Lab Guide

External Web Services –

Integrations 201

Alex Steinwachs & Joseph Evans

Default Login / Password:

admin / Knowledge16

itil / Knowledge16

employee / Knowledge16

This

Page

Intentionally

Left

Blank

Lab Goal

Lab 1.0 Initial Setup and Logistics

Gain basic proficiency in connecting external web services to ServiceNow. At the end of all of these labs you will be able to create external connections via web services. You will learn how to use to use this connection in various features of ServiceNow.

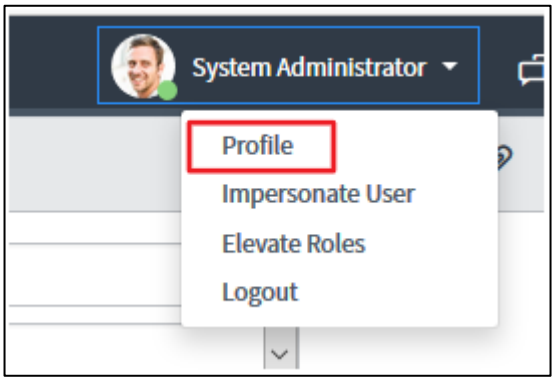
- Lab 1 – Initial Setup and Logistics
- Lab 2 – Creating and testing the connection
- Lab 3 – Use the web services activities in the workflow editor
- Lab 3.1 (Optional) – Additional Fields
- Lab 4 – Setup OAuth authentication
- Lab 5 – Use the web services connection in a ServiceNow script
- Lab 6 (Optional) – Take home practice

Lab 1.0 – Initial Setup and Logistics

At the end of Lab 1.0 you should be able to log in to your ServiceNow Instance and the Slack channel.

Log in to Your Instance and Change the Admin Account Name

1. Log in to the instance provided with the admin credentials.
2. Navigate to the User Profile in the top right by selecting **System Administrator > Profile**.



®2016 ServiceNow, Inc. All rights reserved.

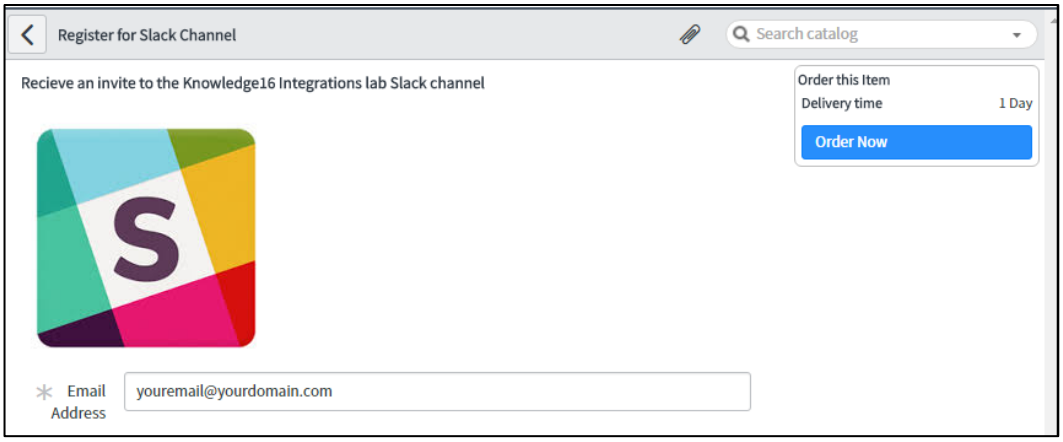
3. Change the **First name** to your **first name** and **Last name** to your **last name@company**.



4. Click **Update**.

Create a Slack Account and Join the Channel

1. Navigate to http://bit.ly/1QPyq2y and enter your **email address**. Click **Order Now**.

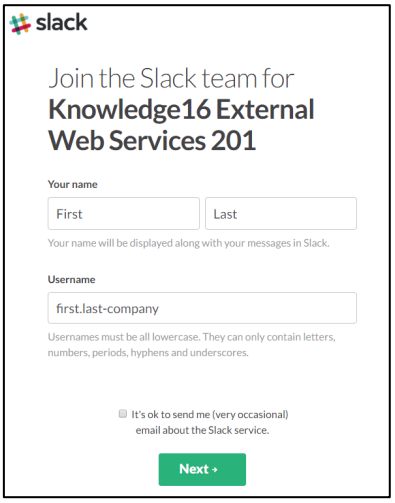


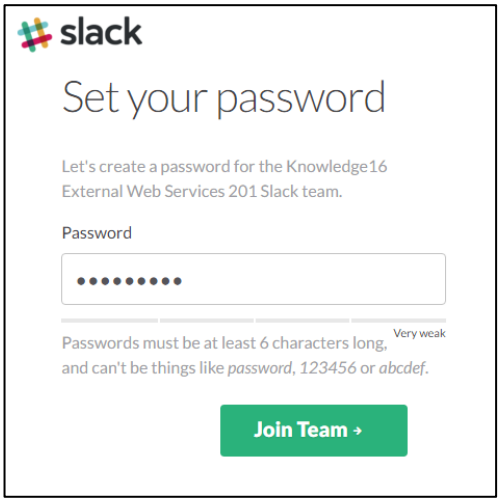
2. You will receive an email inviting you to the Slack Channel. Click Join team in the email.



®2016 ServiceNow, Inc. All rights reserved.

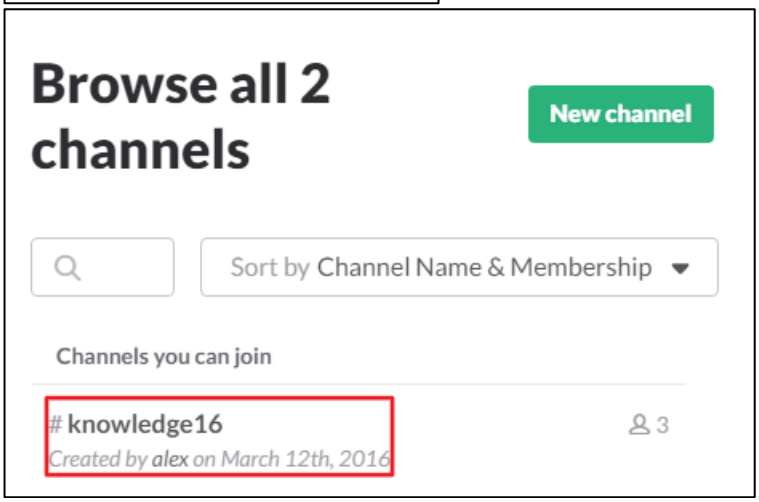
3. Create an account and click **Next**. Set a password then click **Join Team**.





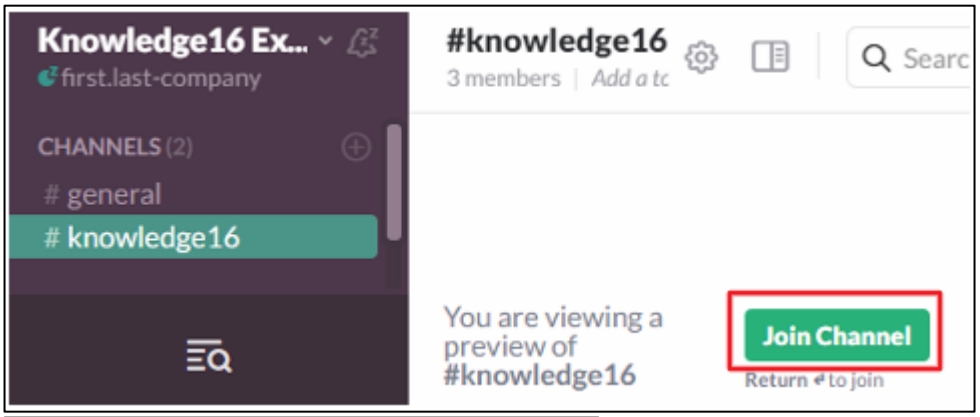
4. Right click **CHANNELS** and select **#knowledge16** to join the channel.

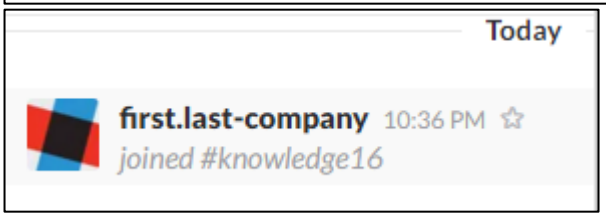




®2016 ServiceNow, Inc. All rights reserved.

5. A **#knowledge16** entry appears in your CHANNELS list. Select **Join Channel** in the lower-right corner. You will receive a message confirming that you successfully joined the channel.





Create a Slack Team and Channel (Optional)

Navigate to https://slack.com/ and enter your **email address**. After you sign up, create a new team and channel. Use these in this lab guide instead of **k16web201.slack.com** and **#knowledge16**.

®2016 ServiceNow, Inc. All rights reserved.

Lab Goal

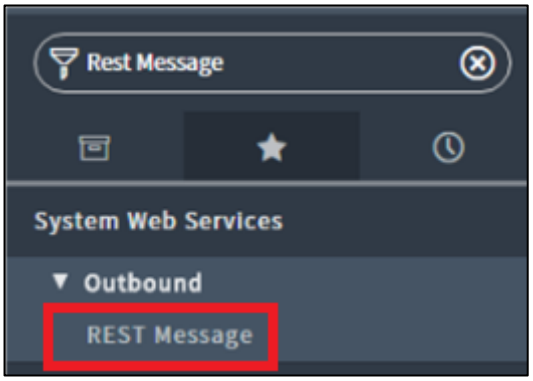
Post a message to a Slack channel using a REST Message connection in ServiceNow. At the end of this lab you will be able to post a test message in the channel through ServiceNow.

Lab 2.0 Creating and testing the connection

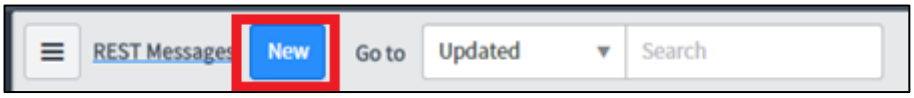
Lab 2.0 – Creating and Testing the Connection

Create a new REST Message

1. Navigate to **System Web Services > Outbound > REST Message**.



2. Select **New**.



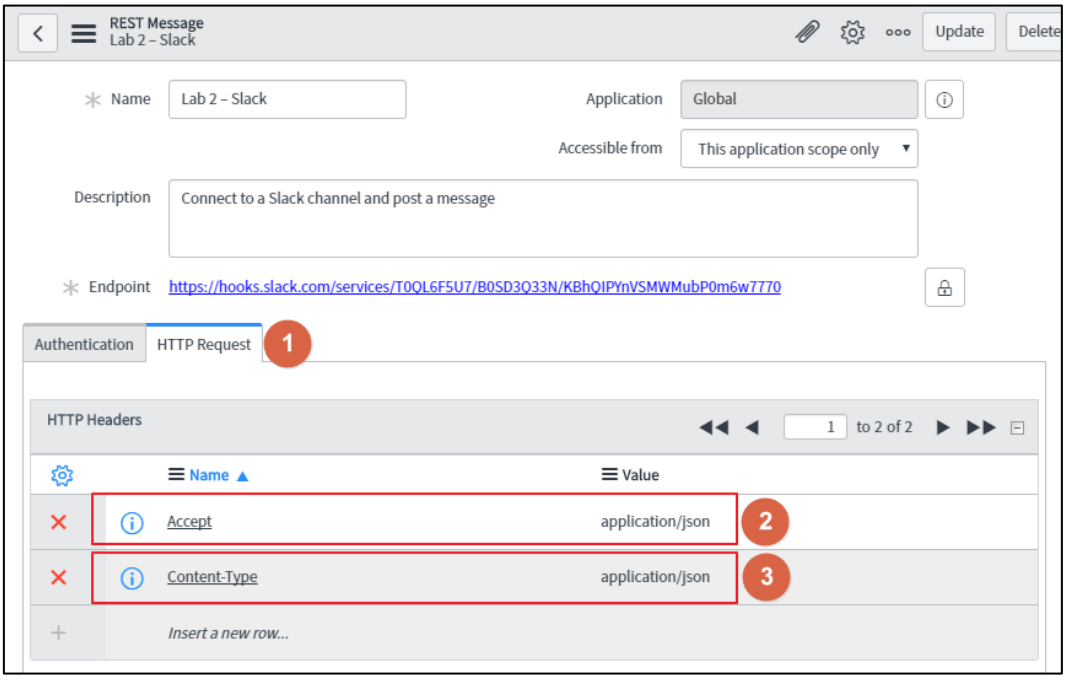
3. Enter the following information:
   Name: **Lab 2 – Slack**
   Accessible from: **All application scopes**
   Description: **Connect to a Slack channel and post a message**
   Endpoint:
   https://hooks.slack.com/services/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
   (PASTEBIN: http://pastebin.com/BNZmsAEQ)

®2016 ServiceNow, Inc. All rights reserved.

4. Select the **HTTP Request** tab then create two HTTP Request Headers by typing in **Insert a new row…**
   Name: **Accept**
   Value: **application/json**

   Name: **Content-Type**
   Value: **application/json**

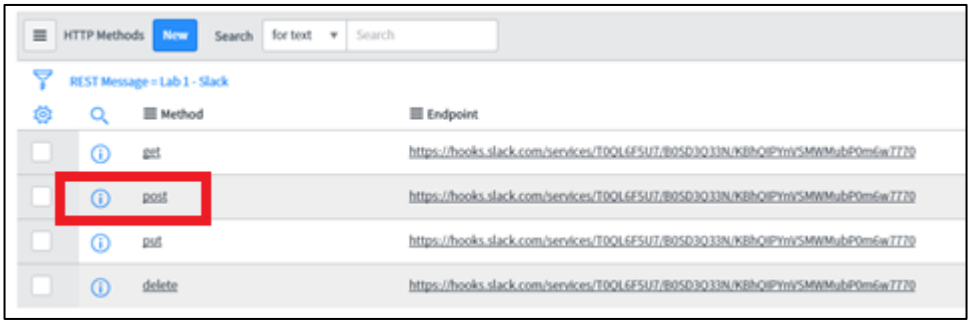
5. Your record should look like the example. Right click in the top bar and click **Save**.



®2016 ServiceNow, Inc. All rights reserved.

