

Programming in the Life Sciences #2: accounts and API keys

Egon Willighagen 

Published October 8, 2013

Citation

Willighagen, E. (2013). Programming in the Life Sciences #2: accounts and API keys. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/hb7ye-mzp21>

Keywords

Pra3006, Openphacts, Javascript, Rest

Abstract

I have outlined the scope of the six-day course: the students will learn to program while hacking on the Open PHACTS' Linked Data API (LDA). The first step is to get an account for the LDA. I have already done that to save time. But these are the steps to take.

Copyright

Copyright © Egon Willighagen 2013. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

chem-bla-ics

I have outlined the scope of the [six-day course](#): the students will learn to program while hacking on the [Open PHACTS' Linked Data API \(LDA\)](#). The first step is to get an account for the LDA. I have already done that to save time. But these are the steps to take. You go to <https://dev.openphacts.org/signup>:

Open PHACTS [Sign in](#) | [Sign up](#)

Developer Home | Want help? | Documentation | Get my API keys! | Featured Apps | Workflow

Signup

User Information

Username *

Email *

First Name *

Last Name *

Password*

Password confirmation *

Contact Details

Organization/Group Name *

By signing up you agree to the following Legal Terms and Conditions ([show](#))

For Heterm

Type the text [Privacy & Terms](#) **ReCAPTCHA**™ stop spam. read books.

© 2013 Powered by scale

You then approve the account via your email account and you are set. The account is needed to get an API key. Using this key, Open PHACTS developers can contact you if your scripts go berserk So, you are kindly invited to make crazy hypotheses and hack the hell out of the platform. That's what I hope my students will do.

To try your new key, go to the documentation page, and open, for example, the *SMILES to URL* method:

chem-bla-ics

SMILES to URL /structure GET

Description
Returns a ChemSpider URL corresponding to an input SMILES string. Driven by ChemSpider.

Response template:

?primaryTopic `chemspider.smiles ?smiles`

PARAMETER	VALUE	DESCRIPTION
app_id	<input type="text"/>	Your access application id
app_key	<input type="text"/>	Your access application key
smiles	<input type="text" value="(required)"/>	A SMILES string. E.g. <chem>CC(=O)Oc1ccccc1C(=O)O</chem>
_format	<input type="text" value="▼"/>	The desired result format.
_callback	<input type="text"/>	For JSONP
_metadata	<input type="text" value="▼"/>	Additional metadata to be included with response.

Here you can see what parameters this LDA method has. We focus now on the `app_id` and `app_key` fields. Each account comes by default with a, um, default `app_id` and default `app_key`. Just click on the field and select them:

PARAMETER	VALUE	DESCRIPTION
app_id	<input type="text"/>	Your access app
app_key	<input type="text"/>	First application key from t Proprietary Unit Tests 6d9 Default 07632cb073eaa8
smiles	<input type="text" value="(required)"/>	
_format	<input type="text" value="▼"/>	

Select the defaults and enter a SMILES (try: CC(=O)NC1=CC=C(C=C1)O). You can select the format you like (I like Turtle) and you get Linked Data back on this [compound](#).

Now, go explore the LDA methods.