Lab Guide

Creating Custom Applications from Scratch 201

Cory Seering & Chris Maloy

Default Login / Password:

admin / Knowledge16

itil / Knowledge16

employee / Knowledge16

This
Page
Intentionally
Left
Blank

Lab Goal

Lab 1.1 Setup, Git, and Create App with Data Model

On your marks, get set up, GO! The goal of this lab is to make sure you are set up properly to complete the subsequent labs.

Be sure that you:

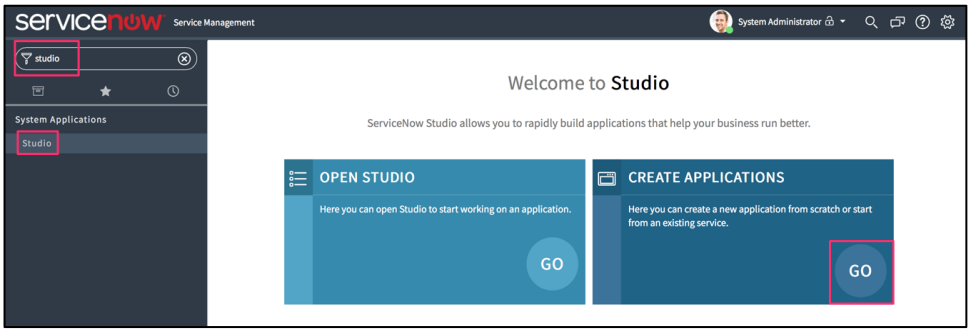
- Have proper access to your assigned instance
- Can find and open Studio
- Create an empty application from Studio

Log in to Your Provided Instance

1. Navigate to the unique instance URL provided to you.
2. Log on with provided credentials.

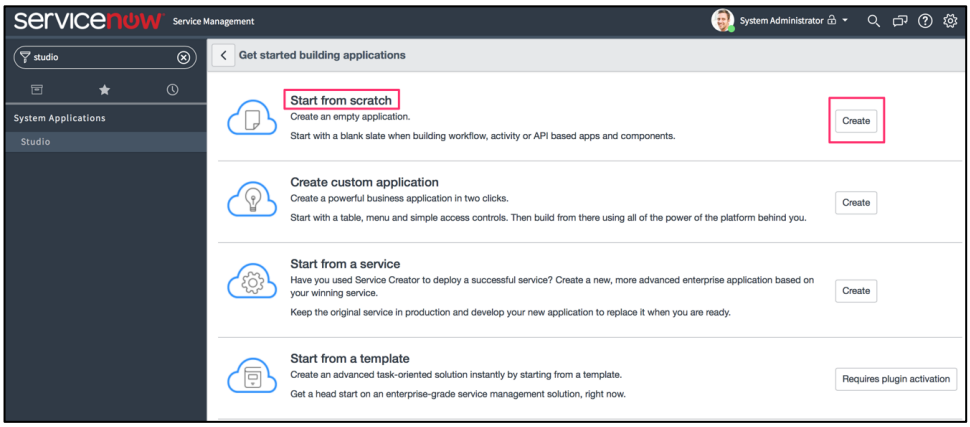
Open Studio, Link to Git Repo, and Create an Application

1. On your instance, locate and find the Studio module and open it.

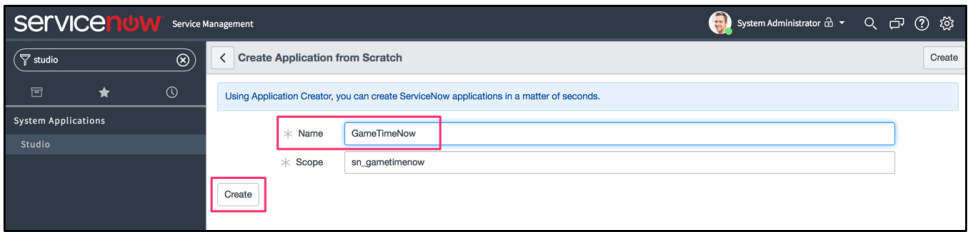


®2016 ServiceNow, Inc. All rights reserved.

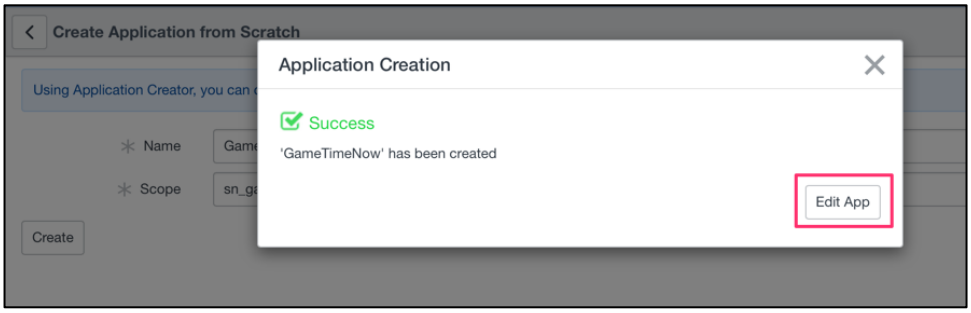
2. Create an application from scratch.



3. Name the application **GameTimeNow** and click the **Create** button.

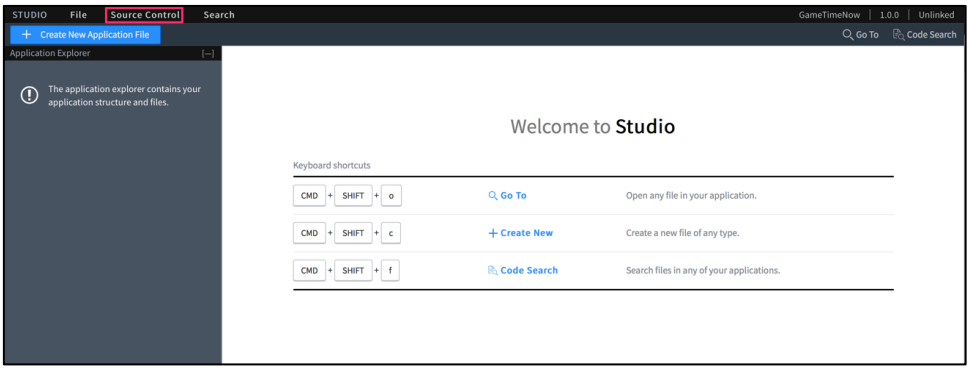


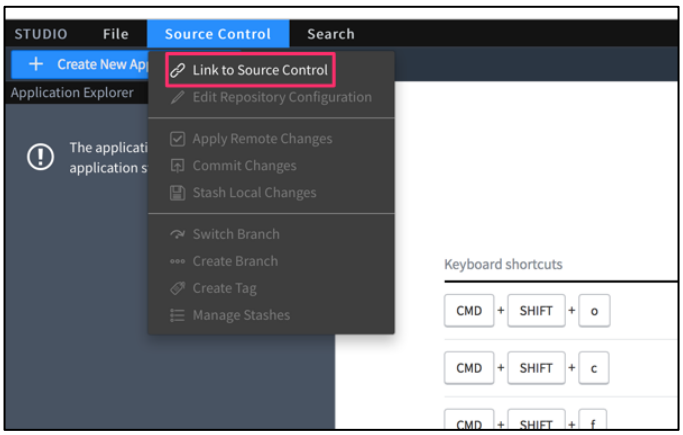
4. Edit this application.

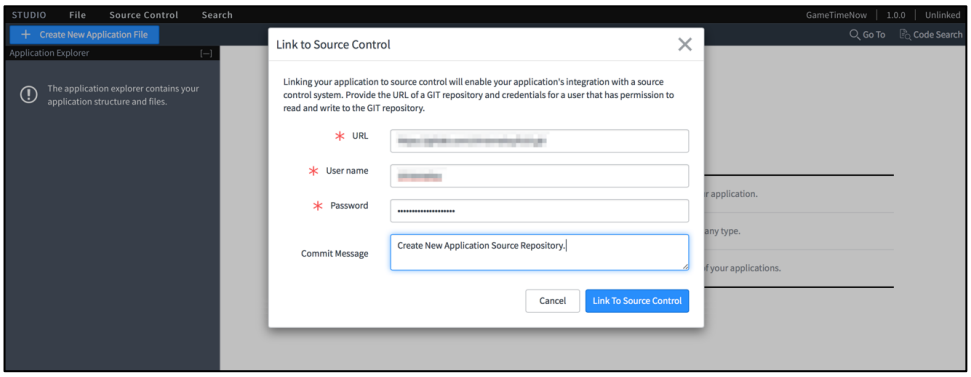


®2016 ServiceNow, Inc. All rights reserved.

5. **(Optional Step)** You are now in the Studio environment ready to start building your application. Notice the Source Control link. If you already have a Git account, you can link it now (so your new application can be pushed up to your source repository for safe keeping). This is not a required step – as you can finish this class without having a repo, but we recommend you do it.

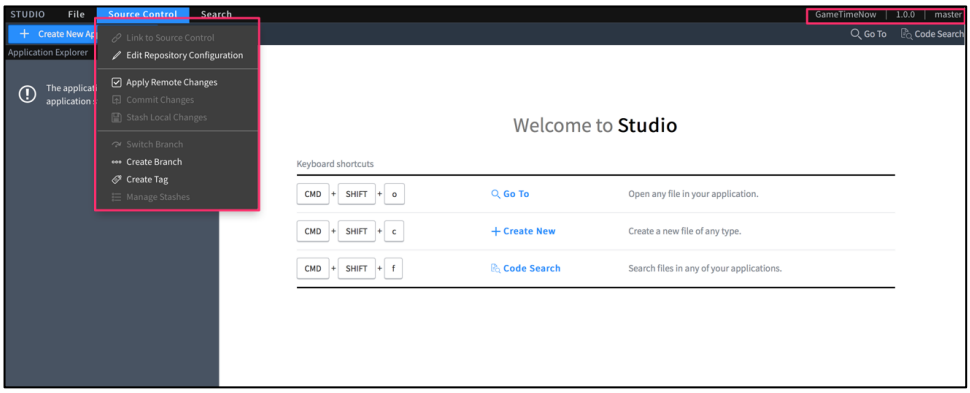




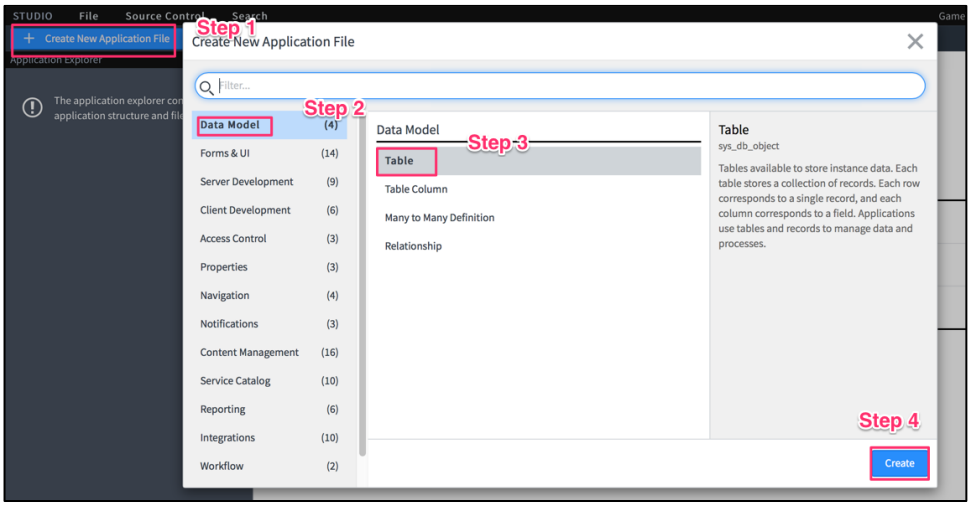


®2016 ServiceNow, Inc. All rights reserved.

6. Now that you are linked to a source repo you have some added functionality. You can attend the Git lab to learn all about this new feature. This is a great way to move your application across environments.



