Lab Guide

Easy No Code Apps: Look Ma, No Code

Arnez Nisperos & Chris Fulton

Default Login / Password: Knowledge17

admin / Knowledge17

itil / Knowledge17

employee / Knowledge17

Lab Sponsored by Amazon Web Services



This
Page
Intentionally
Left
Blank

Lab Goal

Lab 1.1 Setup and Create App with Data Model

On your mark, 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:

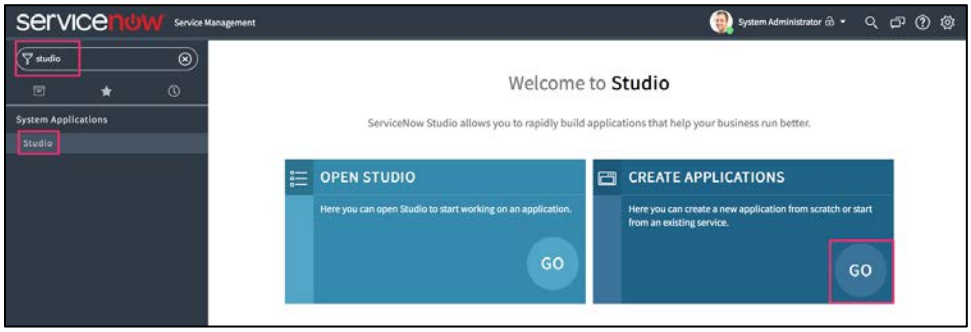
- Have proper access to your assigned instance
- Can find and open Studio
- Create an empty application from Studio

Log in to Your Provided Instance

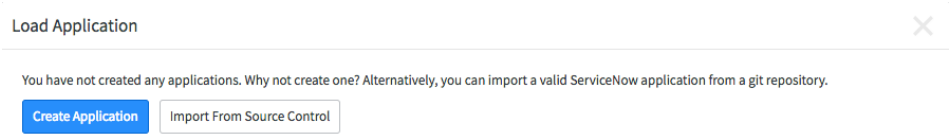
1. Navigate to the unique instance URL provided to you.
2. Log on with provided credentials.

Open Studio and Create an Application

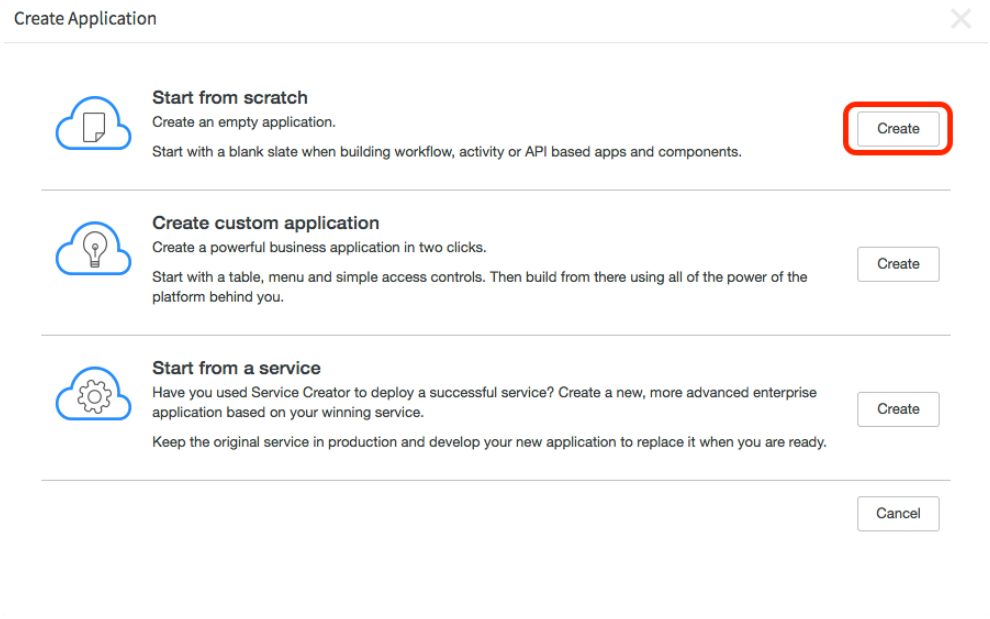
1. On your instance, locate and find the Studio module and open it.



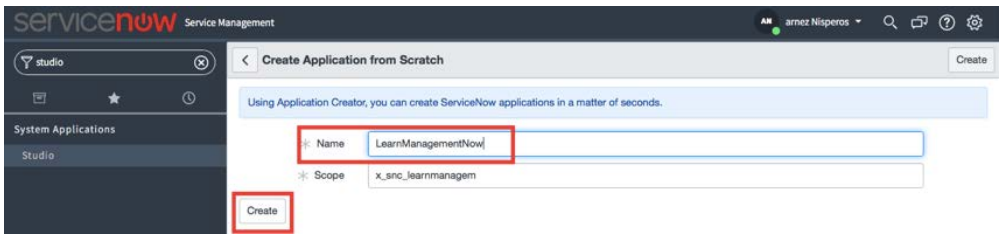
2. Select Create Application from the popup window.



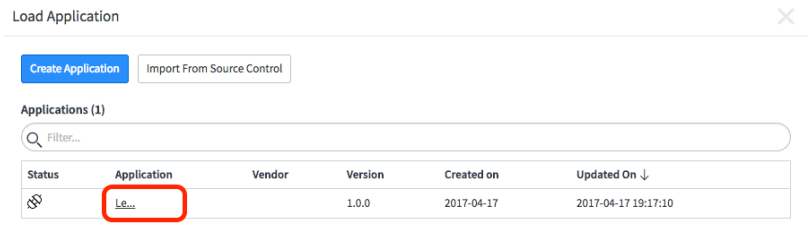
3. Create an application from scratch.



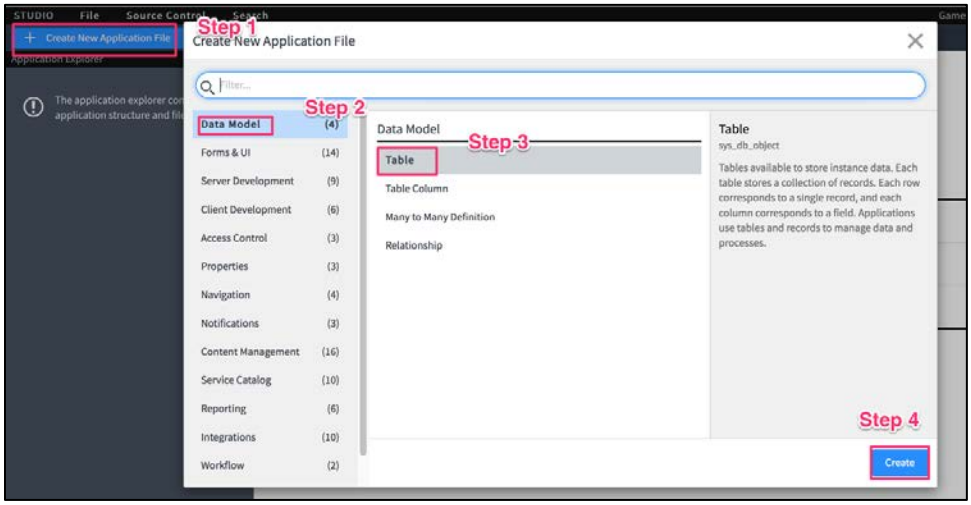
4. Name the application **LearningManagementNow** and click the **Create** button. Click **OK** when the confirm application dialog box appears.



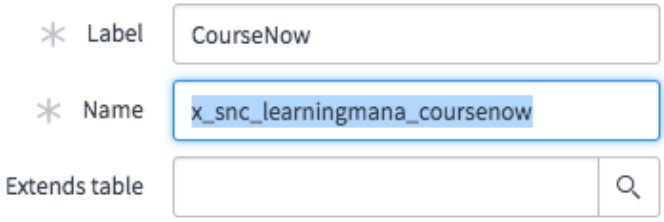
5. Choose **Back to list** and then select the newly created application from the list.



6. Click **Create Application File** (step 1). Data model and Table should be highlighted by default. If not, select them and then click on **Create.**



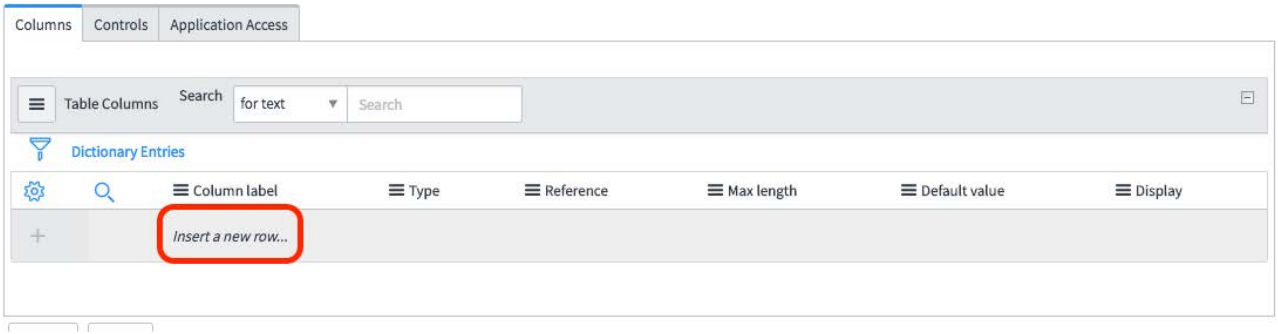
7. Create the **CourseNow** Table first, then create two columns and a menu entry.



Label: **CourseNow**
Name: *automatically populated for you.*

New menu name: **CourseNow**

Toward the middle of the screen locate the three tabs named Column, Controls and Application Access. Ensure that the Column tab is selected. To add a new row, double click on *Insert a new row…* and add the following rows.



Column Label: **Course Name**
Type: **String**
Max Length: **30**
Display: **True**

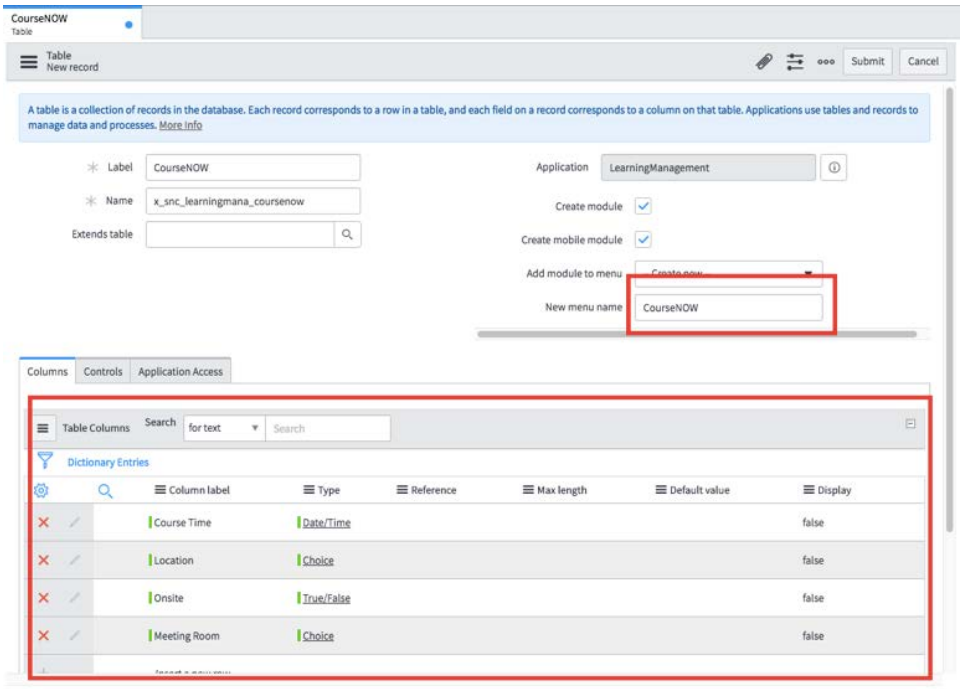
Column label: **Course Date**
Type: **Date/Time**

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

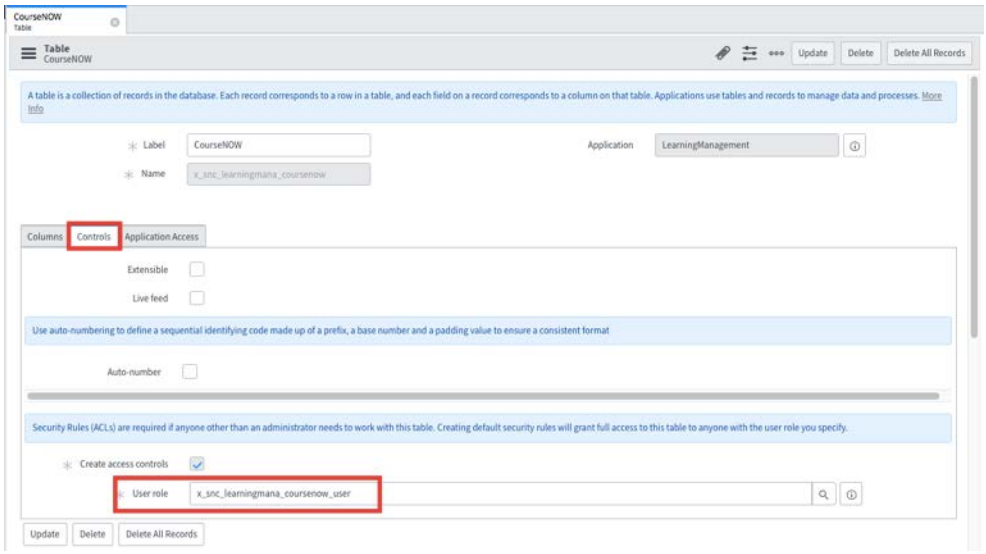
Column label: **Onsite**
Type: **True/False**

Column label: **Meeting Room**
Type: **Choice**

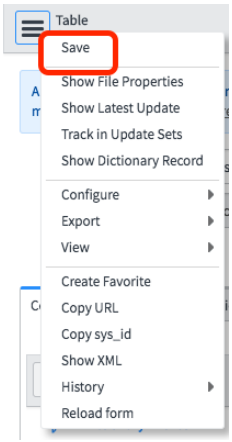
Column label: **Instructor**
Type: **Reference**
Reference: **Users (sys_user)**



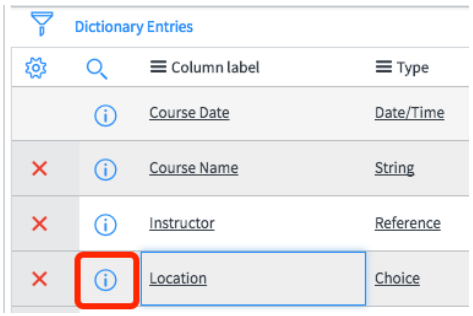
8. Select the Controls tab to see Access Controls which are created by default.



9. Let's save our work before we continue to make additional configuration settings. Save the record by selecting the menu and then selecting save.

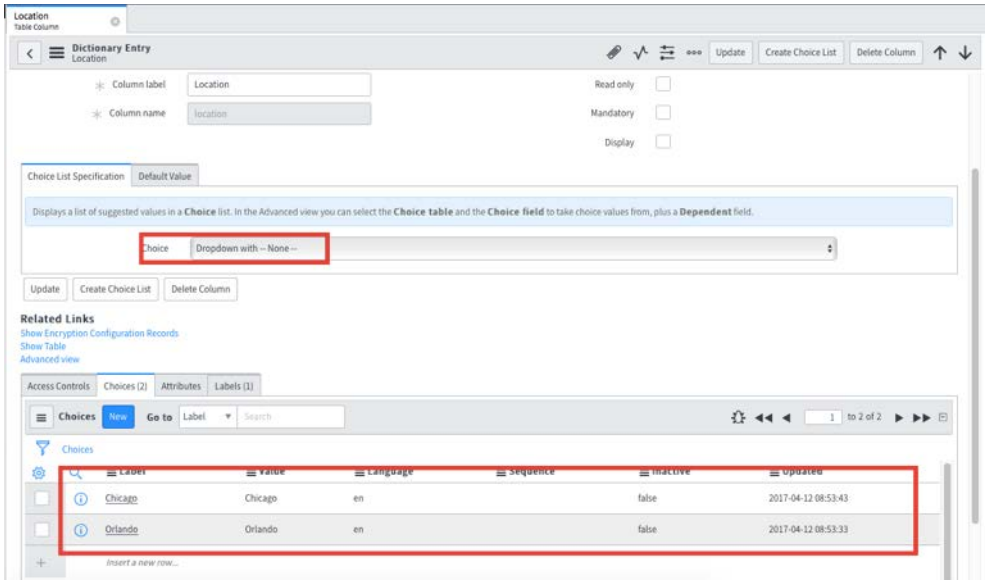


10. Now, we will add some choices to the Location column. This will give the user options to select from a dropdown list. Open the **Location** column you just created.



11. Create two new choices:

    Column label and Value: **Orlando**
    Column label and Value: **Chicago**
    Choice: **Dropdown with – None** –



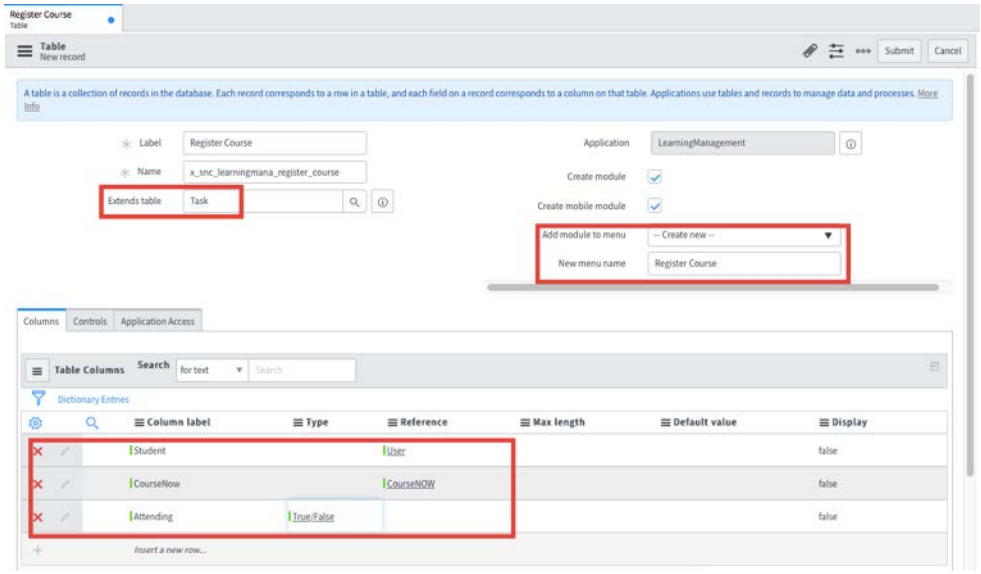
12. Select **Update** to save your work.

13. Create the **Register Course** Table and add the table as a module to the application. This will be used for students to register for a course.

14. Column label: **Student**
    Type: **Reference**
    Reference: **Users (sys_user)**
    Display: **True**

    Column label: **CourseNow**
    Type: **Reference**
    Reference: **CourseNow**

    Column label: **Attending**
    Type: **True/False**



Create the Equipment Request table in a similar manner.

15. Select **Submit** to save your work.

16. Create the **Equipment Request** Table and add the table as a module to the application. This will be used for instructors to have equipment for the course.

