Introduction to SciVerse Applications

Ryan Dietz
ryan.dietz@elsevier.com
What is SciVerse

• Platform to bring together all Elsevier content
  o SciVerse ScienceDirect – full-text articles database (10 million full-text articles, 15 thousand ebooks, & 15 thousand multimedia files)
  o SciVerse Scopus – abstract and citations database (41 million records)
  o SciVerse SciTopics* – a knowledge-sharing community of experts in various fields
  o SciVerse Hub* – a one-stop single search across ScienceDirect, Scopus, SciTopics, and targeted web content
  o SciVerse Applications* – marketplace allowing academic and professional developers to collaborate with researchers to build applications to improve search and discovery

* beta version
SciVerse Applications – The Marketplace

SciVerse Applications Beta

SciVerse Applications Beta lets you integrate search and discovery applications into SciVerse, to help you be more productive in your research. Login or register, find an application and get started - there is nothing to download or install, the applications you've selected will appear immediately within SciVerse.

Developers can create applications for over 15 million SciVerse users worldwide. SciVerse Applications Beta lets you integrate your application directly into the core SciVerse user experience on article, record and search results pages. To learn more, please visit the Developer Network.

SciVerse Applications Beta has just launched and we continue to make improvements. We welcome your feedback on all aspects of this service.

Featured Applications

Expert Search
SciVerse Hub and ScienceDirect
Find experts in Computer Science and view profiles based on information of Arnetminer.org.

eReader Formats
ScienceDirect
Convert articles into ePub or Mobi and select formats for your mobile device.

HealthMash
SciVerse Hub
Find relevant biomedical concepts based on your search term.

Browse applications
SciVerse Applications – The Marketplace

Application Gallery

Choose from many applications available to help you with your research. Recent additions are listed first.

**ODISSea**

**5 stars (1 review)**

SciVerse Hub

Expand your query with standard ontologies and search public data resources via NCBO (National Center for Biomedical Ontology) for additional information on clinical trials, genes, drugs and funding.

**Mendeley Readers**

Be the first to review this application!

SciVerse Hub

Allows you to see how many users of Mendeley have added a ScienceDirect article to their library and information about those users.

**Expert Search**

**5 stars (1 review)**

SciVerse Hub, ScienceDirect

Find experts in Computer Science and view profiles based on information of Arnetminer.org.

**My Oncology Articles**

Be the first to review this application!

SciVerse Hub

Track a summary of the latest oncology articles published in Science Direct. Researchers can identify and track new oncology publications of interest using filters such as tumor and article types.
# My Projects

The My Projects page provides you with a list of all of your applications. The Status column describes what stage your application is in: In Development, Pending Approval or Approved.

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Gadget 2</td>
<td>In Development</td>
</tr>
<tr>
<td>Test Gadget</td>
<td>In Development</td>
</tr>
</tbody>
</table>

**Want to build a new application?**

By clicking “Start a New Project” below, you will begin the process of creating a new SciVerse application.

[Start a New Project]
SciVerse Applications – Edit Project

Edit Project

The Edit Application page is where you will enter information about your application and eventually submit your application for review. Below, you will find the API key which you will require to utilize the SciVerse content APIs.

After updating information below, click Save changes or Submit application for review to add it to the application gallery (* = required field for submitting applications)

Basics

Application Name: Hello World

API Key: 38bb7df1d08e56ba33505e2ac26e35e10e

Application ID: 212259

Version: 

Status: In Development

Type of Application:

