotated
Conditions: Annotations can be done via the content editor or inline. The creation of JSON-LD blocks is mandatory. RDFa and Microdata is desirable.
Goal: Creating JSON-LD blocks placed in content pages or JSON-LD blocks placed in a separate location and linked from the content page

5) Mobile Client for Touristic Services

A client that retrieves offers from the touristic packaging application, based on external user input (e.g. date, type of the offer, price range) and user’s contextual information (geolocation, social media activity).

Input: User profile information (e.g. social media activity, location) and explicit user input such as date, price range and type of the offer.
Conditions: The client should retrieve the data from the packaging service.
Goal: Providing a list of offers to the user on a mobile device based on the criteria.

Umutcan Simsek