SciVerse Applications For Developers

OpenSocial for Science - A SciVerse Primer: MySimpleSearch (Code Example)

A Content API call to retrieve data from ScienceDirect

XML Definition File

SciVerse applications are specified in an xml definition file. SciVerse applications require a "profile" view to be defined, although the xsd schema does not enforce it.

This example gadget demonstrates an application on the SciVerse platform using the SciVerse APIs. MySimpleSearch.xml is the xml definition file for the MySimpleSearch example gadget.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Module>
 <ModulePrefs title="MySimpleSearch">
   <Require feature="opensocial-0.9" />
 </ModulePrefs>
 <Content type="html" view="profile">
  <![CDATA[
   <style type="text/css">
   /* your app's CSS goes here */
   </style>
   <script type="text/javascript">
   // your app's javascript goes here
   </script>
   <!-- your app's html goes here -- >
  ]]>
 </Content>
</Module>
```

Figure 1: XML Definition File for SciVerse Gadget

MySimpleSearch JavaScript

The main code for SciVerse applications is written using JavaScript. When the search results page on SciVerse hub is loaded, the browser also loads the applications that appear on the page in the application toolbar. In your app's javascript you can use opensocial's registerOnLoadHandler function to trigger a particular javascript function.

```
gadgets.util.registerOnLoadHandler(init);
```

Figure 2: JavaScript OpenSocial API gadgets.util.registerOnLoadHandler

Authorization and Authentication

The Sciverse platform uses OAuth (Open Authorization) to hand out tokens (or API keys) for identification via APIs instead of requiring user credentials for the application to access Elsevier APIs. By registering as a SciVerse developer and creating an application, you are given an API key for your app.

When your application is loaded in the browser, the server sends in addition a security token to the application for the duration of the session. This security token or authToken can be retrieved from the app's ContextInfo. When making a call to search or retrieve content via the SciVerse APIs, the application must set both the API key and the authToken in the request headers of the request.

```
function init(){
  gadgets.sciverse.getContextInfo(initCallback);
}
function initCallback(initResponse){
  var context = initResponse;
  var authtoken = context.secureAuthtoken;
  var apikey = "1b11a8765e22vgh09aaa000b4444c8f";
  var requestHeaders ={};
  requestHeaders['X-ELS-APIKey'] = apikey;
  requestHeaders['X-ELS-Authtoken'] = authtoken;
  var view = "COMPLETE";
  var url = "http://api.elsevier.com/content/search/"+
            "index:SCIDIR?query="+searchterm+"&count="+
            nrOfResults+"&view="+view;
  var params ={};
  param[gadgets.io.RequestParameters.HEADERS] =
        requestHeaders;
  gadgets.sciverse.makeRequest(url, scidirCallback, params);
}
function scidirCallback(scidirResponse){
  // process response
}
```

Figure 3: The Use of API-Key and AuthToken in a SciVerse Content API Call

Using JSON and jQuery to Parse the Response Object

The Content API by default returns JSON. You can parse the response object to JSON and use jQuery to extract for instance the DOI (Digital Object Identifier) of the article, which you can then can use to retrieve meta-data about the author (for example his affiliation or the number of citations) from the Scopus cluster.