7. Create your **data** mode.

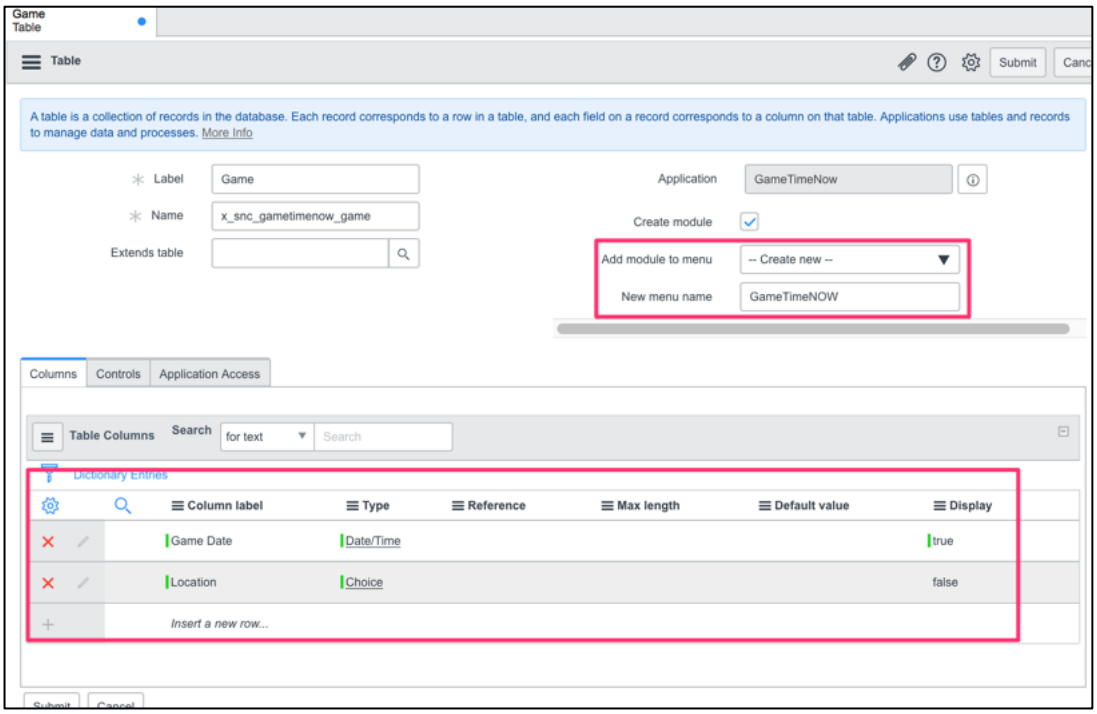


®2016 ServiceNow, Inc. All rights reserved.

8. Create the Game Table. You create two columns and a new Menu with this step;

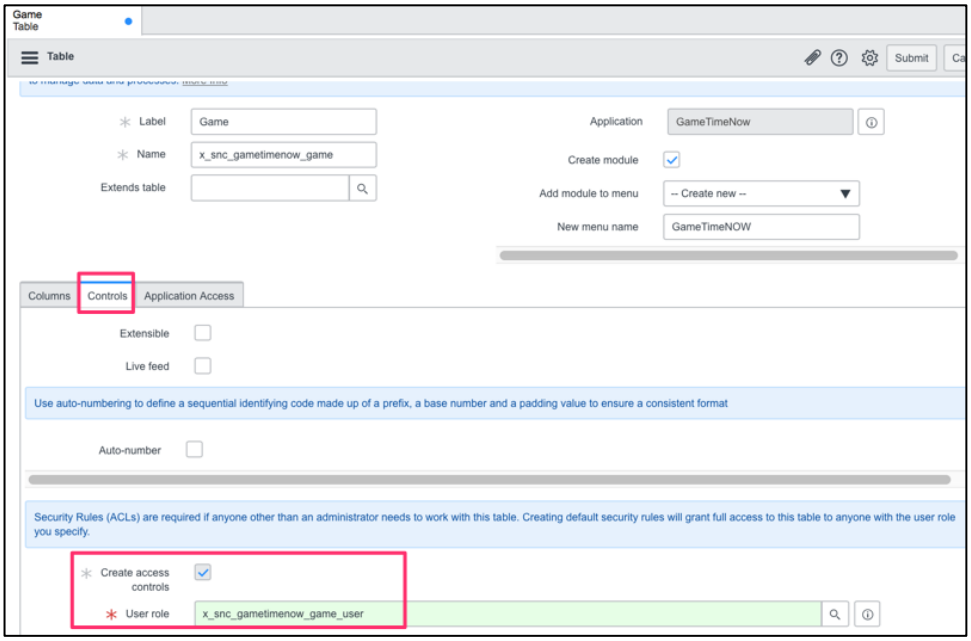
   Column label: **Game Date**
   Type: **Date/Time**
   Display: **True**

   Column label: **Location**
   Type: **Choice**

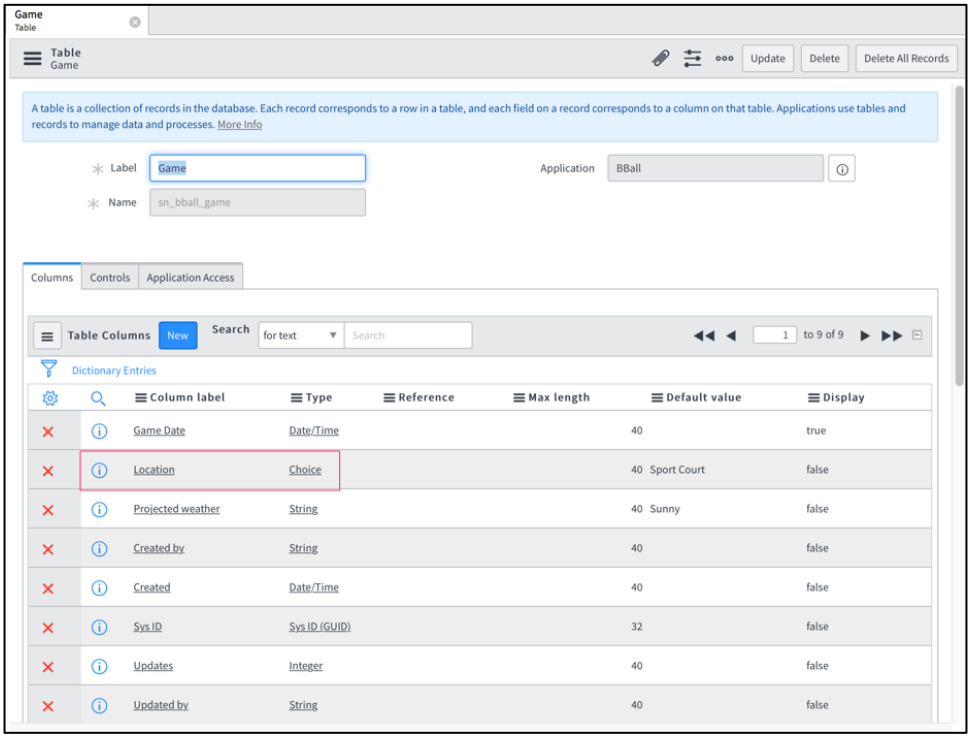


©2016 ServiceNow, Inc. All rights reserved.

9. Access Controls are created by default.



10. Open the **Location** column you just created to set up some choices.



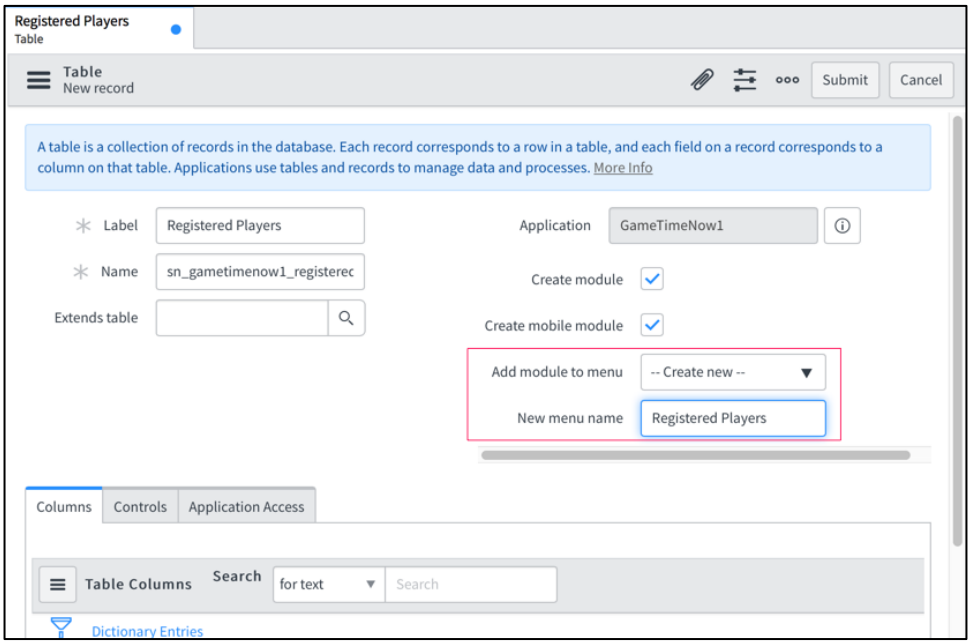
®2016 ServiceNow, Inc. All rights reserved.

11. Create two new choices:

    Column label and Value: **Sport Court**
    Column label and Value: **The Park**
    Choice: **Dropdown with – None** --

12. Create the Registered Players Table and add the table as a module to the application.

    Column label: **Player**
        Type: **Reference**
        Reference: **Users (sys_user)**
        Display: **True**



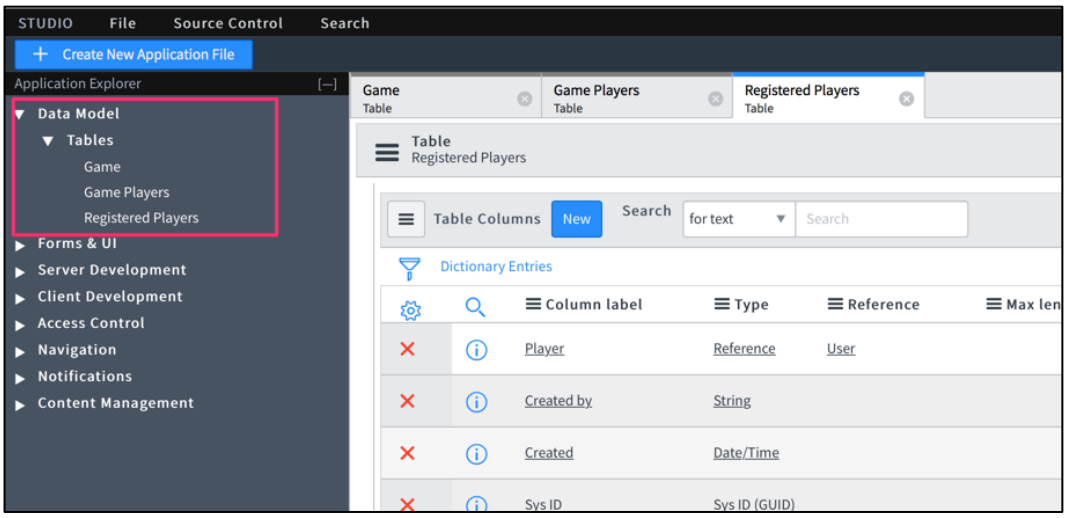
©2016 ServiceNow, Inc. All rights reserved.

13. Create the Game Players Table. Make sure when creating this table you add a new module to the application.

    Column label: **Player**
        Type: **Reference**
        Reference: **Registered Player**
        Display: **True**
    Column label: **Game**
        Type: **Reference**
        Reference: **Game**
    Column label: **Playing**
        Type: **True/False**

Lab Success Verification

1. You should have three tables created in the Application Explorer.



©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

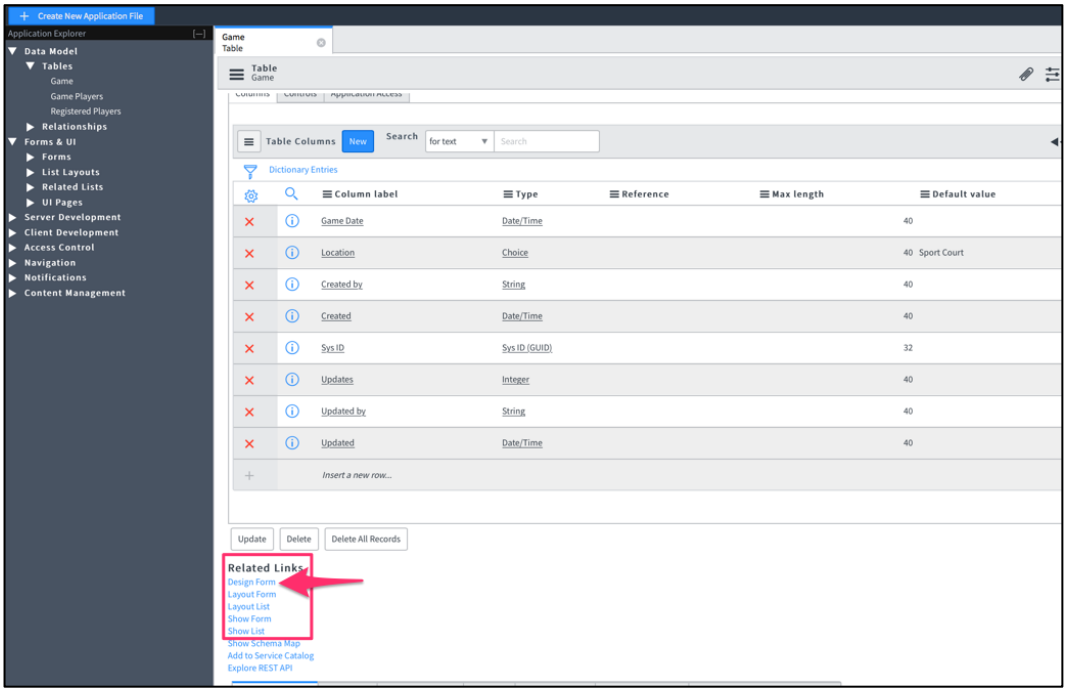
The tables created when setting up the data model dynamically generate a form based on the columns set for it. These forms give users, who have access, a simple UI when interacting with the data model.