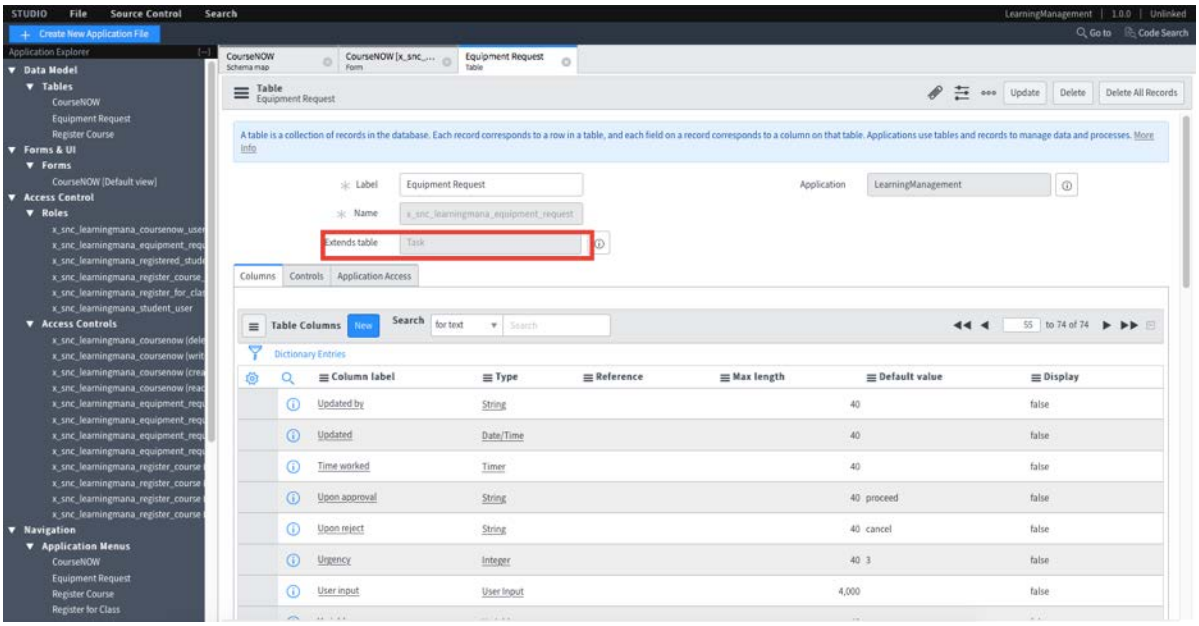
17. In **Extends table** field, enter **Task** and select **Task** from the auto-complete drop-down list. Extending an existing table is a powerful facility that allows easy reuse of existing functionality. In this case, Task provides the underlying functionality for an object to participate in a workflow. Since the Equipment table extends task, it can also participate in a workflow without any additional effort. You build the workflow in a later lab.

18. Column label: **Type**
    Type: **Reference**
    Reference: **Hardware Model (cmdb_hardware_product_model)**
    Display: **True**

    Column label: **Needed From**
    Type: **Date**

    Column label: **Needed To**
    Type: **Date**

    Column label: **CourseNow**
    Type: **Reference**
    Reference: **CourseNow**



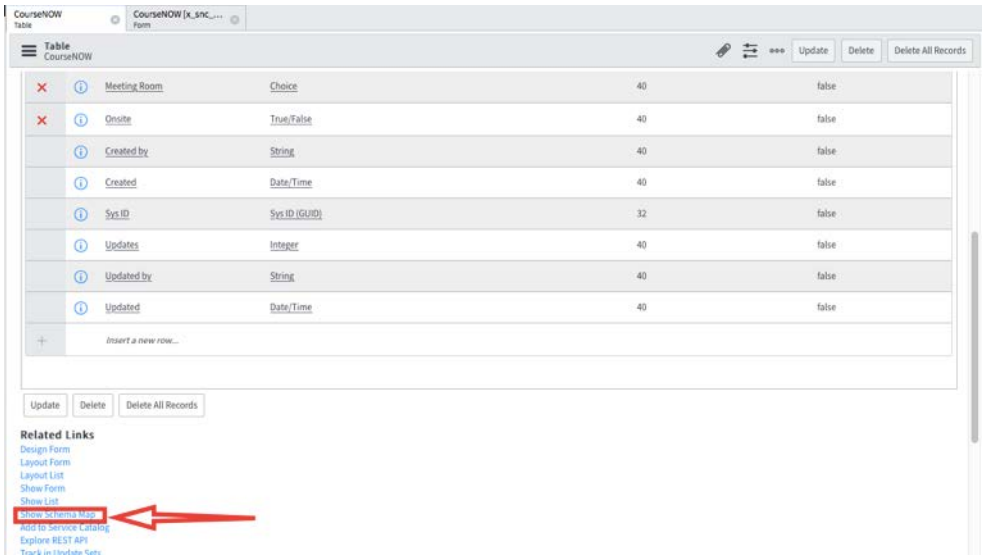
- **Note**: The **CourseNow** field is a **Reference** type field to the **CourseNow** table. A reference field creates a relationship between two tables. In this case, a particular piece of equipment is related to a particular course. It allows you to display information from another table and is like a foreign key in a relational database.

19. Click **Submit**. Now there are three tables in the application.

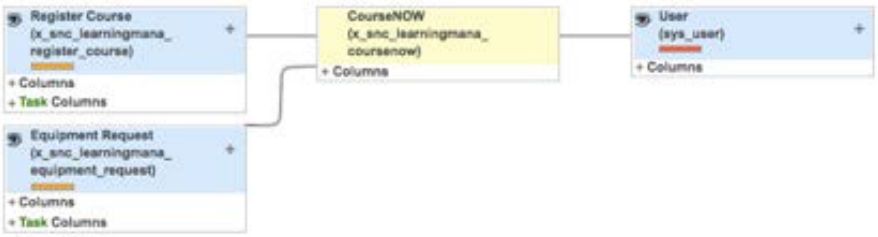
Verify Results

Get another perspective on data model using the schema map.

1. In the Related Links of the **CourseNow** table, click Show **Schema Map**.



This shows the schema map (Entity Relationship diagram) for the application.

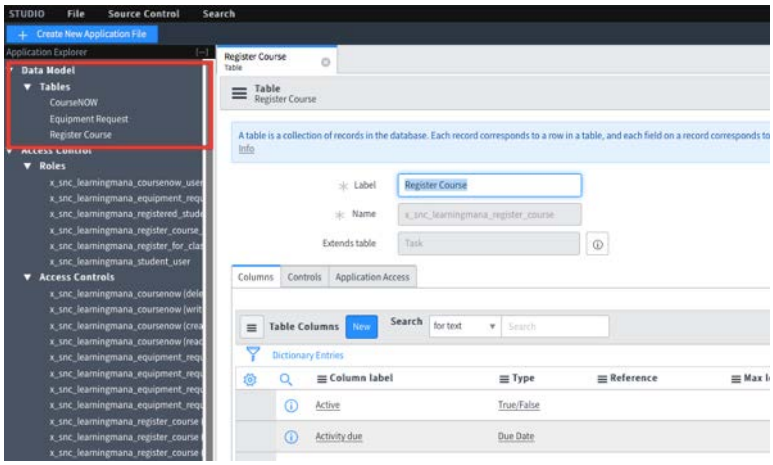


If you do not see the relationship as depicted above, it means that the reference field between the two tables is not properly set. Go back to table creation in the previous section.

- Use the schema map to drill down and look at the application's tables and fields. Notice that there are a few standard system fields added such as Sys ID and Updated. For the Equipment Request table, note the fields included from extending the Task table. The application now has two tables. In the next lab, you modify the user interface so that you can enter data into these tables.

Lab Success Verification

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



Lab Goal

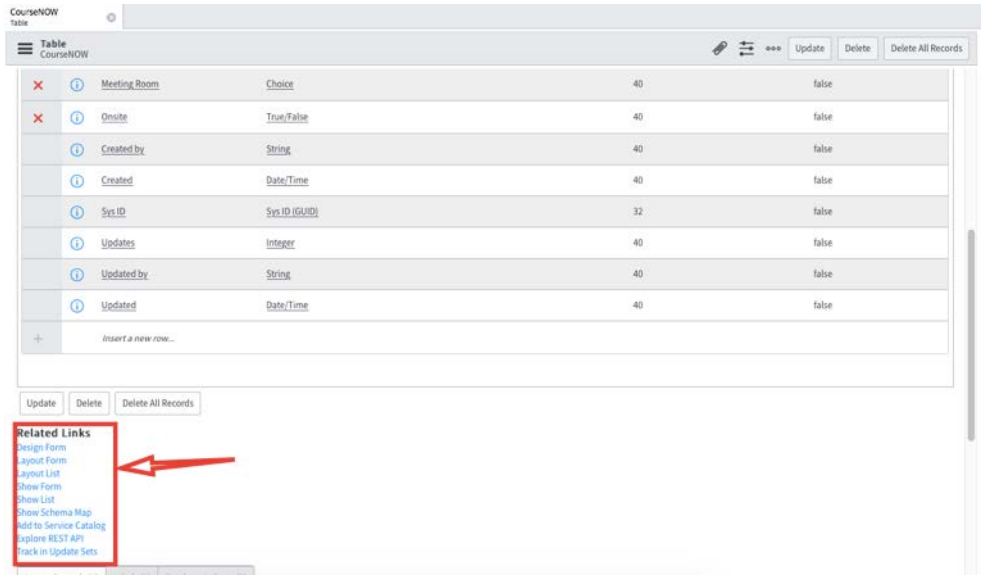
The tables created when setting up the data model dynamically generate a form, based on the columns set for it. These forms give users, who have access, a simple UI when interacting with the data model.

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

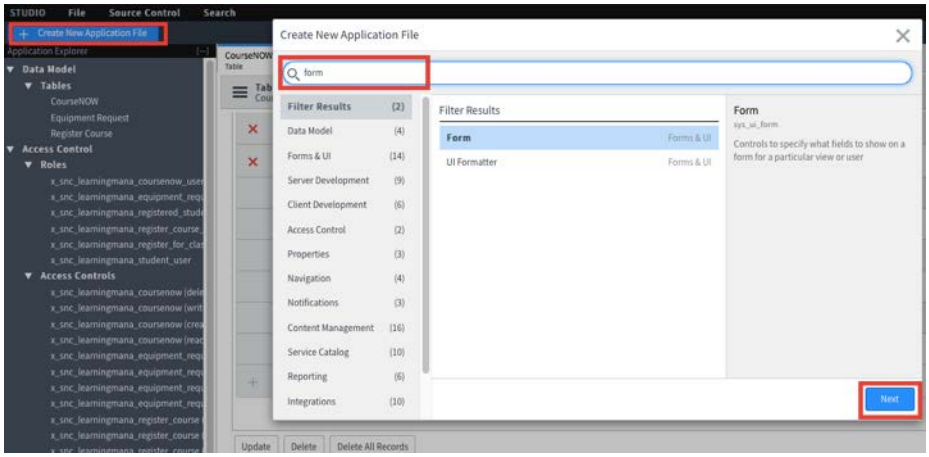
Lab 2.1 Form Designer

Create a Default View for the Records

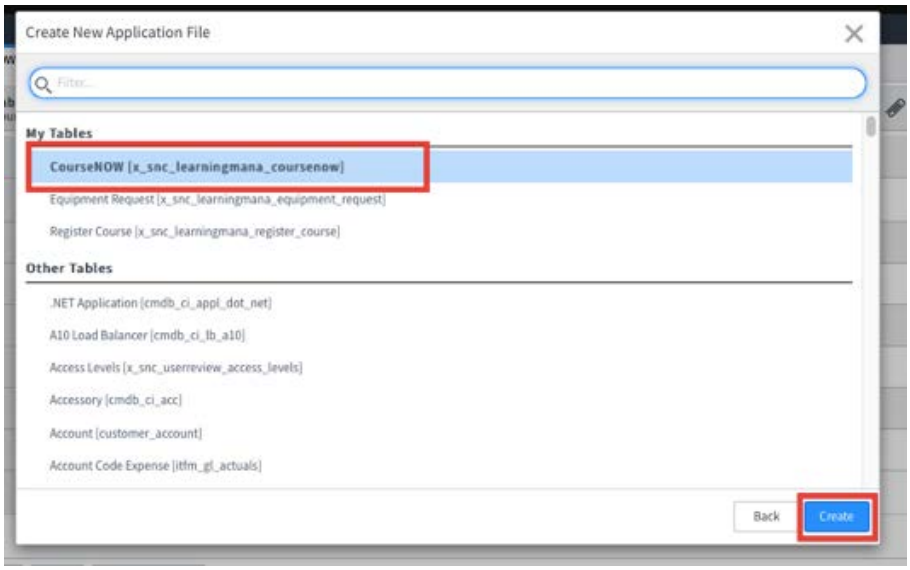
1. Each of the tables you created has related links that help design the default forms and list. We prefer **not** to use these related links as they steal focus away from the rest of Studio.



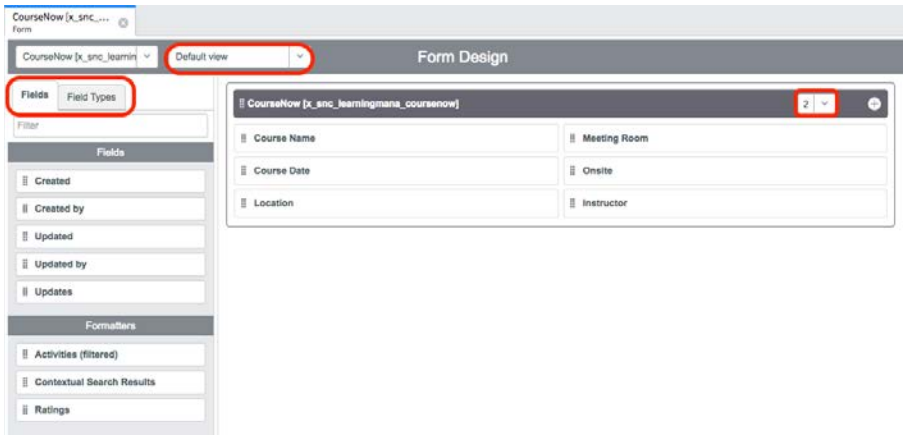
2. Click the **Create New Application File** button and search for **Form**.



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



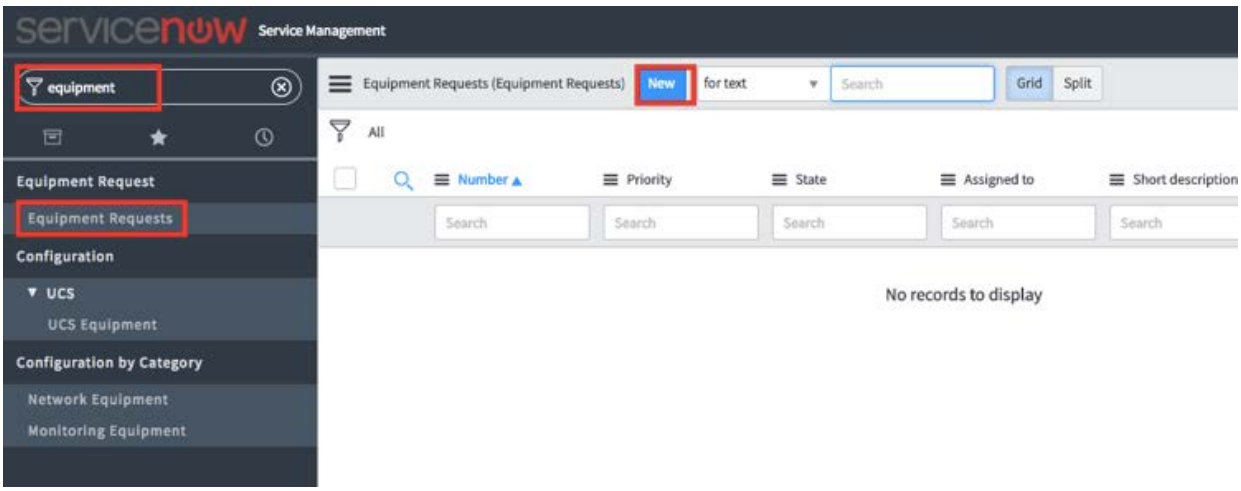
4. Notice the many things you can do in the Form Designer. You can change the data model by adding more fields (columns) to the table. You can update and change some basic properties of the fields you have defined (like making a field mandatory). You can create different views, and change the layout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



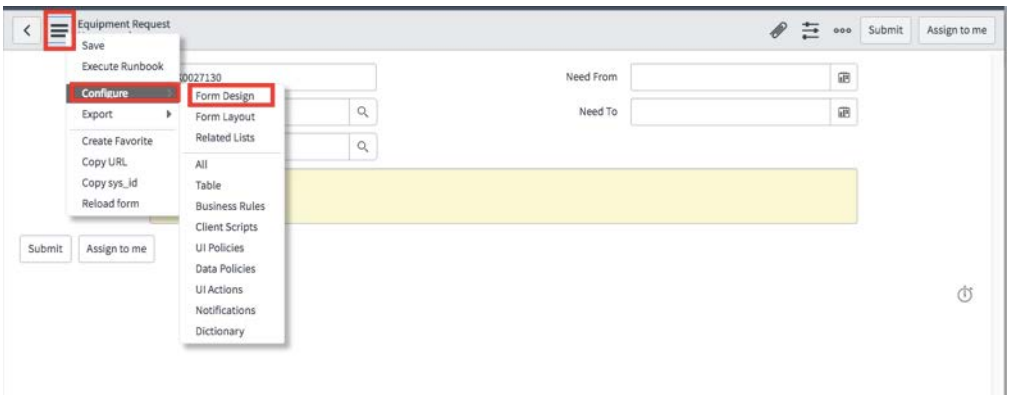
Modify the Equipment Request Form

Now, configure the Equipment Request form.

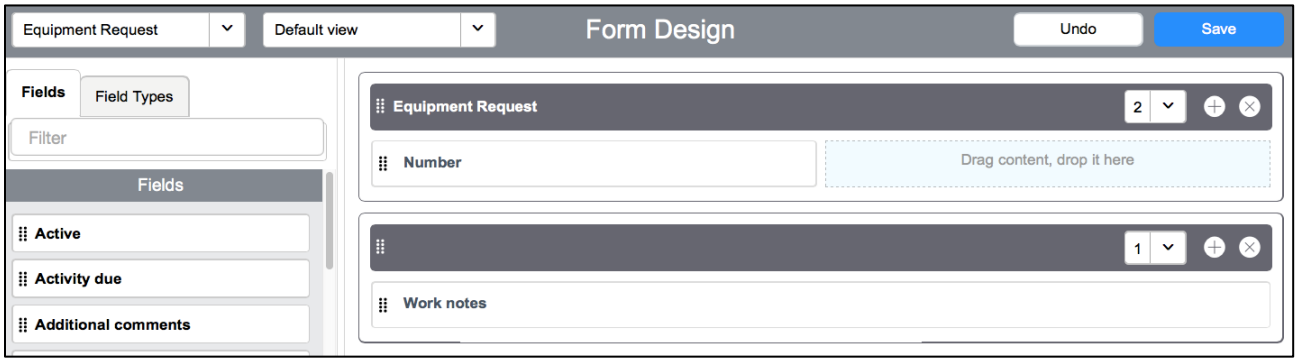
1. Go back to ServiceNow in the other browser tab (not the studio). Click on **Equipment Request** from the left menu. In the main panel, click on **New**.
   ** If you don't see the menu entry, reload the page in your browser.*



2. Click on the **three bars** at the top left of the main panel. Click on **Configure**. Click on **Form Design.**



3. Remove all the default fields in the form except **Number** and **Work notes** by selecting each field and then clicking the **X**.

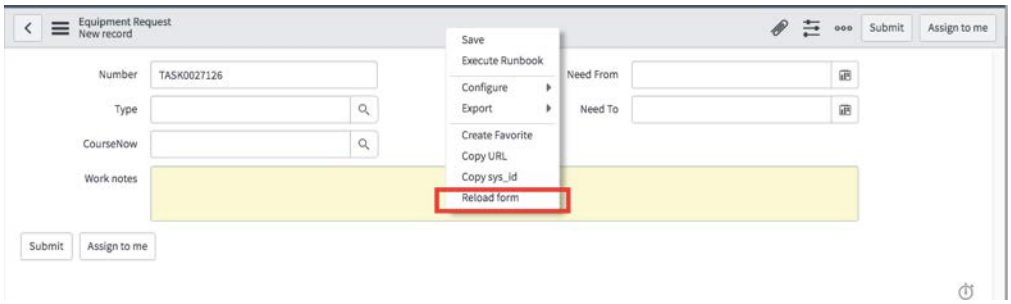


4. Configure the form as follows by dragging the field from **Fields** list on the left and dropping them on the **Equipment Request** form, as specified below. If there is no second section. Create one by clicking on the **+** in the upper-right of the first section.