```
var textJson = gadgets.json.parse(scidirResponse.text);
var entries = textJson['search-results']['entry'];
$.each(entries, function(i1, entry){
  var title = entry['dc:title'];
  var articleDOI = entry['prism:doi'];
});
```

Figure 4: The Use of jQuery to parse JSON data

Integration Points

Integration points are the areas where applications appear in SciVerse. In the xml definition file, the integration points are referenced as “views”. One application can define a number of different views: profile (required), canvas (optional), or sciVerseResultsView (optional)

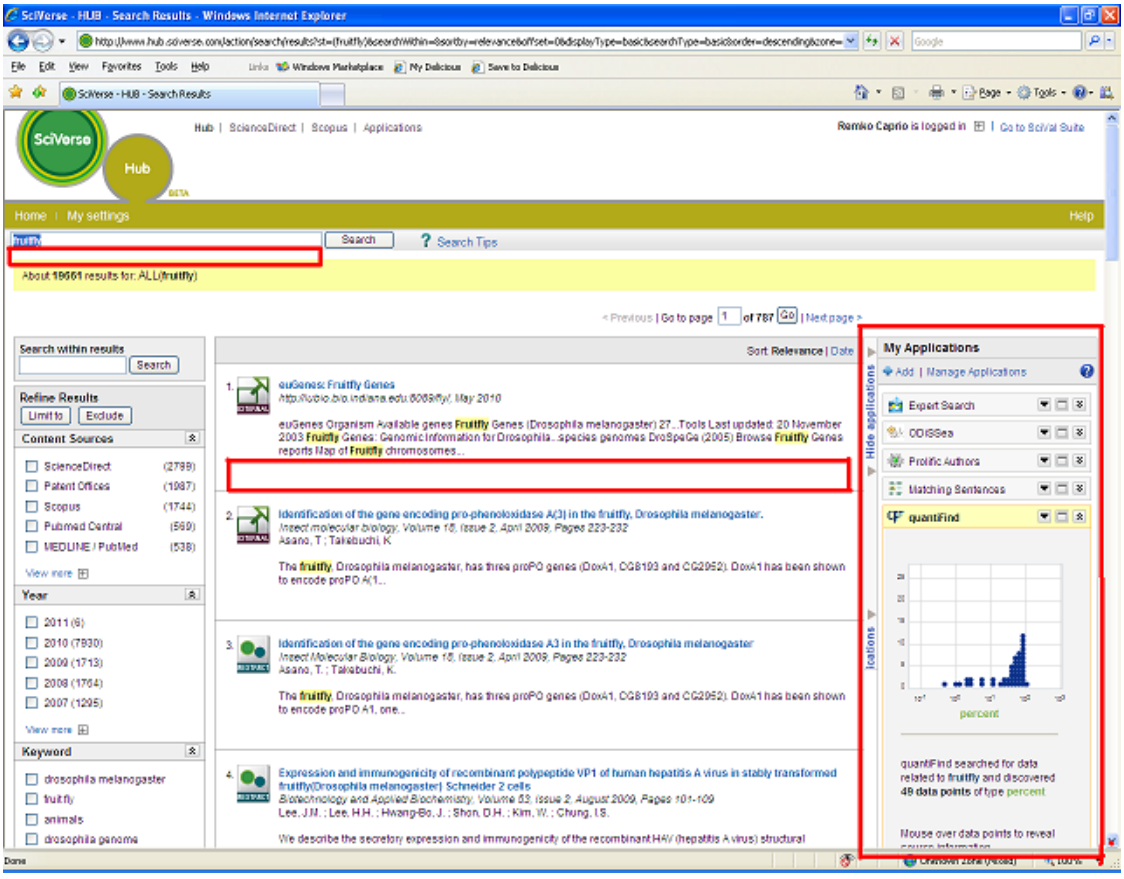

Figure 5: Integration Points in SciVerse Hub Search Results Page

Source Code

The source code for MySimpleSearch can be downloaded from GitHub:

https://github.com/sciversedev/examples/blob/master/contentapi/SciVerseExamples-MySimpleSearch1.xml

Resources

Developer portal:
http://developer.sciverse.com

Application Gallery:
http://www.applications.sciverse.com

General Information:
http://www.info.sciverse.com

SciVerse Hub:
http://www.hub.sciverse.com

SciVerse ScienceDirect:
http://www.sciencedirect.com

SciVerse Scopus:
http://www.scopus.com

Start Building your App
http://developer.sciverse.com

Follow us:
http://twitter.com/sciversedev

Developer Blog
http://developer.sciverse.com/blog

For questions, contact us at:
developer@elsevier.com

For more information please visit: developer.sciverse.com

Copyright © 2010 Elsevier B.V. All rights reserved. SciVerse® is a registered trademark of Elsevier Properties S.A., used under licence.