

# How to Use SWEET

## (Lightweight Version)



### Table of Contents

1. INTRODUCTION	2
2. DOCUMENTATION	3
2.1 INSTALLATION AND CONFIGURATION	3
2.1.1 <i>How to use SWEET</i>	3



## 1. Introduction

SWEET is a Web application-based MicroWSMO editor, which is implemented in two versions. The first version takes the form of a vertical widget displayed within a Web browser. It is very lightweight and can be used to directly annotate RESTful service descriptions visualized in the browser window. The second implementation is part of a fully-fledged dashboard application developed within the SOA4All<sup>1</sup> project, which supports users in performing different tasks related to the service lifecycle. It has some additional functionalities and is more stable. Both SWEET implementations share common main functionalities:

- Insertion of hRESTS microformat tags in the HTML service descriptions in order to mark service properties (operations, address, HTTP method, input, output and labels).
- Integrated ontology search for linking semantic information to service properties.
- Insertion of MicroWSMO model reference tags, pointing to the associated semantic meaning of the service properties.
- Saving of semantically annotated HTML RESTful service description.
- Extraction of MicroWSMO service descriptions based on the annotated HTML.

Each of SWEET's implementations addresses different user needs and supports different use cases. The browser-based lightweight version is suitable for users who are browsing for RESTful services and want to directly annotate the currently displayed description. Whenever a user wants to create MicroWSMO annotations of the currently viewed service, he/she can simply click on the bookmarklet of SWEET, initiate it and directly use it.

---

<sup>1</sup> <http://www.soa4all.eu/>



## 2. Documentation

### 2.1 Installation and Configuration

The here described version of SWEET requires the installation of the Firefox<sup>2</sup> browser, version 3.0 or later, with installed and enabled Firebug<sup>3</sup> Firefox add-on, which supports the communication between the website content and the JavaScript-based MicroWSMO editor implementation. The installation of the tool itself is very simple. It is done by dragging the SWEET link and dropping it in the bookmark panel of Firefox. After this, the tool can be directly started by click on the bookmarklet and no additional installation is required. The current version is available at <http://sweetdemo.kmi.open.ac.uk/>.

#### 2.1.1 How to use SWEET

SWEET is a web application-based MicroWSMO editor (Figure 1), which takes the form of a vertical widget appearing on top of the currently viewed webpage. It consists of four main panels, including the *hRESTS Tags* panel, the *Service Properties* panel, the *Domain Ontologies* panel and the *Annotations* panel. Each of these panels is described in detail in the following sections.

