

HelloWorld GlideAjax

Lab 8.3

10-15 minutes

Lab objectives

You will perform the following in this lab:

- Create a Script Include which extends the AbstractAjaxProcessor class.
- Create a Client Script to call the Script Include.

A. Create a Script Include

1. Create a new Script Include.

Name: **HelloWorld**

Client Callable: **Selected (checked)**

Accessible from: **All application scopes**

Active: **Selected (checked)**

Description: **Lab 8.3 Extends the AbstractAjaxProcessor class**

2. Examine the pseudo-code for the script:
 - Create a new class called HelloWorld.
 - Create an object from the new class with properties inheritable by other objects and which extends the AbstractAjaxProcessor class.
 - Add a method to the new object called alertGreeting.
 - Return "Hello" + parameter containing user's name + "!".
3. Write the Script:

```
1  var HelloWorld = Class.create();
2  HelloWorld.prototype = Object.extendObject(AbstractAjaxProcessor, {
3
4      alertGreeting: function() {
5          return "Hello" + this.getParameter('sysparm_user_name') + "!";
6      },
7
8      type: 'HelloWorld'
9  });
```

4. Select **Submit**.

B. Call the Script Include from a Client Script

1. Create a new Client Script.

Name: **Lab 8.3 HelloWorld Client Script**

Table: **Incident [incident]**

Type: **onLoad**

Active: **Selected (checked)**

Global: **Selected (checked)**

2. Examine the pseudo-code for the script:
 - Create an instance of the GlideAjax object called HelloWorld.
 - Add a param to call the alertGreeting method.
 - Add a param to pass the username.
 - Use the getXML method and the callback function HelloWorldParse to execute the Script Include.
 - Pass the response returned from the Script Include to the callback function.
 - Locate the answer variable in the returned XML and store the value in a variable.
 - Generate an alert to display the value of the answerFromXML variable.
3. Write the script:

```
1  function onLoad() {  
2      var ga = new GlideAjax('HelloWorld');  
3      ga.addParam('sysparm_name', 'alertGreeting');  
4      ga.addParam('sysparm_user_name', '<YOUR NAME>');  
5      ga.getXML(HelloWorldParse);  
6  
7  
8  function HelloWorldParse(response) {  
9      var answerFromXML = response.responseXML.documentElement.getAttribute('answer');  
10     alert(answerFromXML);  
11 }  
12 }
```

4. Select **Submit**.
5. The first parameter in the script you just wrote is **sysparm_name**, which is a reserved parameter name. What information does it pass to the Script Include?

6. Other than beginning with a **sysparm_** prefix, do additional parameters have to use reserved parameter names?

C. Test Your Work

1. Open any Incident.
2. Did an alert appear? Was it populated correctly? If not, debug and re-test.
3. Make the *Lab 8.3 HelloWorld Client Script* **inactive**.

Lab Completion

Great job! You have successfully extended the existing `AbstractAjaxProcessor` class. The `alertGreeting()` method you created in the `HelloWorld Script Include` expects a value to be passed in when it is called. It uses that value to build a string and return it to the client.

You then used the `GlideAjax` class in a Client Script to call the `alertGreeting()` method in the new Script Include.