○ Free
○ Fee Based

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<module>
  <module.name title="Hello World">
    <author.email>draft@your.domain</author.email>
    <require.feature>universal</require.feature>
    <require.feature>ideas</require.feature>
  </module.name>
</module>
```

Test Your Application

Please upload your XML definition file and choose your integration locations, and click "Test in SciVerse" to view your application in SciVerse or "Test in gallery” to view your application in the Application Gallery.

Test in SciVerse

Test in gallery
What are SciVerse Applications

• **JUST GADGETS!!!**
  - Simple HTML and JavaScript applications
  - Embedded in webpages and other apps.

• Gadgets API is a part of the OpenSocial specification
  - iGoogle, LinkedIn, MySpace

• Apache Shindig
  - Reference implementation of OpenSocial API v0.9
  - “Contains” the gadgets on a page
  - Provides basic services like rendering, proxying requests
  - Provides plumbing for developing a framework and framework API’s for gadgets

http://code.google.com/apis/gadgets
http://www.opensocial.org
http://shindig.apache.org
BUT… What ARE SciVerse Applications

• SciVerse Applications can interact with the SciVerse products that contain them

• SciVerse Application Framework – extension of Shindig container
  – Provides access to local data (context information)
  – Provides mechanism for adding UI elements to the SciVerse products
  – Allows developers to integrate their content, data, and tools directly within SciVerse products at key points (integration points).

• SciVerse Content APIs – RESTful APIs that provide direct access to Elsevier content
Where are SciVerse Applications

• Integration points
  – Specific locations in the SciVerse product suite where SciVerse Applications appear

• Views
  – Defined in the gadget specification
  – Represents how the gadget will look
  – In SciVerse, views correspond to multiple Integration points
  – 3 Views supported in SciVerse (profile, canvas, sciverseResultsView)
# Integration Points and Views

<table>
<thead>
<tr>
<th>View</th>
<th>Definition</th>
<th>Integration Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>View when first rendered. Toolbar view. (required)</td>
<td><strong>Hub</strong>- Home page, Search Results page</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ScienceDirect</strong>- Full Text Article page, Search Results page</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Scopus</strong>- Record page</td>
</tr>
<tr>
<td>canvas</td>
<td>View when rendered maximized; covers the full page except header and footer</td>
<td>Opens as the full page wherever the application can be maximized.</td>
</tr>
<tr>
<td>sciverseResultsView</td>
<td>View when rendered under a result in a result set.</td>
<td><strong>Hub</strong>- Search Results page (under each result)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>ScienceDirect</strong>- Search Results page (under each result)</td>
</tr>
</tbody>
</table>
Integration Points – Hub Home Page

- Context free apps (not related to page content) run in grid style toolbar

profile View of each app
Integration Points – Hub Search Results

- Application toolbar along right side of search results (‘profile’ view)
- Applications can display ‘sciverseResultsView’ below each search result.
Integration Points – SD Search Results

- Application toolbar along left side of search results (‘profile’ view)
- Applications can display ‘sciverseResultsView’ below each search result.
Application toolbar along right side of article (‘profile’ view)

• Apps can retrieve the contents of the article
Integration Points – Scopus Abstract

Profile View of each app

• Application toolbar on right side of abstract (‘profile’ view)
Integration Points – Canvas View

- Any application can be “maximized” from the application toolbar
- “canvas” view displayed

The authors of this article have a total of 19 records in Scopus:

Ng, D.C.H., Ng, I.H.W., Yeap, Y.Y.C., Badrian, B., Tsoutsman, T., McMullen, J.R., Sersarian, C., Bogoyevitch, M.A.

Opposing actions of extracellular signal-regulated kinase (ERK) and signal transducer and activator of transcription 3 (STAT3) in regulating microtubule stabilization during cardiac hypertrophy
(2011) Journal of Biological Chemistry

Norris, A.J., Foeger, N.C., Neronne, J.M.

Neuronal voltage-gated K+ (Kv) channels function in macromolecular complexes
(2011) Neuroscience Letters
**Framework APIs**

- ‘sciverse’ feature