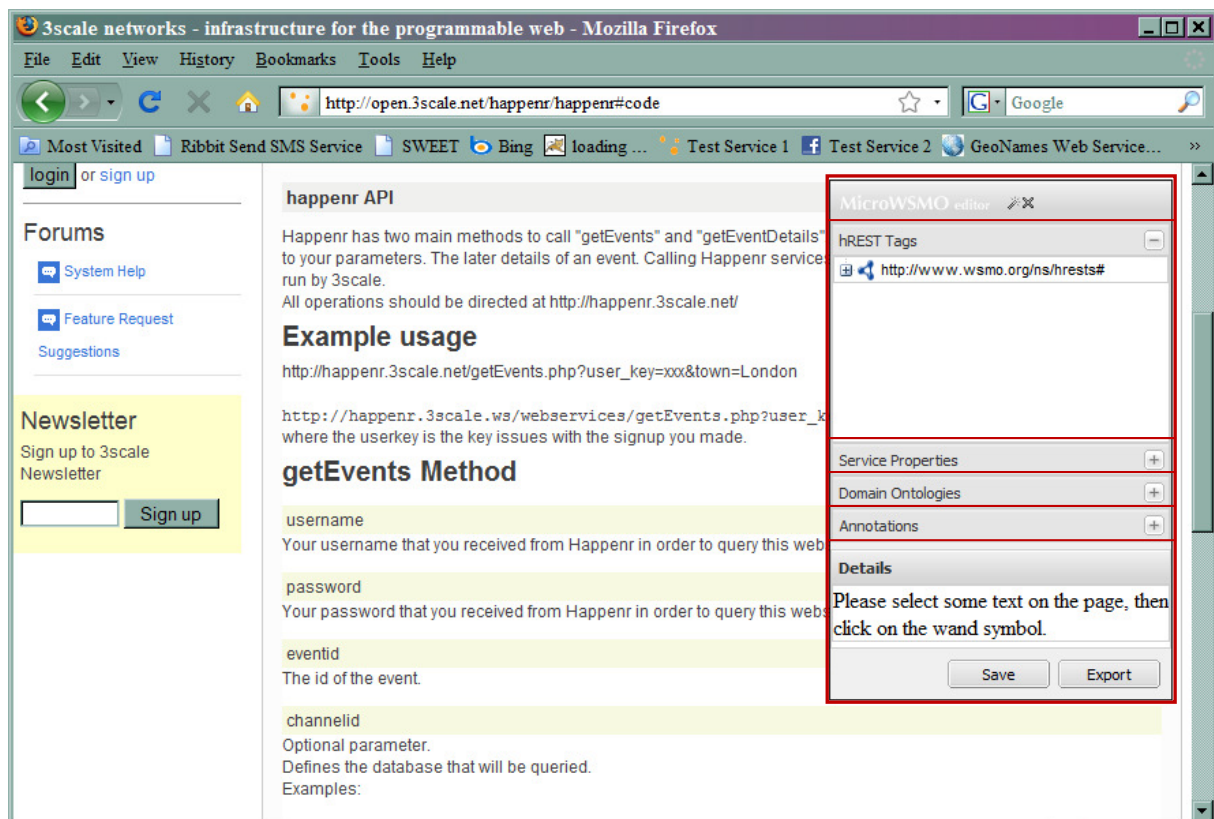


Figure 1: Lightweight SWEET version

The GUI has also a *Save* and an *Export* button as well as two icons for searching for domain ontologies and deleting made annotations. As it can be seen, the implementation is very lightweight and contains only a few elements. Still these elements provide the core functionalities needed in order to support the user in annotating RESTful service descriptions.

<sup>2</sup> <http://www.mozilla-europe.org/en/firefox/>

<sup>3</sup> <http://getfirebug.com/>

The common tasks of the process of annotating RESTful services are:

1. Identifying service properties by inserting hRESTS tags in the HTML service description.
2. Searching for domain ontologies suitable for annotating the service properties.
3. Annotating service properties with semantic information.
4. Saving or exporting the annotated RESTful service.

Each of these tasks is described in detail and is visualized by a screenshot of the SWEET.