5. Click **Save** in the upper-right.

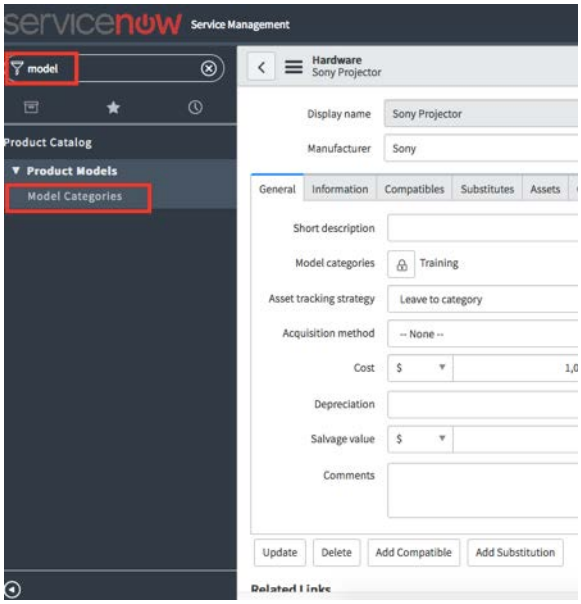
6. Close the Form Design tab and return to the previous tab. Right-click on the header of the Equipment Request. Click on **Reload Form** from the menu options. This will refresh the screen with your changes you made to the form.



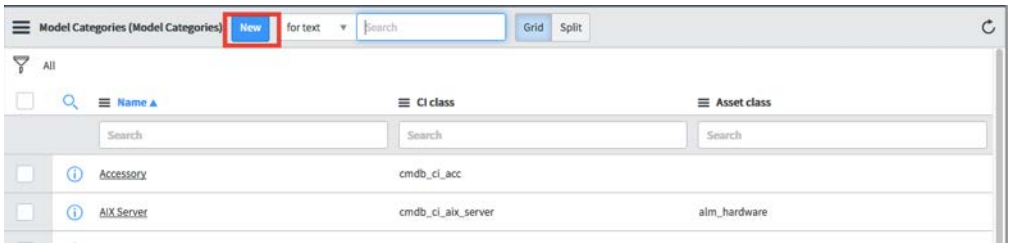
Create Hardware Models

You will be creating hardware model specific for training.

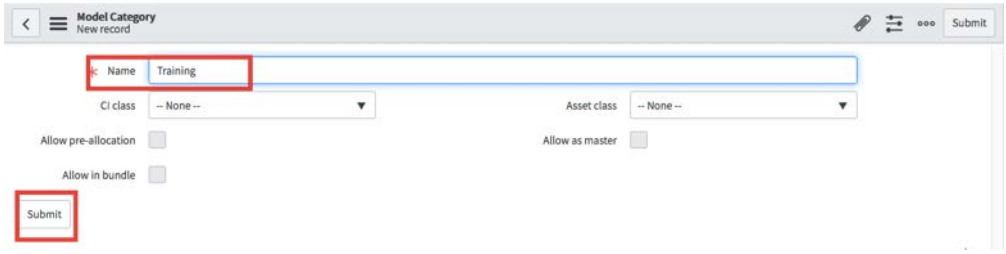
1. First, you need to create a hardware category. In the left menu, search on **model**. Click on **Product Catalog→Product Model→Model Categories.**



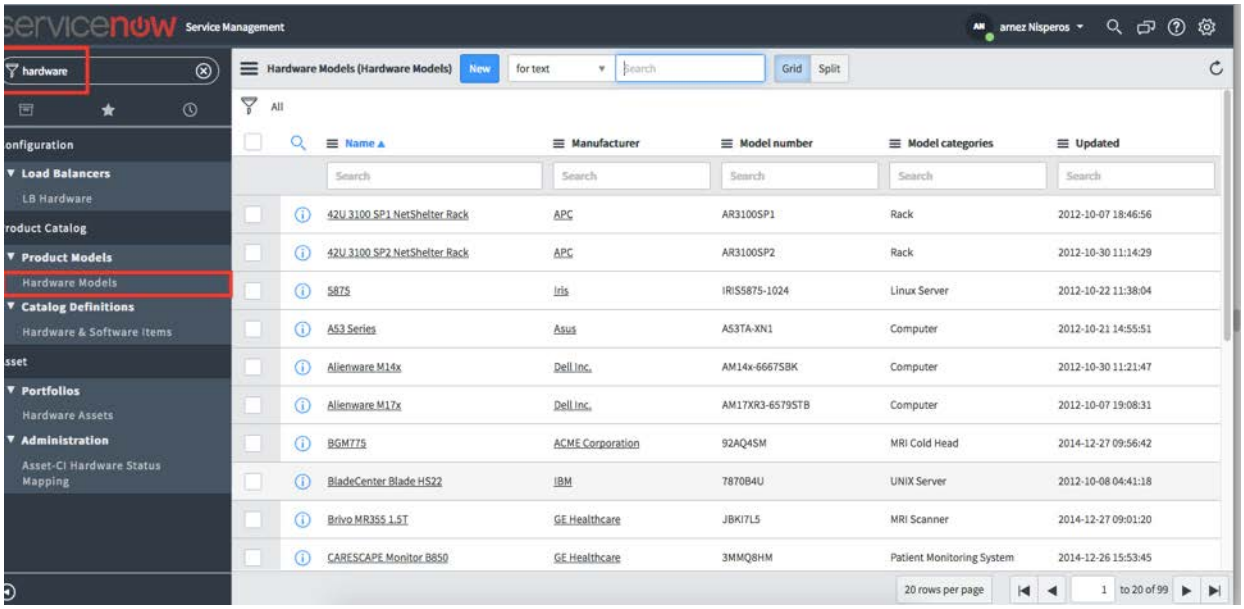
2. On the main panel, Click on **New**.



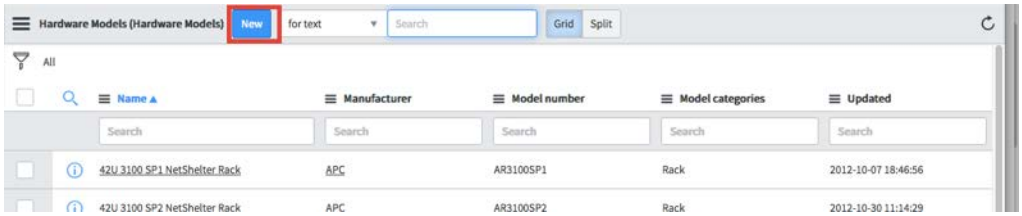
3. In the new **Model Category** form, enter **Training** in the **Name** field. Click on **Submit.**



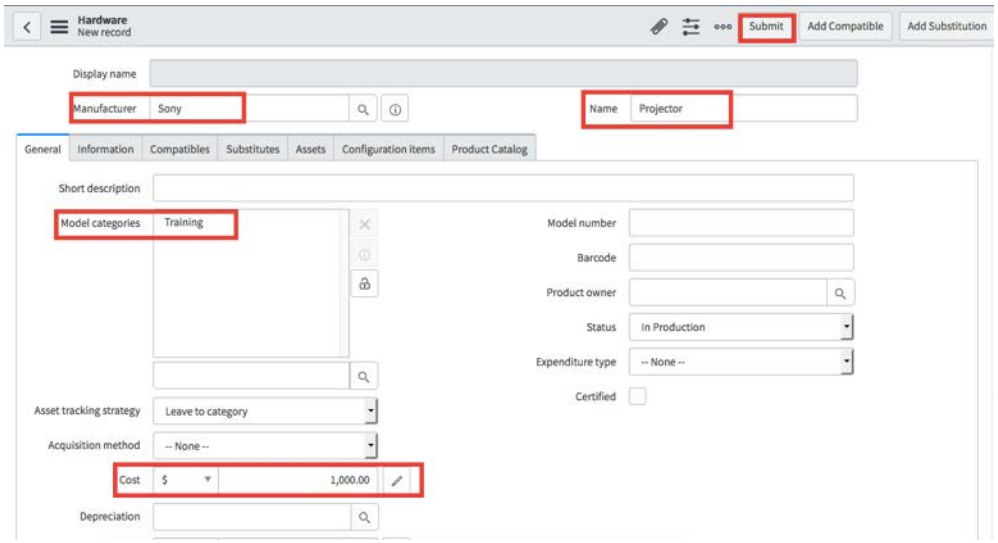
4. On the left menu, search on **Hardware.** Click on **Product Catalog→Product Models→Hardware Models**.



5. On the **Hardware Models** list view main panel, click on **New**.



6. On the **Hardware Model** new form, Next to the **Manufacturer** field, click on . Select **Sony.** In the **Name** field, enter **Projector.** Next to the **Model categories** field, click on . Select the new category you just created, **Training**. In the **Cost** field, enter 1000. Click on Submit on the upper right corner.



7. Repeat creating new Hardware Model:

Hardware Model 2

Manufacturer: Sony
Name: Wireless Microphone
Model categories: Training
Cost: 300.00

Hardware Model 3

Manufacturer: Sony
Name: AV Equipment
Model Categories: Training
Cost: 500.00

Hardware Model 4

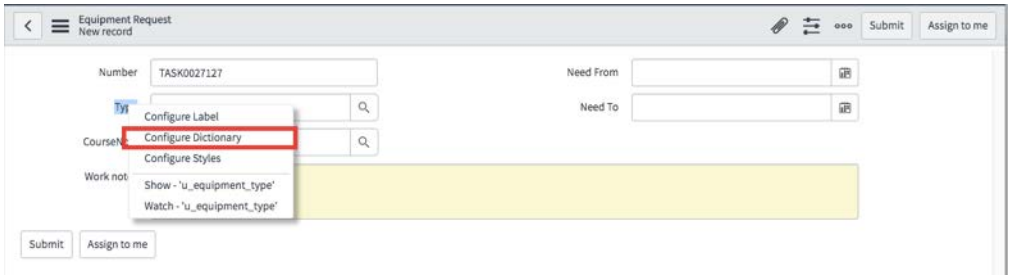
Manufacturer: Brother
Name: Network Ready Laser Printer
Model Categories: Training
Cost: 1200.00

Create a Reference Qualifier

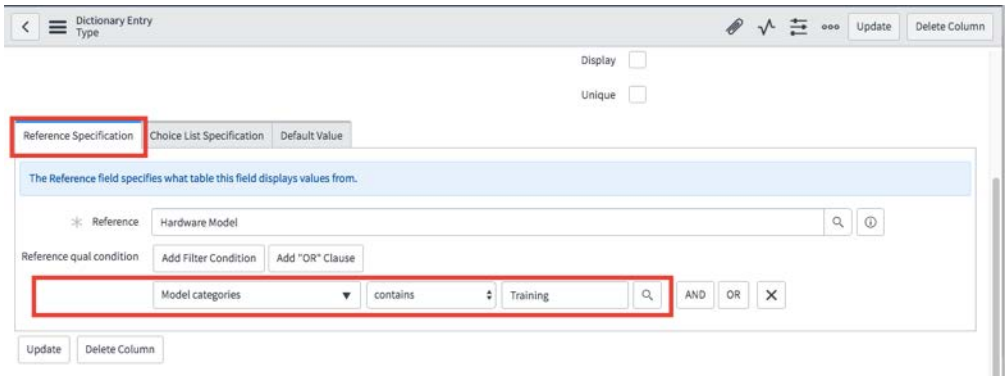
The form looks good. You can optimize the user experience even more by making reference lists more relevant. This reference list filtering can be done with a reference qualifier.

1. Next to the **Type** field, click . Note that over 80 different hardware models are available. Close the pop up window.

2. Right-click on the **Type field label** and select **Configure Dictionary**.

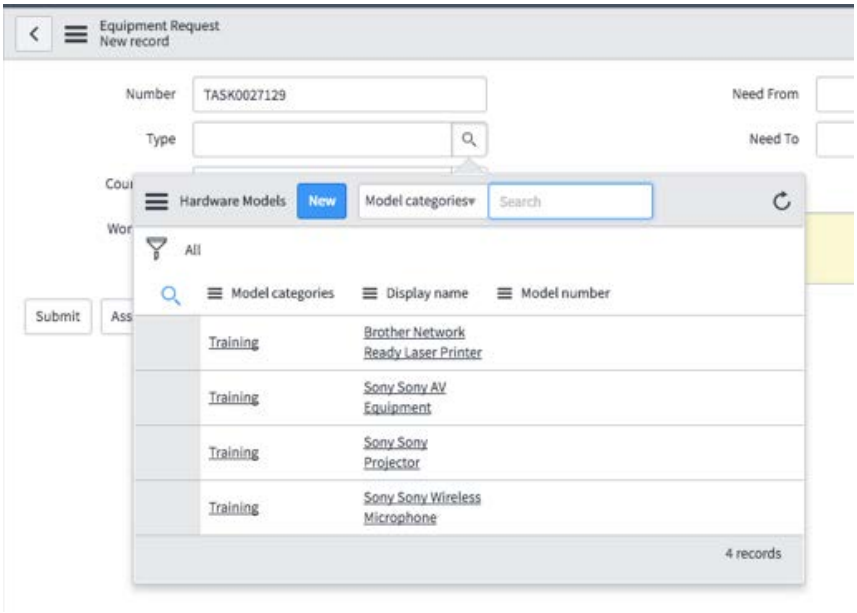


3. In the **Reference Specification** section, in the **Reference qual condition** field, set the filter condition to **Model categories contains Training** as shown below.



4. Click **Update**.

5. Next to the **Type** field, click . Note now only four hardware models are available. Close the pop up window.



8. Using what you've learned, create a form for the **Register** Course table. In the Form Designer, change the property of some fields to be mandatory.

Lab Goal

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.

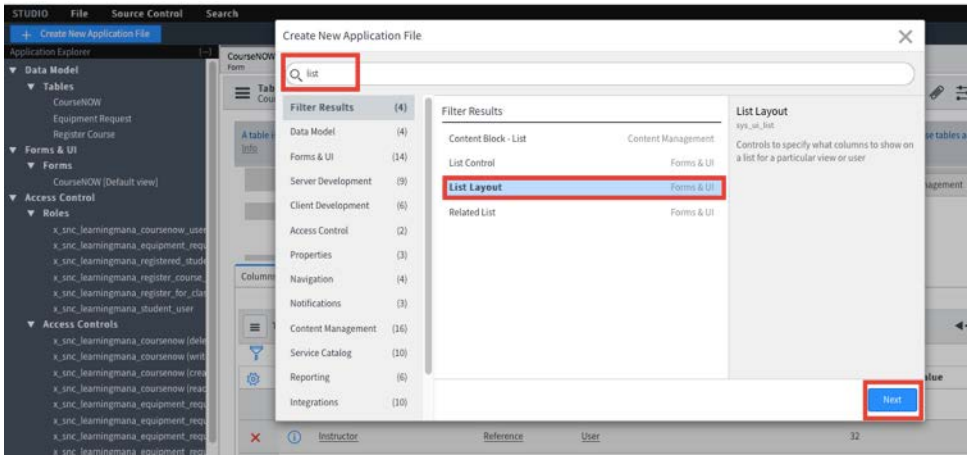
This lab explains how to create a default list layout for your data model.

Lab 2.2
List Layout View

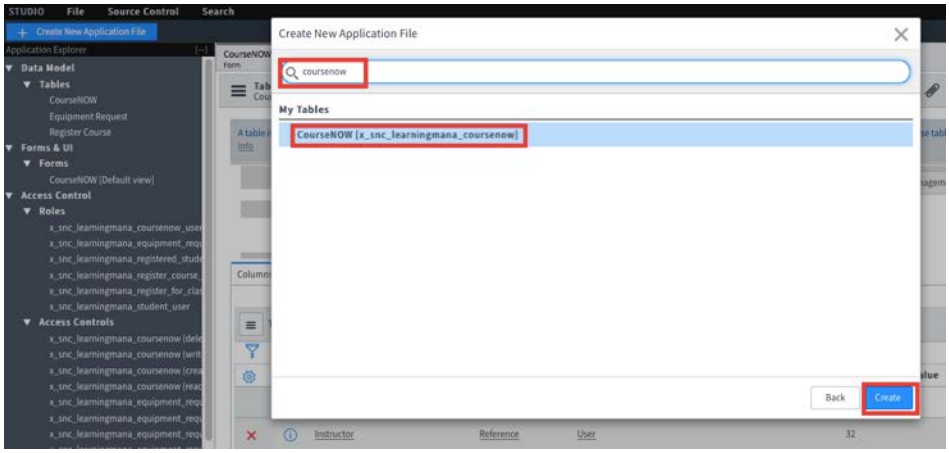
More Formatting Instructions

1. Select the tab in your browser that has the Studio open. If you've closed this tab, you can reopen it by selecting Studio from the menu in the ServiceNow tab.

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

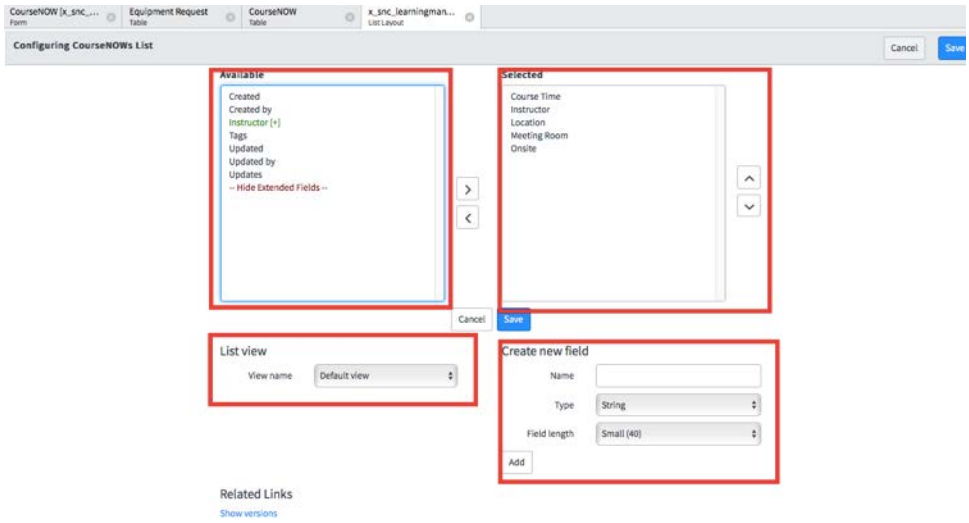


3. Click **Next** and choose the table you wish to use the List Layout on.



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.

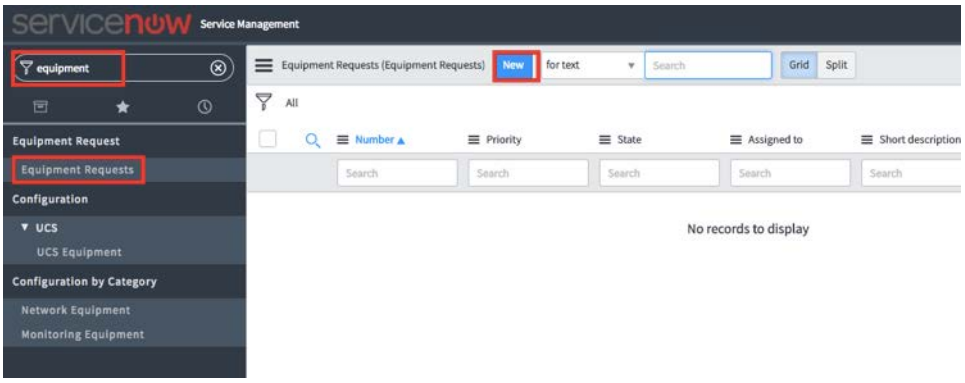
   You can create new data fields or a different view from this screen as well. We will not be doing that in this lab – just know that you can.



