

Calling makeRequest in GWT to Get External Data for an Opensocial Gadget and Creating an Overlay

For this tutorial I use the following parameters:

`$your_namespace_dots` = the namespace of your unit with dots.

`$your_namespace_folders` = the namespace of your unit with folder delimiters.

`$your_website` = the URL of a web site you control.

Since we may want to get information that's not necessarily on the server our gadget is hosted on, cross site scripting restrictions limit us to using [makeRequest](#) to get external data. This tutorial builds on [a previous one](#).

1. Create a Java class `YourRecord.java` for the overlay type in the client folder (you may want to reorganize this class for reuse later) as:

```
package $your_namespace_dots.gwt.client;
```

```
import com.google.gwt.core.client.JavaScriptObject;
```

```
class YourRecord extends JavaScriptObject {  
    // Overlay types always have protected, zero-arg ctors  
    protected YourRecord() { }  
    // Typically, methods on overlay types are JSNI  
    public final native String getTitle() /{  
        return this.data.getElementsByTagName("title").item(0).textContent;  
    }/  
};  
}
```

1. To the `YourGadget` class add a reference to the `EntryPoint`:

```
public EntryPoint entryPoint = this;
```

In Javascript methods that call Java methods, we can usually use “this” like:

```
this.$your_namespace_dots.gwt.client.SomeGadget::handleResponse(L$your_namespace_folders/g
```

However, we’re going to put the method call in a callback so “this” is not what we want.

2. Now add two methods. The JSNI method calls makeRequest and the makeRequest callback calls the Java method which applies the overlay.

```
public void handleResponse(YourRecord rr) {  
    Window.alert(rr.getTitle());  
}  
  
native void callMakeRequest(EntryPoint ep) {  
    $wnd["response"] = function(obj) {  
        ep.$your_namespace_dots.gwt.client.YourGadget::handleResponse(L$your_namespace_folders/g  
    };  
    var params = {};  
    params[gadgets.io.RequestParameters.CONTENT_TYPE] = gadgets.io.ContentType.AM;  
    var url = http://www.clicc.ucla.edu/rss/laptops.php;  
    gadgets.io.makeRequest(url, response, params);  
};
```

3. Add a call to the method in a suitable place. For instance, in the button’s onClick event:

```
callMakeRequest(entryPoint);
```

4. Compile the code, copy it to your server (don’t forget to rename the folder), and check it out at \$your_website/YourGadget/yourgadgetcontainer.html.
5. Add the new file to the repostory, commit then work on adding functionality to the overlay and making handleResponse do what you want with the record.
svn add \$your_namespace_folders/gwt/client/YourRecord.java
svn commit

Revision #1

Created Fri, Oct 10, 2008 1:47 PM by Hunt, Carl

Updated Sun, Oct 12, 2008 4:57 PM by Hunt, Carl