This lab explains how to get familiar with the Form Designer and create a default view for the tables just created.

Lab 2.1 Form Designer

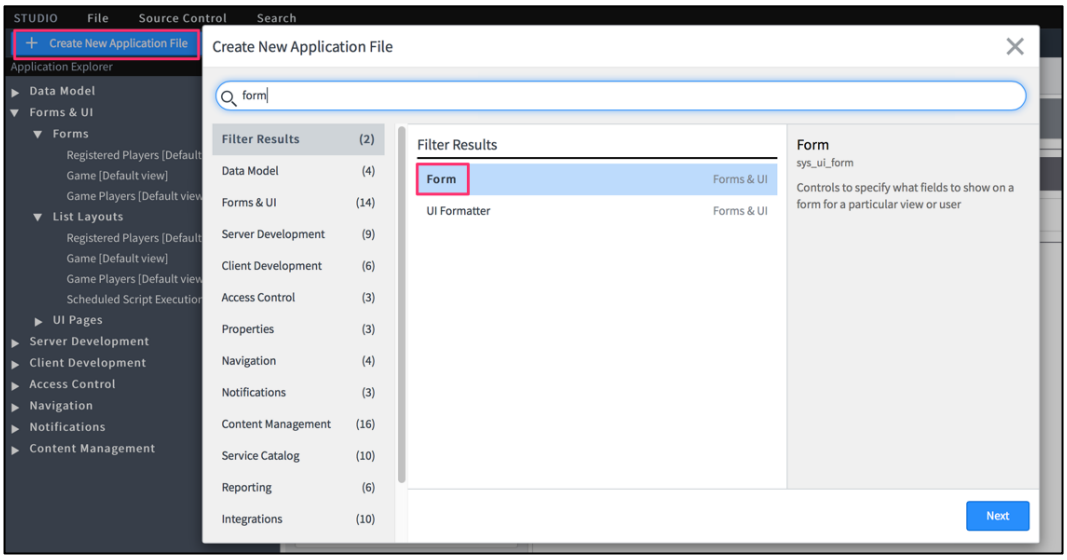
Create a Default View for the Records

1. Each of the tables you created has related links that help design the default forms and list. We prefer **not** to use these related links as they steal focus away from the rest of Studio.

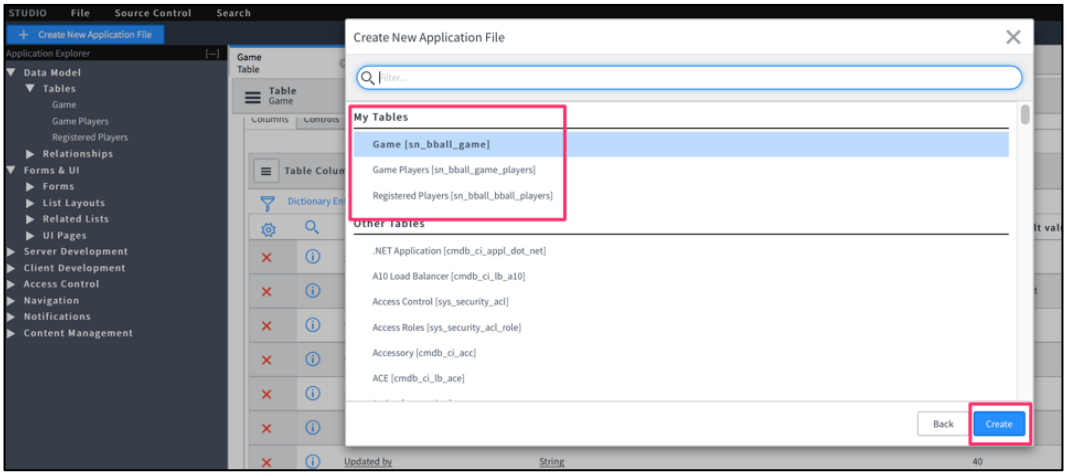


®2016 ServiceNow, Inc. All rights reserved.

2. Click the **Create New Application File** button and search for **Form**.

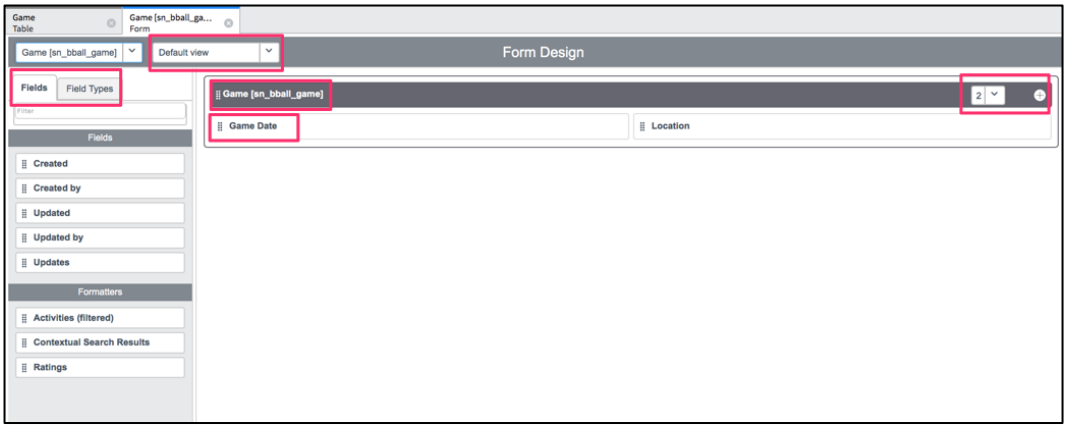


3. Click **Next** and choose the table you wish to use the Form Designer on.



®2016 ServiceNow, Inc. All rights reserved.

4. Notice the many things you can do in the Form Designer. You can change the data model by adding more fields (columns) to the table. You can update and change some basic properties of the fields you have defined (like making a field mandatory). You can create different views, and change the layout of the form by adding new sections and columns to the form.



5. Modify the three tables you created in this fashion to have the layout you want. Change the property of some fields to be mandatory. You need to do this by using the **Default View** for the three tables you created in Lab 1.

®2016 ServiceNow, Inc. All rights reserved.

Lab Goal

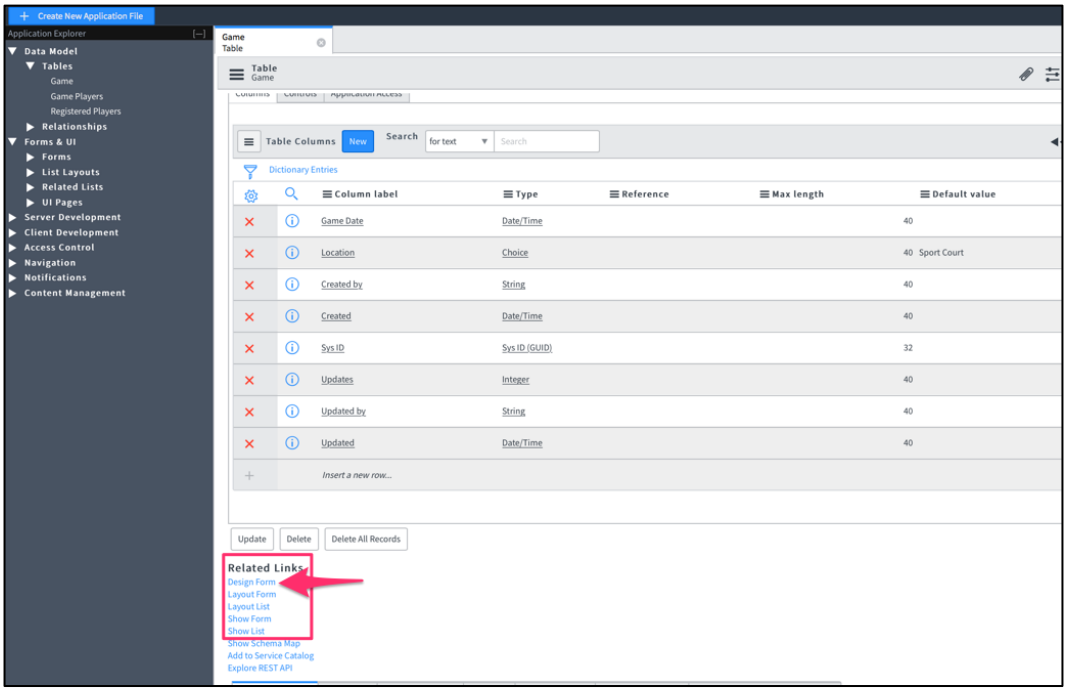
Lab 2.2
List Layout View

Remember that a form represents a record in ServiceNow, but when you query for a list of records on a particular table you get a spreadsheet-looking grid view. This is called a list view. Often, especially with tables that have a lot of columns, you do not want every column on this view, just the most important fields from left to right.

This lab explains how to create a default list layout for your data model.

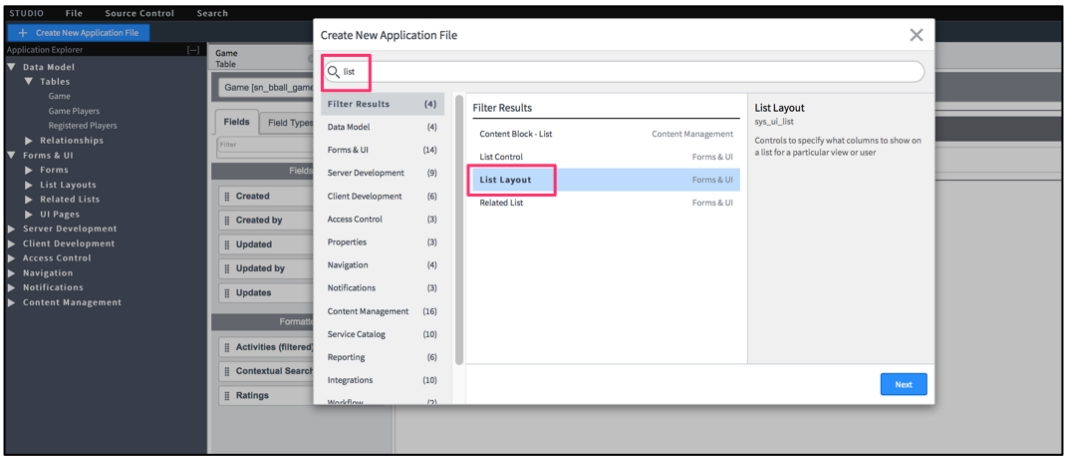
More Formatting Instructions

1. Each of the tables you created has related links that help design default forms and lists. We prefer not to use these related links as they steal focus away from the rest of Studio.

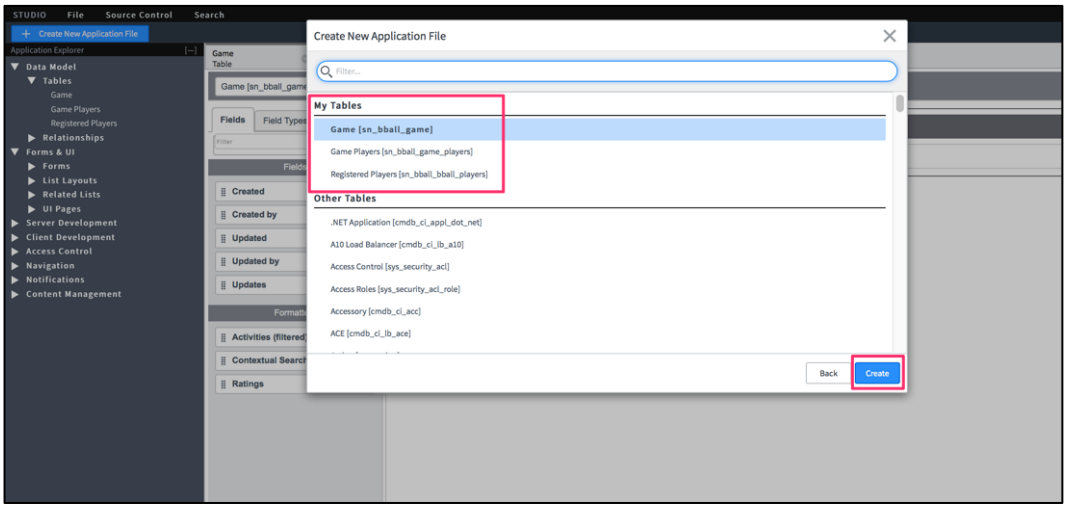


©2016 ServiceNow, Inc. All rights reserved.