```
<Require feature="sciverse"/>
```

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gadgets.sciverse.getAllResults</td>
<td>Retrieves all search results.</td>
</tr>
<tr>
<td>gadgets.sciverse.getArticleContent</td>
<td>Retrieves the article content (SD article page only)</td>
</tr>
<tr>
<td>gadgets.sciverse.getContextInfo</td>
<td>Retrieves the context information</td>
</tr>
<tr>
<td>gadgets.sciverse.getPageUrl</td>
<td>Returns the URL of the current host page</td>
</tr>
<tr>
<td>gadgets.sciverse.getResults</td>
<td>Retrieves specific search results</td>
</tr>
<tr>
<td>gadgets.sciverse.invokeResultsView</td>
<td>Opens the application’s ‘sciverseResultsView’</td>
</tr>
<tr>
<td>gadgets.sciverse.makeContentApiRequest</td>
<td>Use this function to make calls to the Content API’s</td>
</tr>
<tr>
<td>gadgets.sciverse.makeRequest</td>
<td>Use this function to make requests to external resources</td>
</tr>
</tbody>
</table>
Framework APIs

• ‘hub’ feature

```html
<Require feature="hub"/>
```

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>gadgets.hub.executeSearch</td>
<td>Executes a new Hub search as if the user had typed the search query in the actual Hub application. The page reloads with new search results.</td>
</tr>
</tbody>
</table>

• Detailed information on the Developer Portal at http://developers.sciverse.com/frameworkapireference
Developing SciVerse Applications

• Software Developer Kit (SDK) vs Application Market Place (AMP)

• SDK
  – Work offline
  – Simulates a “live” environment
  – Work in Eclipse environment (extension of the OSDE plugin)

• AMP
  – Live environment
  – Develop and test applications
  – Submit for approval
Building a SciVerse Application

• Hello World

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Module>
  <ModulePrefs title="HelloWorld" author_email="dietzrd@your.domain"/>
  <Content type="html" view="canvas">
    <!-- The code for Canvas view is here. -->
    <div>Canvas view for HelloWorld.</div>
  </Content>
  <Content type="html" view="profile">
    <!-- The code for Profile view is here. -->
    <div>Profile view for HelloWorld.</div>
  </Content>
  <Content type="html" view="sciverseResultsView">
    <!-- The code for Sciverse Results View view is here. -->
    <div>Sciverse Results View view for HelloWorld.</div>
  </Content>
</Module>
```
Our First SciVerse Application

- URL vs HTML gadgets

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Module>
  <ModulePrefs title="HelloWorld" author_email="dietzrd@your.domain"/>
  <Content type="url" view="canvas" href="http://www.njit.edu/hackathon"/>
  <Content type="html" view="profile">
    <![CDATA[
      <!-- The code for Profile view is here. -->
      <div>Profile view for HelloWorld.</div>
    ]]]></Content>
  <Content type="html" view="sciverseResultsView">
    <![CDATA[
      <!-- The code for Sciverse Results View view is here. -->
      <div>Sciverse Results View view for HelloWorld.</div>
    ]]]></Content>
</Module>
```
canvas View

profile View

Hello World – Profile & Canvas

profile View

profile View

Profile view for HelloWorld.
<?xml version="1.0" encoding="UTF-8"?>
<Module>
  <ModulePrefs title="HelloWorld" author_email="dietzrd@your.domain">
    <Require feature="sciverse"/>
  </ModulePrefs>
  <Content type="html" view="canvas"><![CDATA[
    <!-- The code for Canvas view is here. -->
    <div>Canvas view for HelloWorld.</div>
  ]]></Content>
  <Content type="html" view="profile"><![CDATA[
    <script>
      function showResultsView() {
        gadgets.sciverse.invokeResultsView(1,"Hello World from below result 1");
      }

      gadgets.util.registerOnLoadHandler(showResultsView);
    </script>
    <!-- The code for Profile view is here. -->
    <div>Profile view for HelloWorld.</div>
  ]]></Content>
  <Content type="html" view="sciverseResultsView"><![CDATA[
    <!-- The code for Sciverse Results View view is here. -->
    <div>Sciverse Results View view for HelloWorld.</div>
  ]]></Content>
</Module>
Hello World - a bit more interesting

```xml
<ModulePrefs title="HelloWorld" author_email="dietzj@your.domain">
  <Require feature="sciverse"/>
  <Require feature="views"/>
</ModulePrefs>