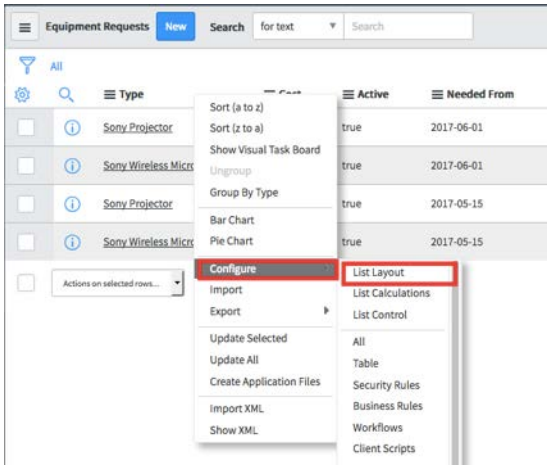
Modify the Equipment Request List Layout

Now, customize the Equipment Request list and form views following a similar process. Make sure you select the ServiceNow tab first.

1. Navigate to Equipment Request > Equipment Requests.



2. Right-click on any header (i.e.. **Type**). Click on **Configure** from the menu list. Click on **List Layout** from the menu list. The list configuration form appears.

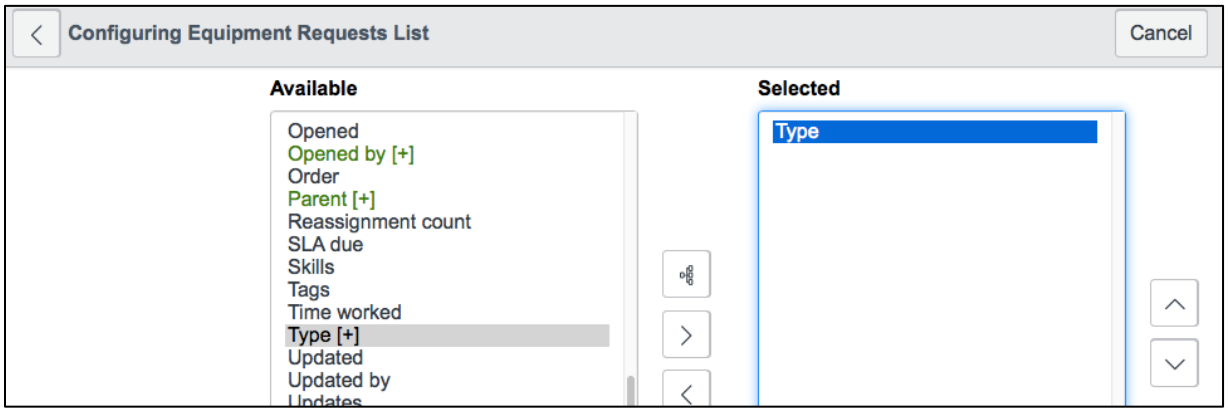


- Since the Equipment Request table extends from the Task table, you will see a lot of fields in the available fields list and some fields from the Task table in the **Selected** list.

  35. Remove all the default fields in the **Selected** list by double-clicking on the field names. Alternatively, hold down the **SHIFT** key and select all these fields and then press the **<** (remove) button.

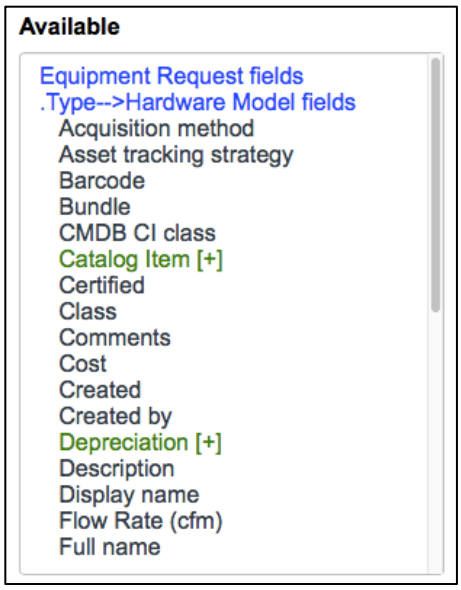
- **Note**: You can safely remove fields from a list or form view. Removing fields with configuration does not change the actual table structure.

3. In the **Available** list, double-click on the **Type** field to add them to the Selected list.

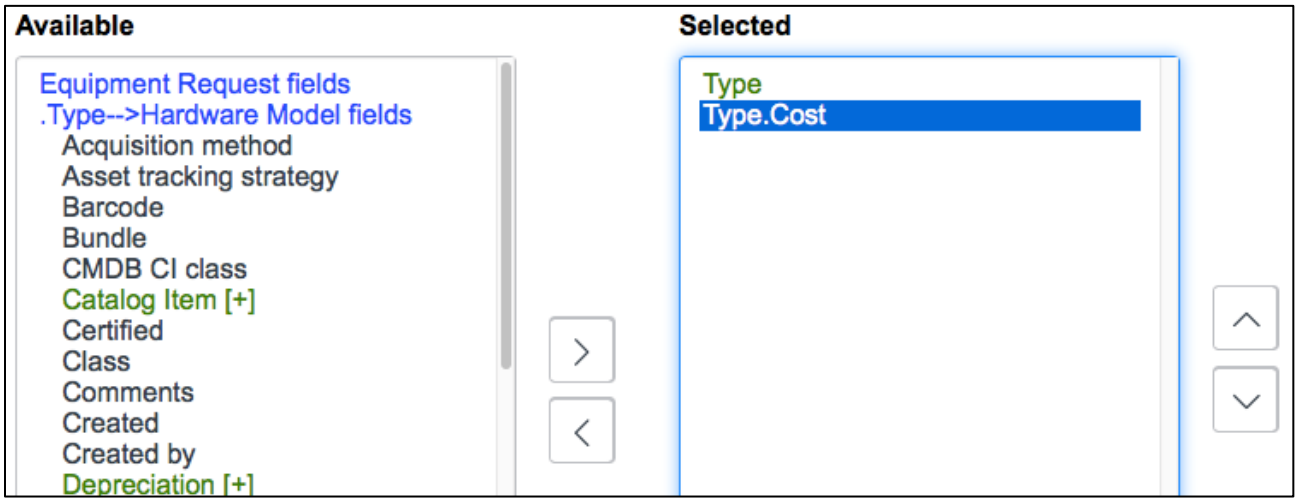


4. Display in the Equipment List a field from the related Hardware Model table. To see fields from this related table, select the **Type** field from the **Available** fields list. ServiceNow displays tables in green text with a [+] next to the table name.

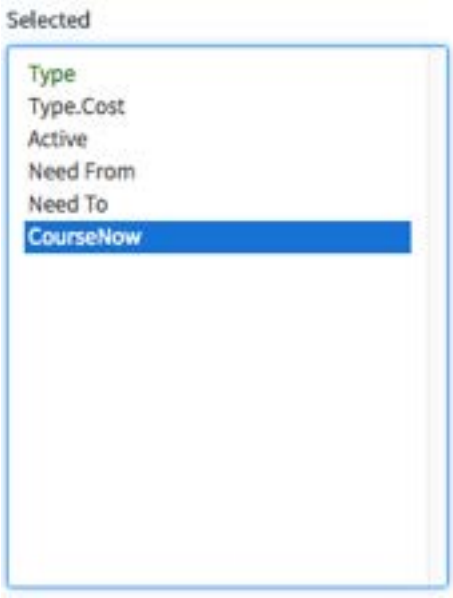
   Click on  to show a list of the fields in the related table.



5. Double-click the **Cost** field to add it to the Selected list.



6. In the **Available** list, click **Equipment Request fields** at the top of the list to return to the fields directly on the Equipment Request table.
7. Add and order the **Needed from**, **Needed until**, and **CourseNow**.



8. Click **Save**.
9. Repeat this exercise for the other tables you created using the format you like best.

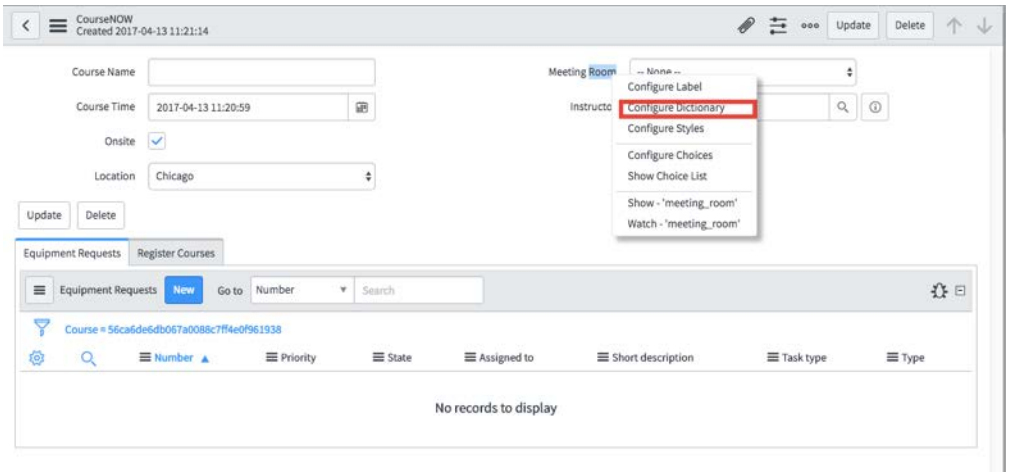
Lab Goal

Dependent fields allow you to change fields based on the selected value of a related field. As an example, when filling out your address on a form and you select your state (Florida as an example), the list box for City should only contain values that are valid for Florida. We can use this method to enforce data quality and reduce user error. Let's apply that same principle to our application.

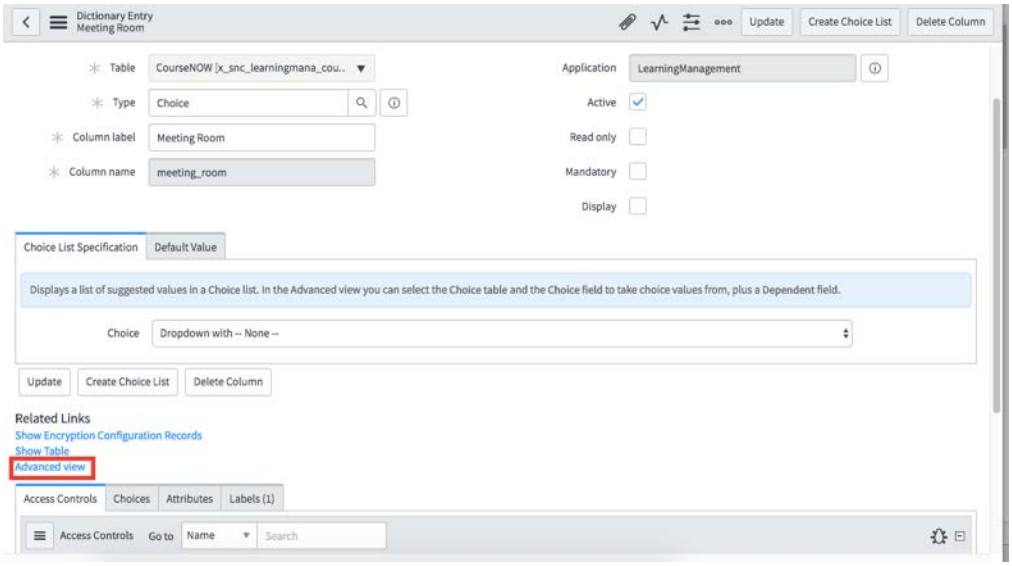
Lab 2.3
Dependent Fields & List

Configure a Field to be Dependent on Another Field

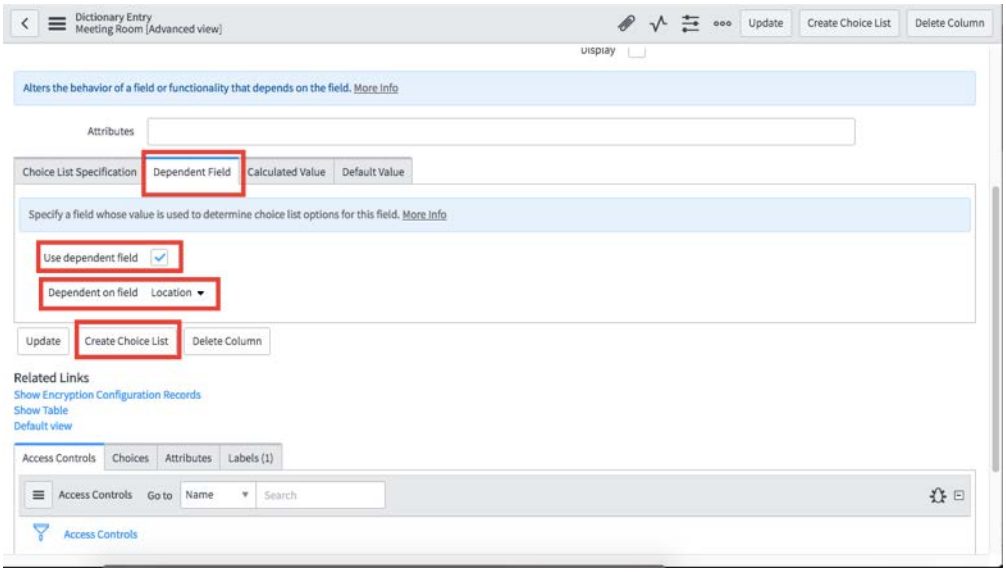
1. On the CourseNow form, right-click on the **Meeting Room** label. Click on **Configure Dictionary** on the menu list.



2. You will be on the Dictionary Entry of **Meeting Room**. In the Related Links section, click on **Advanced View.**

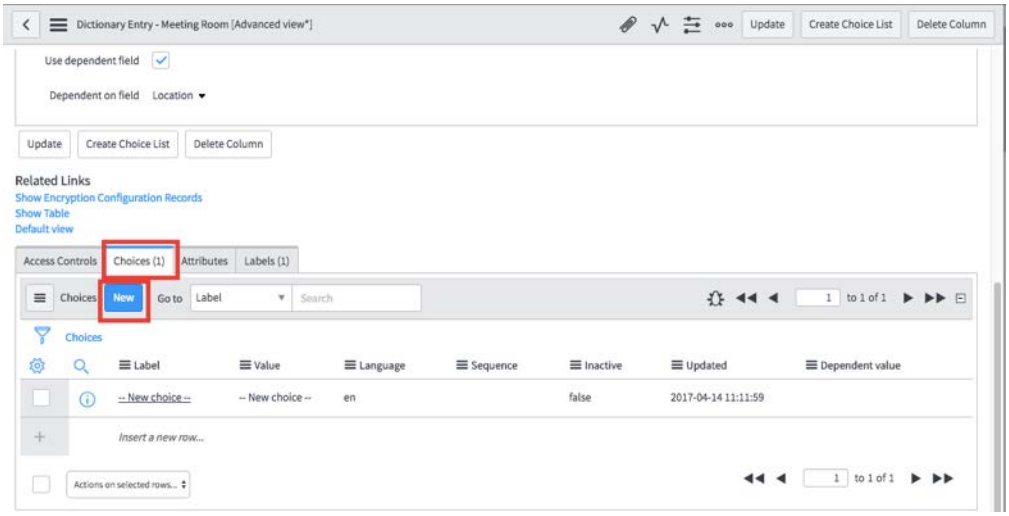


3. Click on the **Dependent Field** tab. Check **Use dependent field**. Select the field location from the list in **Dependent on field**. Right-click on the header and click on **Save** from the menu list.

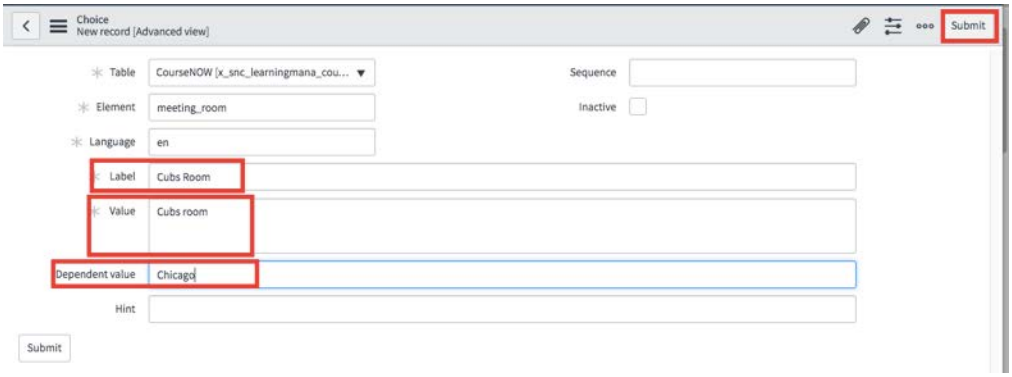


Add Dependent Choices to a Choice field

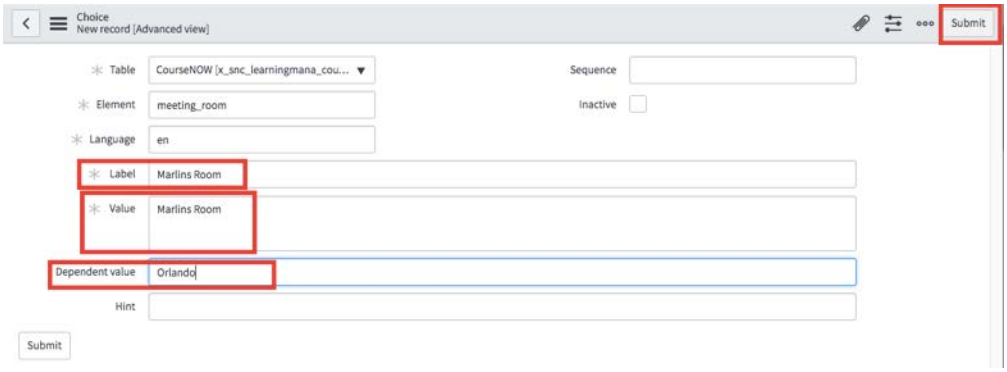
1. Click on the **Choices** tab. Click on **New** to create a new choice.



2. In the **Label** field, enter **Cubs Room**. In the **Value** field, enter **Cubs Room**. In the **Dependent value field**, enter **Chicago.** This choice will only show if Chicago is chosen in the Location field.



3. Repeat and create another choice. **Label: Marlins Room, Value: Marlins Room, Dependent Value: Orlando.**



Verify Results

1. Go back to the **CourseNow** form and test. Select **Chicago** in the **Location** dropdown. Click on the list of **Meeting Room**. **Cubs Room** should be the only option. Select **Orlando** in the Location dropdown. Click on the list of Meeting **Room**. **Marlins Room** should be the only option.

2. Add more choices to **Meeting Room** for each **Location**.

Lab Goal

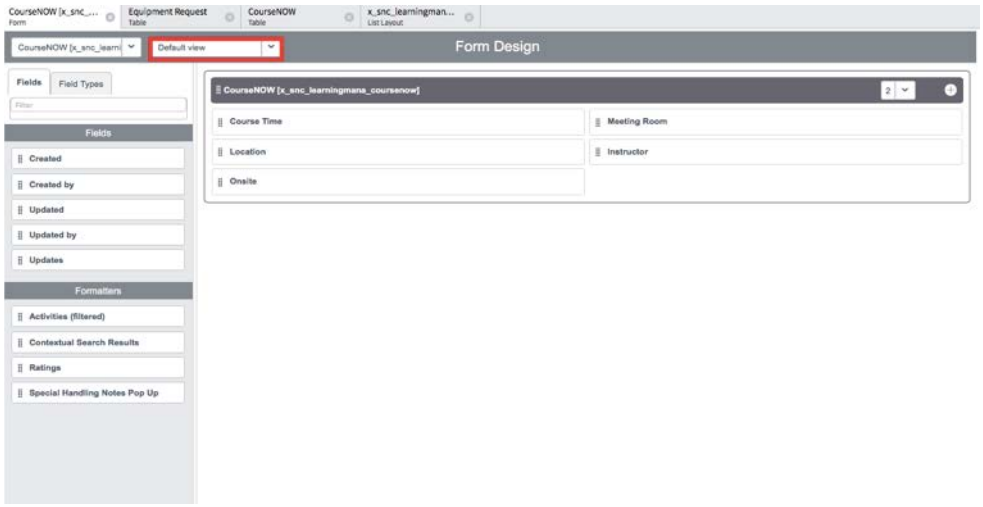
Lab 2.4
Mobile View

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.