2. Click the **Create New Application File** button and search for **List**.



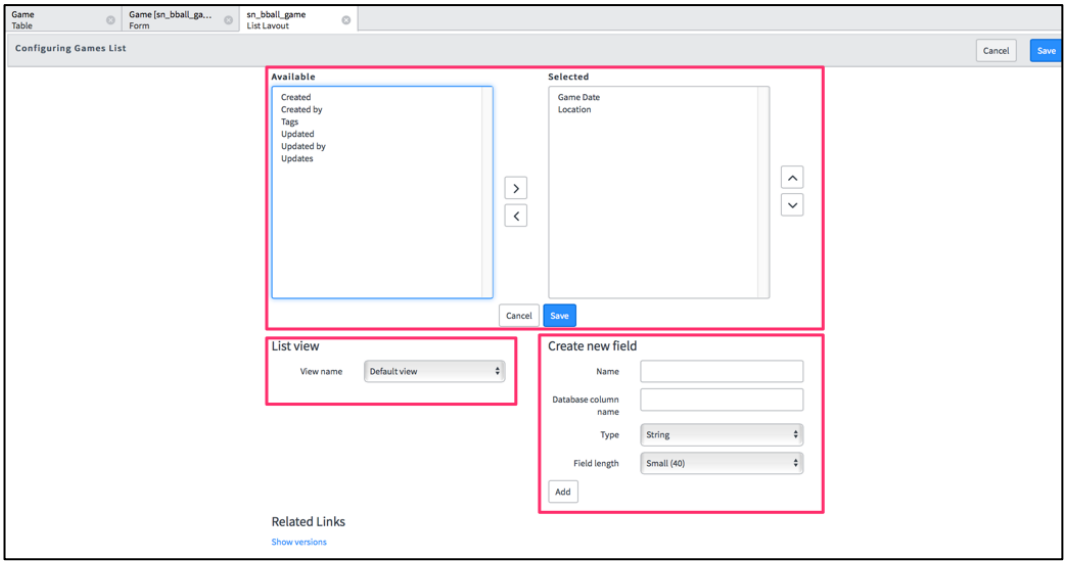
3. Click **Next** and choose the table you wish to use the List Layout on.



©2016 ServiceNow, Inc. All rights reserved.

4. There are many sections to this designer. You can select fields that your data model already contains. The fields selected show up from left to right on the list based on the order you placed them in the selected slush bucket.

   You can create new data fields or a different view from this screen as well. We will not be doing that in this lab – just know you can.



5. Repeat this exercise for the three tables you created using the format you like best.

©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

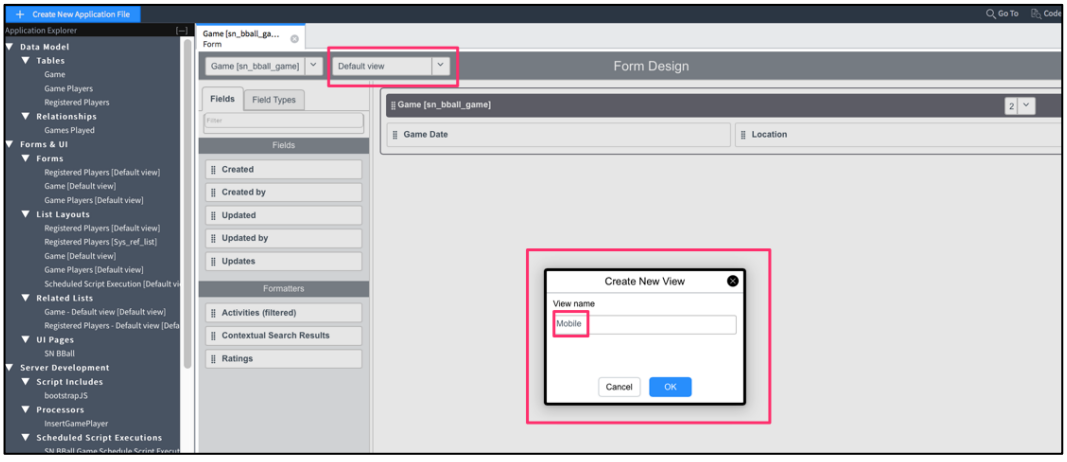
Lab 2.3 Mobile View & Related List

Applying the principles learned earlier, this lab explains how to create a custom view for mobile. ServiceNow automatically renders data in the best mobile format it can, but you have the option of overriding this with a custom view.

This lab also explains how to create a related list to show up when the form is rendered. Related lists are tables of data that are related in some way (either by foreign key in the database or an explicit relationship you create) to the data / form you are rendering.

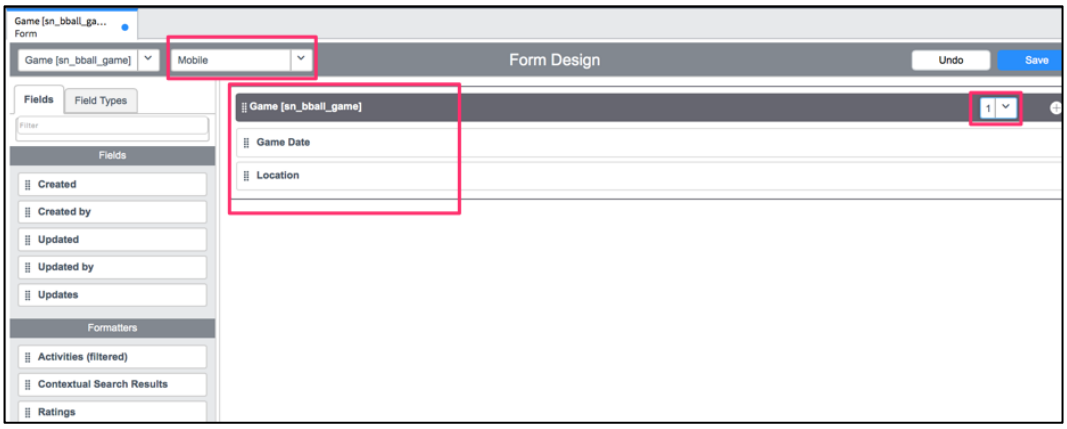
More Formatting Instructions

1. Open the form designer again for our Game table. Do you remember how to do that? Create a new Mobile view from Form Designer.



®2016 ServiceNow, Inc. All rights reserved.

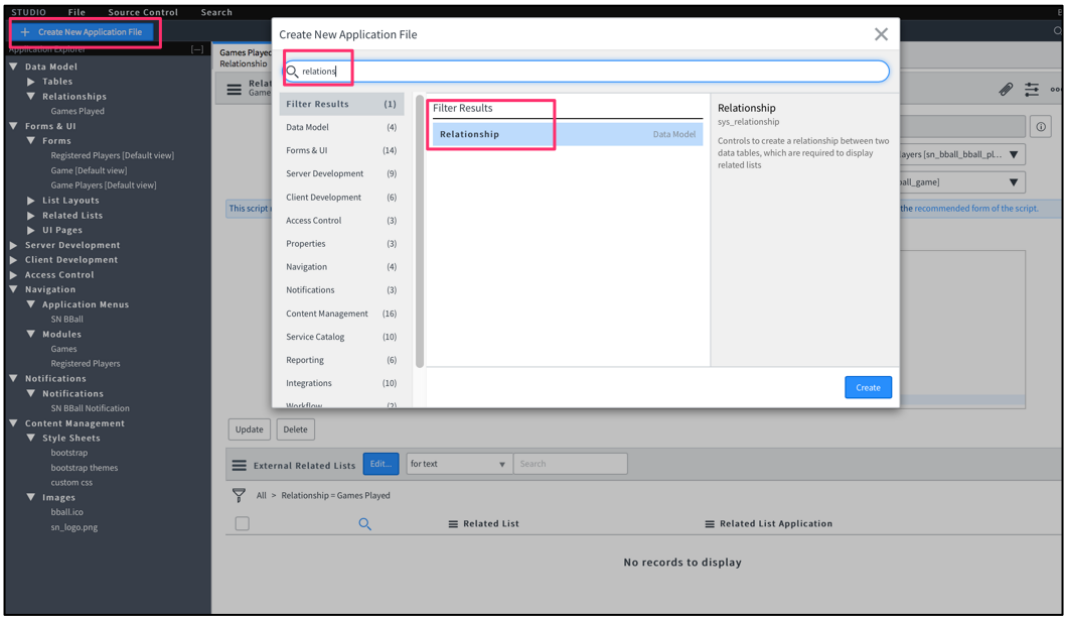
2. Change our main section to have **1** column to render instead of 2.



3. Click **Save**.

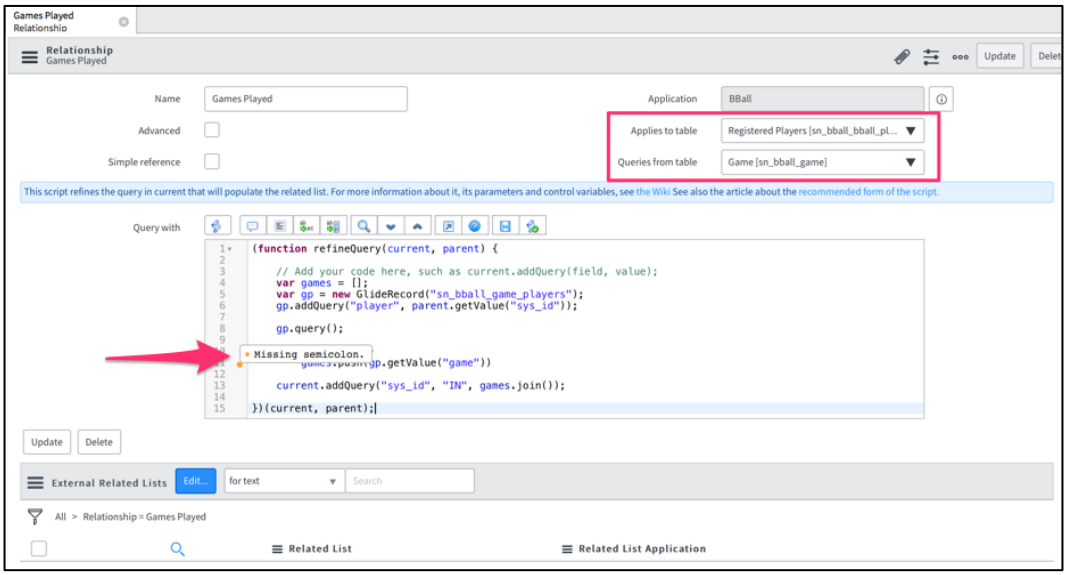
4. Now build the related list. We wanted to throw in the concept of explicitly defined relationships into this lab, so you need some code for this part (do not worry, we wrote it and supply it to you).

   Build the relationship: Create a new **relationship application** file.



©2016 ServiceNow, Inc. All rights reserved.

5. The **Applies to table** field needs to be set to **Registered Players**. The Queries from table needs to be set to **Game**. Copy and paste the code provided into the **Query with** field.



//NOTE – your new GlideRecord table may have a different name prefix.

```
(function refineQuery(current, parent) {

  // Add your code here, such as current.addQuery(field, value);
  var games = [];
  var gp = new GlideRecord("sn_bball_game_players");
  gp.addQuery("player", parent.getValue("sys_id"));

  gp.query();

  while(gp.next())
        games.push(gp.getValue("game"));

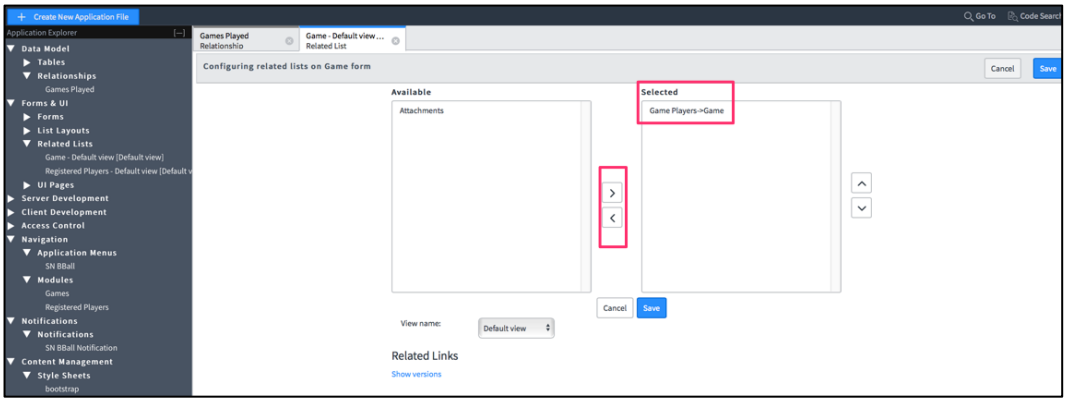
  current.addQuery("sys_id", "IN", games.join());

})(current, parent);
```

©2016 ServiceNow, Inc. All rights reserved.