Working with HTTP Methods

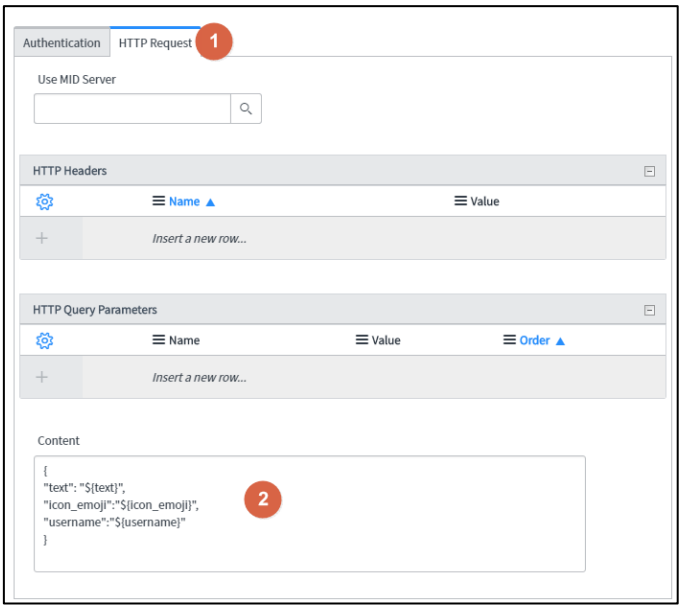
1. After clicking **Save,** four HTTP Methods appear. You can view these in the related list at the bottom of the page. Select **post**.



2. Navigate to the **HTTP Request** tab then enter the following into the **Content** text box.

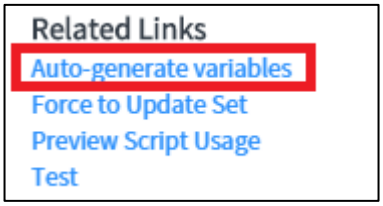
```
{
"text": "${text}",
"icon_emoji":"${icon_emoji}",
"username":"${username}"
}
```

(PASTEBIN: http://pastebin.com/EYNKnTwr)

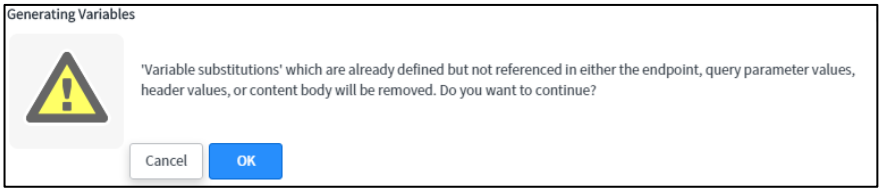


®2016 ServiceNow, Inc. All rights reserved.

3. Right click in the top bar and click **Save**. Click **Auto-generate variables** in the Related Links section.

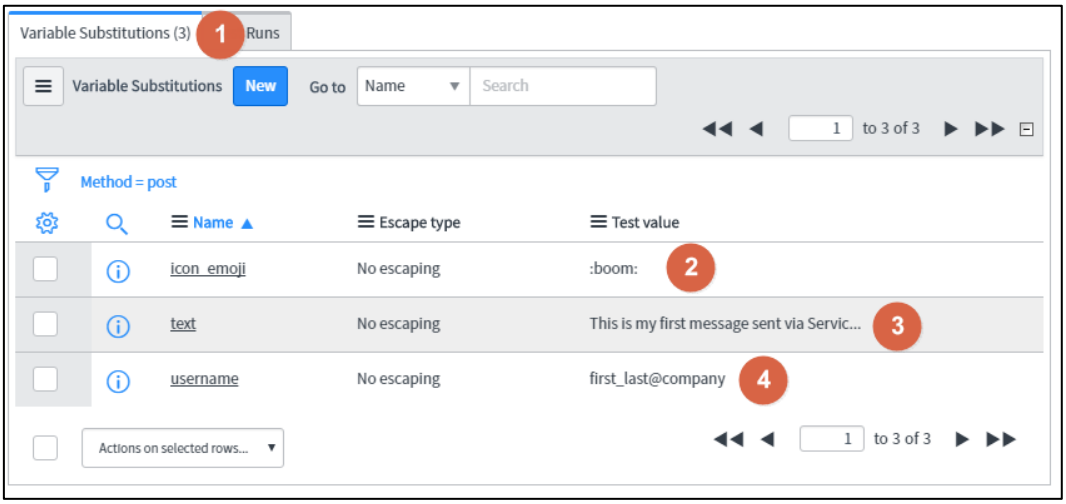


4. If you get a message about Generating Variables, click **OK**.



5. This action generates three entries in the **Variable Substitutions** related list. Enter the following values into the **Test value** column. Feel free to change to your own but please keep them appropriate. Use http://emojisearch.com/ for a list of emojis.

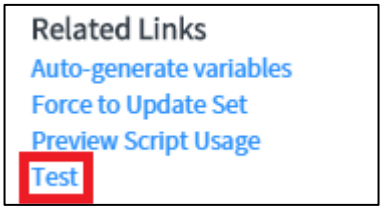
   icon_emoji: **:boom:**
   text: **This is my first message sent via ServiceNow's REST API**
   username: **first_last@company**



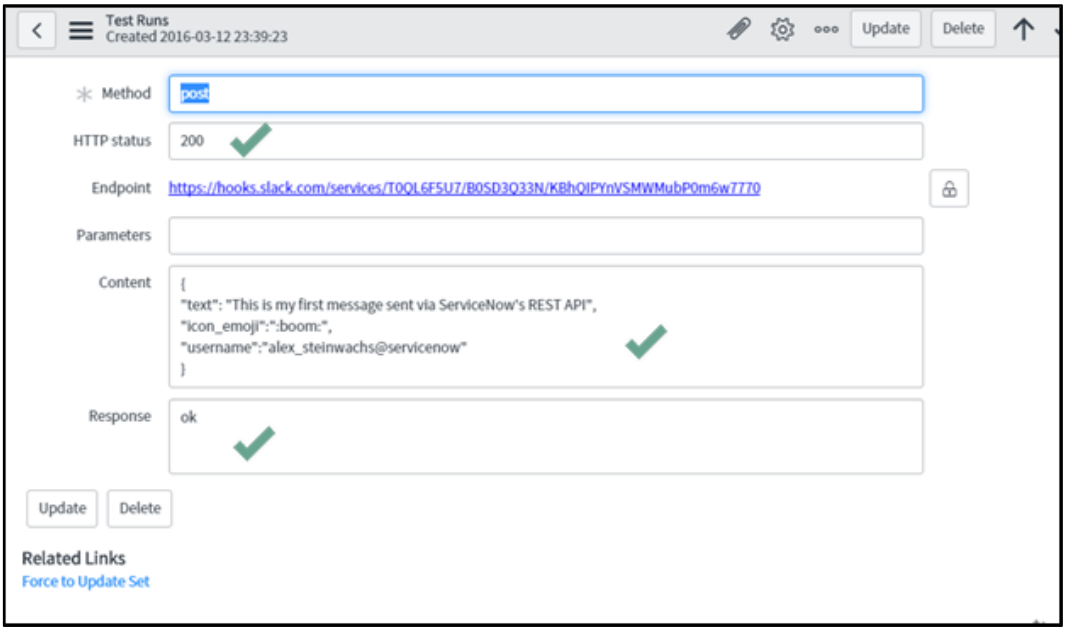
®2016 ServiceNow, Inc. All rights reserved.

Lab Success Verification

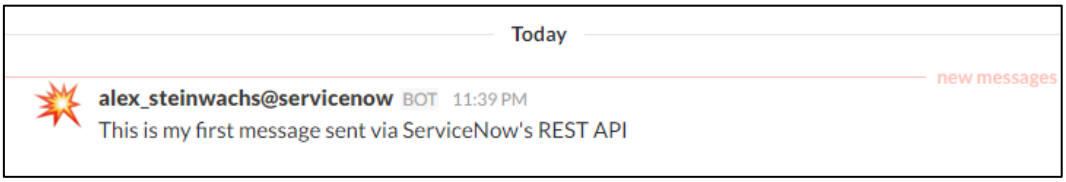
1. Under the Related Links, click the **Test** link.



2. If everything ran correctly, then you should get a HTTP status of **200** with a response of **ok.**



3. Check the Slack channel #knowledge16 we joined in **Lab 1.0** for your message.



®2016 ServiceNow, Inc. All rights reserved.

Lab Goal

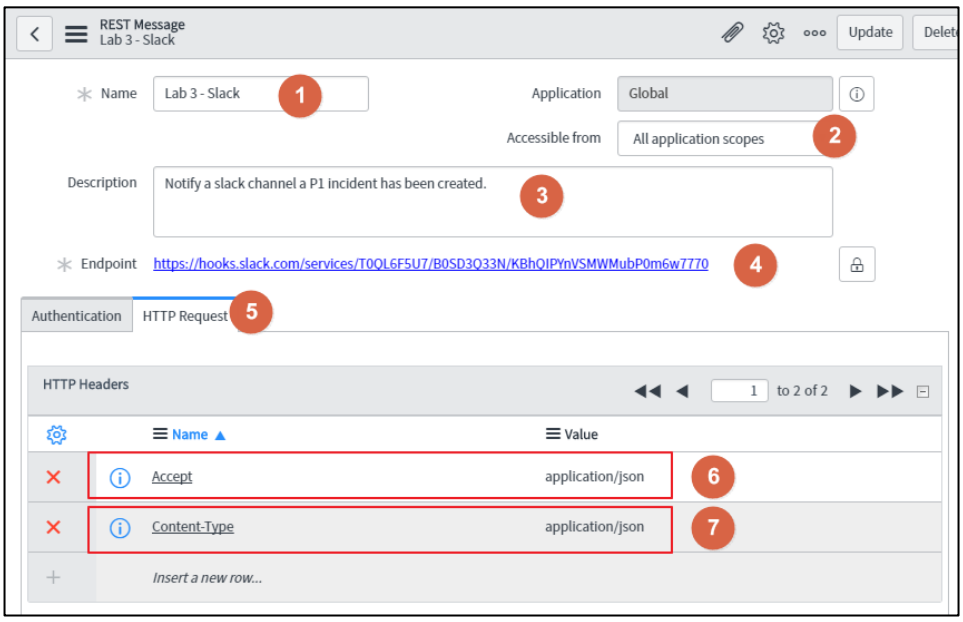
This lab explains how to create a new REST Message and use it in a workflow activity in the Workflow Editor. At the end of Lab 3.0 you post a message into the Slack channel each time a Priority 1 Incident is created.

Lab 3.0 Using Web Services in the Workflow Editor

Lab 3.0 – Use Web Services in the Workflow Editor

Create and Test a New REST Message

1. Create a new **REST Message** with a name of **Lab 3 – Slack** using the same **endpoint, scopes** and **HTTP Headers** as **Lab 2.0**.



®2016 ServiceNow, Inc. All rights reserved.

2. Navigate to the **post** method. Enter the following content into the **Content** text box.

```
{
  "icon_emoji":"${icon_emoji}",
  "username":"${username}",
  "attachments":[
    {
      "fallback":"${short_description}",
      "color":"${color}",
      "pretext":"New P1 Incident created in ServiceNow",
      "author_name":"${caller_name}",
      "author_link":"${url}/nav_to.do?uri=sys_user.do?sys_id=${caller_sys_id}",
      "title":"${number}",
      "title_link":"${url}/nav_to.do?uri=incident.do?sysparm_query=number=${number}",
      "text":"${short_description}",
      "unfurl_links":true,
      "fields":[
        {
          "title":"Priority",
          "value":"${priority}",
          "short":true
        },
        {
          "title":"State",
          "value":"${state}",
          "short":true
        }
      ]
    }
  ]
}
```

(PASTEBIN: http://pastebin.com/y0zTGt15)

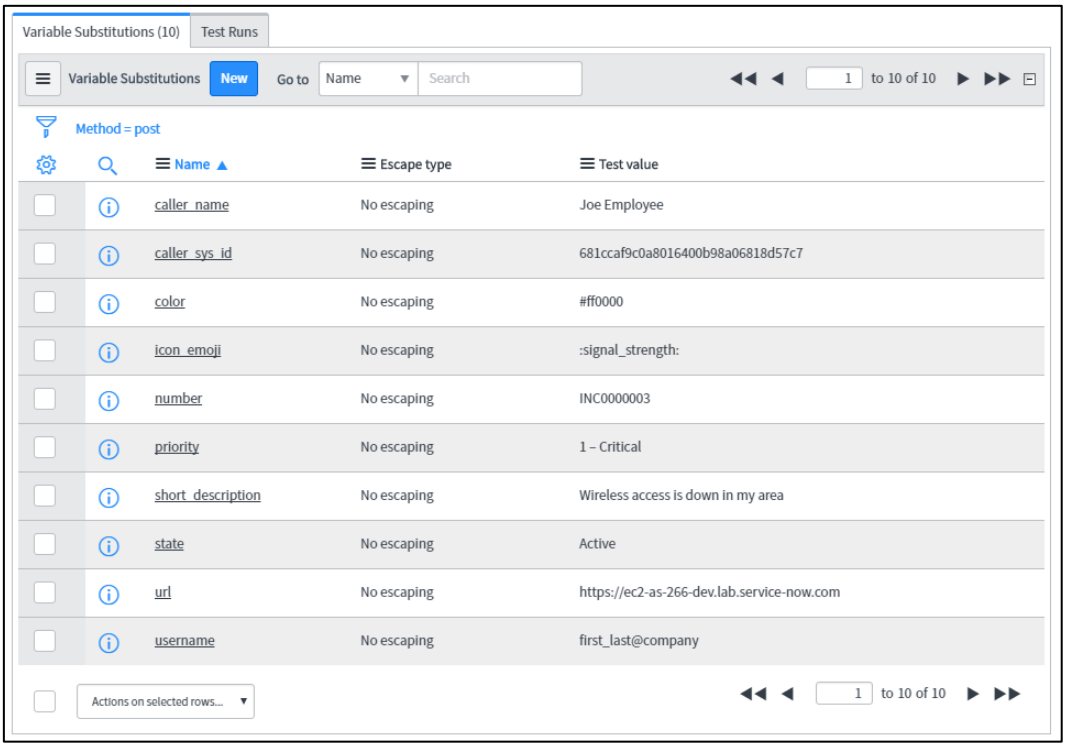
3. Right click in the top bar and click **Save**.

