

# ServiceNow Application Developer

## Server-side Scripting > Exercise: Extend GlideAjax

In this exercise, you will write a Script Include to create a client-callable Script Include by extending *GlideAjax*. The new Script Include retrieves the *Requested for's* email address from the *User* table.

### Create the Script Include

---

1. Create a Script Include.
  - a. In Studio, click the **Create Application File** link.
  - b. In the *Filter...* field enter the text **Script** OR select **Server Development** from the categories in the left hand pane.
  - c. Select **Script Include** in the middle pane as the file type then click the **Create** button.

2. Configure the Script Include:

*Name:* **GetEmailAddress**

*API Name:* **(this field value is automatically populated)**

*Client callable:* **Selected (checked)**

*Application:* **(this field value is automatically populated)**

*Accessible from:* **This application scope only**

*Active:* **Selected (checked)**

*Description:* **Script Include to return an email address. The calling client-side script passes a sys\_id for a User table record.**

3. Delete the template from the *Script* field. Copy this script and paste it into the *Script* field.

```
var GetEmailAddress = Class.create();
// Extend the global.AbstractAjaxProcessor class
GetEmailAddress.prototype =
Object.extend(Object(global.AbstractAjaxProcessor,{
    // Define the getEmail function.
    // Create a GlideRecord for the User table.
    // Use the sysparm_userID passed from the client side to retrieve a
    record from the User table.
    // Return the email address for the requested record
    getEmail: function() {
        var userRecord = new GlideRecord("sys_user");
        userRecord.get(this.getParameter('sysparm_userID'));
        return userRecord.email + '';
    },
    type: 'GetEmailAddress'
}));
```

4. Click the **Submit** button.

► **QUESTION:** The Script include references *sysparm\_userID*. Where does the value of *sysparm\_userID* come from at runtime?

## Create the Client Script

---

1. Create a Client Script.
  - a. In Studio, click the **Create Application File** link.
  - b. In the *Filter...* field enter the text **Client** OR select **Client Development** from the categories in the left hand pane.
  - c. Select **Client Script** in the middle pane as the file type then click the **Create** button.

## 2. Configure the Client Script:

*Name:* **NeedIt Populate Email Field**

*Table:* **NeedIt [x\_<your\_company\_code>\_needit\_needit]**

*UI Type:* **Desktop**

*Type:* **onChange**

*Field name:* **Requested for**

*Active:* **Selected (checked)**

*Description:* **Use the GetEmail Address Script Include to retrieve the Requested for's email address from the database when the Requested for value changes.**

3. Delete the template from the *Script* field. Copy this script and paste it into the *Script* field.

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {

    // Modified the if to return if the newValue == oldValue to avoid
    // unnecessary trips to the server
    if (isLoading || newValue === '' || newValue == oldValue) {
        return;
    }

    // Instantiate the GetEmailAddress Script Include
    var getEmailAddr = new GlideAjax('GetEmailAddress');
    // Specify the getEmail method
    getEmailAddr.addParam('sysparm_name', 'getEmail');
    // Pass the Requested for sys_id
    getEmailAddr.addParam('sysparm_userID',
g_form.getValue('u_requested_for'));
    // Send the request to the server
    getEmailAddr.getXML(populateEmailField);

    // When the response is back from the server
    function populateEmailField(response){
        // Extract the email address from the response, clear any value from
the email field,
        // set new value in the email field
        var emailFromScriptInclude =
response.responseXML.documentElement.getAttribute("answer");
        g_form.clearValue('u_requested_for_email');
        g_form.setValue('u_requested_for_email', emailFromScriptInclude);
    }
}
```

4. Click the **Submit** button.

► **QUESTION:** How does ServiceNow know which Script Include to call? How does it know which method to use?

## Test the Client Script and Script Include

---

1. Switch to the main ServiceNow browser window and use the browser reload button to reload ServiceNow.
2. Open an *existing NeedIt* record for editing (for this part of the exercise, do not create a new record).
3. Change the value in the *Requested for* field to **Fred Luddy**.
4. The *Requested for email* field value should change to ***fred.luddy@example.com***. If not, debug using the debugging strategies explained earlier in this module. If you see a message about cross-scope privileges, it was caused by ServiceNow detecting allowed use of an out-of-scope file (*AbstractAjaxProcessor*). You will see the message only once.
5. Change the value in the *Requested for* field to **Beth Anglin**. The value in the *Requested for email* field should change again.

## Challenge

---

1. Create a *NeedIt* record.
  - a. Examine the *Requested for* field and notice it has a value.
  - b. Examine the *Requested for email* field and notice it does not have a value.

Modify the Client Script logic so the *Requested for email* field is also populated for new *NeedIt* records. You may find the *GlideForm* [isNewRecord\(\)](https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=r_GlideFormIsNewRecord) ([https://developer.servicenow.com/app.do#!/api\\_doc?v=madrid&id=r\\_GlideFormIsNewRecord](https://developer.servicenow.com/app.do#!/api_doc?v=madrid&id=r_GlideFormIsNewRecord)) method useful. Do not simply remove the *isLoading* check from the *if* statement in the existing Client Script. Removing this check means the Script Include is called every time the *NeedIt* form loads. Round trip calls to the server are "expensive" from a load time perspective and should be avoided if possible.

### ► CHALLENGE SOLUTION:

