chem-bla-ics

Programming in the Life Sciences #9: APIs and Web Services

Published October 30, 2013

Citation
Willighagen, E. (2013, October 30). Programming in the Life Sciences #9: APIs and Web Services. *Chem-bla-ics*. https://doi.org/10.59350/7skws-4f170

Keywords
Pra3006

Abstract
Continuing on the theory covered in this course, this part will talk about application programming interfaces (APIs) and web services.

Continuing on the theory covered in this course, this part will talk about application programming interfaces (APIs) and web services.

Application Programming Interfaces

APIs define how programs can be used by other programs. An API defines how methods are called and what feedback you can expect. It basically is the combination of documentation and the program itself. But, unlike any piece of software, an API is aimed at users, rather than use in the same program. The API is how you communicate between programs.

Now, in this course we will see two key types of APIs. The first are the APIs provided by the libraries that we use. For example, we already indicated that we will be using at least the following two libraries, ops.js and d3.js. These libraries are a collection of functional bits (e.g. classes and methods). For example, ops.js defines an API which wraps closely the Open PHACTS Linked Data API (LDA) itself. The API requires as to do a few things: 1. create a wrapper for the LDA; 2. define a call back function; 3. invoke the actual

```
call.var searcher = new Openphacts.ConceptWikiSearch(
  "https://beta.openphacts.org", appID, appKey
);
var callback=function(success, status, response){
    searcher.parseResponse(response);
};
searcher.findCompounds('Aspirin', '20', '4', callback);
```

Web Services

Web services are a special kind of APIs: they expose an API over the web. That imposes some features of these APIs: first, they are based on a web transport layer, commonly HTTP, but XMPP is possible too. HTTP is used by your web browser too. Secondly, the web server needs a common communication language to serialize the method call. Here, two key standards are used, XML and JSON. We will cover these in more detail later. For now, it suffices to think of these as envelopes in which are message is sent. Now, another aspect standardized is how to call the web services. For that, SOAP and REST are the most important standards for the life sciences (though I still think Wagener's XMPP approach is still worthwhile checking out!). SOAP and REST use XML and JSON are underlying serialization format.

So, web services are theoretically complex. For this course, most of it is hidden by the client library that will take care of the HTTP and SOAP/REST layers. The students who wish to use Java instead of JavaScript, will face the problem that you first need to find a Java client library for the LDA. There is this library, but that needs exploring for use with the latest Open PHACTS LDA. Higher stakes, higher rewards.

Take home message

Practically, you do not need to know much of the technologies behind web services, just like you do not need to know machine instructions CPUs follow to run your program. But, it is important to have seen these terms. You will run into them, and need enough context to know where and how to find answers to the questions that you will have.

There is one exception: JavaScript Object Notation, JSON. That is the format in which the data is returned by the service, and you will have to handle that. JSON will be the topic of the next post.