<Content type="html" view="profile"><![CDATA[
  <script>
    function showResultsView() {
      gadgets.sciverse.invokeResultsView(1,"msg":"Hello world from result 1");
      gadgets.sciverse.invokeResultsView(2,"msg":"Hello world from result 2");
    }
    
    gadgets.util.registerOnLoadHandler(showResultsView);
  </script>

  <!-- The code for Profile view is here. -->
  <div>Profile view for HelloWorld.</div>

  ]]></Content>

<Content type="html" view="sciverseResultsView"><![CDATA[
  <!-- The code for Sciverse Results View view is here. -->
  <script type="text/javascript">
    function init() {
      var prefs = gadgets.views.getParams();
      var msg = prefs.msg;
      document.getElementById("content").innerHTML = msg;
    }
    
    gadgets.util.registerOnLoadHandler(init);
  </script>

  <div id="content"></div>

  ]]></Content>
</Module>
```

New required feature (line 31)

Pass parameters to the ‘sciverseResultsView’

Get the parameters
Hello World – more interesting???
SciVerse Content APIs

• RESTful APIs that provide direct access to Elsevier content
  Request Endpoint → http://api.elsevier.com/content

• Search APIs
  – ScienceDirect
  – Scopus
  – Hub

• Retrieval APIs
  – Full-text
  – Abstract

• Two parameters required for authentication
  – API Key
  – Authentication token

More info → http://developers.sciverse.com/api
SciVerse Content APIs

```javascript
function callContentAPI(callback) {
    var requestHeaders = {};
    requestHeaders['X-EIS-APIKey'] = "a6ed256352770ff0d6d5a0bcf8d95a8c ";
    requestHeaders['X-EIS-AuthToken'] = contextInfo.secureAuthToken;
    requestHeaders['Accept'] = "application/json, application/atom+xml, text/xml";

    var url = encodeURI("http://api.elsevier.com/content/search/index:SCOPUS?query=
        + contextInfo.searchQuery) + ";&view=STANDARD";

    gadgets.sciverse.makeContentApiRequest(url, callback, requestHeaders);
}
```

- Where do `contextInfo.secureAuthToken` and `contextInfo.searchQuery` come from?

CONTEXT INFORMATION
Context Information

• What is it?
  – Information about the current page (e.g., current search terms)
  – Provides a way for SciVerse application to gain knowledge about what's happening in the SciVerse product.

• Examples: `secureAuthToken`, `searchTerms`, `doi`, `pii`, `issn`

• Full list available at [http://developers.sciverse.com/framework#context](http://developers.sciverse.com/framework#context)
Retrieving Context Information

- Two ways
  1. Using `getContextInfo()` – JSON object
  2. User Preference

- Using `getContextInfo()`

```javascript
function contextCallback(contextInfo){
  // Get context information from the JSON object.
  var secureAuthTokenVal = contextInfo.secureAuthToken;
  var searchQueryVal = contextInfo.searchQuery;

  // Do something with the values
}

// Run this function of the page has loaded.
gadgets.util.registerOnLoadHandler(function() {
  gadgets.sciverse.getContextInfo(contextCallback);
});
```
Retrieving Context Information

- Context info as a user preference

```xml
<?xml version="1.0" encoding="UTF-8"?>

<Module>
  <ModulePrefs title="contextInfoAsPrefGadget" author_email="youremail@your.domain">
    <Require feature="sciverse" />
    <ModulePrefs>
      <!-- Define a user preference with the same name as the context info as a user preference. -->
      <UserPref name="secureAuthToken" datatype="hidden" />
      <UserPref name="searchQuery" datatype="hidden" />
      <Content type="html" view="canvas,profile">
        <![CDATA[
          <script type="text/javascript">
            var prefs = new gadgets.Prefs();
            var secureAuthTokenVal = prefs.getString("secureAuthToken");
            var searchQueryVal = prefs.getString("searchQuery");
          </script>
        ]]>
      </Content>
  </ModulePrefs>
</Module>
```

- Hidden user prefs. Names match context information
- Access context information as prefs
References


Questions??