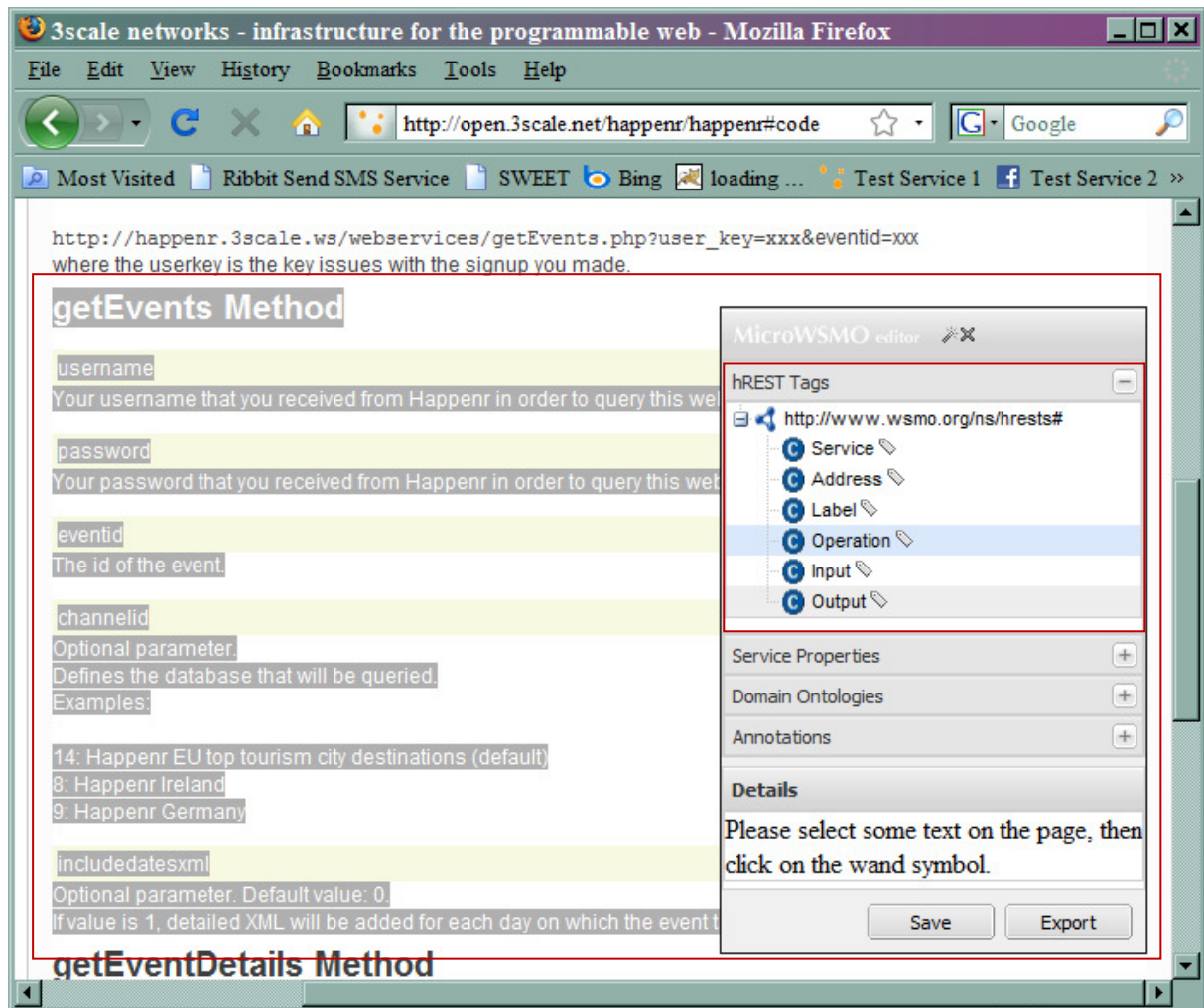


Figure 2: Inserting hRESTS Tags

### Identifying service properties by inserting hRESTS tags

This version of SWEET is designed for use cases when a user is browsing the Web and finds an interesting RESTful service, which he/she wants to semantically annotate. The tool can simply be started by clicking on a bookmarklet, which executes the JavaScript initializing the web application. After the tool is started, the user can begin to insert hRESTS tags for marking the service properties.

In contrast to services described in WSDL, which have a clear structure and definition of the operations, inputs and outputs, RESTful service descriptions are commonly given in plain text. Therefore, it is necessary to somehow mark the service properties in the text description. The MicroWSMO standard specification does this by introducing the hRESTS microformat. The user can insert hRESTS tags by selecting the part of the description related to a particular property and clicking on the appropriate tag. Figure 2 visualizes how an

operation can be marked. The *getEvents* operation content is selected and is marked by clicking on the *operation* node in the hrests tree in the hRESTS Tags panel. As a result the HTML tag class="operation" is inserted in the corresponding HTML <div> element. In this way, the service properties are recognizable and machine processable. Naturally, in order to create a meaningful service structure, the user should start by marking the complete service body, followed by the operations, the inputs and outputs of the operations and the corresponding address. The user can also mark the names of the service and of the operations as labels, in order to make the resulting MicroWSMO description more user-friendly. After the user builds up the service structure by marking all service properties with hRESTS tags, he/she can continue by searching for suitable domain ontologies for semantically annotating the service.