4. Click **Auto-generate variables** in the Related Links section.

®2016 ServiceNow, Inc. All rights reserved.

5. Enter the following values into the **Test value** column:

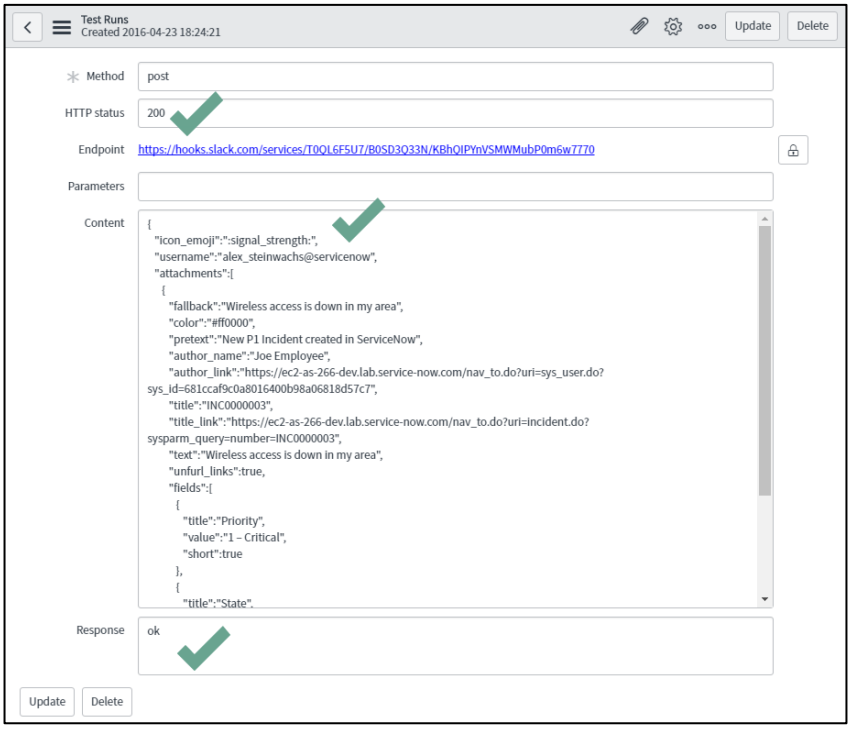
   username: **first_last@company**
   state: **Active**
   caller_name: **Joe Employee**
   color: **#ff0000**
   short_description: **Wireless access is down in my area**
   number: **INC0000003**
   caller_sys_id: **681ccaf9c0a8016400b98a06818d57c7**
   icon_emoji: **:signal_strength:**
   priority: **1 – Critical**
   url: **https://<INSTANCE>.service-now.com/**



Note: If you are not able to see the Incident or User record, replace the test values with ones that exist in your instance.

®2016 ServiceNow, Inc. All rights reserved.

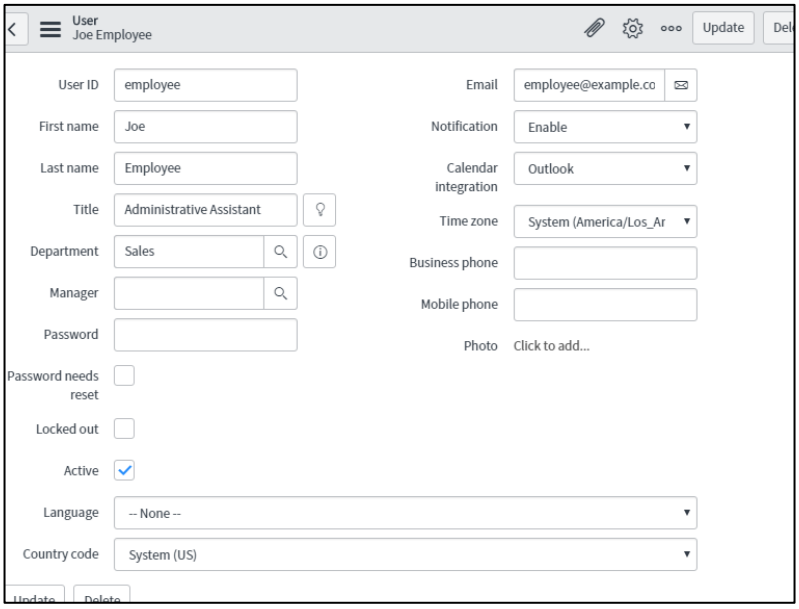
6. Click **Test** in the Related Links section. You should see a HTTP status of **200** with a response of **ok**. You should also see a notification in the Slack channel.



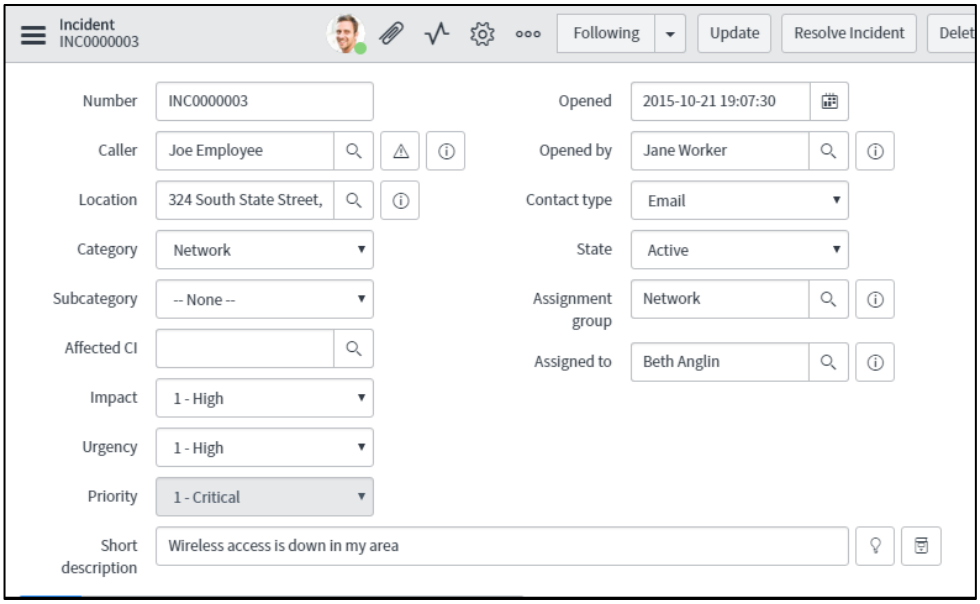


®2016 ServiceNow, Inc. All rights reserved.

7. Clicking the caller name- (**Joe Employee**) in the Slack channel should open a new window with the user record.



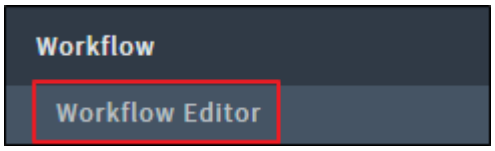
8. Clicking the Incident number (**INC0000003**) in the Slack channel should open a new window with the Incident record.



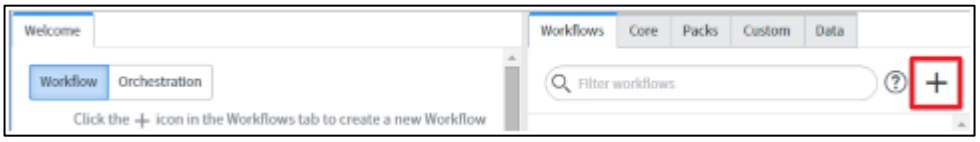
®2016 ServiceNow, Inc. All rights reserved.

Add REST Message to Workflow Activity

1. After you confirm that the test messages are reaching the channel, create a workflow to send notifications for new P1 incidents. Navigate to **Workflow > Workflow Editor**.



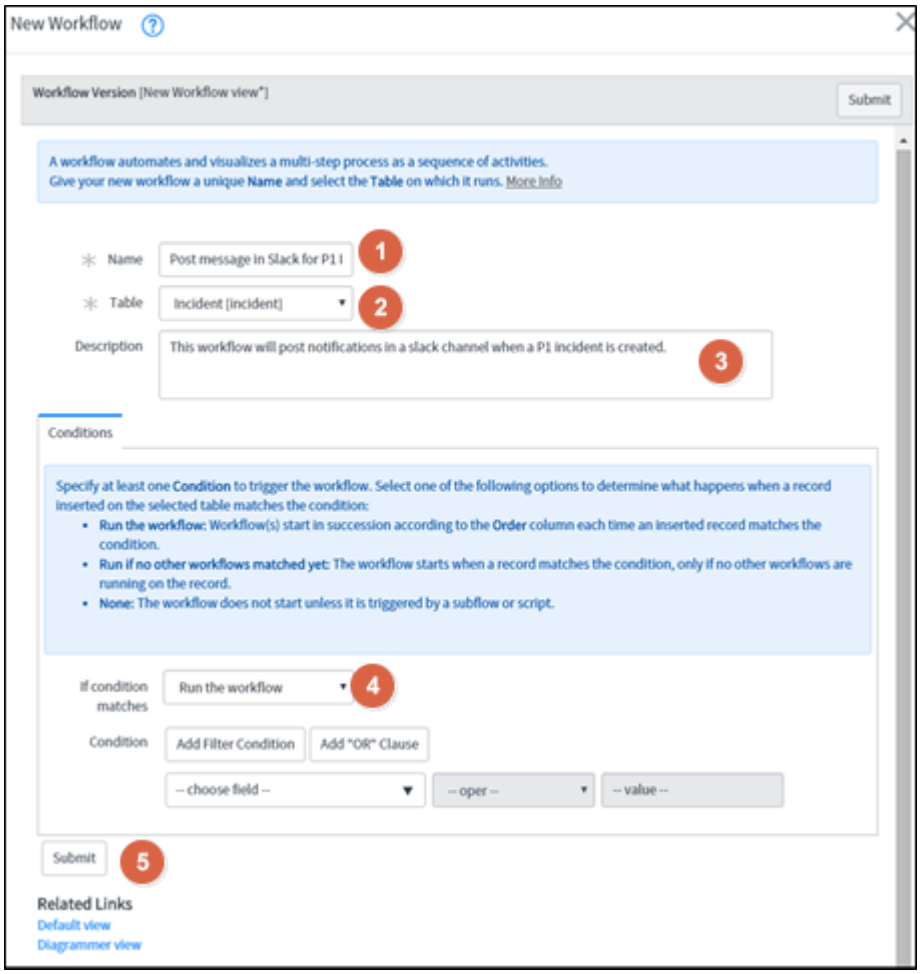
2. Select the **+** to create a new workflow.



®2016 ServiceNow, Inc. All rights reserved.

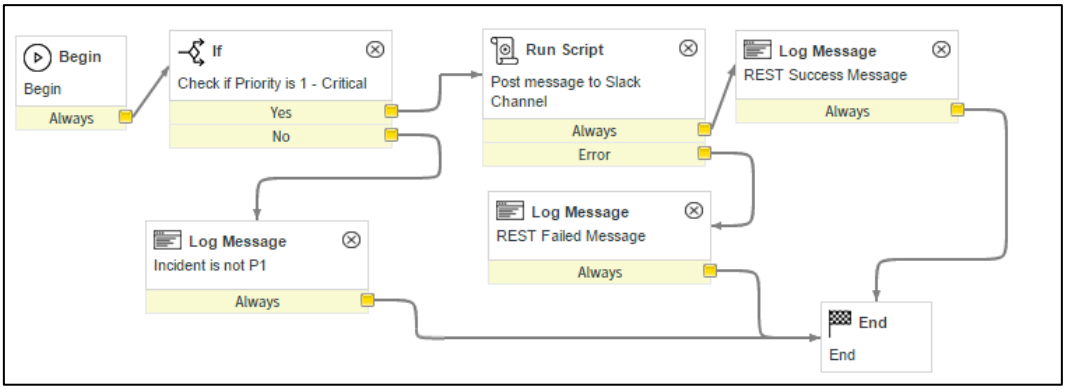
3. Use these workflow settings then click **Submit**:

   Name: **Post message in Slack for P1 Incidents**
   Table: **Incident**
   Description: **This workflow will post notifications in a slack channel when a P1 incident is created.**
   If condition matches: **Run the workflow**
   Condition: **Leave this blank**

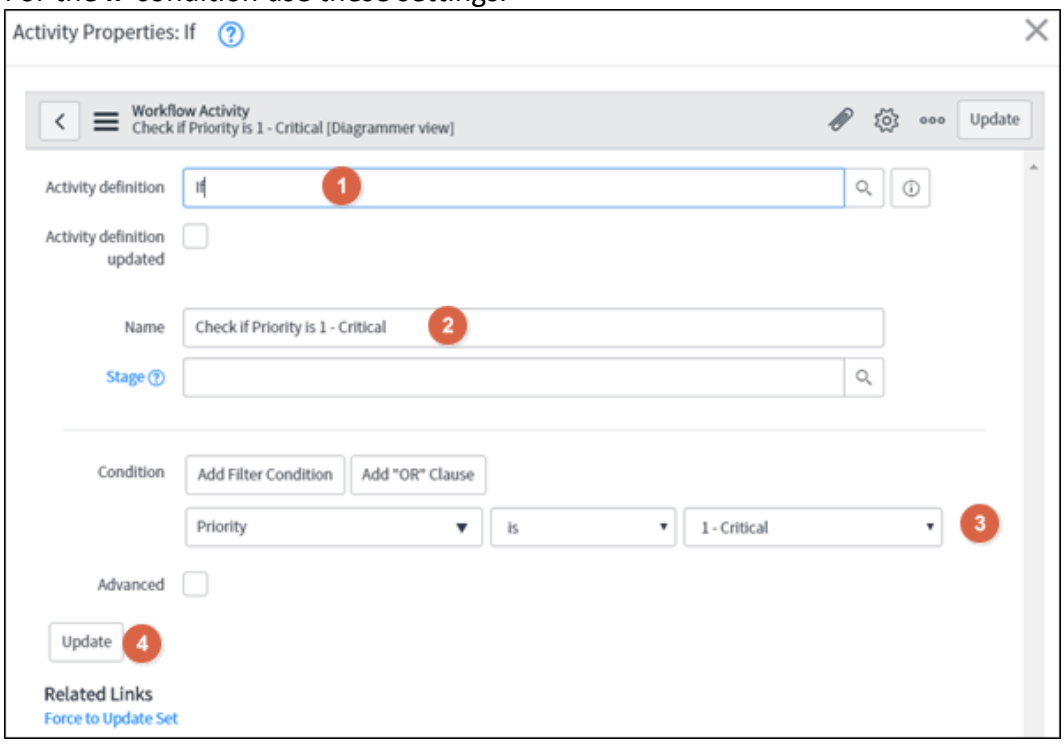


®2016 ServiceNow, Inc. All rights reserved.