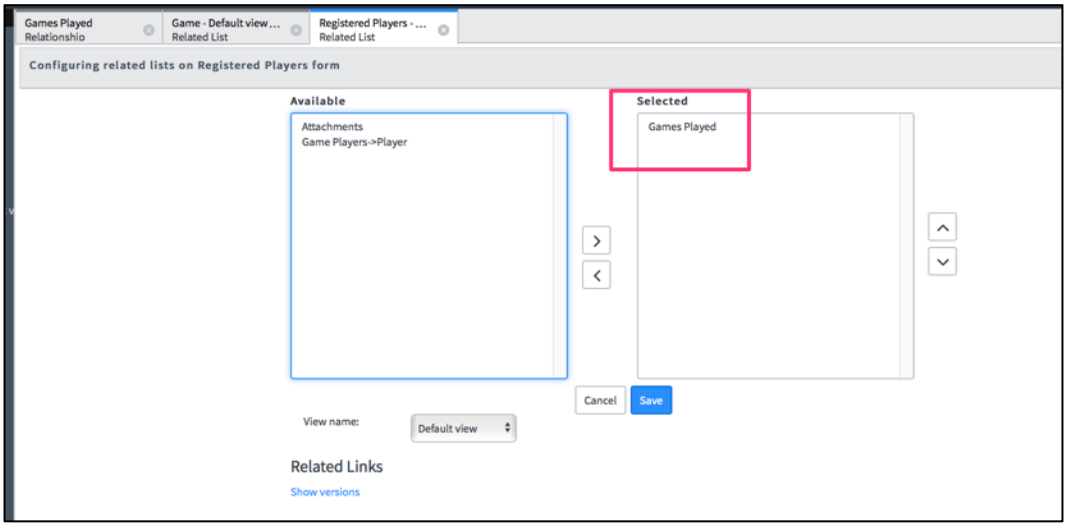
6. Now that you created a relationship, you can update the forms with these related lists (to show that the related data will render on the default table forms).

   Create a new related list application file (hopefully we don't have to screenshot this step anymore) for the Game table and select **Game Players > Game relationship**.



7. Click **Save**.

8. Now create a related list on the registered players table. Select the **Games Played** relationship you created.



9. Click **Save**.

®2016 ServiceNow, Inc. All rights reserved.

Lab Success Verification

1. You should be able to browse to the newly created menu for GameTimeNOW and open each module. The modules should be laid out as you designed and the related list for the Game and Registered players table is there. You may need to populate some demo data to get a good look at how the form is rendered.

©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

Lab 3.1
UI Policy

This lab explains how to use a UI policy to enforce the mandatory fields necessary for the tables. You could have easily have accomplished this in the Form Designer, but in our efforts to give you more knowledge of the platform we do this a different way.

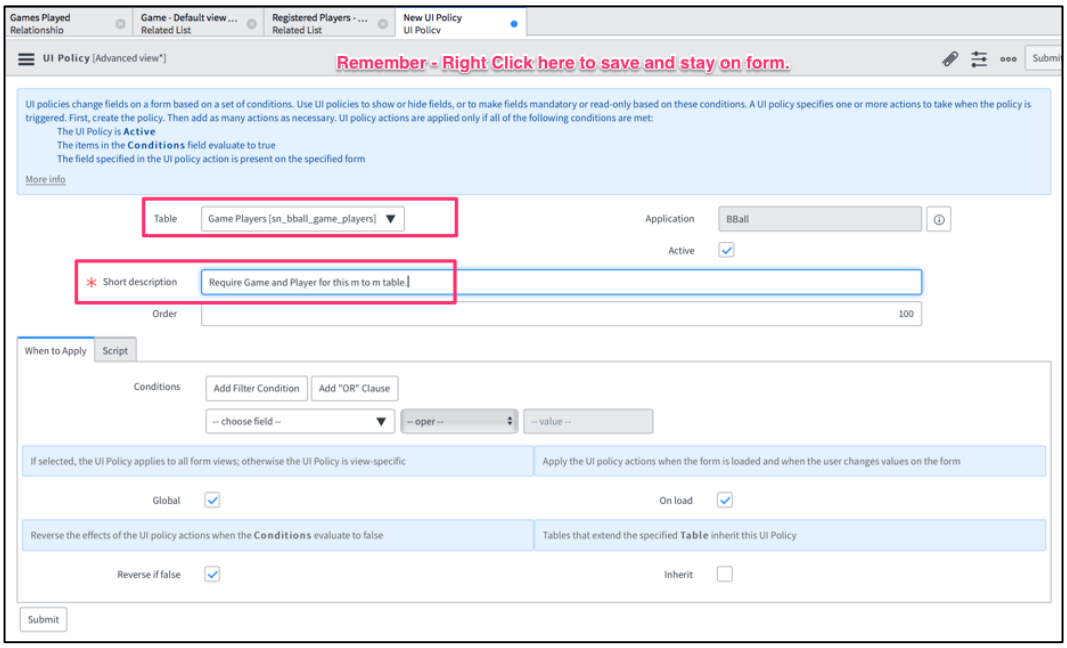
UI policies are useful for controlling the behavior of a form based on a specific condition. Another example – you may have a field you want visible only if it pertains to the state of another field. This keeps the form complexity low and prevents the form showing more than is necessary.

Create Mandatory Field on the Games Players and Games Table

1. Create a new UI Policy application file. This UI Policy enforces the mandatory fields you need on the Game Players table.

   ***Note** – Close the related list tab before creating a new application file.

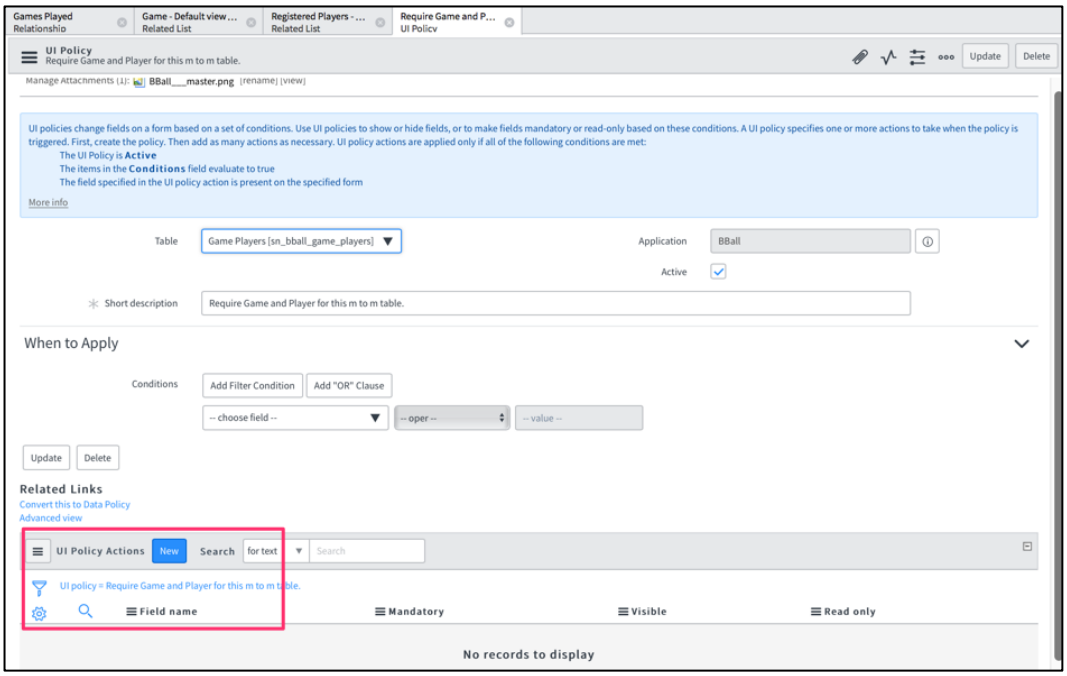
2. Select the **Game Players** table and give a short description.



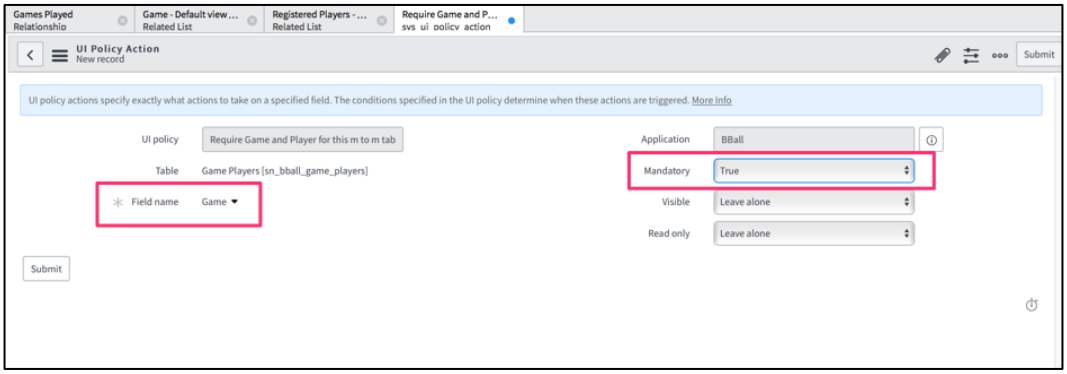
®2016 ServiceNow, Inc. All rights reserved.

3. Right-click **Save**.

4. Notice your form changed after the save. Now you have the option to create a few policy rules. Create the UI Policy Actions by clicking **New**.



5. Select the field for **Game** and set the **Mandatory** flag.



6. Click **Submit**.

7. Repeat **step 5** for the Player reference field as well.

©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

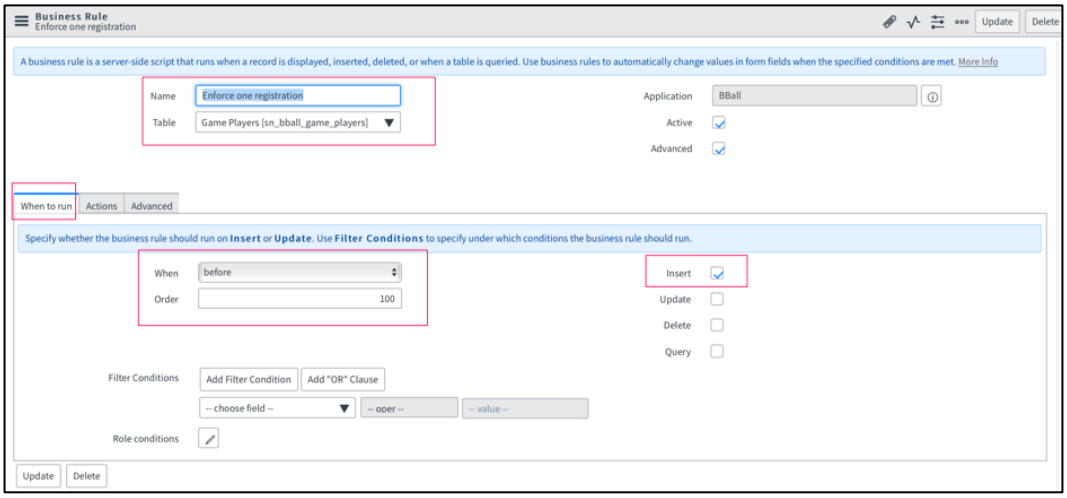
Lab 4.1 Server-Side Scripting – BR & Processor

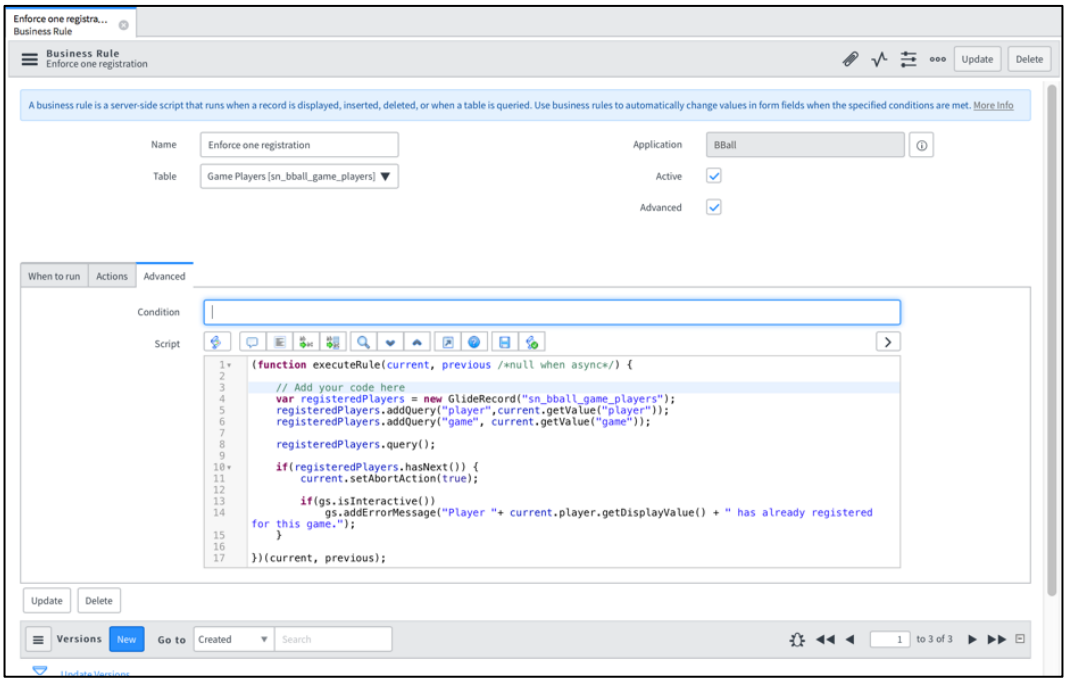
There are so many places in the platform you can script to get the behavior you want – business rules, script actions, inbound email actions, scripted REST and SOAP web services and processors, etc. This scripting engine is what gives the ServiceNow Platform the flexibility to build pretty much whatever you want. We do not go into when, why and where is the best time to use these scriptable components of ServiceNow, but any power user of ServiceNow will take the time to try as many as they can.