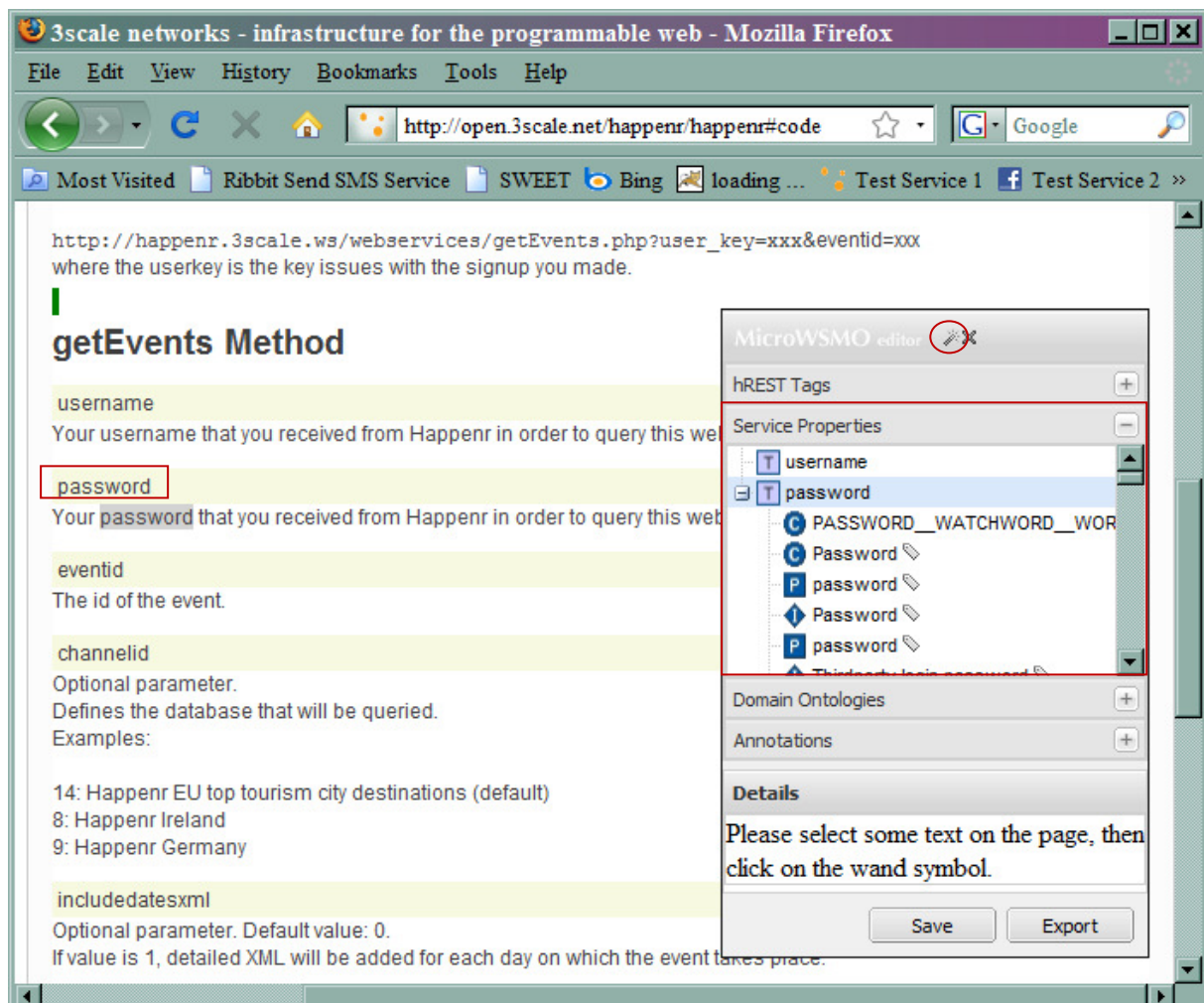


Figure 3: Searching for Domain Ontologies

### Searching for suitable domain ontologies

The user can start searching for suitable domain ontologies by selecting a service property, for example the input parameter “password” and clicking on the magic wand symbol. This sends a request to Watson’s API <sup>4</sup>for retrieving matching ontologies. The result is a tree structure, where each of the nodes represents the matching concept, property or instance and the corresponding ontology. For example, as it can be seen in Figure 3, the search based on “password” returned a concept with label “PASSWORD\_WATCHWORD\_WOR”,

<sup>4</sup> <http://watson.kmi.open.ac.uk/>

followed by a “Password” concept, followed by a “Password” property, followed by a “Password” instance, etc.