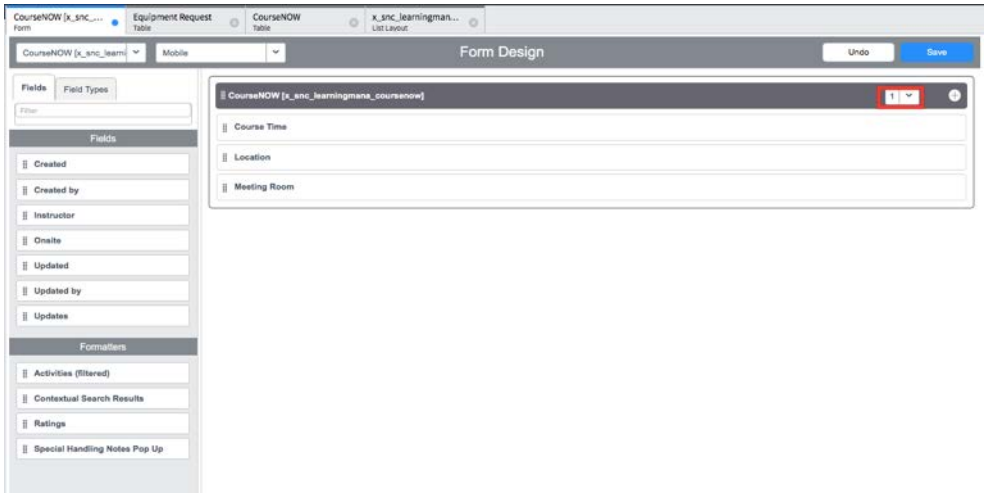
This lab also explains how to create a related list to show up when the form is rendered. Related lists are tables of data that are related in some way (either by foreign key in the database or an explicit relationship you create) to the data / form you are rendering.

More Formatting Instructions

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



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



3. Click **Save**.

Lab Success Verification

1. You should be able to browse to the newly created menu for CourseNOW 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.

Lab Goal

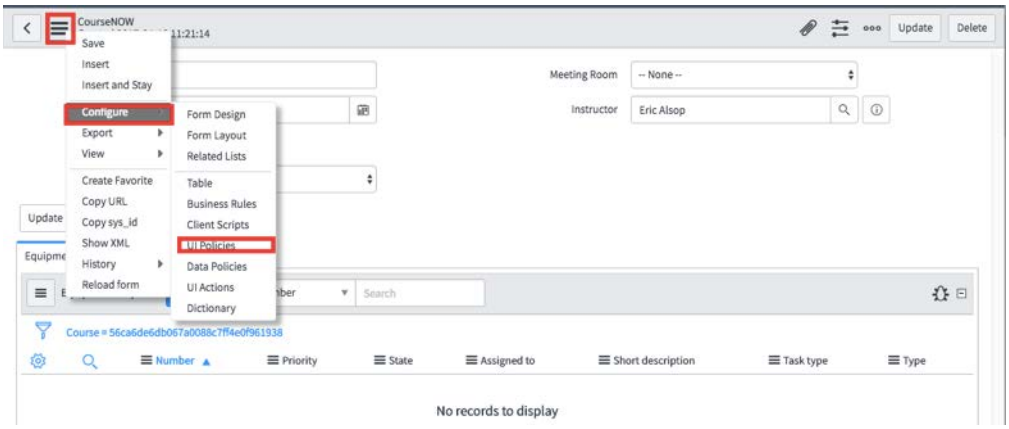
Lab 3.1
UI Policy

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.

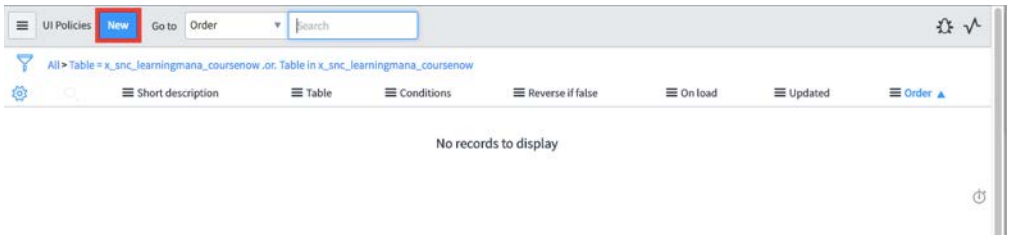
UI policies are useful for controlling the behavior of a form based on a specific condition. Another example – you may have a field you want visible only if it pertains to the state of another field. This keeps the form complexity low and prevents the form showing more than is necessary.

Create UI Policy on the CourseNow Table

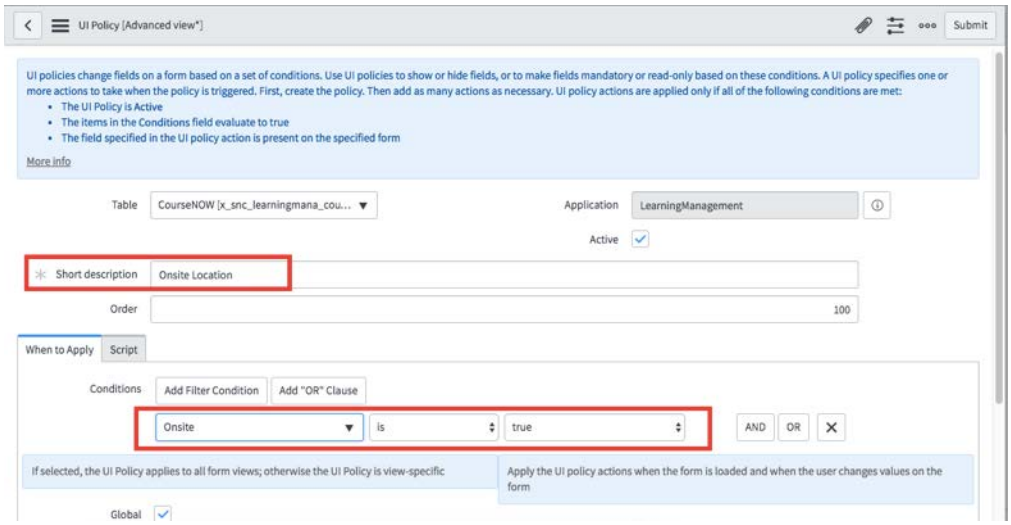
1. From the **CourseNow** form, click on the three bars in the upper-left corner in the main panel. Click on **Configure** from the menu list. Click on **UI Policies**.



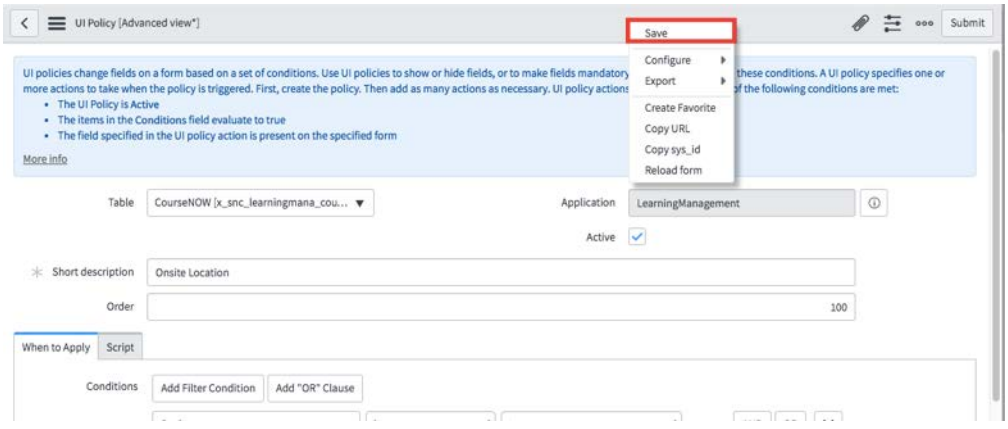
2. In the **UI Policies** list view, click on **New.**



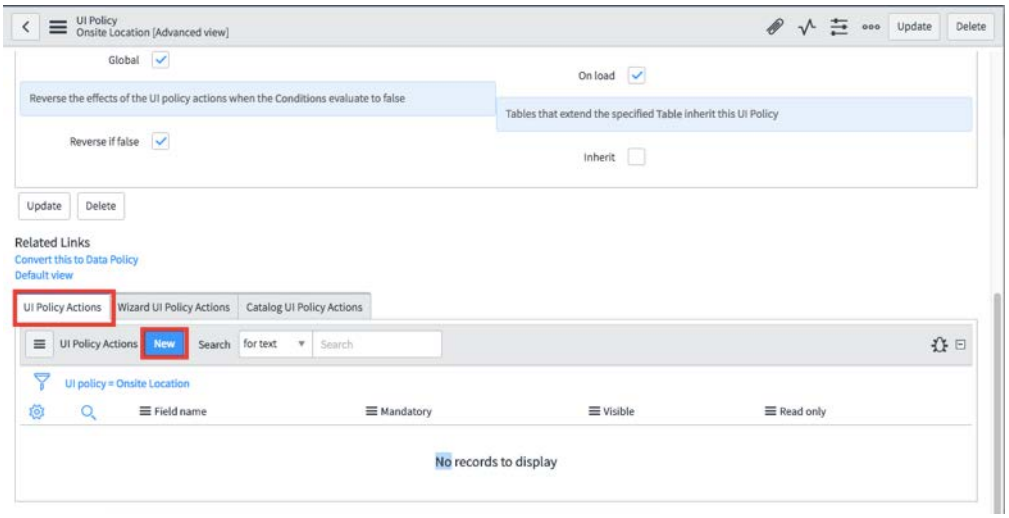
3. For the **Short Description**, enter **Onsite Location**. In the **When to Apply** Tab, select **Onsite** in the dropdown list and **is true**. This UI Policy will only apply when **True** is selected in the **Onsite** field.



4. Right-click on the header of **UI Policy** and select **Save** from the menu list.



5. In the **UI Policy Actions** tab, click on **New**.



6. In the **UI Policy Action**, select **Location** field in from the list. Select **True** in the **Visible** field. Click on **Submit** to save. When **Onsite** is set to **true**, the **Location** field will show.



7. Repeat creating a UI Policy Action. **Field name: Meeting Room, Visible: True**.



Verify Results

1. Go back to the **CourseNow** form and test. **Location** field and **Meeting Room** field should not be shown. Select True in the **Onsite** field. **Location** field and **Meeting Room** field should appear.

Lab Goal

Lab 4.1 Business Rules

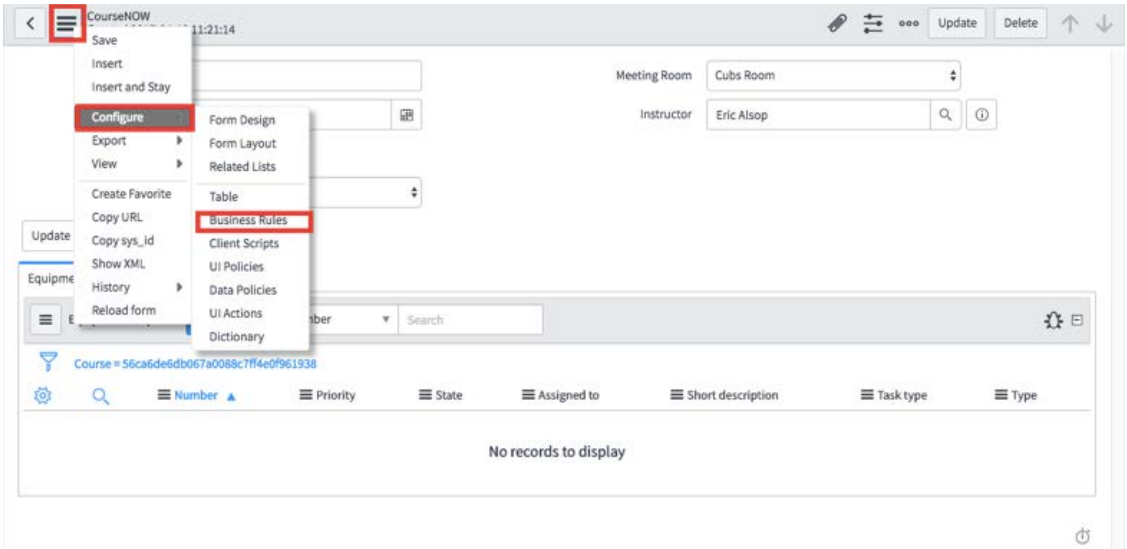
There are so many places in the platform you can script to get the behavior you want – business rules, script actions, inbound email actions, scripted REST and SOAP web services and processors, etc. This scripting engine is what gives the ServiceNow Platform the flexibility to build pretty much whatever 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 enforce rules especially the ones where no code in involved.

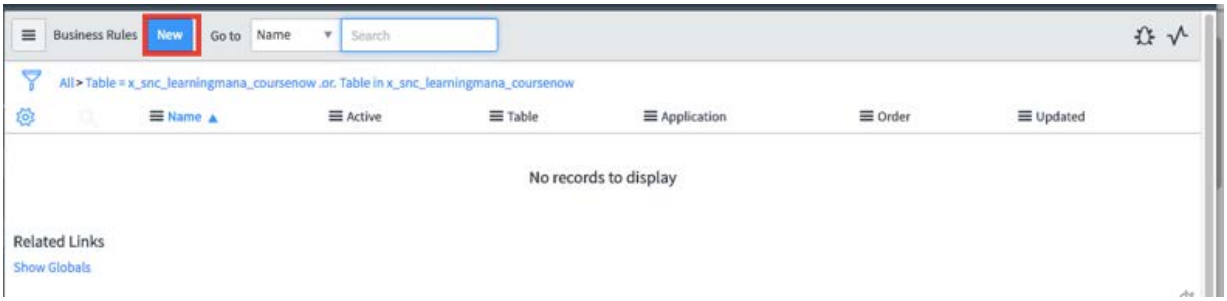
Use Case: An Instructor does not travel and can only have a course in the Chicago Location. A business rule can be written to ensure that all courses assigned to a particular instructor will always be in a certain location.

Create a Business Rule on the CourseNow Table

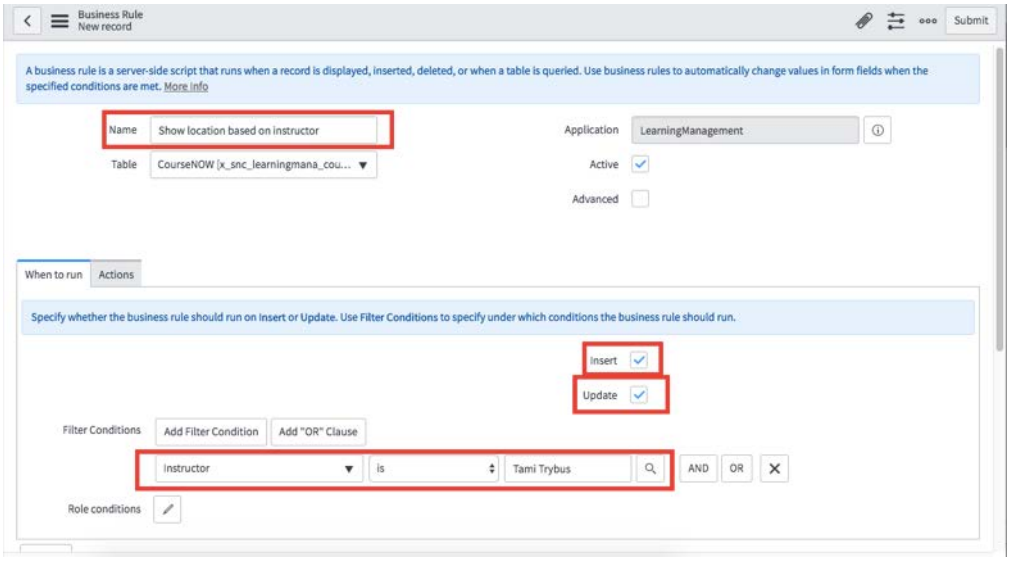
1. On the **CourseNow** form, click on the **three lines** at the upper-left corner of the main panel. Click on **Configure** from the menu list. Click on **Business Rules** from the menu list.



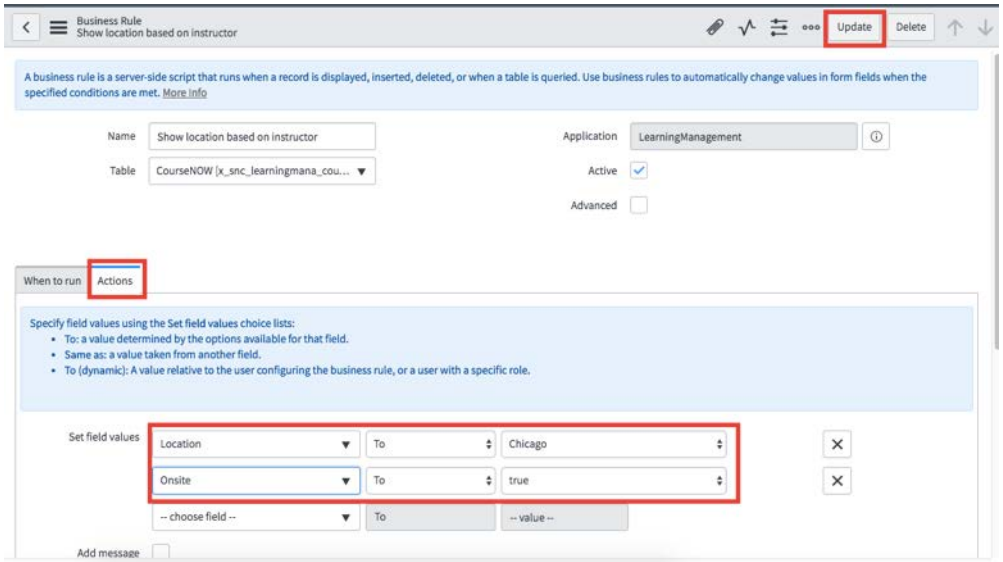
2. In the **Business Rules** list view, click on **New**.



3. In the **Business Rule New record** form, enter **Show location based on instructor** in the **Name** field. In the **When to run** tab, check **Insert** and **Update**. In the Filter Conditions, select **Instructor** in the dropdown list. Select **Is.** Click on . Select **Tami Tybus**.



4. In the **Action**s tab, in the **Set field values**, select **Location** from the dropdown list. Select **To**. Select **Chicago** from the dropdown list. Select **Onsite** from the download list. Select **To**. Select **True** from the download list. When **Tami Trybus** is the **Instructor**, the **Onsite** field will be set to **True** and the **Location** field will be set to **Chicago**.



Verify Results

1. Go back to **CourseNow** form. Set the **Instructor** field to **Tami Trybus**. Set the **Onsite** field to **False**. Save the **CourseNow**. Verify that the **Onsite** field is **True** and the **Location** field is set to **Chicago**.

Test Drive the Application

Lab 5.1
Test the Application

This lab explains how to verify what you have done in the previous labs.

Related Story
As a Training Manager, when I am looking at a Courses, I need to see the equipment requests related to that course so that I can make sure all the requests for that course have been processed in a timely manner.

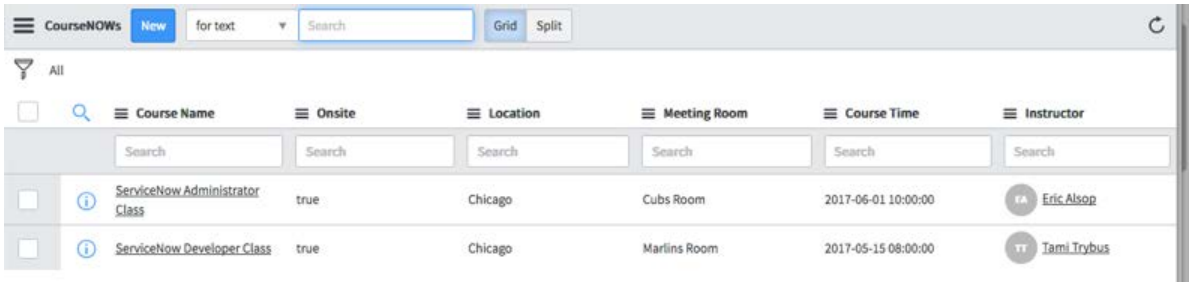
Enter Courses

It is time to start enjoying the fruits of your labor. Add a few course records.