This lab explains how to build a processor. You want the ability to update a glide record and redirect a user based on a link they click in a notification email. Processors are a good candidate to accomplish this use case.

®2016 ServiceNow, Inc. All rights reserved.

1. Create a **business rule application** file. This business rule prevents players from registering more than once for a particular game.





Note: If you chose different names for your tables, you need to modify the code.

©2016 ServiceNow, Inc. All rights reserved.

```
(function executeRule(current, previous /*null when async*/) {

      // Add your code here
      var registeredPlayers = new GlideRecord("sn_bball_game_players");
      registeredPlayers.addQuery("player",current.getValue("player"));
      registeredPlayers.addQuery("game", current.getValue("game"));

      registeredPlayers.query();

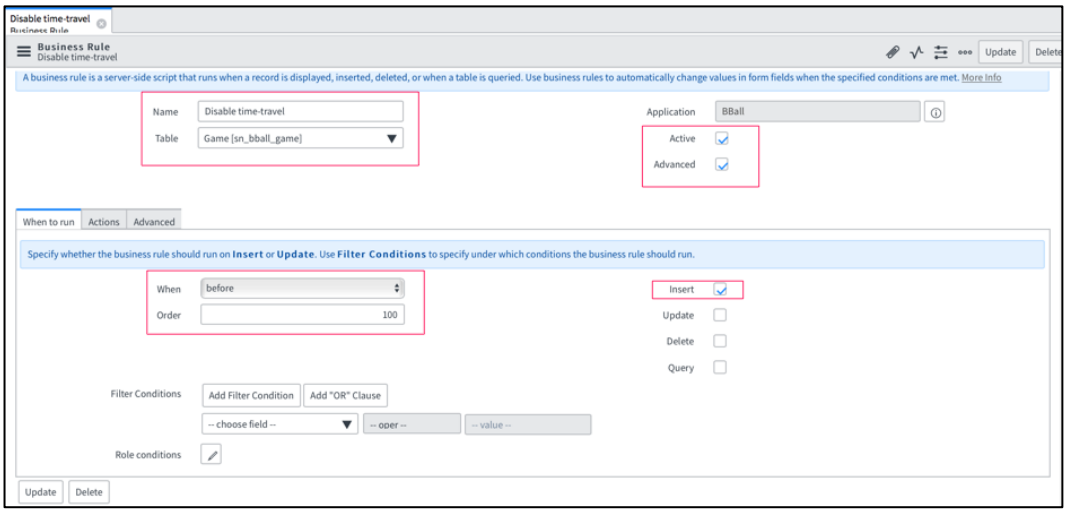
      if(registeredPlayers.hasNext()) {
            current.setAbortAction(true);

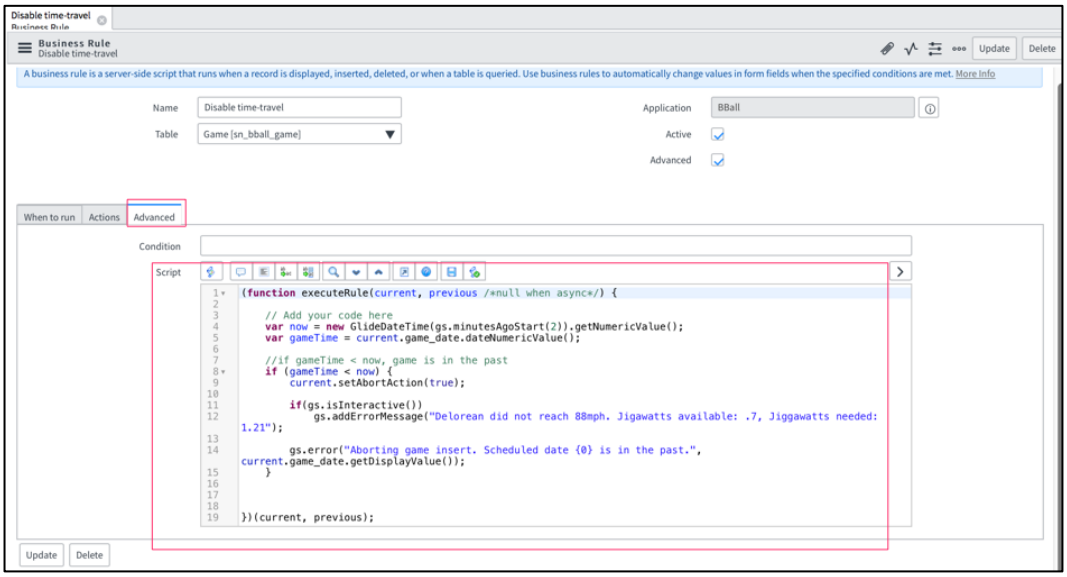
            if(gs.isInteractive())
                  gs.addErrorMessage("Player "+ current.player.getDisplayValue()
+ " has already registered for this game.");
      }

})(current, previous);
```

©2016 ServiceNow, Inc. All rights reserved.

2. Create an **application** file for your second business rule. This business rule prevents games from getting created in the past.





©2016 ServiceNow, Inc. All rights reserved.

```
(function executeRule(current, previous /*null when async*/) {

	// Add your code here
	var now = new GlideDateTime(gs.minutesAgoStart(2)).getNumericValue();
	var gameTime = current.game_date.dateNumericValue();

	//if gameTime < now, game is in the past
	if (gameTime < now) {
		current.setAbortAction(true);

		if(gs.isInteractive())
			gs.addErrorMessage("Delorean did not reach 88mph. Jigawatts
available: .7, Jiggawatts needed: 1.21");

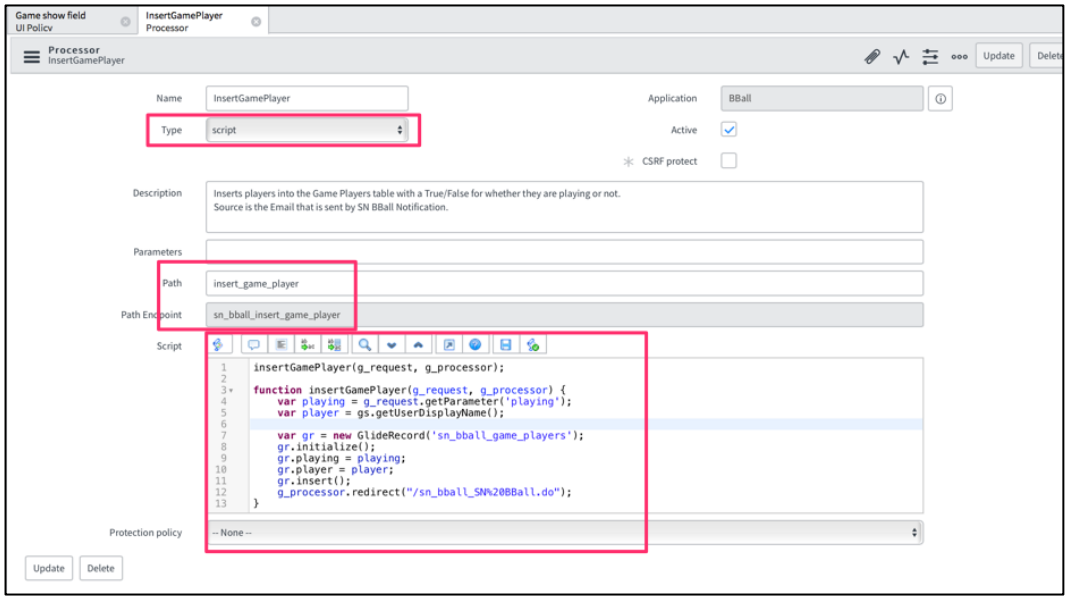
		gs.error("Aborting game insert. Scheduled date {0} is in the past.",
current.game_date.getDisplayValue());
	}



})(current, previous);
```

©2016 ServiceNow, Inc. All rights reserved.

3. Create a processor application file. Set the **type** to **script** and the **path** to **insert_game_player**. Use the code provided for the script field. You may need to change the GlideRecord name to the correct table name you created.




```
insertGamePlayer(g_request, g_processor);

function insertGamePlayer(g_request, g_processor) {
      var playing = g_request.getParameter('playing');
      var player = gs.getUserDisplayName();

      var gr = new GlideRecord('sn_bball_game_players');
      gr.initialize();
      gr.playing = playing;
      gr.player = player;
      gr.insert();
      //g_processor.redirect("/sn_bball_SN%20BBall.do");
      g_processor.redirect("/aPageThatExist.do");
}
```

4. Click **Update**.

©2016 ServiceNow, Inc. All rights reserved.

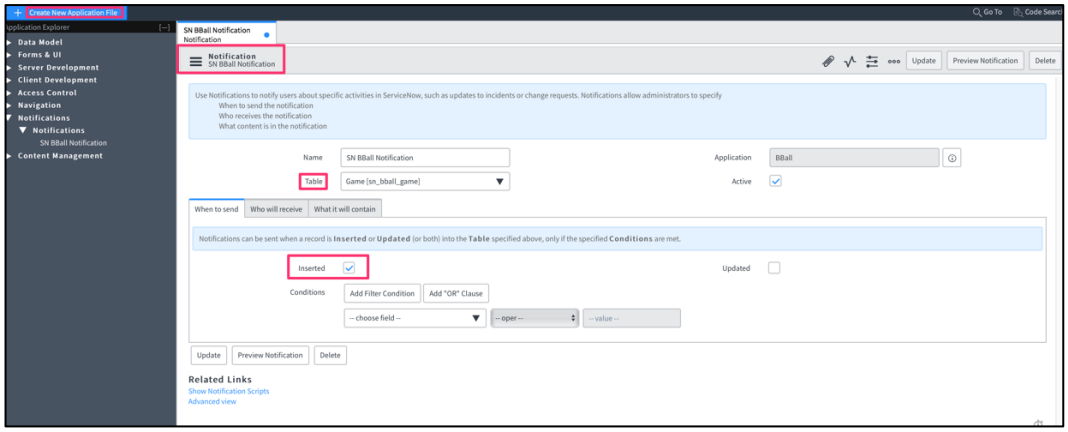
Lab Goal

This lab explains how to use email notifications to communicate a game invite.

Lab 4.2
Notifications

More Formatting Instructions

1. Create a new Notification Application File. Fill out the form as indicated below. Remember there are three tabs on the Notification form that need to be filled out. Use the screen shots below to help you out.



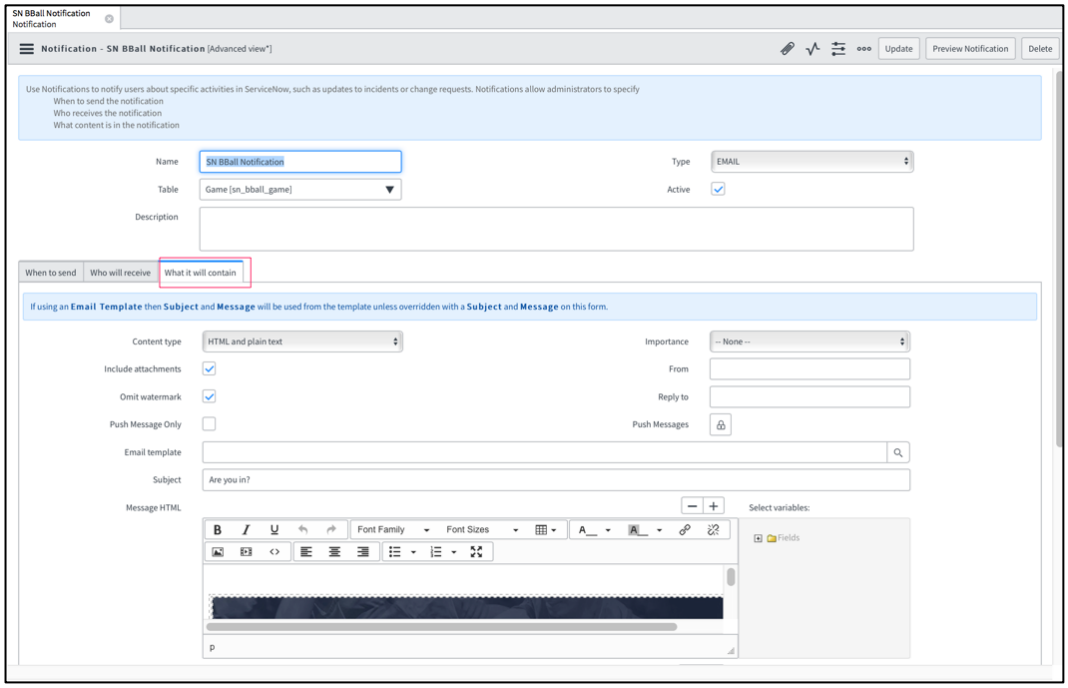
2. In the Message HTML, you want to insert/edit links on the Yes and No Text. The link points to the URL of the processor created above.