4. Create this workflow.



5. For the **IF** condition use these settings.



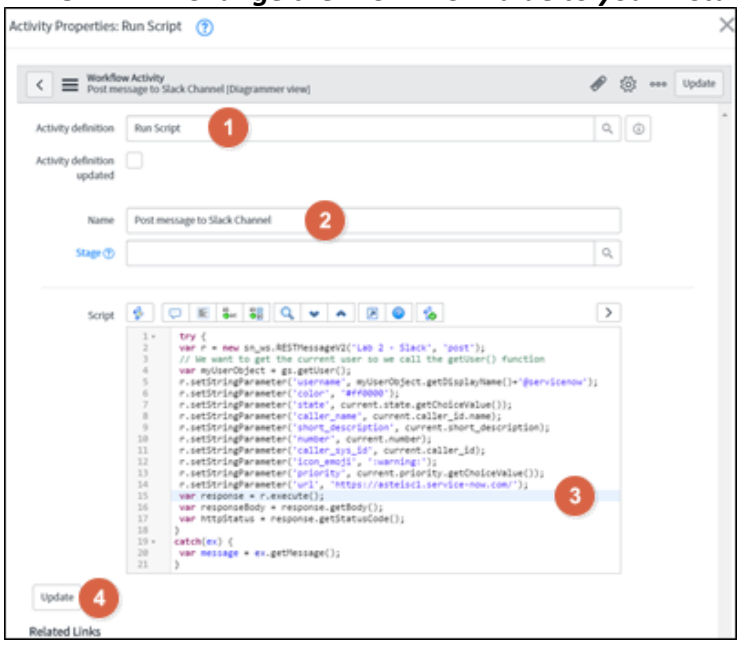
©2016 ServiceNow, Inc. All rights reserved.

6. For the **Run Script** activity, use these settings.

```
try {
   var r = new sn_ws.RESTMessageV2('Lab 3 - Slack', 'post');
   var myUserObject = gs.getUser();
   r.setStringParameter('username', myUserObject.getDisplayName() + '@servicenow');
   r.setStringParameter('color', '#ff0000');
   r.setStringParameter('state', current.state.getChoiceValue());
   r.setStringParameter('caller_name', current.caller_id.name);
   r.setStringParameter('short_description', current.short_description);
   r.setStringParameter('number', current.number);
   r.setStringParameter('caller_sys_id', current.caller_id);
   r.setStringParameter('icon_emoji', ':warning:');
   r.setStringParameter('priority', current.priority.getChoiceValue());
   // Change INSTANCE to your instance name
   r.setStringParameter('url', 'https://INSTANCE.service-now.com/');
   var response = r.execute();
   var responseBody = response.getBody();
   var httpStatus = response.getStatusCode();
} catch (ex) {
   var message = ex.getMessage();
}
```

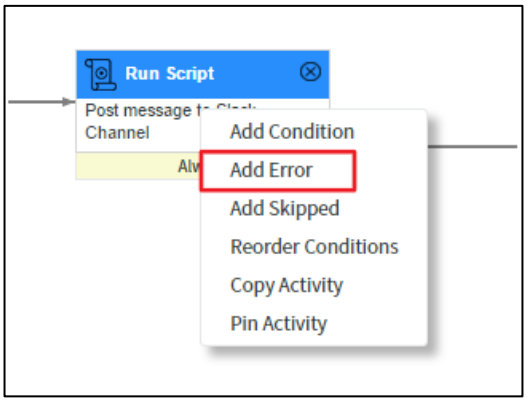
(PASTEBIN: http://pastebin.com/qi0kE4ZY)

IMPORTANT: Change the INSTANCE value to your instance name

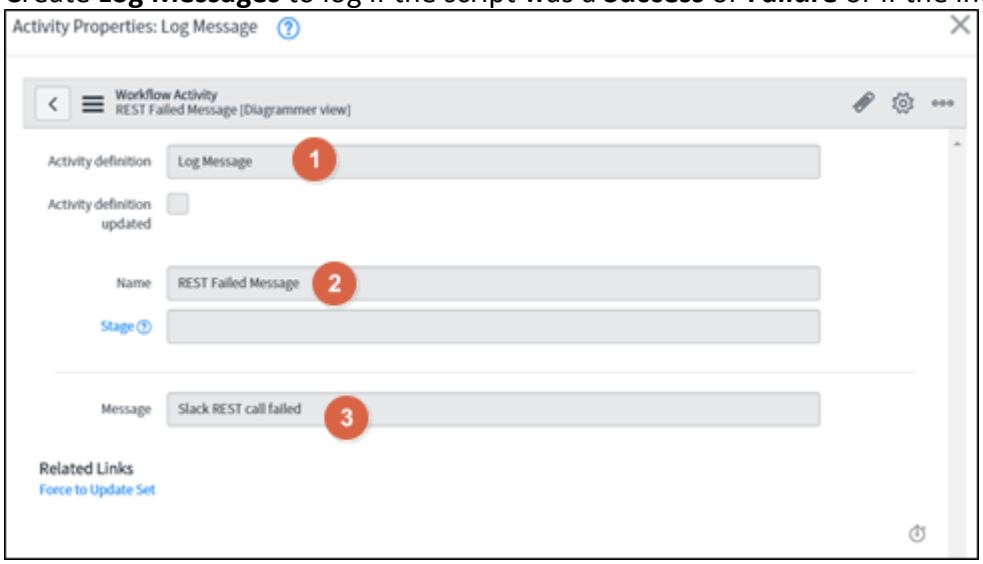


®2016 ServiceNow, Inc. All rights reserved.

7. Right-click **Run Script** and select **Add Error**.



8. Create **Log Messages** to log if the script was a **Success** or **Failure** or if the incident is not **P1**.



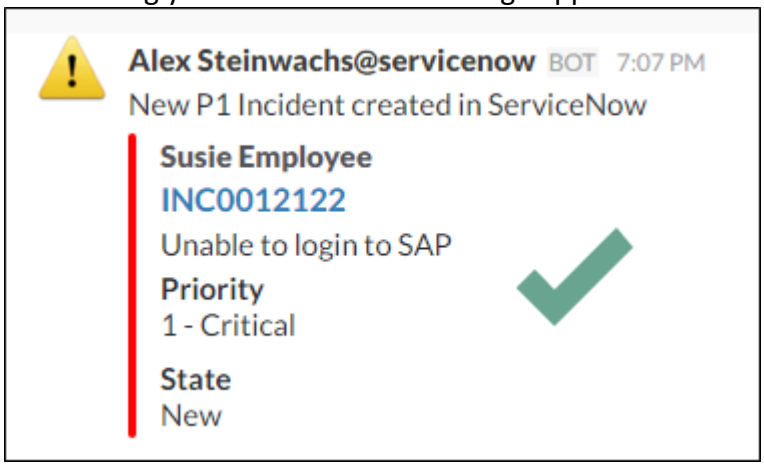
9. **Publish** the workflow.

®2016 ServiceNow, Inc. All rights reserved.

Lab Success Verification

1. Navigate to **Incident > Create New**
   Caller: **Pick a caller**
   Short Description**: Something unique**
   Impact: **1 (all)**
   Urgency **1 (High)**
   Priority: **1 – Critical** (auto set based on Impact and Urgency)

2. After saving you should see the message appear in the Slack channel.



®2016 ServiceNow, Inc. All rights reserved.

Lab Goal

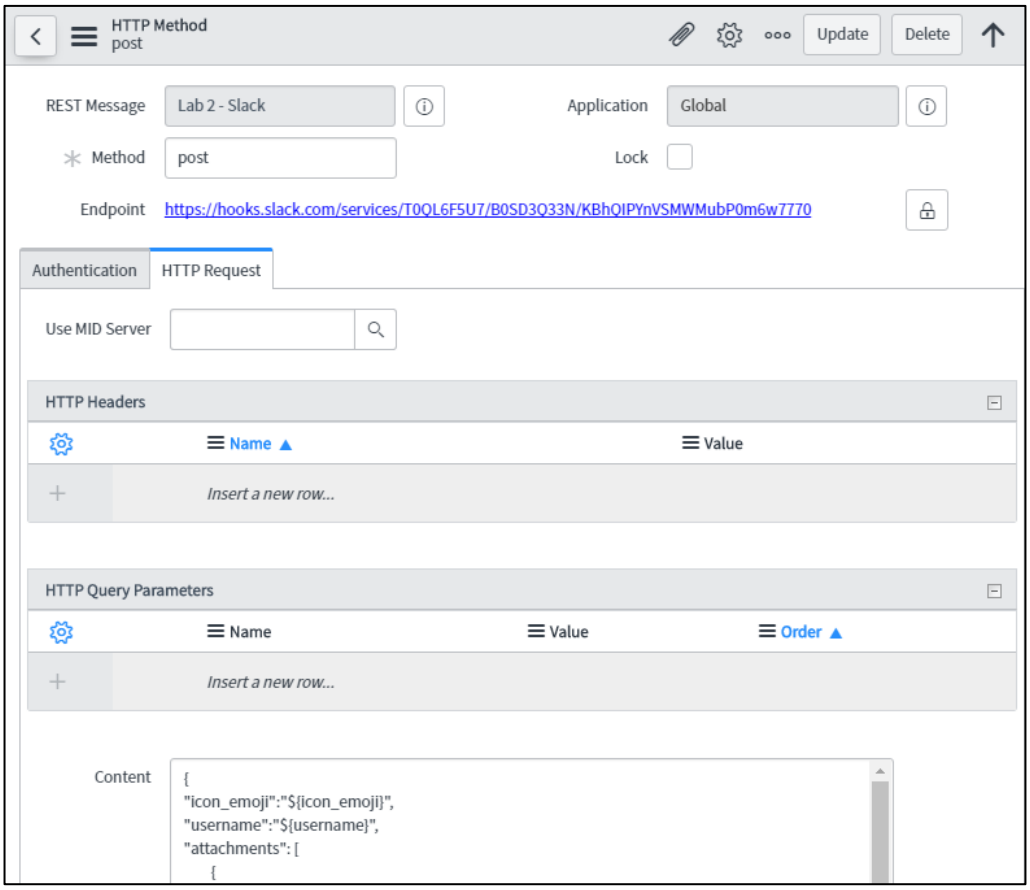
This optional lab explains how to add information based on the Affected CI in the Incident. At the end of Lab 3.0 you post a message into the Slack channel each time a Priority 1 Incident is created with the Affected CI.

Lab 3.1 Additional Fields (Optional)

Lab 3.1 – Additional Fields (Optional)

Modify the REST Message

1. Modify the **Content** text box in the post message created in **Lab 3.0**.



®2016 ServiceNow, Inc. All rights reserved.

```
{
  "icon_emoji":"${icon_emoji}",
  "username":"${username}",
  "attachments":[
    {
      "fallback":"${short_description}",
      "color":"${color}",
      "pretext":"New P1 Incident created in ServiceNow",
      "author_name":"${caller_name}",
      "author_link":"${url}/nav_to.do?uri=sys_user.do?sys_id=${caller_sys_id}",
      "title":"${number}",
      "title_link":"${url}/nav_to.do?uri=incident.do?sysparm_query=number=${number}",
      "text":"${short_description}",
      "unfurl_links":true,
      "fields":[
        {
          "title":"Priority",
          "value":"${priority}",
          "short":true
        },
        {
          "title":"State",
          "value":"${state}",
          "short":true
        },
        {
          "title":"Affected CI",
          "value":"${cmdb_ci}",
          "short":true
        }
      ]
    }
  ]
}
```

(PASTEBIN: http://pastebin.com/R24ATzkd)

2. Add the variable with a **Test value** to the same post message.



3. **Save** and **Test** to make sure the new Affected CI field displays in the Slack Channel.

©2016 ServiceNow, Inc. All rights reserved.

Modify the Workflow

1. Checkout the workflow created in **Lab 3.0** and modify the **Run Script** step to add the Affected CI Name, CI type, and icons based on the CI type.

```
try {
   var r = new sn_ws.RESTMessageV2('Lab 3 - Slack', 'post');
   // We want to get the current user so we call the getUser() function
   var myUserObject = gs.getUser();
   r.setStringParameter('username', myUserObject.getDisplayName() + '@servicenow');
   r.setStringParameter('color', '#ff0000');
   r.setStringParameter('state', current.state.getChoiceValue());
   r.setStringParameter('caller_name', current.caller_id.name);
   r.setStringParameter('short_description', current.short_description);
   r.setStringParameter('number', current.number);
   r.setStringParameter('caller_sys_id', current.caller_id);
   r.setStringParameter('icon_emoji', ':warning:');
   r.setStringParameter('priority', current.priority.getChoiceValue());
   // Change INSTANCE to your instance name
   r.setStringParameter('url', 'https://INSTANCE.service-now.com/');
   // Display different information in the channel based on CI class
   switch (current.cmdb_ci.sys_class_name.getChoiceValue()) {
      case 'Computer' : r.setStringParameter('cmdb_ci', ' :computer: ' + current.cmdb_ci.name); break;
      case 'Software' : r.setStringParameter('cmdb_ci', ' :floppy_disk: ' + current.cmdb_ci.name); break;
      case 'Printer' : r.setStringParameter('cmdb_ci', ' :printer: ' + current.cmdb_ci.name); break;
      case 'Service Offering' : r.setStringParameter('cmdb_ci', ' :briefcase: ' + current.cmdb_ci.name); break;
      case 'Business Service' : r.setStringParameter('cmdb_ci', ' :briefcase: ' + current.cmdb_ci.name); break;
      default : r.setStringParameter('cmdb_ci', ' :gear: ' + current.cmdb_ci.name);
   }
   var response = r.execute();
   var responseBody = response.getBody();
   var httpStatus = response.getStatusCode();
} catch (ex) {
   var message = ex.getMessage();
}
```

(PASTEBIN: http://pastebin.com/jCPXsdsx)

Lab Success Verification

1. Create a New Incident like in **Lab 3.0** but include an **Affected CI** with a type of Software. What do you get in the Slack channel? What other types of CMDB classes could you add? Try adding a case for "Tomcat" and use the ":cat:" emoji.