1. Navigate to **CourseNow > CourseNow**.

2. Create CourseNow records using the following information.

Name	Onsite	Location	Meeting Room	Course Time	Instructor
ServiceNow Administrator Class	Yes	Chicago	Cubs Room	10:00:00 2017-06-01	Tami Trybus
ServiceNow Developer Class	Yes	Orlando	Marlins Room	08:00:00 2017-05-15	Eric Schroeder

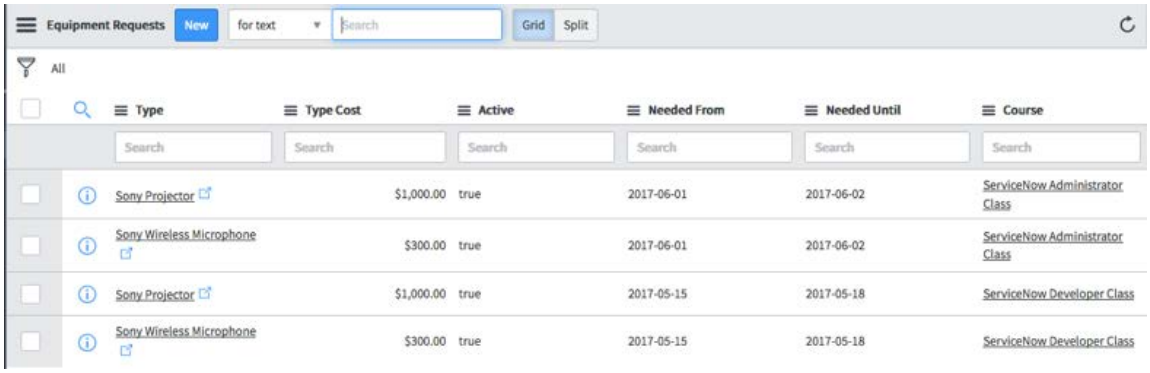


Now add some corresponding equipment request records.

Enter Equipment Requests

1. Navigate to **Equipment Request > Equipment Requests.**
2. Create equipment request records using the following information.

Type	CourseNow	Needed from	Needed Until
Sony Projector	ServiceNow Administrator Class	2017-06-01	2017-06-02
Sony Wireless Microphone	ServiceNow Administrator Class	2017-06-01	2017-06-02
Sony Projector	ServiceNow Developer Class	2017-05-15	2017-05-18
Sony Wireless Microphone	ServiceNow Developer Class	2017-05-15	2017-05-18



Note: You can sort by the column headings in the list view as well as edit any field by double-clicking. Give it a try.

Add a Related List

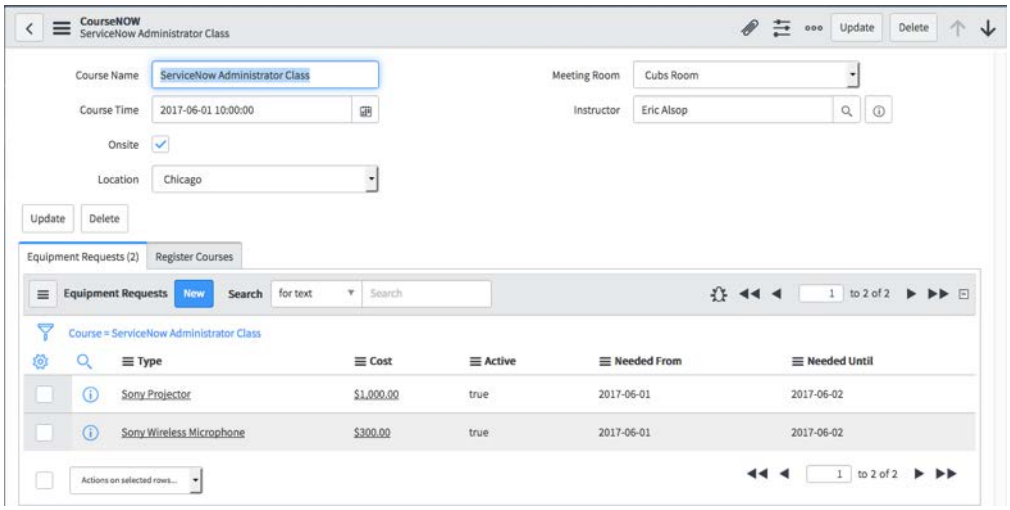
Make one small enhancement to the application. When you look at a particular course, you want to be able to see the equipment requested for that event on the same page. Since the Equipment Request table is related to the CourseNow table, you can do this easily.

1. Navigate to **CourseNow > CourseNows**.
2. Open the **ServiceNow Administrator Class**.
3. Right-click on the gray header bar and select **Configure >Related Lists**.
4. From the **Available** list, select **Equipment Request-> CourseNow** and click .



- **Note**: Also add the Attachments related list as desired. The platform has built-in functionality that allows you to add a file, document, or image to any record.

5. Click **Save**. Notice that the related equipment requests for the event now show up in the same view.



You now have a functioning application hosted securely in the cloud. This is already a huge improvement over ad-hoc spreadsheet sharing. But you are not done with the application yet.

Record Producers

Lab 6.1 Record Producers

To this point we have worked on building and configuring the application components that will be used by those in the back office. Now we want to shift our focus to how perspective students will work with the application to register and view available courses.

To register for a course, students will need to fill out an online form. This will be done using a record producer which will present a form to the user and add a record to the table of registrations.

Then, we will create a simple self-service page for users to interact with using the ServiceNow Service Portal. The Service Portal allows you to create rich, responsive interfaces that are easy to use. We will also expose the record producer in the Service Portal so people are able to register for classes.

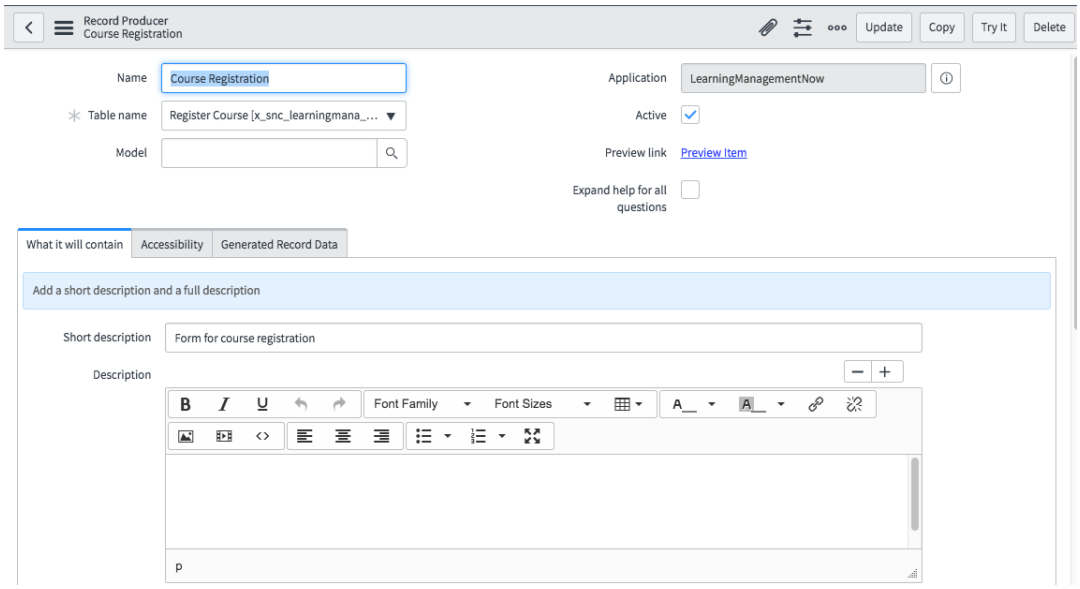
Record Producer

We are going to create a simple form using a record producer.

1. Navigate to **Service Catalog > Record Producers**
2. Select the New button to create a new record producer.

Fill in the fields as follows:

Name: **Course Registration**
Table Name: **Register Course**
Short Description: **Form for course registration**



3. Save the record.

4. After the record saves, you will then have a set of tabs at the bottom of the form which are the related lists. One of these tabs is labeled variables which is where we define the form. Navigate to the variables tab and select **New.**



5. Fill out the form as follows:

   Map to field: **Checked**
   Field: **CourseNow**
   Type: **Reference**
   Order: **100**
   Question: **Course Name**

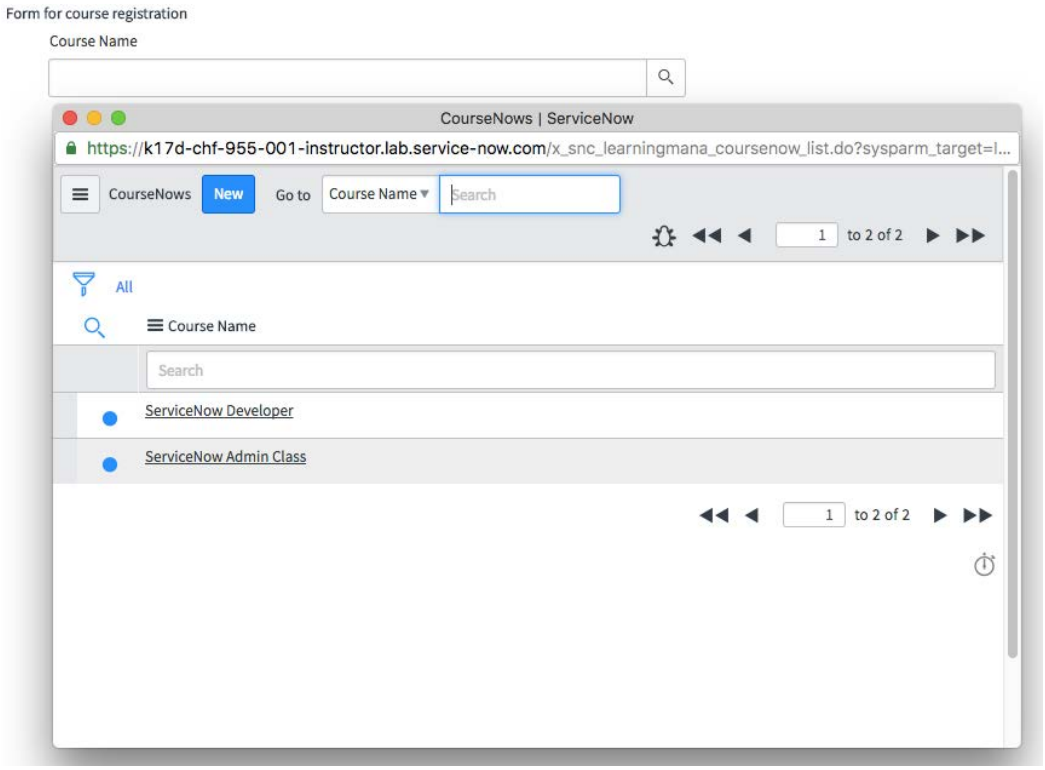
6. Select the Type Specifications tab and fill it in as follows:

   Reference: **CourseNow**
   Click the Update button to save.

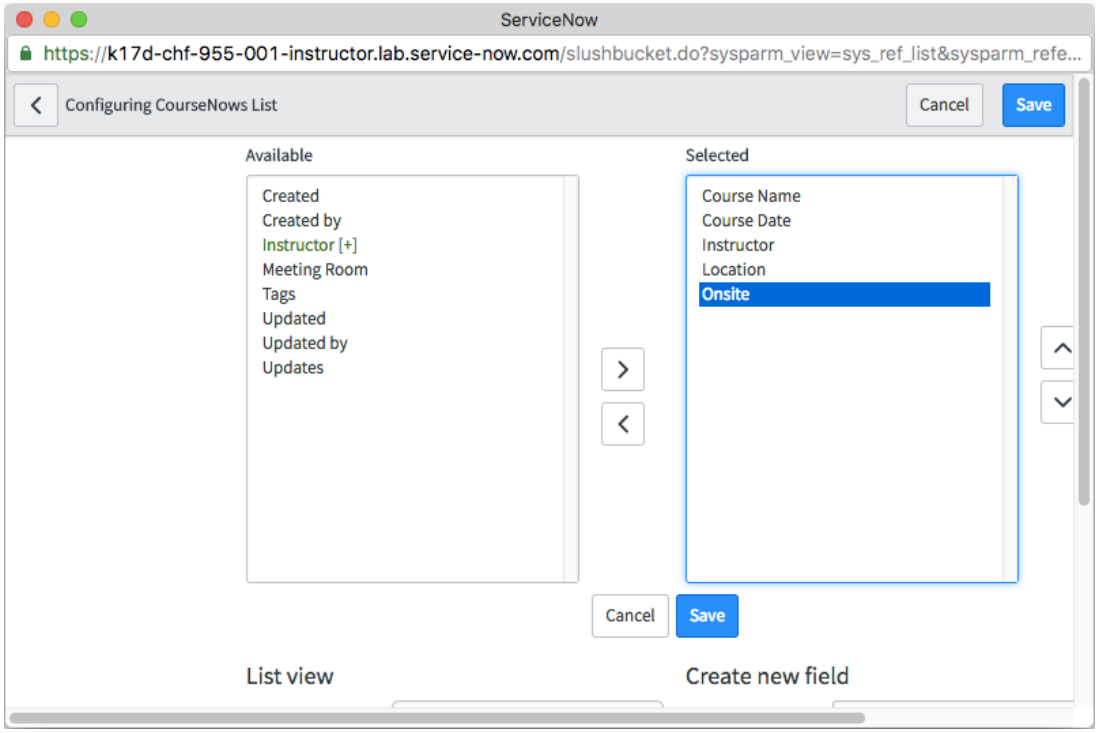
7. Now let's take a look at the form so far and see what it looks like. Select **Try It** from the upper right corner. This will display the form with a the variable that we defined which is the course for registration. Select the magnifying glass next to the field to see a list of courses.

   After the record saves, you will then have a set of tabs at the bottom of the form which are the related lists. One of these tabs is labeled variables which is where we define the form.

   Navigate to the variables tab and select **New.**



8. We see the list of courses but there is only one field displayed. It would be helpful if the user could see more detail such as the course date and instructor. Let's modify the list to show those fields. Right-click on the column heading entitled **Course Name.** Choose **Configure > List Layout**. Select a few fields from the available list and move them to the selected list and click Save. These fields will display in the popup list of courses.



Click **Save**, select a course and then click **Submit**.

9. Our form only has one field on it so far and needs to have more information in order to register a user for the class. Using the steps above, add the **Student** field to the form so it looks like this:

   Hint: Student will be a reference to sys_user and Attending will be a reference.

   *Hint: If the fields don't show in the correct order on the form check to see that you specified a numeric value for the **order** field.*



Lab Success Verification

Test your form once you've added the other field. Submit the form and then look at the list view of your **Register Courses** table. You will see an entry in the list for your registration record that you just created using the record producer.

Do you notice any fields that are missing or have missing data? We didn't include all the fields on this simple form so this is to be expected. Robust forms with required fields, default values, and a little bit of simple code can dramatically change the way a form works and drives user behavior. We've only scratched the surface with this simple example of what a record producer can do.

Service Portal

Students need to have a dashboard where they can view available courses and see which courses they are enrolled in. Instead of directing users to the platform UI we will utilize the Service Portal which is a platform feature with in ServiceNow. The Service Portal provides a response-rich user interface to display any type of data.

Lab 6.2
Service Portal

Service Portal is built on Angular.js and Bootstrap but you don't have to know either to create a great experience thanks to the included widgets. Widgets are reusable components that you drag-and-drop onto a canvas and configure through forms. Let's get started by creating a page that lists all the current courses.

1. Navigate to **Service Portal > Service Portal Configuration.** A new tab will open in your browser and gives you several options for configuration. We are going to use the Designer to create a new page. Click **Designer.**

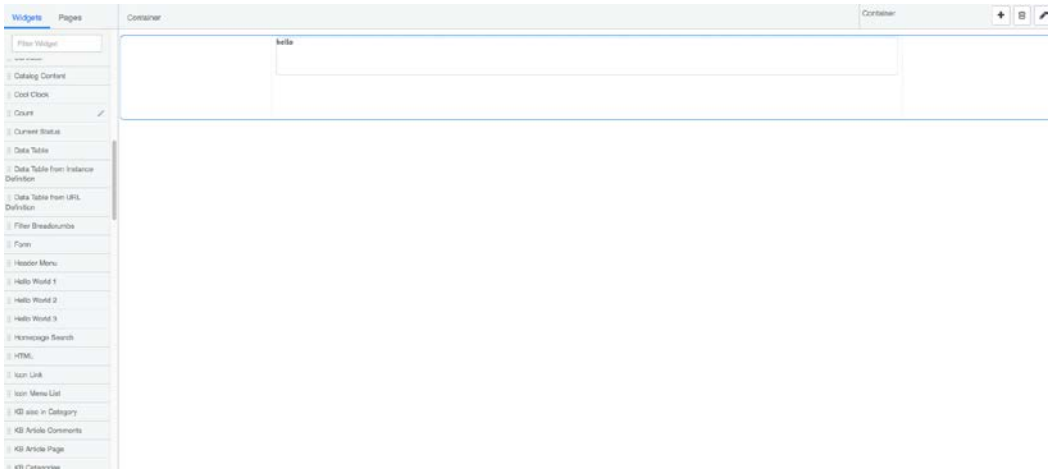
2. In the new tab you will see all the pages that are part of the current Service Portal. You will also see an option highlighted in blue to **Add a new page.** Click that hyperlink.

   Enter the Page Title of **CourseNow** then click **Submit.**

3. You are now ready to drag and drop items onto the canvas to create your page. The left side of the screen shows your layouts and widgets while the main portion of the screen is the canvas which represents a WYSIWYG view of the page you are creating.

   In the center you will see a blue + symbol. Click on it and select the top item labeled **12**. *This determines the screen layout as defined by Bootstrap. We won't go into detail here but there are other labs at Knowledge where you can explore the Service Portal in more detail and get more information.*

4. On the left, scroll down in the widgets section and drag the HTML widget onto the 12 width column that you just created. Your screen should look like this.



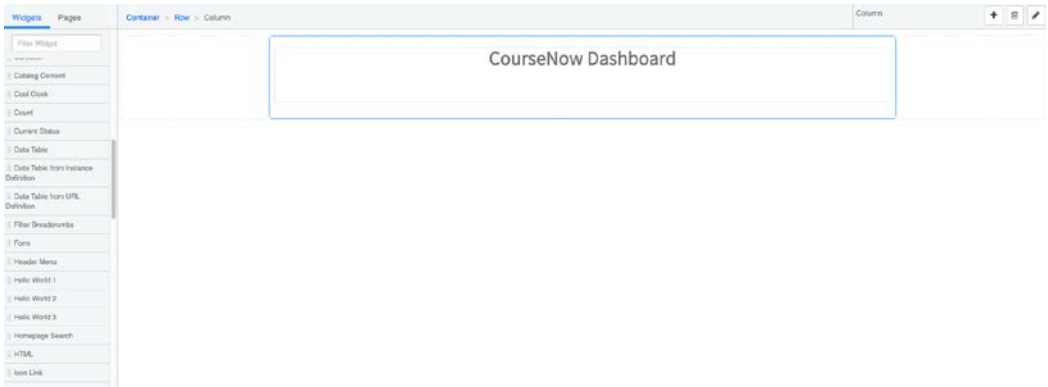
5. Click inside the blue square where you see the word “hello” to select the HTML widget. Select the pencil (edit) icon in the upper right corner of the HTML widget.

   In the HTML section of the form type in **CourseNow Dashboard.**

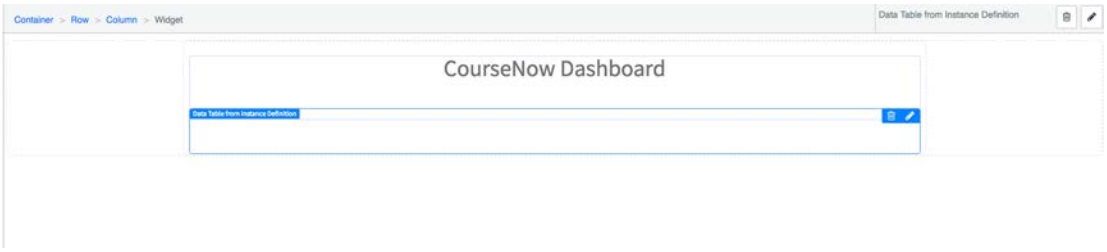
   Highlight the text and change the drop down that says paragraph to the **Heading 1** option.