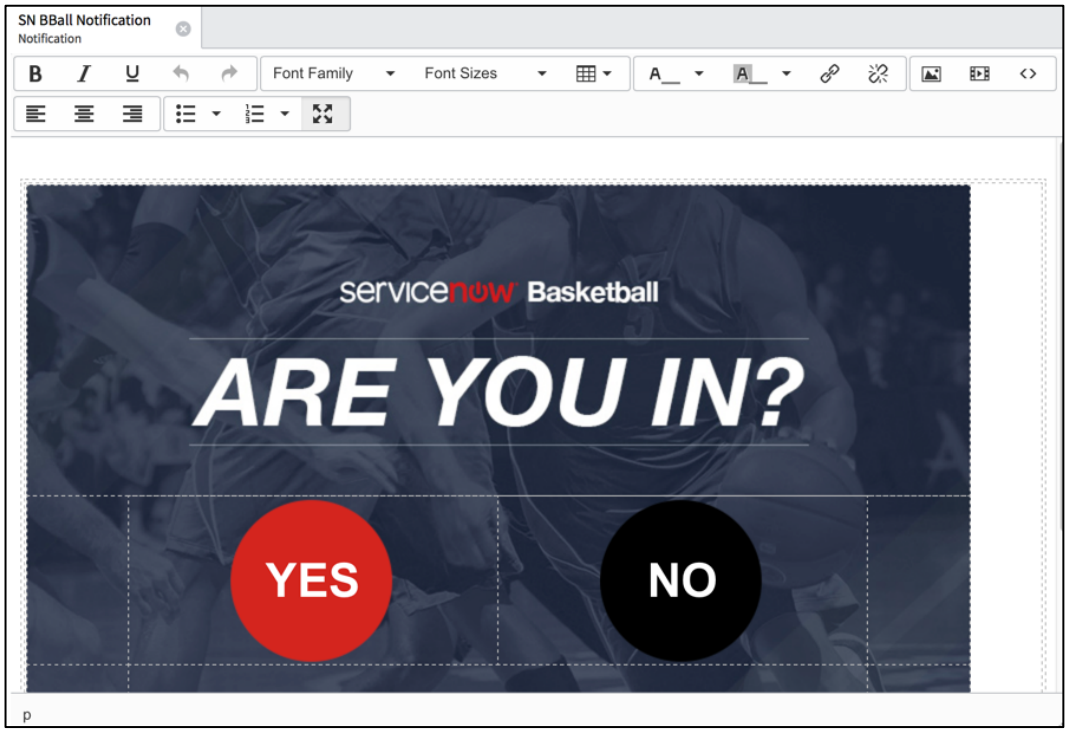
   /sn_bball_insert_game_player.do?playing=true

   /sn_bball_insert_game_player.do?playing=false

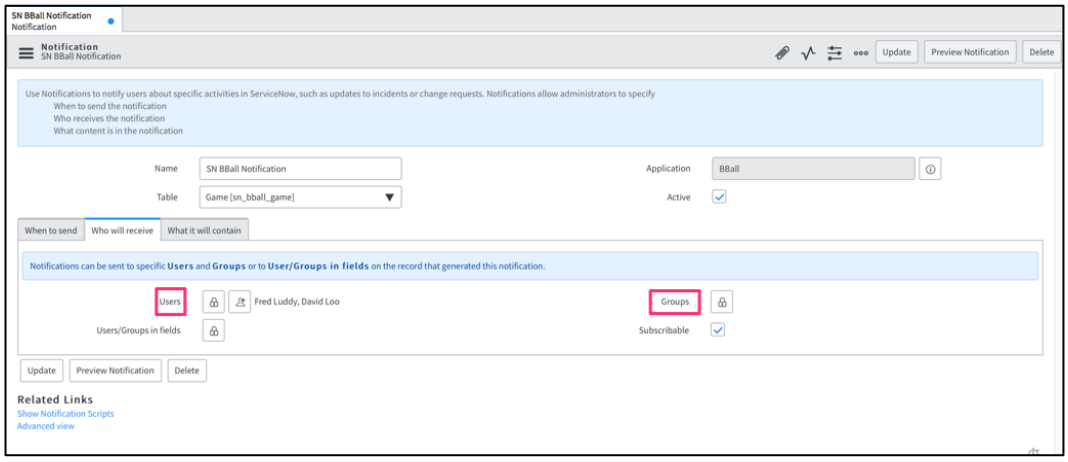
®2016 ServiceNow, Inc. All rights reserved.

3. Here is the HTML - copy provided. We hate typing.





®2016 ServiceNow, Inc. All rights reserved.




```
<p> </p>
<table width="100%">
<tbody>
<tr>
<td>
<table style="width: 800px; height: 575px; background:
url('http://s30.postimg.org/ghj65vrep/sn_basketball_bg.jpg'); border-
collapse: collapse;" width="800">
<tbody>
<tr>
<td colspan="4" height="260"> </td>
</tr>
<tr>
<td> </td>
<td id="button-wrapper" style="background: url('/bball_email_in.png')
no-repeat center; text-align: center;" width="310" height="140"><a
style="text-decoration: none; line-height: 140px; margin: 0; padding:
0; font-family: Arial, Helvetica; font-size: 40px; font-weight: bold;
color: #fff;"
href="/sn_bball_insert_game_player.do?playing=true">YES</a></td>
<td style="background: url('/bball_email_out.png') no-repeat center;
text-align: center;" width="310" height="140"><a style="text-
decoration: none; line-height: 140px; margin: 0; padding: 0; font-
family: Arial, Helvetica; font-size: 40px; font-weight: bold; color:
#fff;" href="/sn_bball_insert_game_player.do?playing=false">NO</a></td>
<td> </td>
</tr>
<tr>
<td> </td>
<td style="height: 100px; text-align: center; color: white;"
colspan="2">Click <a style="color: #d11e26;"
href="/sn_bball_SN%20BBall.do">HERE</a> to see who's playing.</td>
<td> </td>
</tr>
<tr>
```

®2016 ServiceNow, Inc. All rights reserved.

```
<td style="background-color: black; text-align: center; height: 75px;"
colspan="4">
<p style="margin: 0; padding: 0; font-family: Arial, Helvetica; font-
size: 12px; color: #fff; padding-top: 16px; text-align:
center;">ServiceNow Basketball</p>
</td>
</tr>
</tbody>
</table>
</td>
</tr>
</tbody>
</table>
<p> </p>
```

©2016 ServiceNow, Inc. All rights reserved.

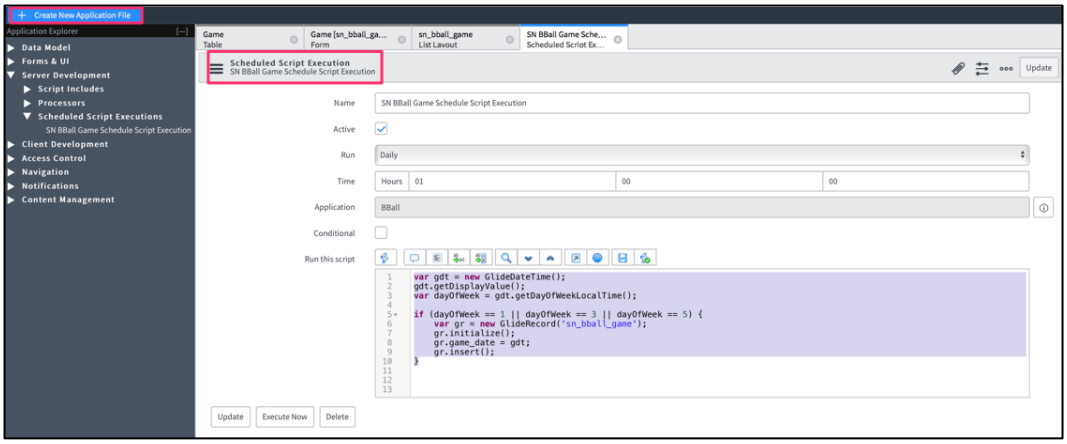
Lab Goal

This lab explains how the scheduler works and how to create a scheduled script to run daily, creating games for the day based on the day of the week you run the game.

Lab 4.3 Scheduled Script Executions

More Formatting Instructions

1. Create a new **Application** file for a Scheduled Script Execution and fill in the details as shown. Give it any name, but it runs daily at 1 a.m. to create the games for the day:




```
var gdt = new GlideDateTime();
gdt.getDisplayValue();
var dayOfWeek = gdt.getDayOfWeekLocalTime();

if (dayOfWeek == 1 || dayOfWeek == 3 || dayOfWeek == 5) {
  var gr = new GlideRecord('sn_bball_game');
  gr.initialize();
  gr.game_date = gdt;
  gr.insert();
}
```

©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

Lab 5.1 Integrations (OPTIONAL LAB)

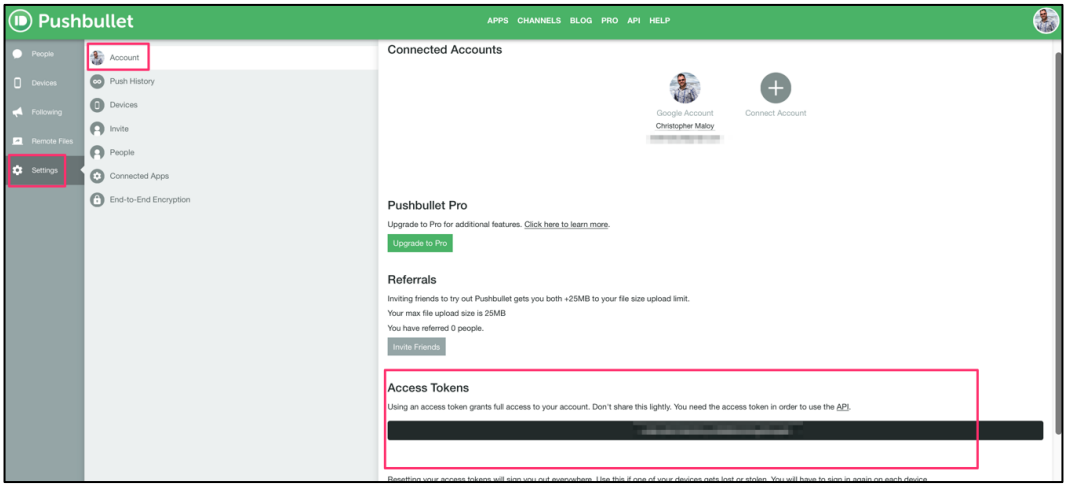
This lab explains how to do a Pushbullet Integration from the ServiceNow instance to get targeted reminders from the ServiceNow instance to personal devices.

No need to spend too much time during class reading and understanding the Pushbullet API documentation. We walk you through this section and you can read up on this cool product later.

https://www.pushbullet.com

https://docs.pushbullet.com

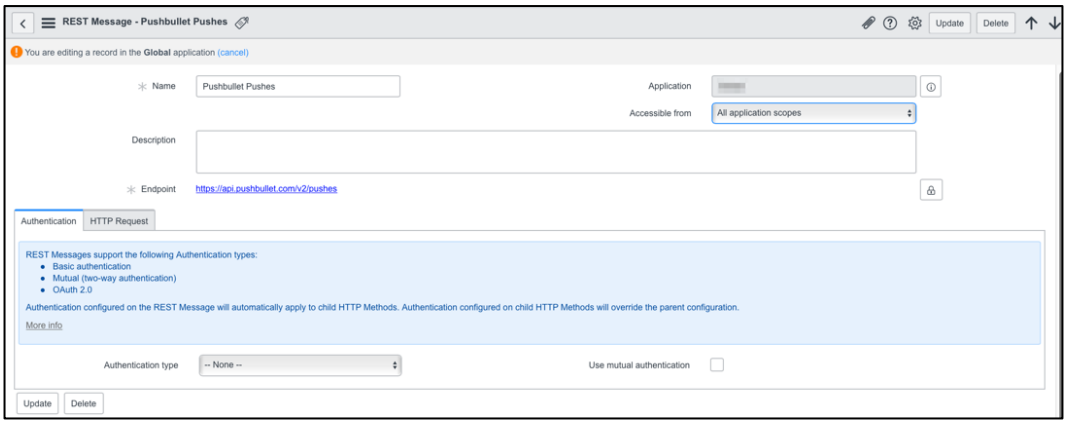
1. Create a **Pushbullet** account if you do not have one.
2. After the account is created, you need to have Pushbullet generate an access token. You can also do this with Oauth, but for sake of simplicity use your authorization token.
3. After generating the authorization token (see image of where in Pushbullet to do this) copy and paste it. You need this when creating the REST message.
4. Navigate to the **Settings > Account** section and click the **Create Access Token** button.



©2016 ServiceNow, Inc. All rights reserved.

5. Return to the ServiceNow instance to create a REST application file. Name this REST Message Pushbullet Pushes.

   Make sure the Application is in your GameTimeNow application and make sure this REST message is available to all application scopes.