®2016 ServiceNow, Inc. All rights reserved.

Lab Goal

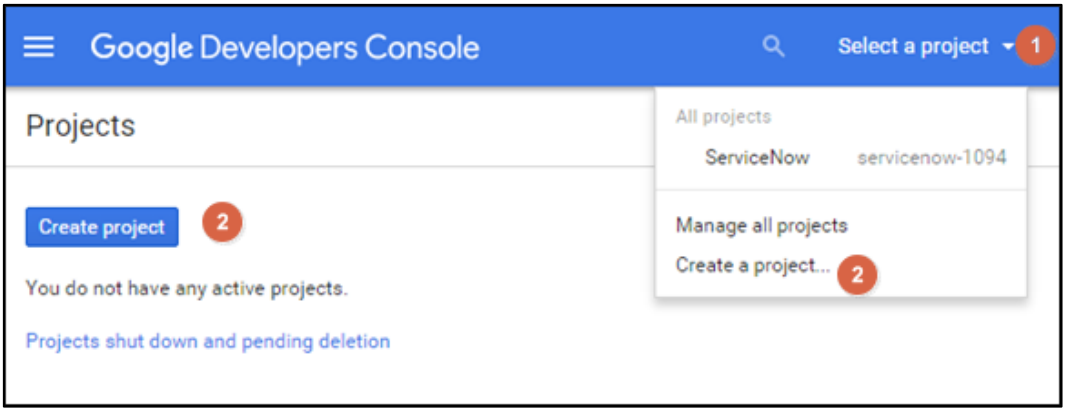
Many modern APIs no longer support basic authentication and require the more secure **OAuth** authentication method. This lab explains how to set up **OAuth** authentication with a new Google Project. At the end of **Lab 4.0** you successfully shorten a URL using the Google URL Shortener API.

Lab 4.0 Setup and configure OAuth

Lab 4.0 – Setup and configure OAuth

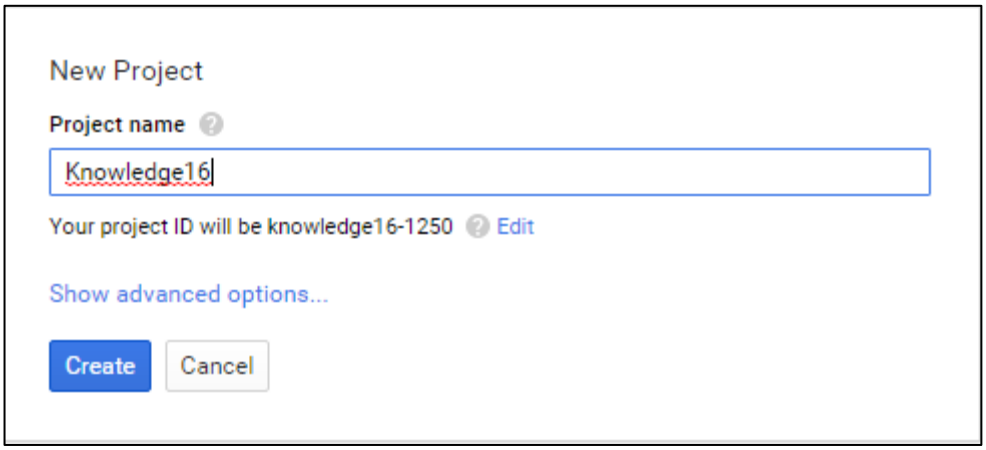
Create a New Google Project

1. Navigate to the Google Developer Console - https://console.developers.google.com
2. Log in using your Google credentials, or create them if you do not have Google credentials.
3. Click **Select a project** > **Create a project**



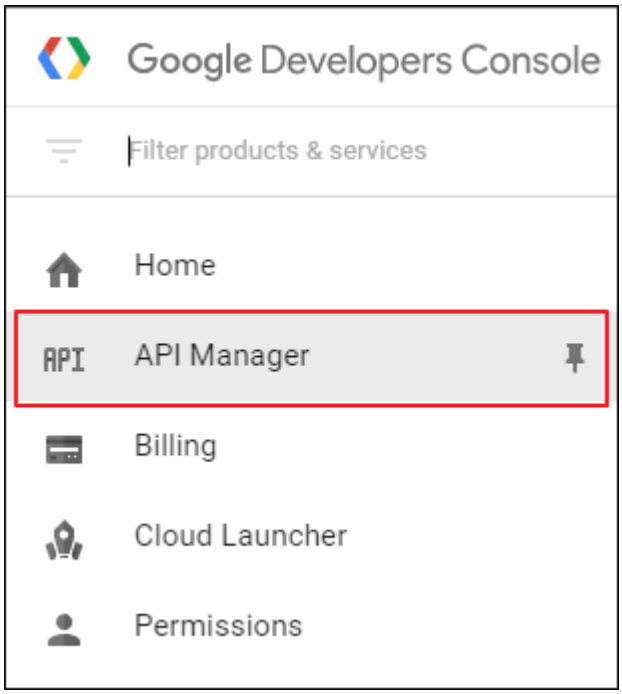
®2016 ServiceNow, Inc. All rights reserved.

4. Enter the **Project name** Knowledge16 then click Create.



After Google creates the project, the Project Dashboard appears.

5. Select the **Products & Services** icon  on the top left, and open the **API Manager**:

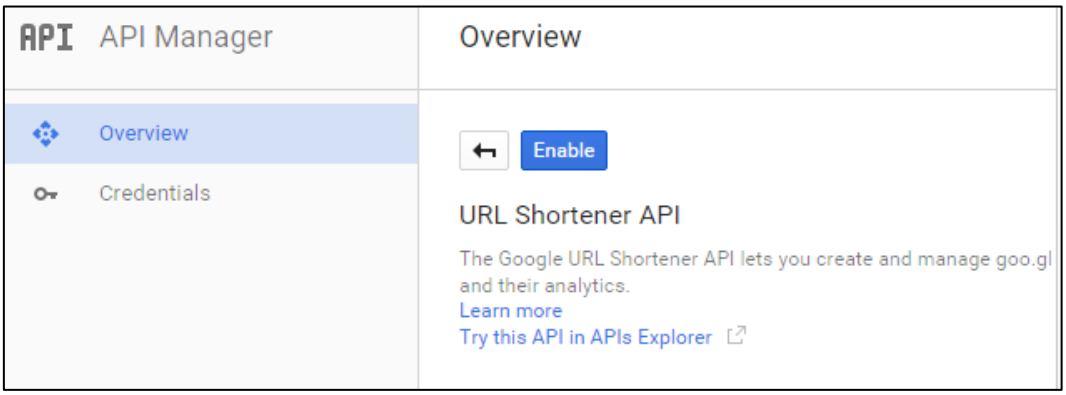


©2016 ServiceNow, Inc. All rights reserved.

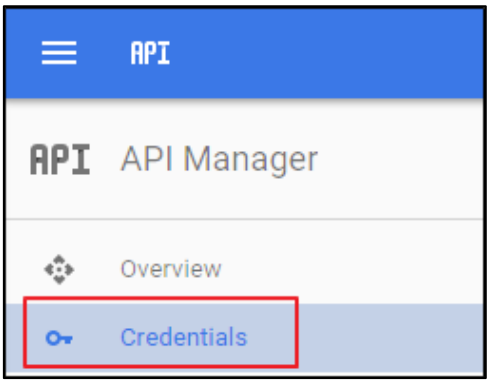
6. In the search box, enter **URL Shortener API**, and open the API from the results



7. Click **Enable**.

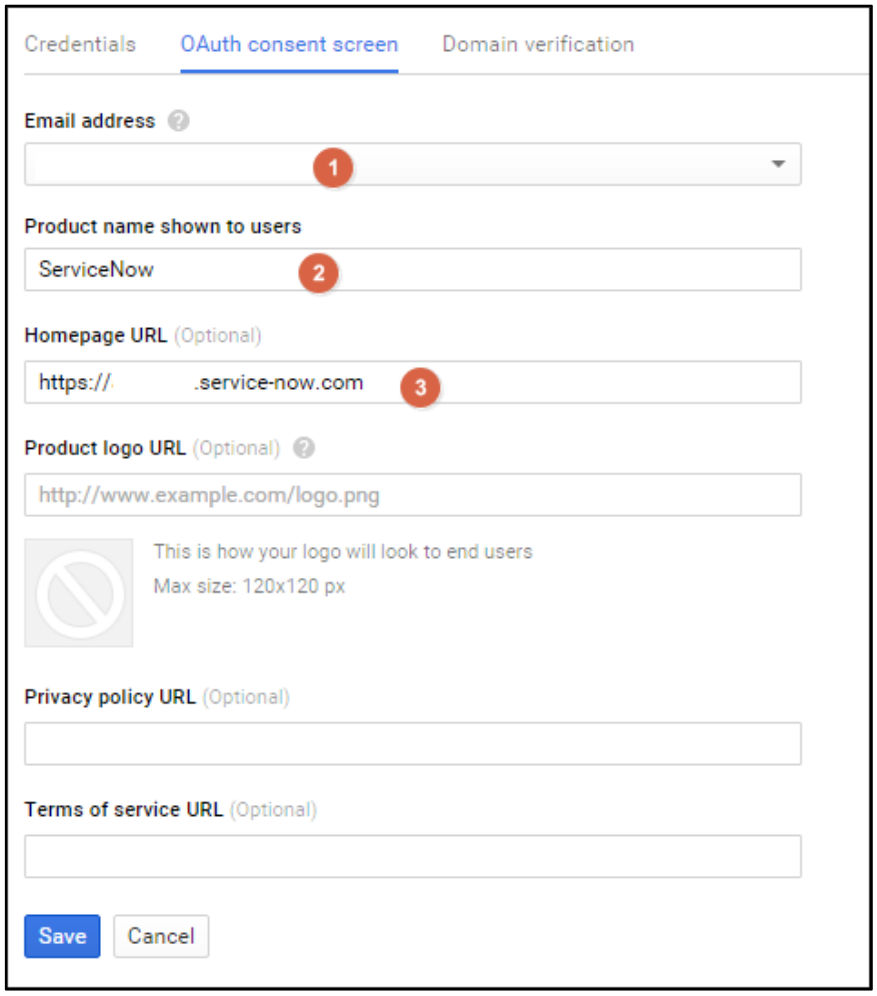


8. Navigate to **API Manager > Credentials**.



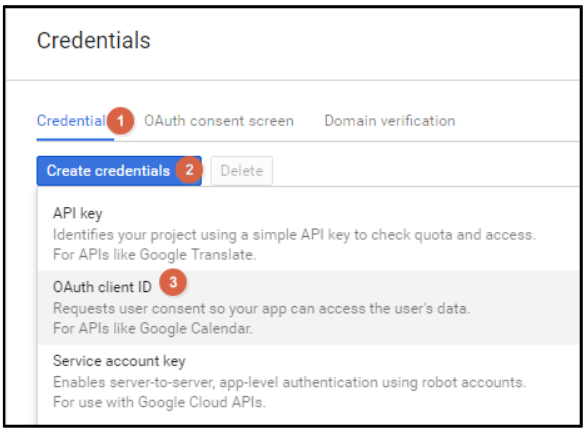
©2016 ServiceNow, Inc. All rights reserved.

9. Navigate to the **Oauth** consent screen.
   - Enter your **Product name shown to users** as **ServiceNow**
   - Enter the **Homepage URL** as your instance



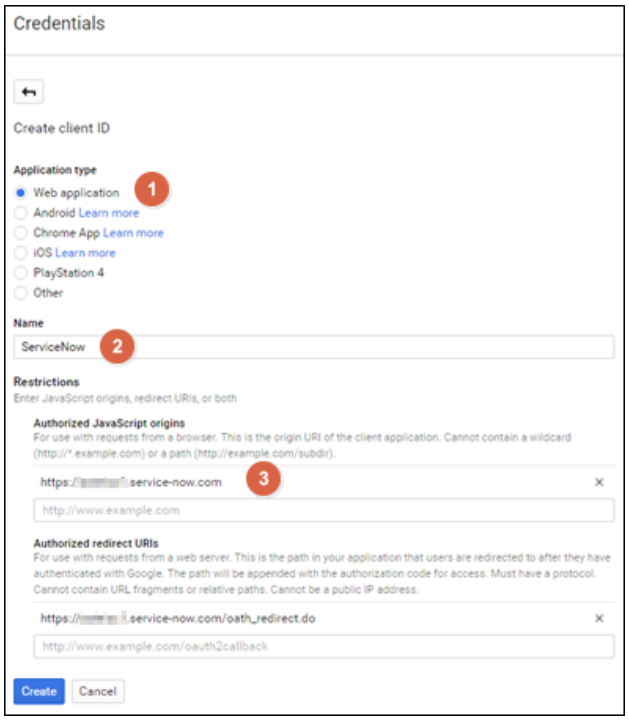
®2016 ServiceNow, Inc. All rights reserved.

10. Click **Credentials > Create Credentials > OAuth Client ID**.



11. In the **Create Client ID** window:
    - Ensure the **Web application** radio button is selected
    - Enter a descriptive **Product name**
      This name appears when you authorize the OAuth token in your instance
    - Add your instance URI in the **Authorized JavaScript origins** field
      https://<instance>.service-now.com
    - Add the OAuth redirect URI for your instance to the **Authorized redirect URIs** field
      This URL follows the format https://<instance>.service-now.com/oauth_redirect.do

®2016 ServiceNow, Inc. All rights reserved.



12. Click **Create**. The Client ID information appears.

13. Record the **Client ID** and the **Client Secret** values.

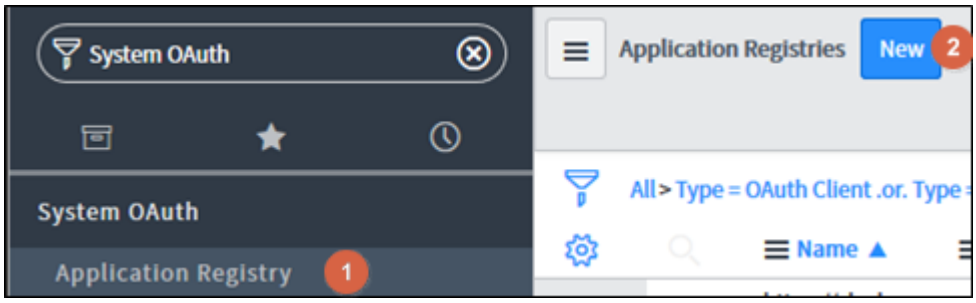


You will need these values to configure the Google service as an OAuth provider in your instance.