   Center the text using the icon in the tool bar.

   Click Save.



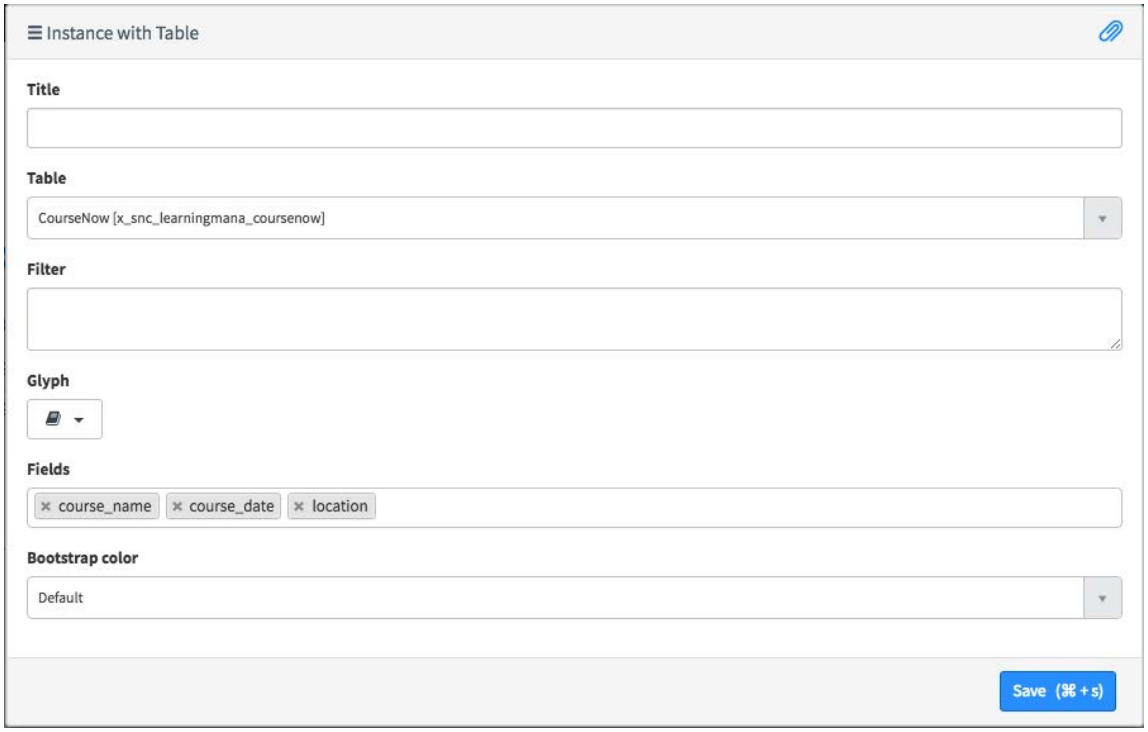
6. Now that we have a title, let's add some data to display the list of courses available. To do this we will search for a widget called **Data Table from Instance Definition** in the left hand window. You can use the search bar at the top left or scroll to find the widget. Drag and and drop that widget just below the HTML widget.



7. This same widget can be added to the form multiple times to create multiple instances of the same widget. Each instance can be configured differently to display whatever data you would like. We're going to configure this widget to display the list of courses.

   With the Data Table widget selected, click on the pencil icon to edit.

   Fill out the form as shown here. Then click the Save button.



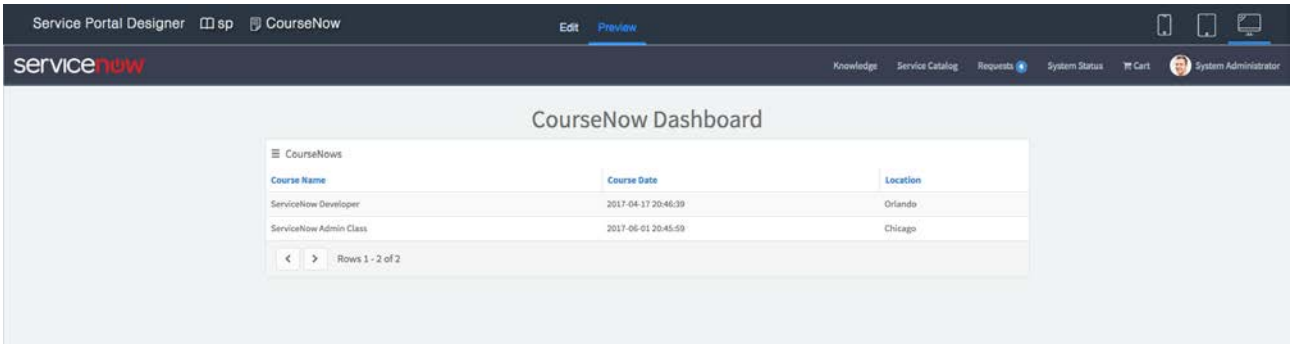
8. Notice how the widget now contains live data? This is part of the power of using the Service Portal and widgets. As you design your pages you will be able to immediately see the results using your data from your instance.

9. Click on the **Preview** text at the top center of the window. This allows you to view the page with out any of the container and widget controls/outlines. Notice how the title bar is now displayed to give you full context of how the page will display.

   In the upper-right corner, you will see three icons which represent a mobile device, a table and a desktop. By clicking on each icon you will see the preview change to show how the page will appear on each device.

Lab Success Verification

Your page should look like the screenshot below – a simple HTML title and a table of courses from the CourseNow catalog.



Automate with Workflows and Approvals

Lab 7.1
Workflow

Too often, business processes are implemented by a combination of email, ad-hoc conversations and other inefficient, manual methods. Workflows can automate this, resulting in increased efficiency and reduced errors. Additionally, workflows provide an audit trail—this feature can be critical for an application that has to comply with internal or external regulations.

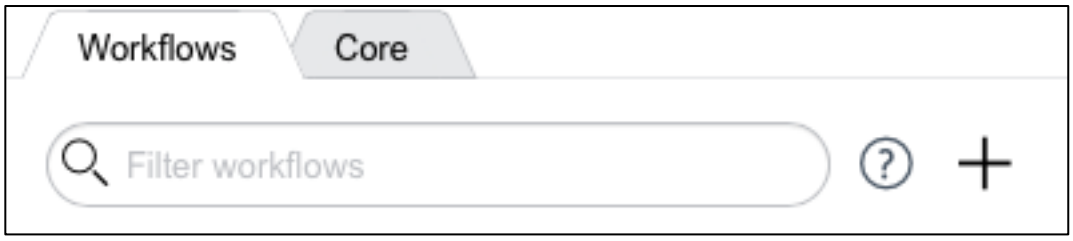
In this sample scenario, a manager and the Training VP have to approve the request for any equipment that costs over $500. You will build a workflow to automate this process. As soon as a user enters a request for equipment costing over $500, the workflow will get triggered. This is a simple (but common) scenario. The platform workflow capabilities allow complex criteria for triggering approvals and branching logic as well as integration with external systems.

Related Story
As a Training VP, I need to approve requests for any equipment that costs over $500 so that I can control my budget and make sure the request is justified.

Start the Workflow

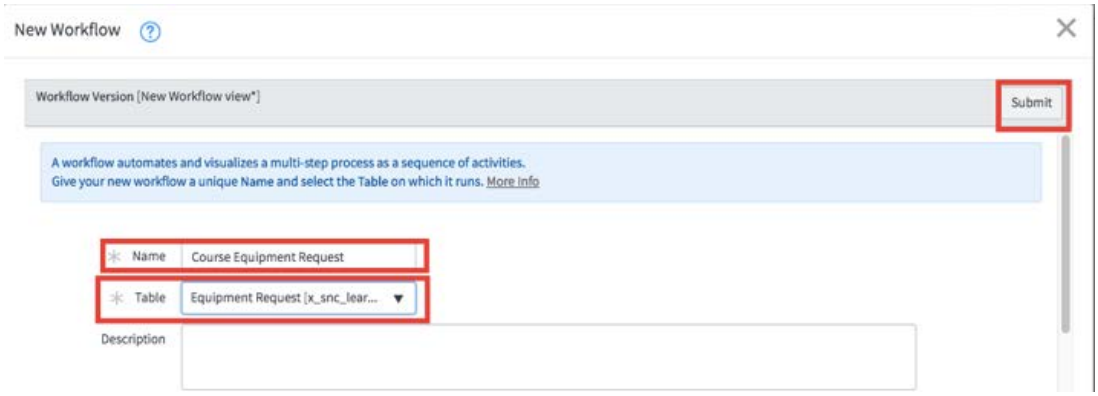
In this section, create the workflow container.

1. Navigate to **Workflow > Workflow Editor**. A new browser tab opens with the workflow editor.
2. In the **Workflows** tab, click + . The **New Workflow** dialog opens.



3. In the Name field, type **Course Equipment Request**.
4. In the **Table** field, select **Equipment Request [x…_equipment_request]** (…=Application you created).

5. You want the workflow to run only for active requests. Set the **Condition** to **Active is true**.

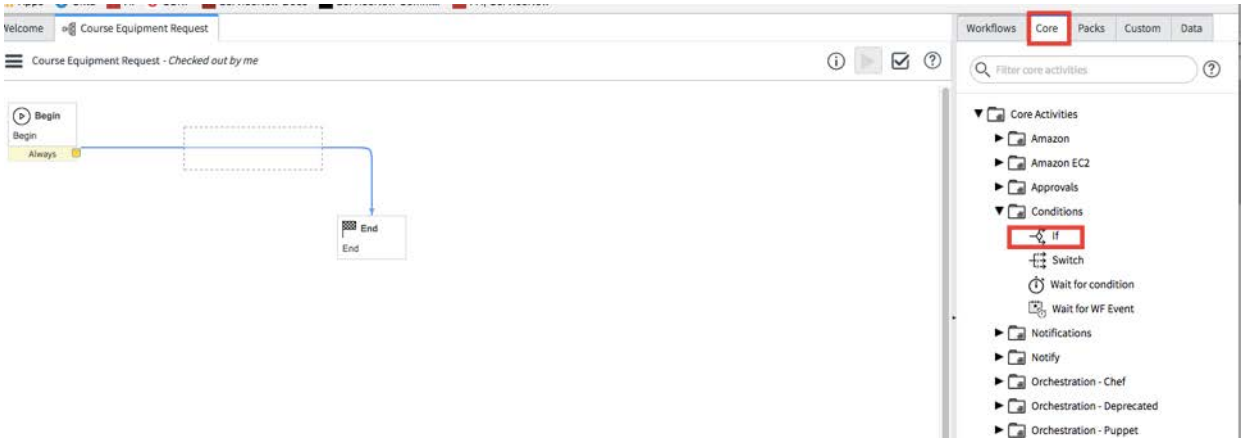


6. Click **Submit**. An empty workflow is shown. On the right, in the **Core** tab, is the palette of tasks and other activities you can add to the workflow.

Add an If Activity

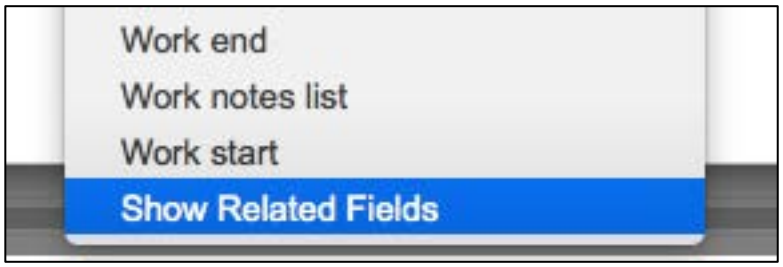
In this section, you add an activity to check if approval is required based on the equipment cost.

1. In the **Core** tab, expand the **Conditions** folder. Drag the **If** activity between the **Begin** and **End** activities. The connecting line will turn blue. Drop it there.



2. In the **New Activity: If** dialog **Name** field, enter **Over $500?**.

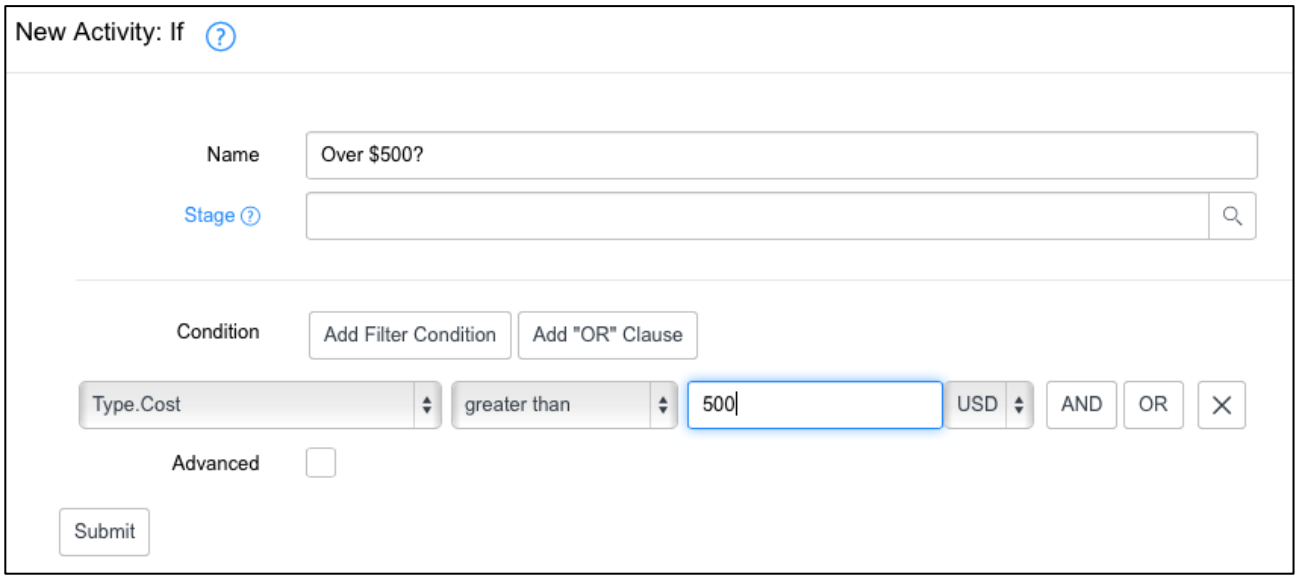
3. For the **Condition**, you will use the **Cost** field from the related **Hardware Model** table. To access the fields on the related table, from the **field** list, select **Show Related Fields**.



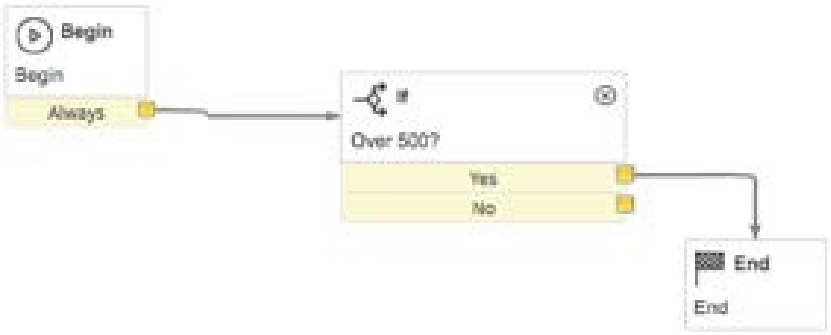
4. Select **Type → Hardware Model fields**.

5. Select **Cost**.

6. Complete the condition to read **Type.Cost greater than 500 USD**. The completed form is shown below.

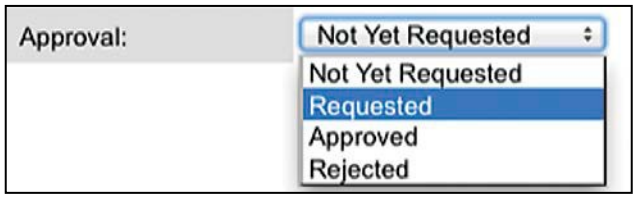


7. Click **Submit**.



Add an Approval - User Activity

You want to track the status of the workflow in a field on the record itself so that the user who submitted the request can quickly know whether their request was approved or not. For this you use a field from the task table called Approval (remember that Equipment extends Task so you have access to all the fields in the task table). The Approval field can have four possible values as shown below—this works nicely for what you want to do.



Fortunately, by using approval activities, the workflow will set this field automatically as it progresses. In this section, you add manager approval.

1. To prepare for the workflow for the new activity, click the line that connects the **If** activity and the **End** activity. Press **Delete**. Move the **End** activity down and to the right.



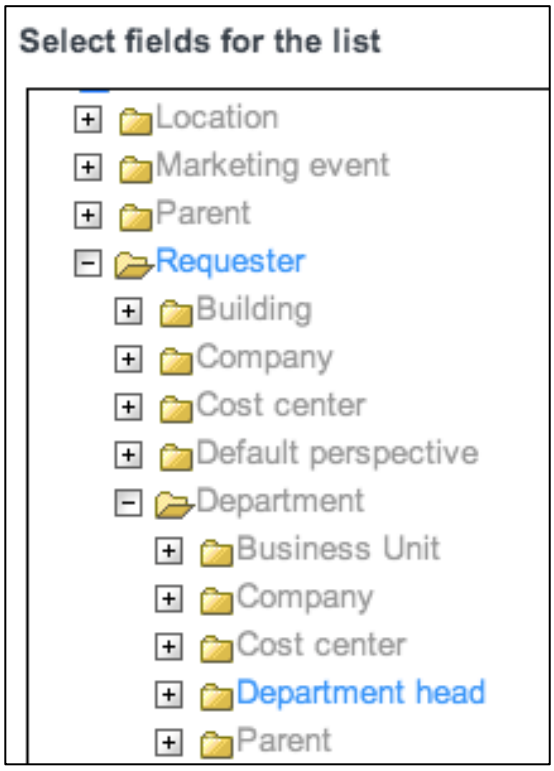
2. In the **Core** tab, expand the **Approvals** folder. Drag and drop the **Approval-User** activity to the right of the **If** activity.

3. In the **New Activity: Approval – User** dialog, in the **Name** field, type **Manager Approval**.

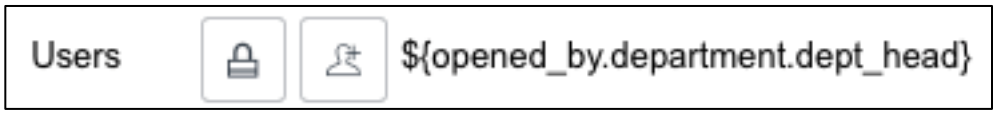
4. You can send the request to the requester's department manager. Click on  next to **Users.**

5. Click .

6. In the **Select fields for the list** dialog, expand **Requester**, expand **Department**, and click **Department head**.



7. Click . You should see the variable **${opened_by.department.dept_head}** in the **Users** field.



8. Click **Submit**.

Add Approval - Group Activity

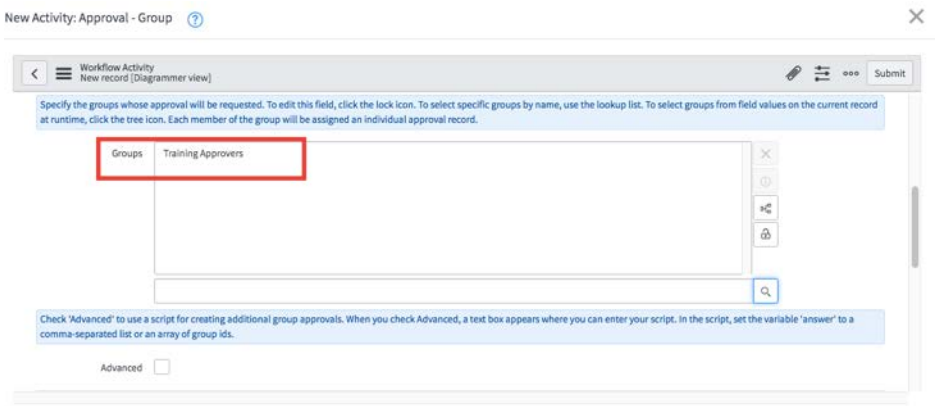
Now add the second level of approval by the Marketing Approvers group.