However, the user needs some additional information in order to be able to decide if this semantic annotation is suitable for the service property. This additional information is provided by clicking on the matching node, which expands and provides details about the concept and the corresponding ontology. The user can receive more information about each of the ontologies by switching to the *Domain Ontologies* panel (seen in Figure 4), which includes a list of all concepts as well as a list of all the matching properties, which were found for this ontology. For example, the ontology starting with “<http://csd.abdn.ac.uk/research/>” is a match for both the password and username input parameters and contains a list of concepts, which can be viewed by clicking on the “All concepts” folder. Based on this information the user can decide to associate a particular service property with semantic information.

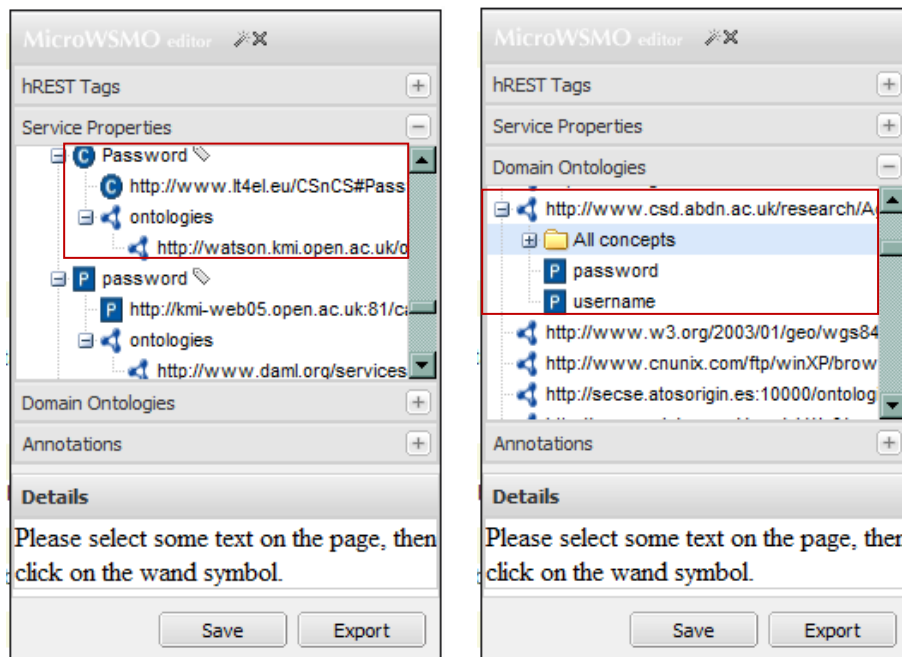


Figure 4: Choosing a Domain Ontology

### Annotating service properties

The annotation of service properties can be easily done by selecting the service property and clicking on the corresponding semantic concept. For example, as it can be seen in Figure 5, “password” is automatically highlighted in red and can be associated with the “password” concept by clicking on it. As a result, the `rel="model"` and `href` HTML tags are inserted and the modified HTML is the following:

```
<h3 id="nid2" background-color="lightgrey" rel="model" href="http://www.daml.org/services/owl-  
s/1.1/Concepts.owl#Password">password</h3>
```

The user can annotate as many service properties as necessary and use only one or different ontologies. The resulting HTML service annotation is in accordance with the MicroWSMO specification and enables the automatic extraction of RDF MicroWSMO service descriptions from the annotated HTML.