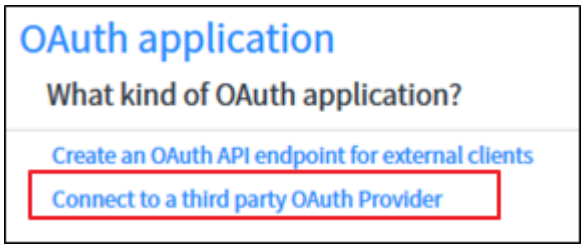
®2016 ServiceNow, Inc. All rights reserved.

Configure the Google Service as an OAuth Provider in Your Instance

1. Navigate to **System OAuth > Application Registry**. Click **New**



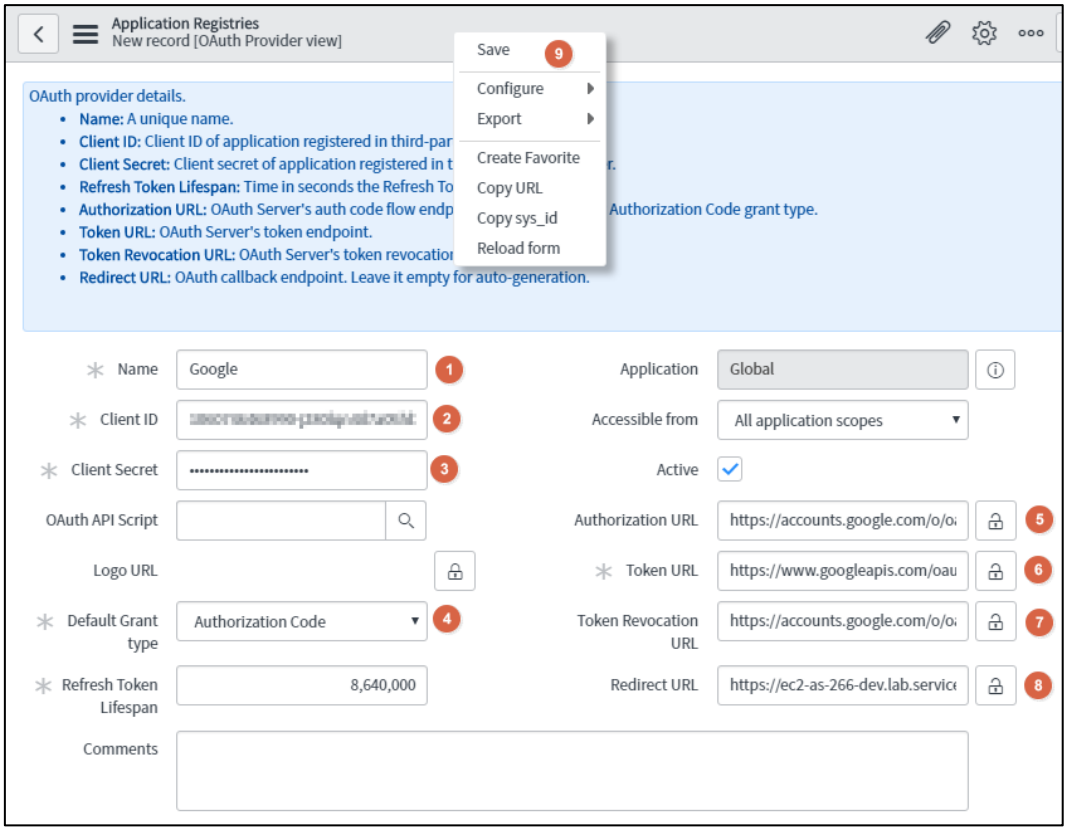
2. Select **Connect to a third-party OAuth Provider.**



3. Fill out the Application Registries form as follows:

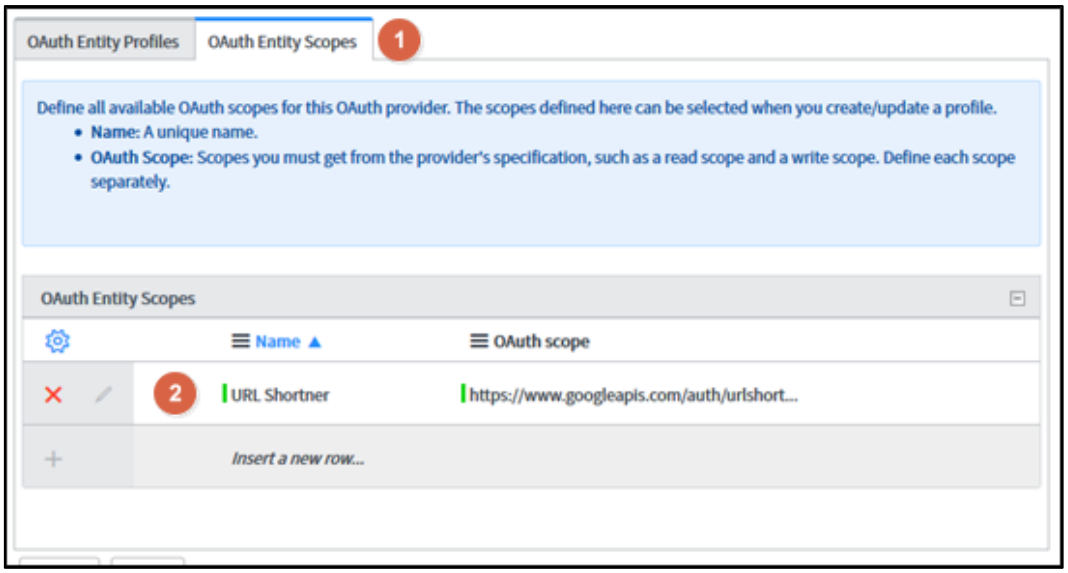
- Enter a **Name** for the OAuth provider – for this example, use **Google**
- Enter the **Client ID** that you obtained from Google
- Enter the **Client Secret** that you obtained from Google
- Set the **Default Grant type** to **Authorization Code**
- In the **Authorization URL** field, https://accounts.google.com/o/oauth2/auth
- In the **Token URL** field, https://www.googleapis.com/oauth2/v3/token
- In the **Token Revocation URL** field, https://accounts.google.com/o/oauth2/revoke
- In the **Redirect URL** field, https://<instance>.service-now.com/oauth_redirect.do
- Right-click the form header and click **Save**

®2016 ServiceNow, Inc. All rights reserved.



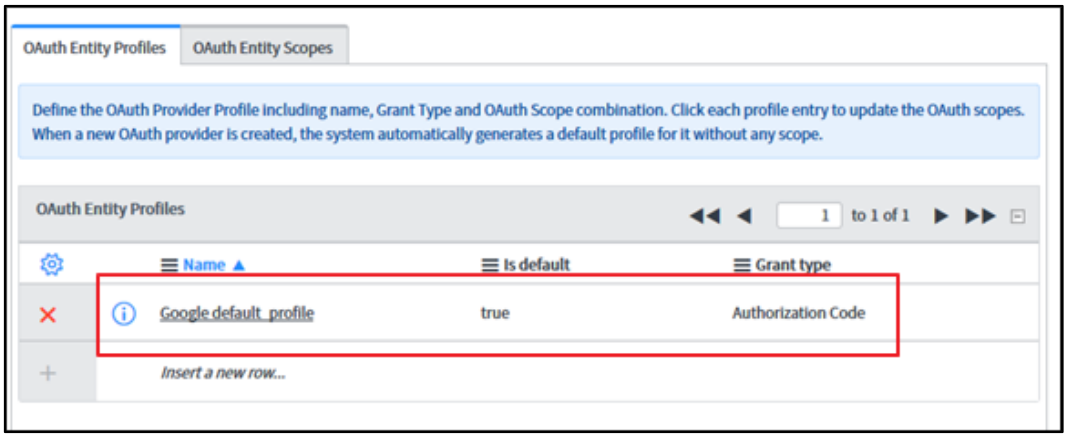
4. A new OAuth Entity Profile record is created. In the **OAuth Entity Scopes** embedded list, add a new row with the **Name** of **URL Shortener** and **OAuth scope** value set to https://www.googleapis.com/auth/urlshortener.

®2016 ServiceNow, Inc. All rights reserved.



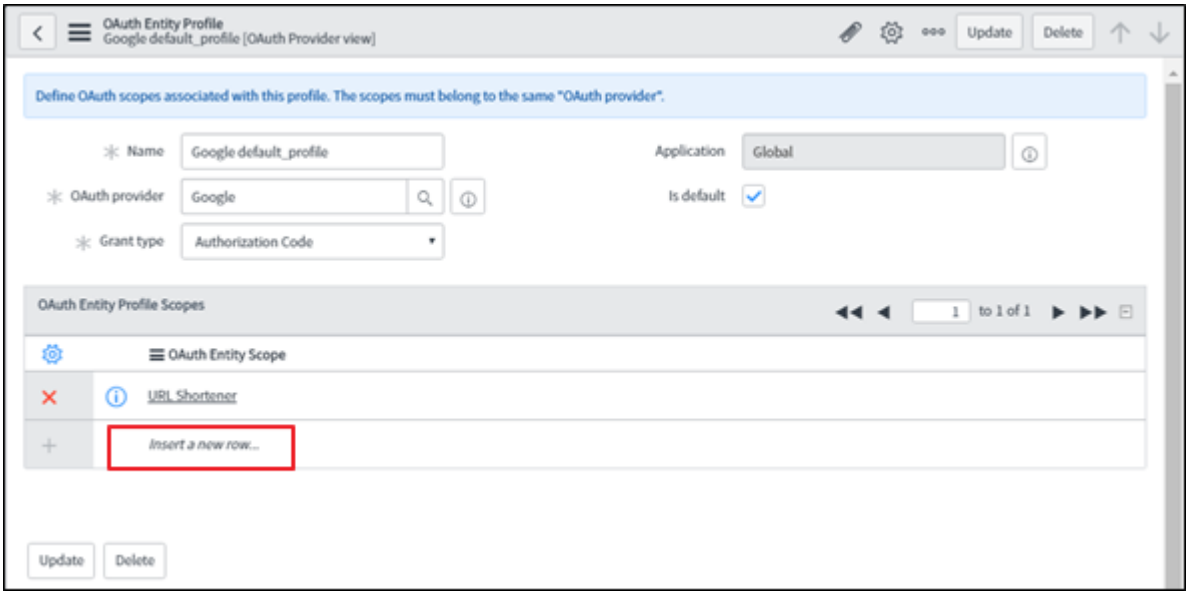
5. Right-click the form header and select **Save**.

6. In the **OAuth Entity Profiles** embedded list, select the automatically-created profile.



®2016 ServiceNow, Inc. All rights reserved.

7. In the **OAuth Entity Profile Scopes** embedded list, add a new row and select the **URL Shortener** scope.



8. Click **Update**.

Configure a New REST Message with the URL Shortener API

1. Navigate to **System Web Services > Outbound > REST Message**.
2. Click **New**.
3. On the REST Message form:

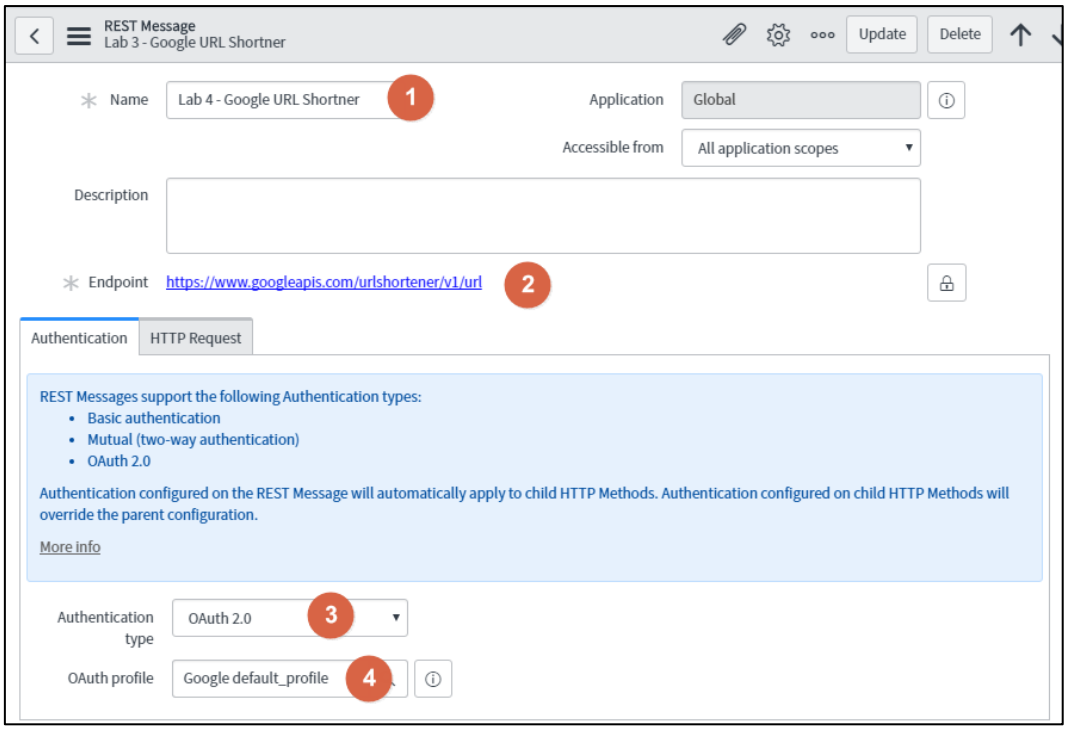
   Enter **Lab 4 – Google URL Shortener** for the **Name**

   In the **Endpoint** field, enter
   https://www.googleapis.com/urlshortener/v1/url

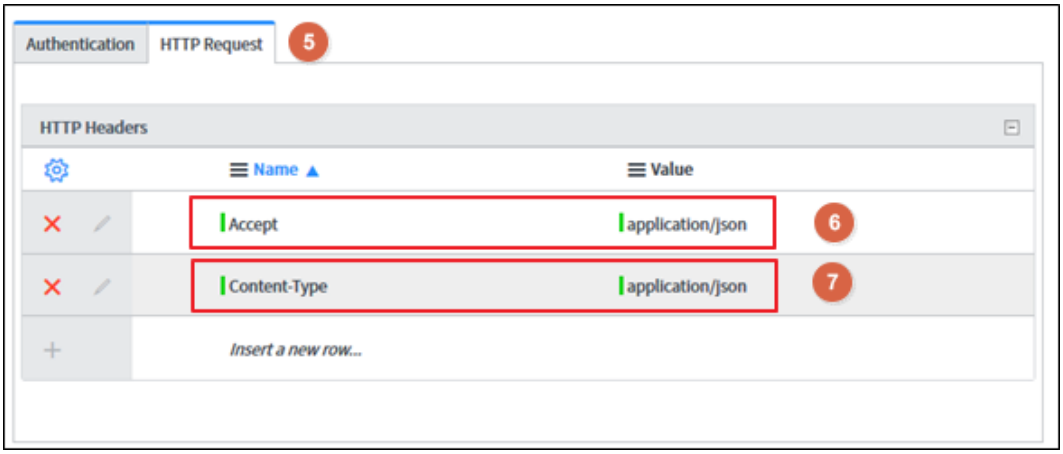
   In the **Authentication** tab, set the **Authentication type** to **OAuth 2.0**

   In the **Oath profile** field, select **Google default_profile**

®2016 ServiceNow, Inc. All rights reserved.