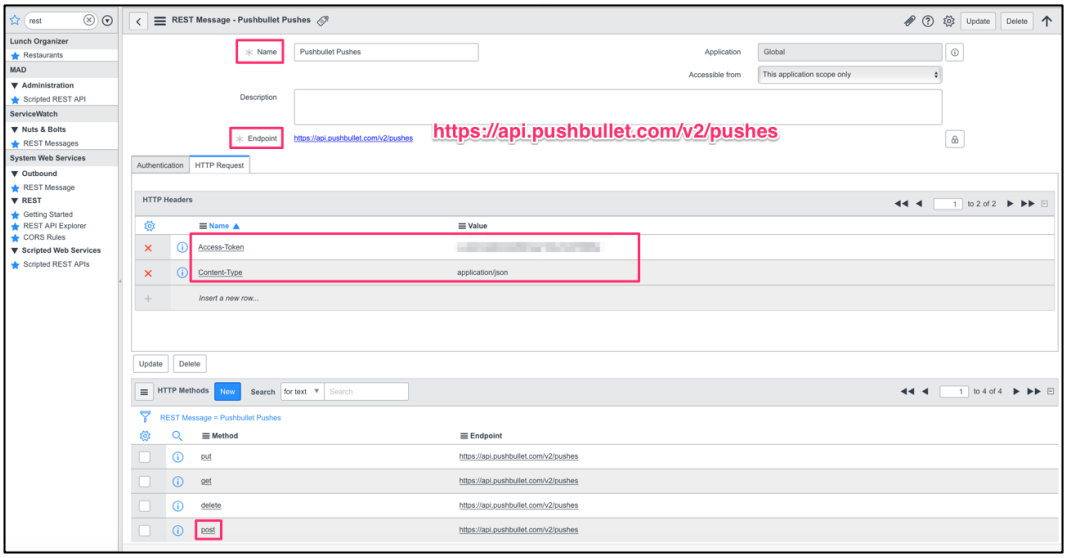


6. Right-click on the header and **Save** to stay on this form.

7. After you right-click/save, the REST methods are automatically generated. Note the PUT, GET, DELETE, and POST.

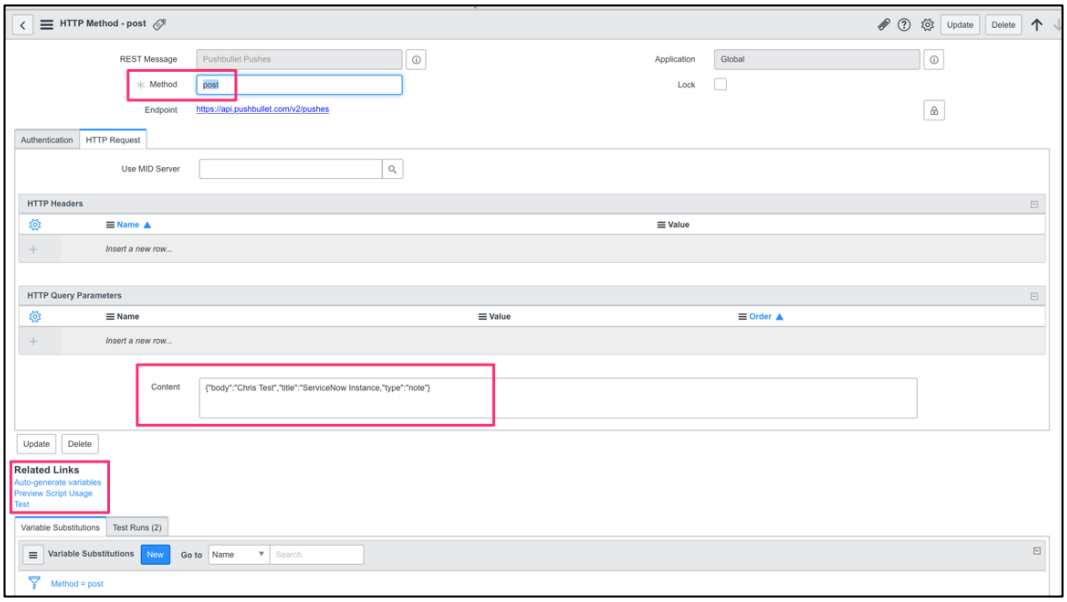
8. Fill out the HTTP Request fields first. You need two fields in the HTTP Request:

   Access-Token. The value is the Pushbullet access token you generated on the Pushbullet site.
   Content-Type. The value is application/json.



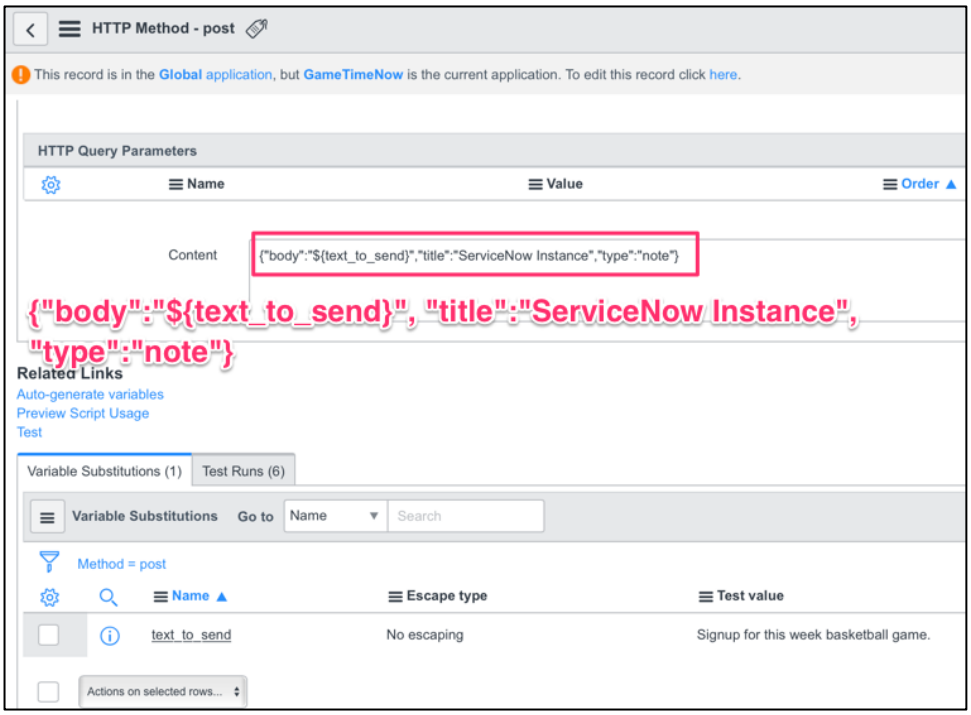
©2016 ServiceNow, Inc. All rights reserved.

9. Right click and **Save** again (because we are OCD and need to do this again). Click into the POST method. A form similar to this one appears. Fill out the Content section of this form on the HTTP Request tab (of the POST method).



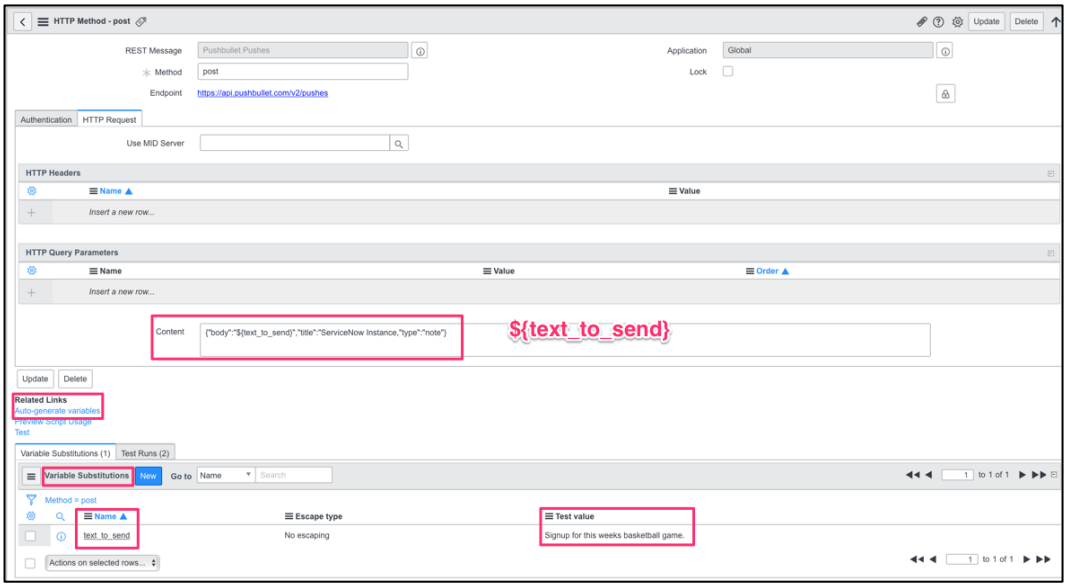
©2016 ServiceNow, Inc. All rights reserved.

10. Take a closer look. Make the Content match what is shown in the JSON format. This is all dictated by the service you are talking to (in this case Pushbullet) and is explained in their documentation.

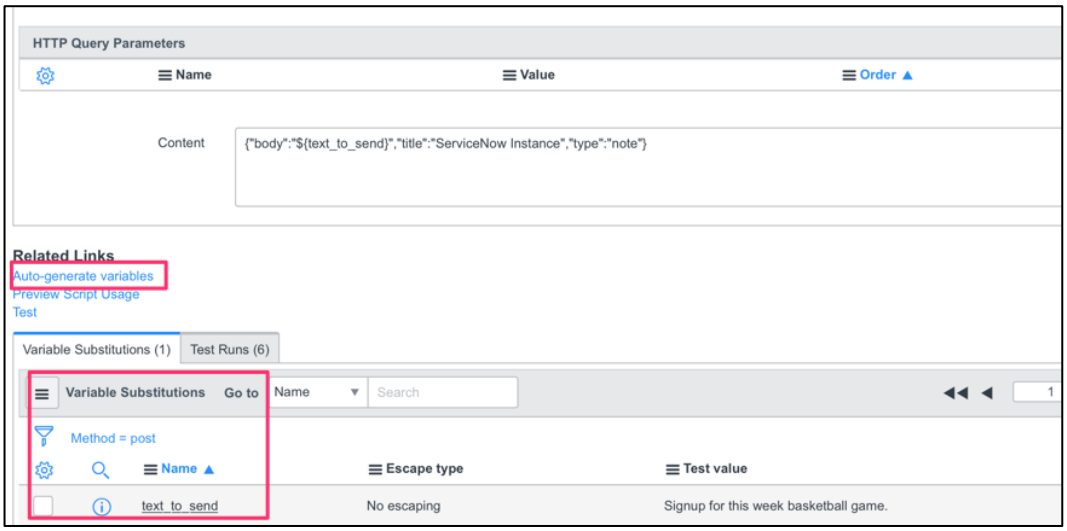


®2016 ServiceNow, Inc. All rights reserved.

11. Notice that the **text_to_send** in the content field has a weird format. This format lets ServiceNow know that we will use some dynamic content supplied at runtime. It is a placeholder for data to come.



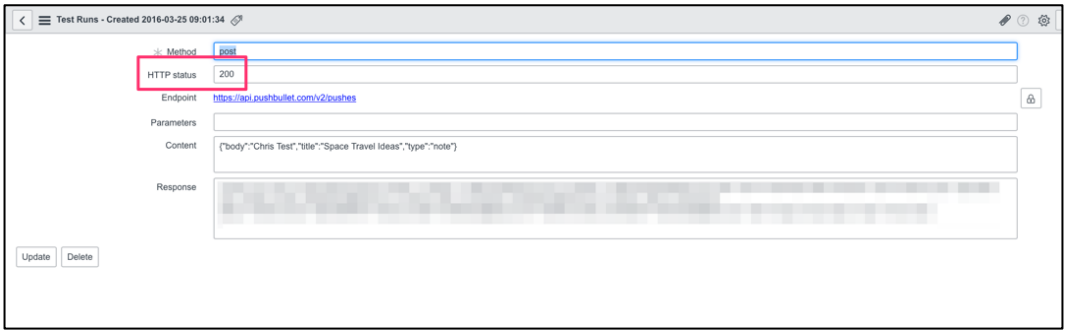
12. Test this API to see if it is working. There are some related links on this page. If you click the Auto-generate variables link, it creates some test variables you can use, based on the special syntax used above. In this case it generates a variable for **text_to_send**. After it generates the variable, give it some text value to test. Put anything you want there.



©2016 ServiceNow, Inc. All rights reserved.

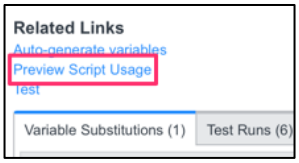
13. Click the test related link. You should get back a **200** response (and all your devices are notified– if you have any devices registered with Pushbullet).





®2016 ServiceNow, Inc. All rights reserved.

14. Finally, you are ready to incorporate this integration into the rest of our GameTimeNow application. There is a preview script usage related link that auto generates the JavaScript you can use in a business rule or any place you write script in ServiceNow.





15. Copy the contents of this script to use it to notify Players when a game is created, when a game is about to start, or to mention winners, or special announcements. You can use this on top of the standard email notifications you get with ServiceNow.

    One thing to note here, of course, is that we set this integration up with an access token–so it only sends notifications to the account tied to that token. If you want this to be more dynamic, implement it in a user-based way with **oauth** or dynamic access tokens for users.

©2016 ServiceNow, Inc. All rights reserved.