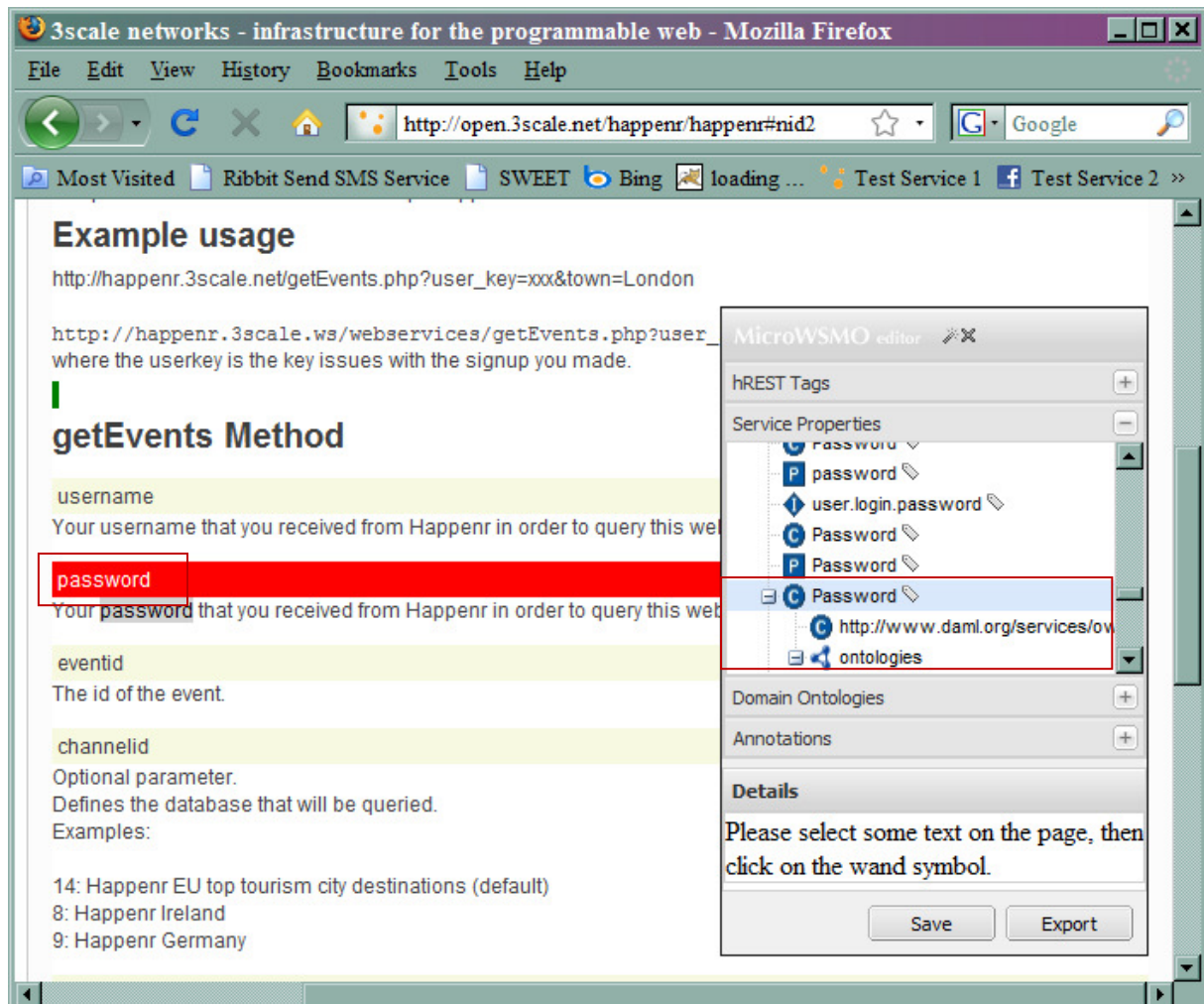


Figure 5: Making Semantic Annotations

### Saving or exporting the annotated RESTful service

After the insertion of hRESTS tags is complete and the required service properties are linked to semantic information, the resulting annotated HTML can be saved. The data can be stored locally by the user or be posted directly in a service repository. The annotated HTML can be used for extracting RDF MicroWSMO service descriptions, which can easily be done by clicking on *Export*.