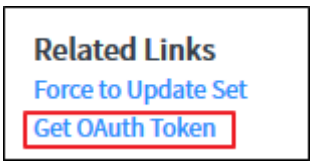
4. Add the **Accept** and **Content-Type** HTTP headers.



5. Right-click the form header and click **Save**.

®2016 ServiceNow, Inc. All rights reserved.

6. Click the **Get OAuth Token** related link to request an authorization token from Google using the configured client ID.



7. In the **Request for Permission** window that appears, click **Allow** to obtain an access token.

8. In the **HTTP Methods** related list, select the **POST** method, and populate the form as follows:

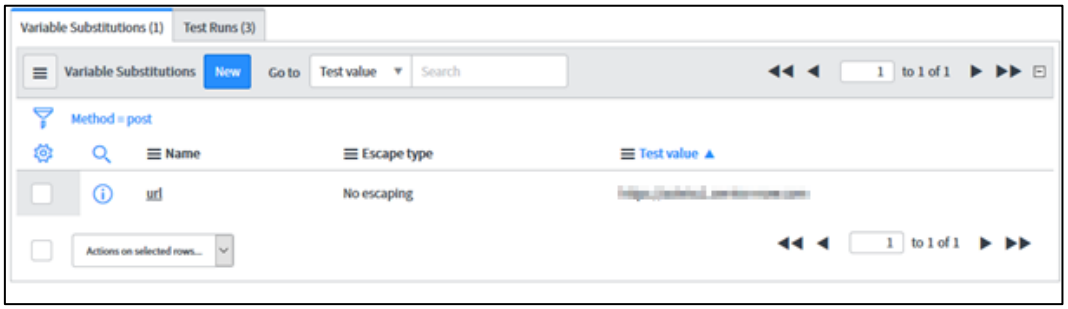
   Leave the HTTP method **Authentication type** as **-- None --** to use the OAuth profile from the parent REST message record

   Add the following to the **Content** block:

```
{
 "longUrl" : "${url}"
}
```

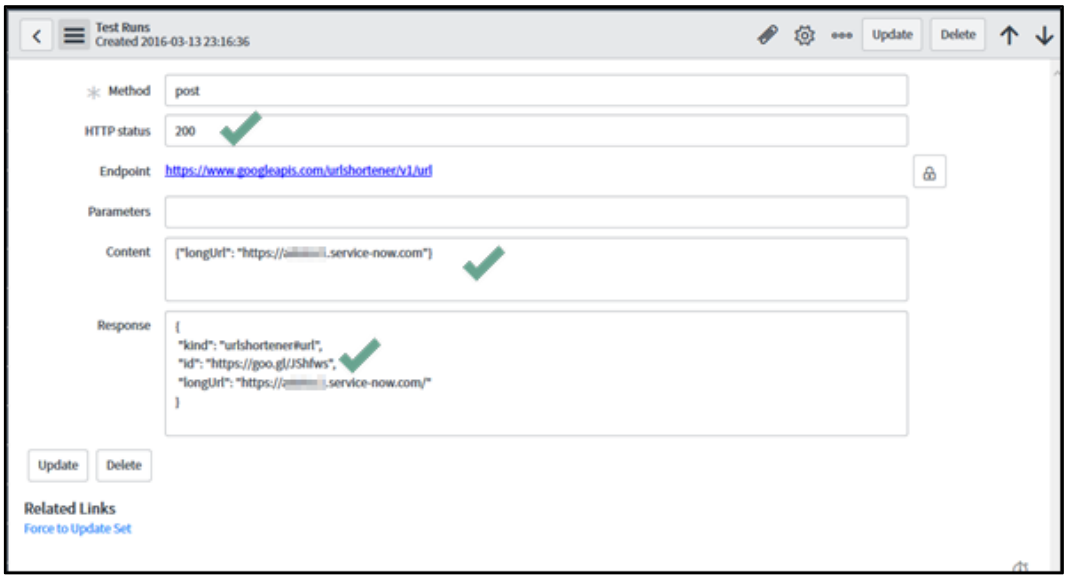
9. Right-click the form header and click **Save**.

10. Auto-generate the variable then put your instance URL as the test value.



®2016 ServiceNow, Inc. All rights reserved.

11. Click the **Test** related link. The test result should display an **HTTP Status** of 200, and the result of the contacts API call.



12. Write down the shortened URL to use in the next lab.

®2016 ServiceNow, Inc. All rights reserved.

Lab Goal

This lab explains how to use the Web Services connection in a ServiceNow script. You explore and practice using Web Services to manipulate record data. At the end of Lab 5.0 you will be able to convert a large URL into a shortened version using Google's URL Shortener API.

Lab 5.0 Using a Web Services Connection in a Script

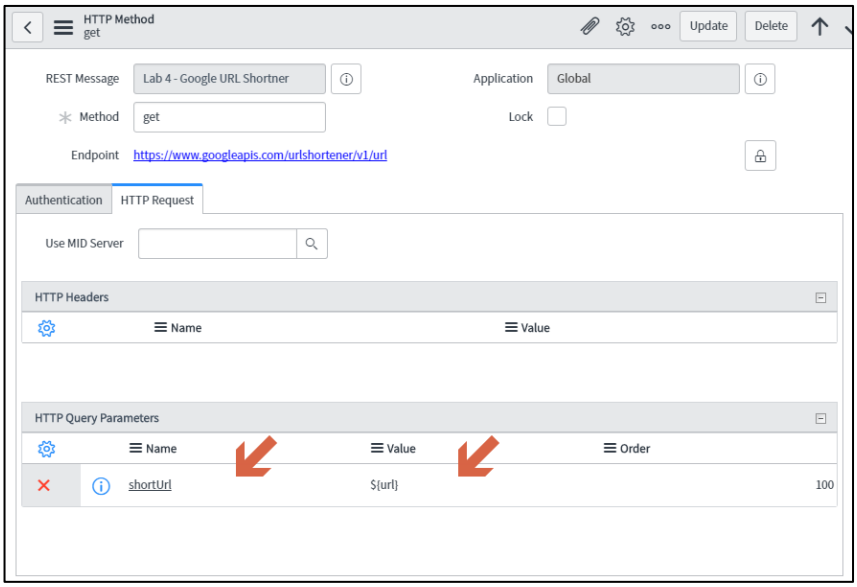
Lab 5.0 – Using Web Services Connections in a Script

Configure the GET Message

1. Navigate to **Web Services > Outbound > REST Message > Lab 4 – Google URL Shortener**. Open the **get** HTTP Method in the related list.

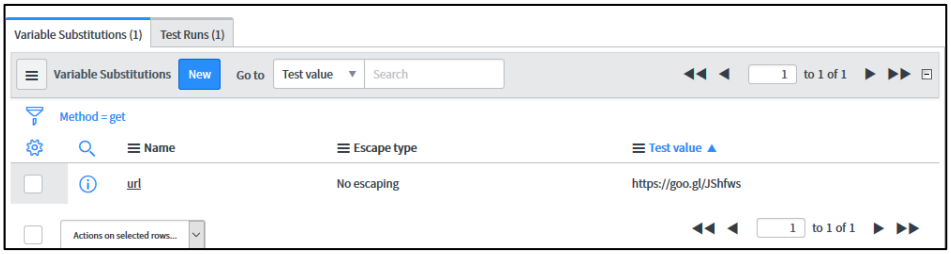
2. Add an **HTTP Query Parameter** on the **HTTP Request** tab:

   Name: **shortUrl**
   Value: **${url}**

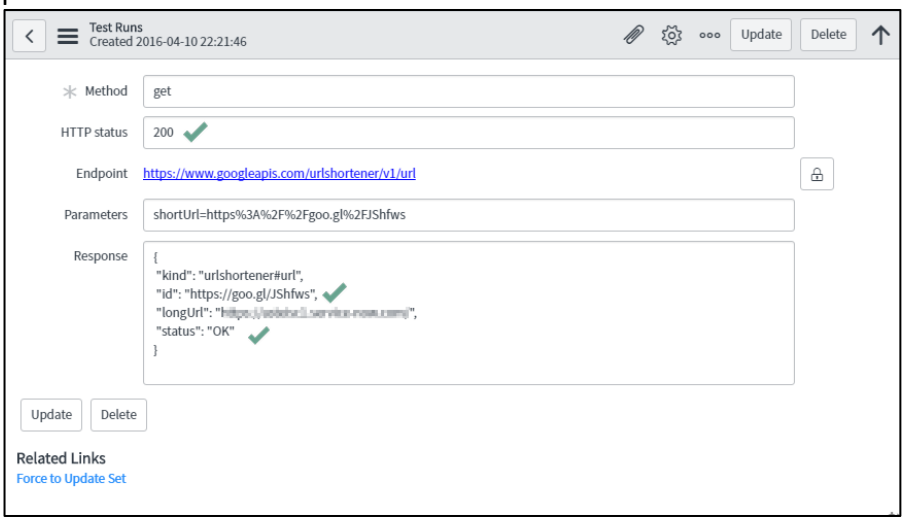


®2016 ServiceNow, Inc. All rights reserved.

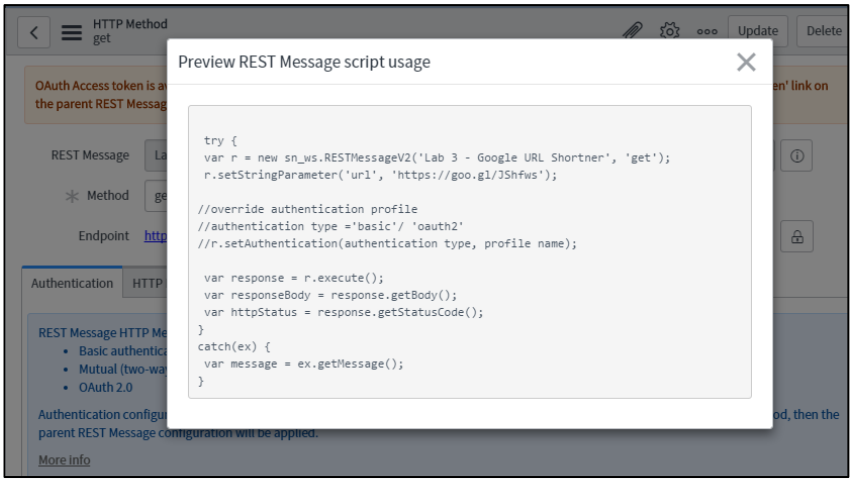
3. Auto-generate the variables and put the shortened URL generated at the end of **Lab 4.0** into the **Test value**.



4. Test the message. You should get an HTTP status of **200** and a **longURL** with a value of the URL provided in **Lab 4.0**.



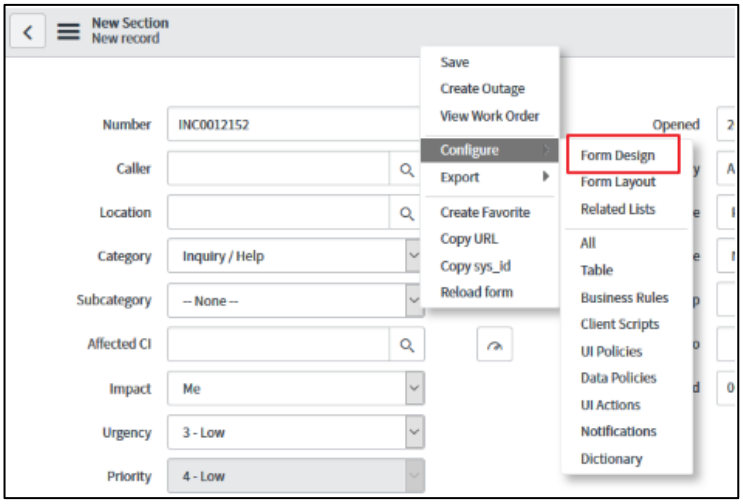
5. Return to the HTTP Method and click **Preview Script Usage** in the Related Links to use this as a template for the script.



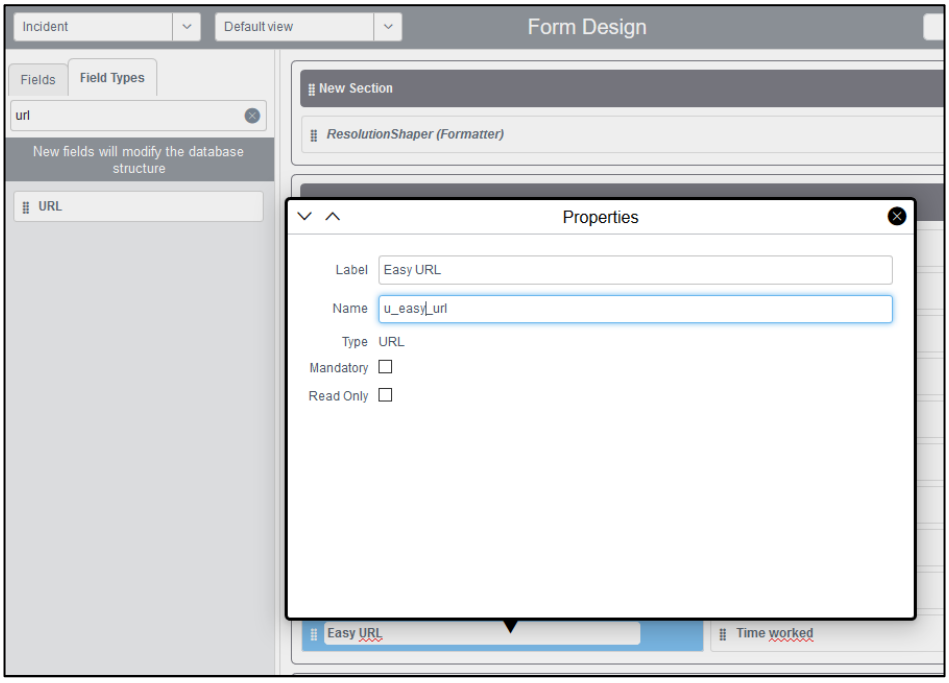
®2016 ServiceNow, Inc. All rights reserved.