1. This time you add a group approval. Drag and drop the **Approval - Group** activity to below the **Approval - User** activity.

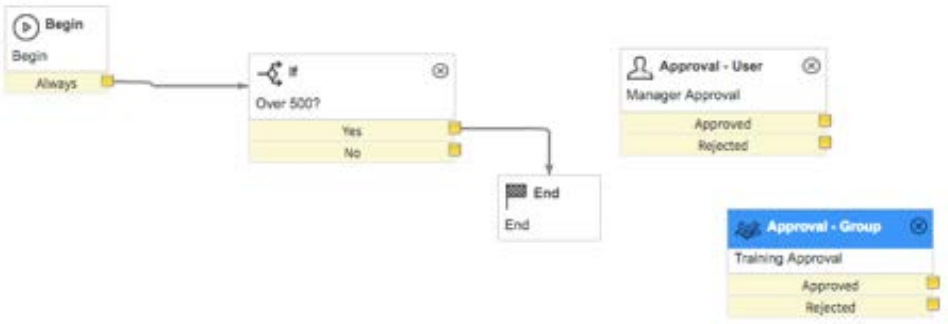
2. In the **Name** field, type **Training Approvers**.

3. Next to the **Groups** field, click .

4. In the **search** field, type **Training Approvers** and press **Enter**. Click .



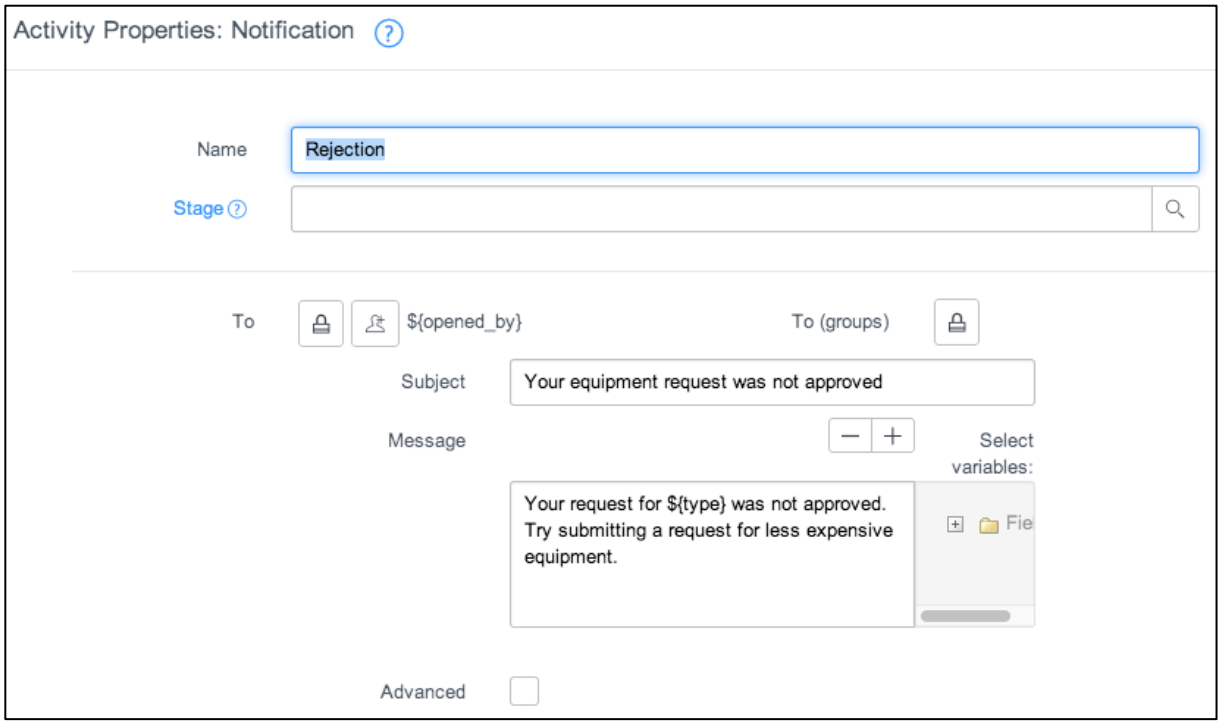
5. Click **Submit**. The workflow should now look like the example.



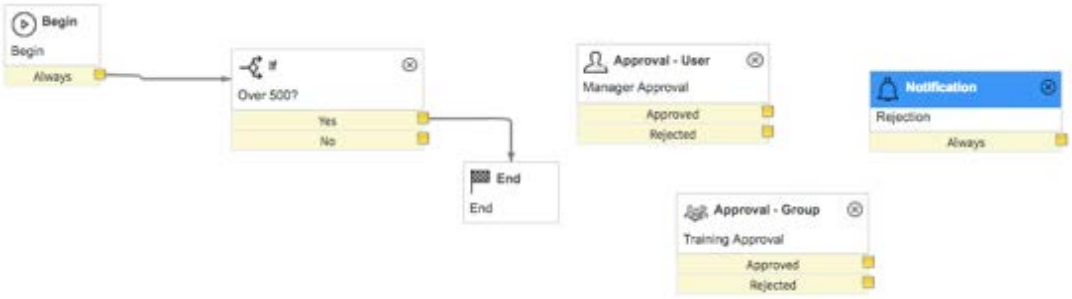
Add a Notification Activity

It would be nice for the requester to be notified if the request is rejected. Use a Notification activity for requester notification.

1. In the **Core** tab, expand the **Notifications** folder, drag the **Notification** activity to the canvas and drop it above the **End** activity.
2. In the **New Activity: Notification** dialog, in the **Name** field, type **Rejection**.
3. Unlock the **To** field. Using the **Select fields dialog box** as you did before, select **Requester**.
4. In the **Subject** field, type **Your equipment request was not approved**.
5. In the **Message** field, type **Your request for ${type} was not approved. Try submitting a request for less expensive equipment**.



6. Click **Submit**. Your workflow should look like the example.



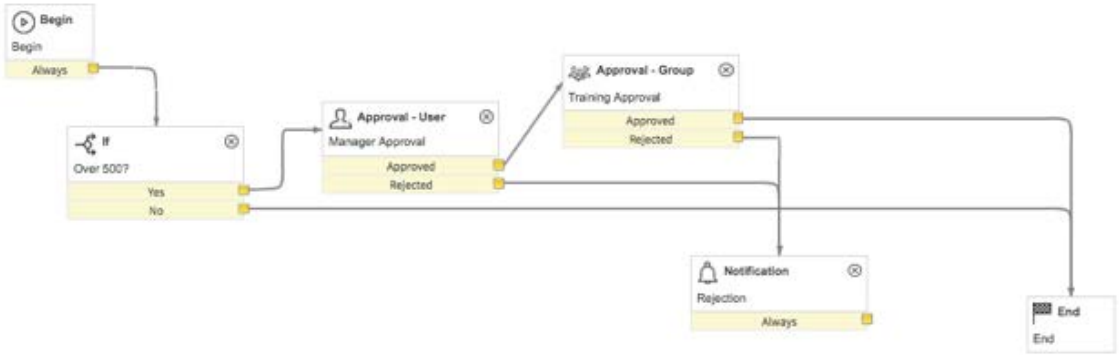
Connect the Activities

Now that the activities are in place, connect activity results to the appropriate next step in the process.

1. To connect activities, drag the square next to the activity result to the target activity. Alternatively, you can right-click on the activity result and select **Link to…** to use a dialog to make the connections. Use this table to make the connections:

From activity…	Drag result…	To activity…
If	Yes	Approval - User
If	No	End
Approval – User	Approved	Approval - Group
Approval – User	Rejected	Notification
Approval – Group	Approved	End
Approval – Group	Rejected	Notification
Notification	Always	End

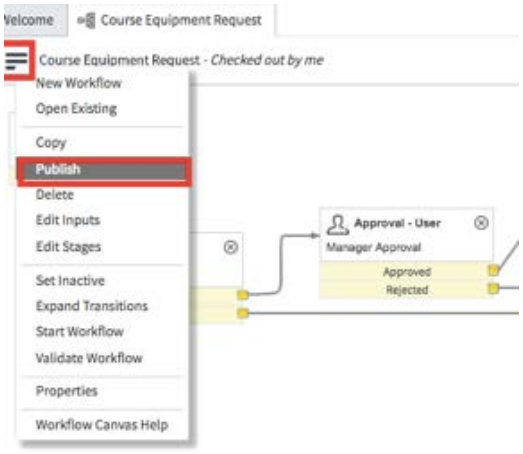
2. Rearrange the activities to make the process flow more readable.



Publish and Test the Workflow

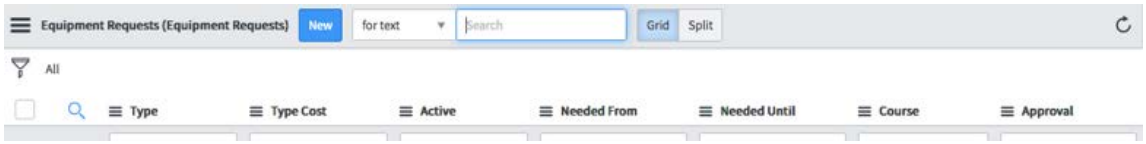
The workflow is complete and ready to be activated. In this section, you will publish the workflow.

1. In the **Workflow Actions** menu, select **Publish**. The status at top should now show **Published**.

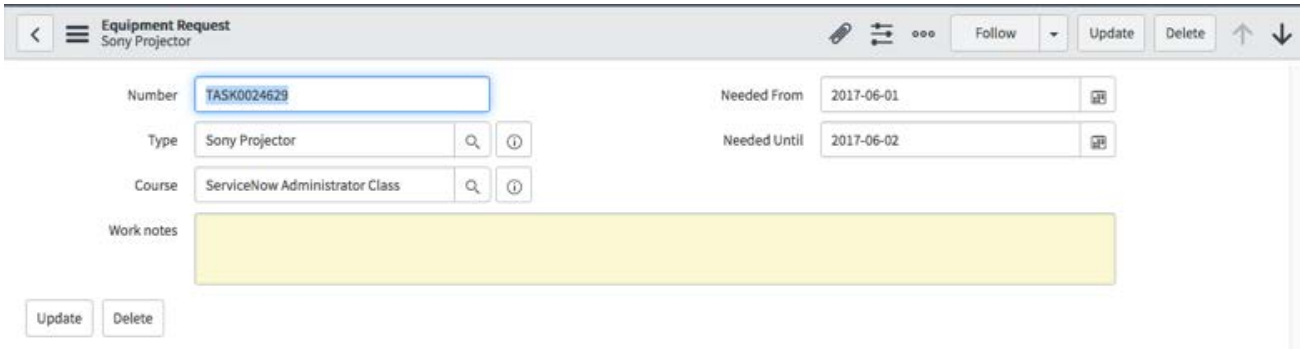


2. Before you test the workflow, make a simple addition to the Equipment list view. In the browser tab with the main application, navigate to **Equipment Request > Equipment Requests**.

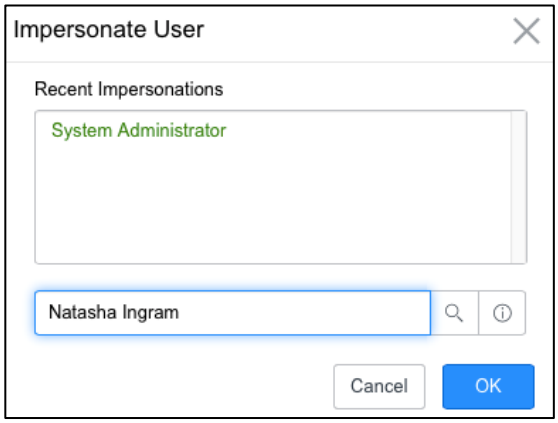
3. Configure the **Equipment Requests** list view and add the **Approval** field.



4. Now test the approval process. Enter a new equipment request for an equipment **Type** that costs more than **$500** (see sample below) and click **Submit**.

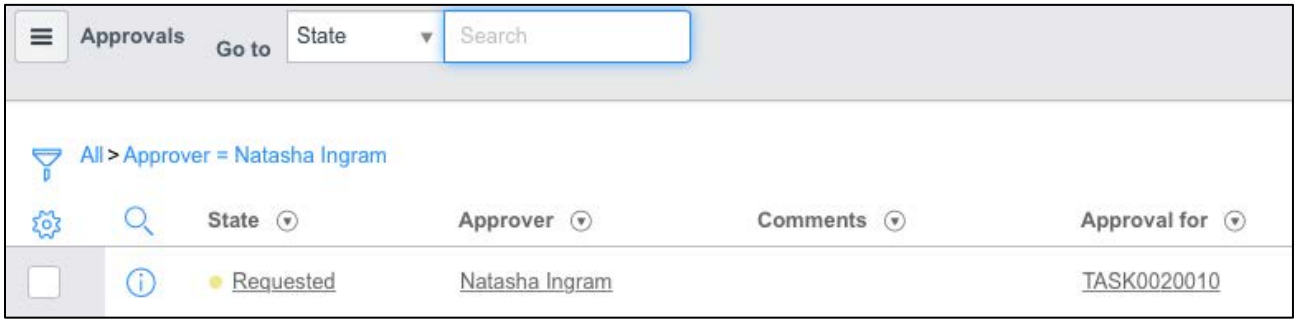


5. To see if someone got the approval request, you first need to log in as Natasha Ingram who part of the approval chain. As a System Administrator, you can impersonate other users in the system. Click  in the banner area and in the dialog that appears search for **Natasha Ingram**.



- You should now be impersonating Natasha Ingram.

6. Navigate to **Self-Service > My Approvals**. A list view of all the approval requests appears.



Click the **Requested** link for the recently submitted request.

7. Click **Approve**.

8. Now go back to the role of System Administrator. Click  again and double-click **System Administrator**.

9. Navigate to **Equipment Request > Equipment Requests**. The recently approved item should appear with the status **Approved**.

Notice, that in this example, you can change the approval field status directly. Though this may be fine in this example, in a real world scenario, you may want to make this field read-only for the user who submits the request.

Workflows and approvals are extremely powerful features. With the ServiceNow Service Automation Platform, this functionality is easily accessible by your applications.

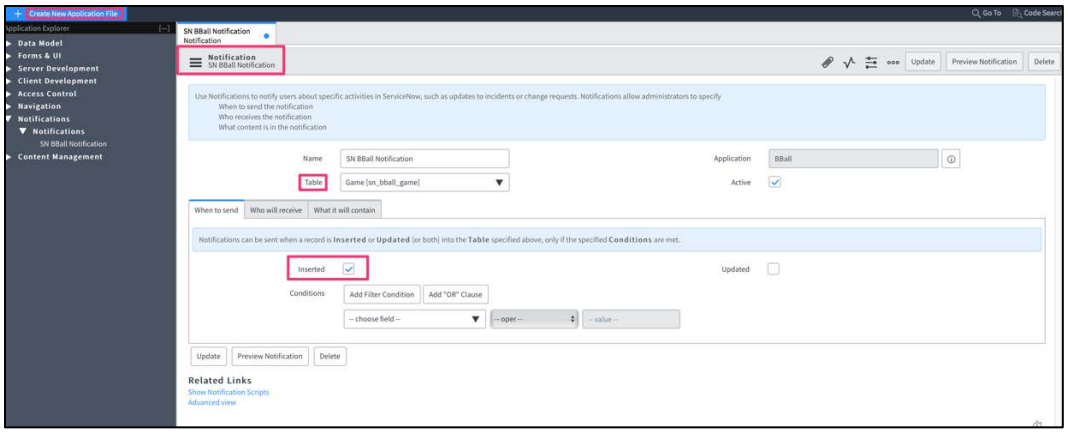
Lab Goal

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

Lab 8.1 Notifications (OPTIONAL LAB)

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.

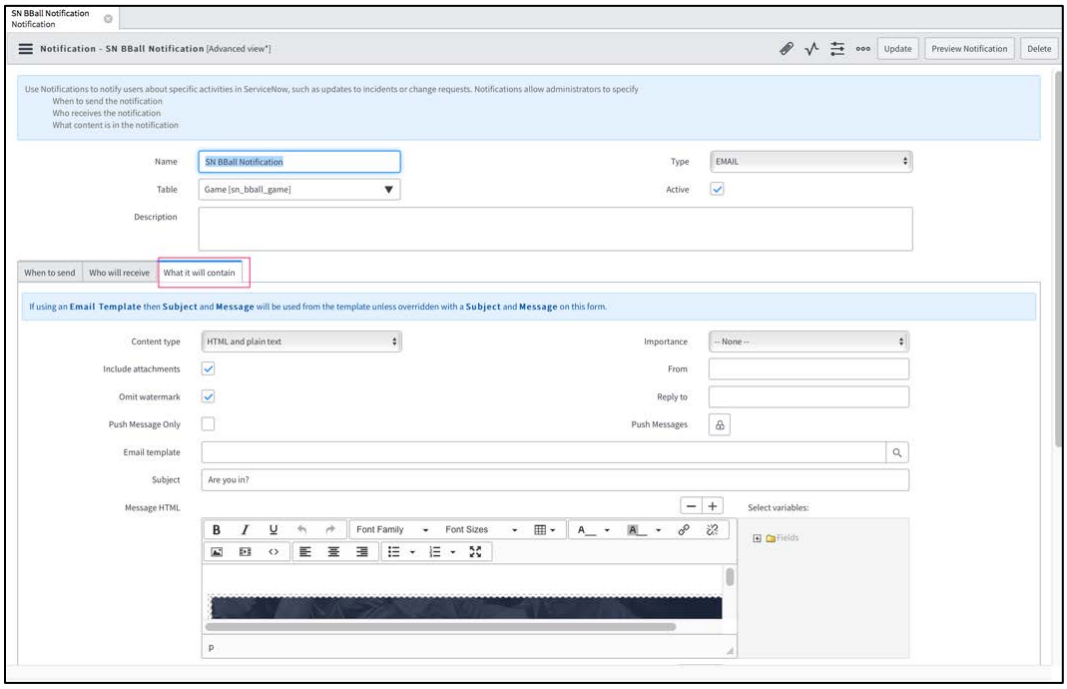


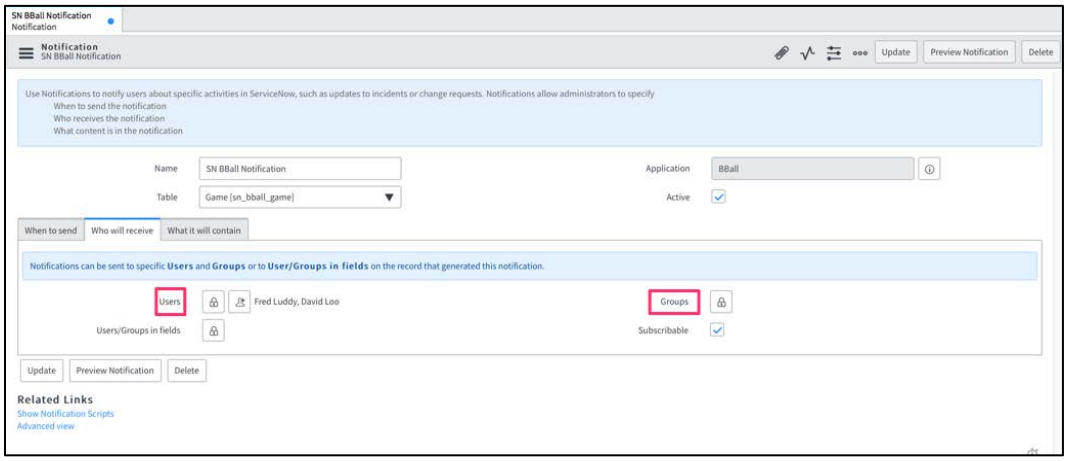
2. In the Message HTML, you want to insert/edit links on the Yes and No Text. The links point to the URL of the processor created above.

   /sn_course_add_student.do?add=true

   /sn_course_add_student.do?add=false

3. Here is the HTML (copy provided; we hate typing).





```
<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>
<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>
```