Configure the Incident Schema

1. Navigate to **Incident > Create New**. Right-click in the top and **Configure > Form Design**.



2. Create a new field with a type of URL. Label it **Easy URL** and name it **u_easy_url**.



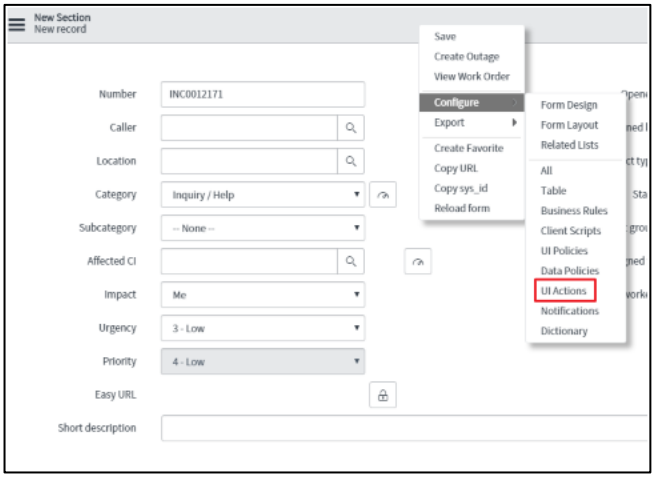
3. **Save** and reload the Incident form. You should see the **Easy URL** field on the page.



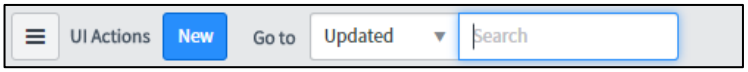
®2016 ServiceNow, Inc. All rights reserved.

Create the UI Action

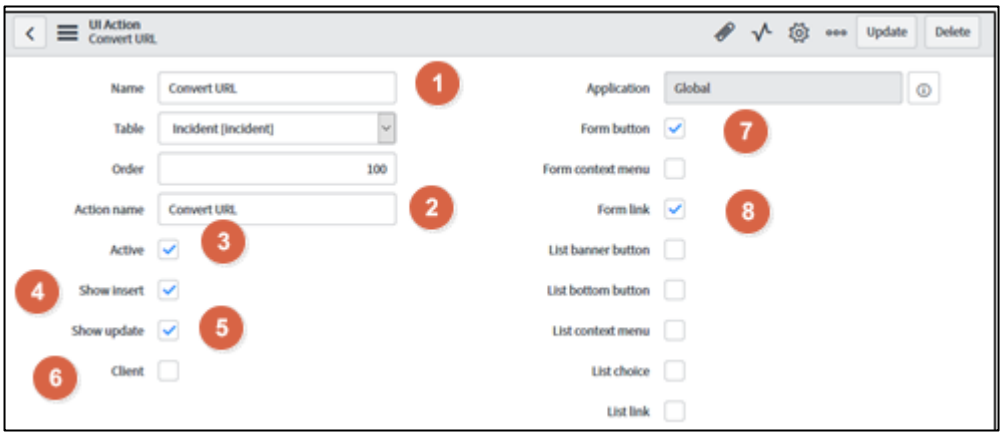
1. Right click in the top and **Configure > UI Actions**.



2. Select **New**.



3. Create a new UI Action with the following values then right click in the top and click **Save**.
   Name: **Convert URL**
   Action Name: **Convert URL**
   Active: **True**
   Show insert: **True**
   Show update: **True**
   Client: **False**
   Form button: **True**
   Form link: **True**



®2016 ServiceNow, Inc. All rights reserved.

4. Enter the following script into the Script field then **Save**. This is based on the script generated at the end of Step 1.

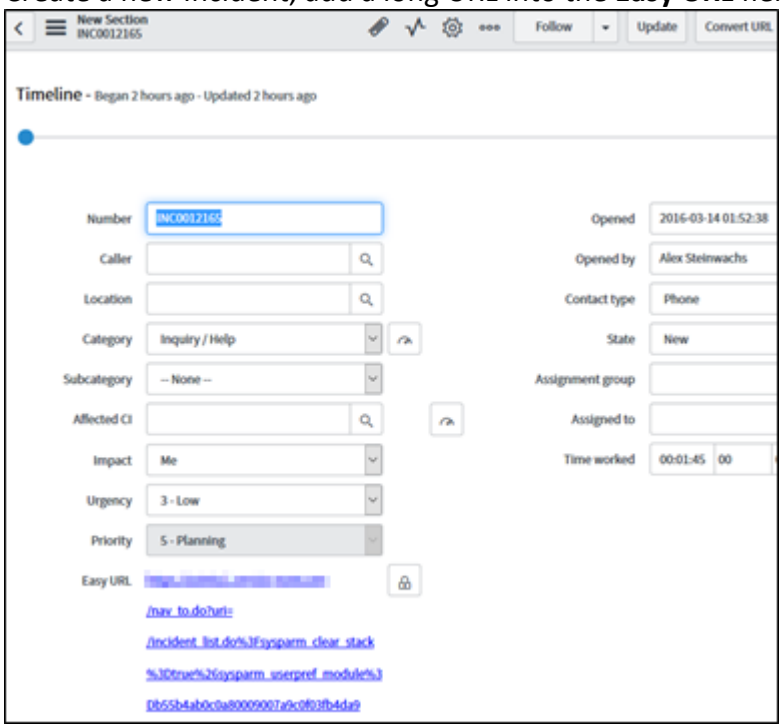
```
if (current.u_easy_url.indexOf("goo.gl") >= 0) {
   var type = 'get';
} else {
   var type = 'post';
}
try {
   var r = new sn_ws.RESTMessageV2('Lab 4 - Google URL Shortner', type);
   r.setStringParameter('url', current.u_easy_url);
   var response = r.execute();
   var responseBody = response.getBody();
   var httpStatus = response.getStatusCode();
   if (httpStatus == '200') {
      var jp = new JSONParser();
      var responseObj = jp.parse(responseBody);
      var newurl = responseObj;
      if (type == 'get') {
         current.u_easy_url = newurl.longUrl;
      } else {
         current.u_easy_url = newurl.id;
      }
   }
   current.update();
   action.setRedirectURL(current);
} catch (ex) {
   var message = ex.getMessage();
}
```

(PASTEBIN: http://pastebin.com/aW7LDEkW)

®2016 ServiceNow, Inc. All rights reserved.

Lab Success Verification

1. Create a new Incident, add a long URL into the **Easy URL** field, and **Save**.



2. Click the **Convert URL** UI Action.



3. You should see the Easy URL convert to a shortened goo.gl URL.



4. Click the **Convert URL** UI Action again. Each time you click the UI Action the URL should change form.



®2016 ServiceNow, Inc. All rights reserved.

Lab Goal

This lab is designed to give you extra practice using Web Services by re-creating the Slack Invite Request Item used in Lab 1.0. At the end of Lab 6.0 you will be able to send an invite for the slack channel to an email address.

Lab 6.0 Take Home Practice (Optional)

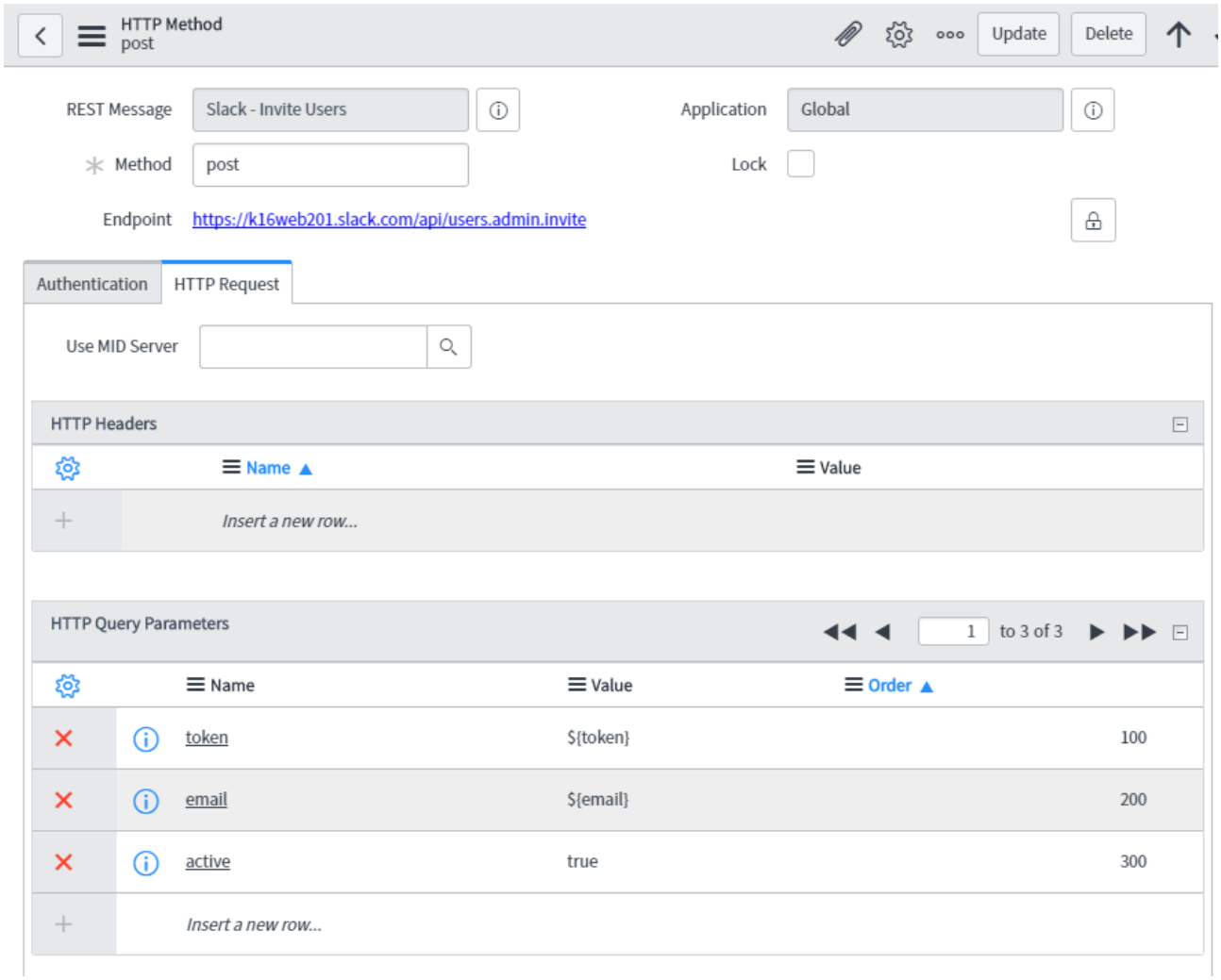
Lab 6.0 – Take Home Practice

Configure the GET message

1. Create a new Slack team and Channel using the instructions in Lab 1.0.
2. Create a new REST message with a URL of: https://YOURSLACKTEAM.slack.com/api/users.admin.invite
3. Generate a test API token here: https://api.slack.com/docs/oauth-test-tokens

®2016 ServiceNow, Inc. All rights reserved.

4. Use the following post query parameters (change the Endpoint to https://YOURSLACKTEAM.slack.com/api/users.admin.invite):

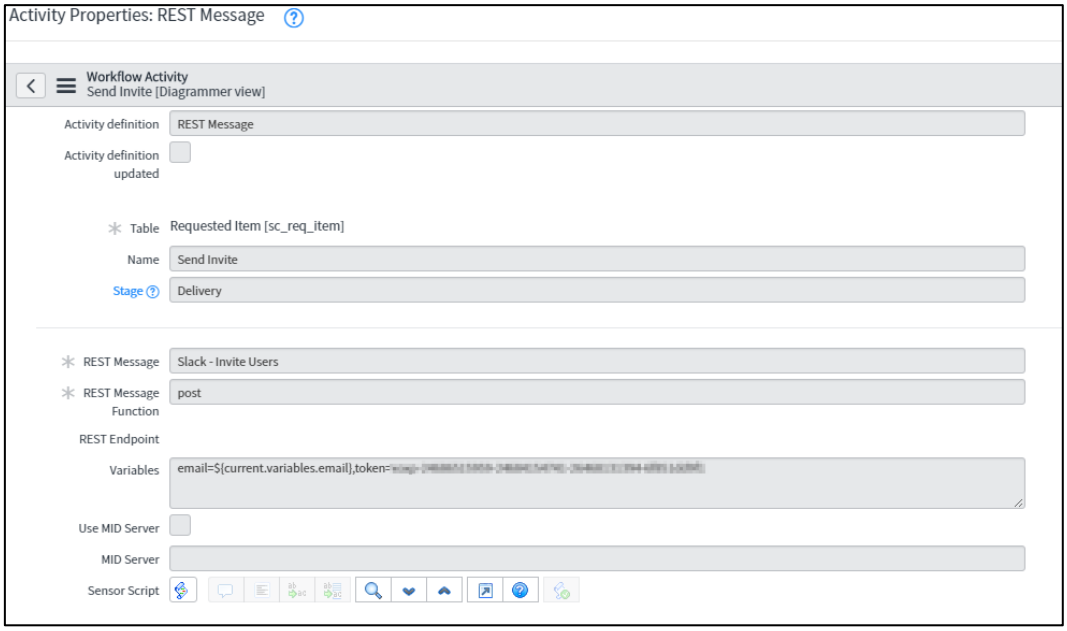


5. Use this to create a Catalog Item that can send an email value into Slack's API that has Slack send that address an invite.

®2016 ServiceNow, Inc. All rights reserved.

Answer





®2016 ServiceNow, Inc. All rights reserved.