



HCL Domino Volt

HCL Domino Volt 1.0.4 Step by Step Tutorial

Marty Lechleider, Product Manager
martin.lechleider@hcl.com

Table of Contents

1.	<i>Introduction</i>	3
1.1.	Design Overview	3
1.2.	Design Palette.....	4
2.	<i>Create the course catalog app from a spreadsheet</i>	7
3.	<i>Create an App Page for viewing the catalog data</i>	8
3.1.	Adding a Data Grid	9
3.2.	Setting the App Page to Home Page	11
3.3.	Testing the App Page and Data Grid	12
3.4.	Customizing the App Page for course selection and registration	13
3.5.	Add JavaScript to the Data Grid onRowSelect event	14
3.6.	Improving the presentation of selected course information.....	16
4.	<i>Create the Request Form</i>	19
4.1.	Improving the presentation of the form items	20
4.2.	Prepopulating values in the form.....	21
4.2.1.	User and manager information	21
4.2.2.	Course information.....	23
5.	<i>Creating a registration link on the App Page</i>	25
5.1.	Adding a dynamic parameter to pass the course number.....	25
6.	<i>Add workflow to the request form</i>	27
6.1.	Dynamically assignment to the manager role via service call	29
7.	<i>Add 'My Requests' view to the home page</i>	32
7.1.	Adding a Tab Section	32
7.2.	Adding a 2nd Data Grid for 'My Requests'	33
8.	<i>Test the finished app</i>	35
9.	<i>Appendix: rowData Schema</i>	37

1. Introduction

This step by step guide helps understand how to develop Web applications using HCL Domino Volt.

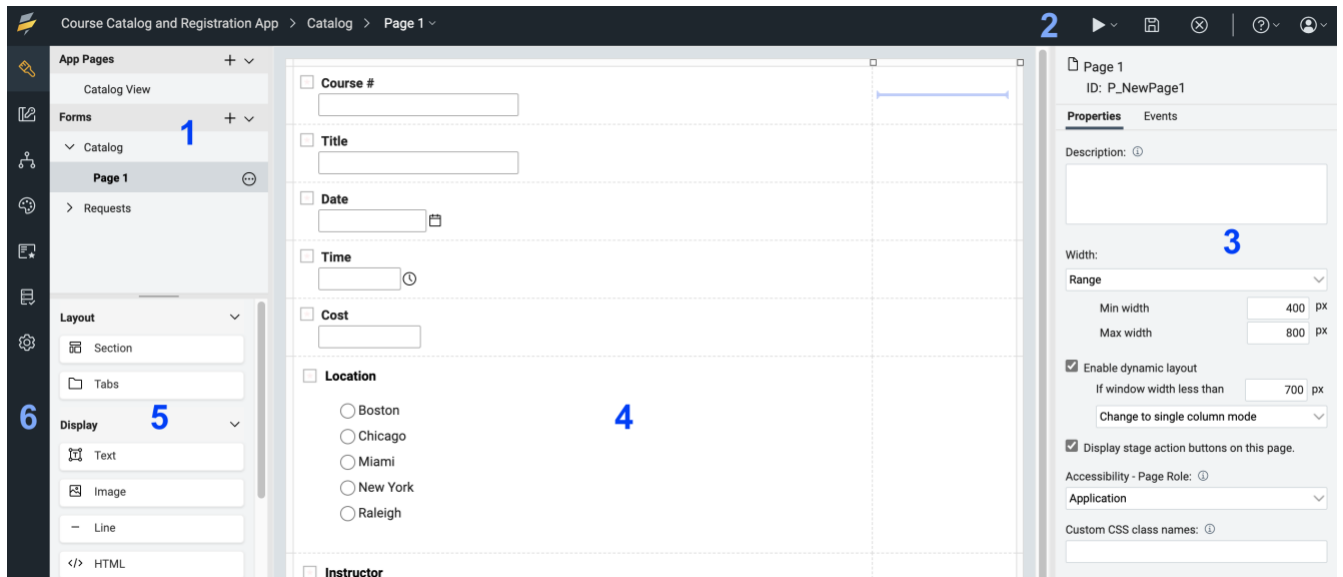
It is assumed that the reader will have access to an HCL Domino Volt server, or has downloaded the Docker Workstation, container image and related files to this step-by-step guide.

Before we delve on with each of the steps, let's deal with some basics, in an effort to help make the steps easier to comprehend.

1.1. Design Overview

This is the part of the product you will spend most of your time. It consists of



1. The top left side shows you an outline of your app which can consist of App Pages and Forms.
2. In the upper right you will find icons to save and preview your work.
3. The properties panel on the right lets you set properties for what is currently selected.
4. The design canvas in the middle is where you design forms and app pages.
5. The design palette on the lower left includes items that be can be added to you design canvas.
6. On the left side you are able to select between different design modes – design, access, workflow stages, style and more.






1.2. Design Palette

Below is a list of all the items on the palette which you can add to your app.




Layout Items

 Section	Provides a way to group items together and change the properties, rules, visibility, and visual layout of the entire group.
 Tabs	Creates a tabbed folder containing one or more sections for adding form items.



Display Items








 Text	Provides information to guide users, such as header, or instructions.
 Image	Displays images in your app, such as company logos. A list of supported file types is provided.
— Line	Inserts a line as visual element to help in app design.
</> HTML	Allows for advanced display and visual formatting in your app, with HTML markup.
 Media	Adds a media file such as .avi, .mpg, and .mp4 into your application, and creates a media object to show the file.

Action Items




 Button	Inserts a button in the app. You can set the button to navigate to another page in the app, trigger a service call, display an image, advance a workflow, or any custom event.
 Link	Places a link on the app for navigating to another page in the app, or a URL when clicked. The link opens in a new browser window or tab.
 Page Navigation	Inserts Back and Next buttons for moving users through pages in the form.

Entry Items







 Single Line Entry	Capture short pieces of information from users, such as names, or addresses.
 Multi-Line Entry	Capture longer blocks of information from users, such as descriptions of events, or feedback.

 Rich Text Entry	Capture formatted rich text from users that can include links, lists and tables
 Name Picker	Lookup and select names of user and groups from the directory
 Email	Capture e-mail addresses. The field validates that an email address format is entered.
 Number	Capture decimal and integer number such as amount, quantities and ages.
 Currency	Capture currency amounts. Displays a currency symbol, based on the form's regional settings or item's currency type.
 Website	Allows users to enter Web URLs, and checks that the provided URL is a valid format.
 Password	Capture a value that is visually indicated on the form by asterisks.

Calendar Items

 Date	Captures a date through direct entry or calendar picker. You can set a date range and modify the style of the displayed date.
 Time	Captures time values through manual entry, or selection from a time picker. You can set a time range and modify the style of the displayed time.
 Time Stamp	Capture date-time combination. Time stamps can have a range set and are useful when forms are required by a specific deadline.

Selection Items

 Check Box	Captures a single confirmation-related input. For example, "I approve" or "I confirm".
 Drop Down	Creates a drop-down menu. Users select one of the provided options from the list.
 Select One	Creates a radio button list of options. Users must select one option from the list of choices.
 Select Many	Creates a check mark list of options. Users can select multiple options from the list of choices.
 Choice Slider	Creates a visual slider with a set of choices defined.
 Numeric Slider	Creates a visual slider to gather a number. The range can include decimal places.



Survey

Creates a question, a list of options, and a range of responses for the question. Users can rank each option by choosing one of the provided responses.

Advanced Items



Table

Creates a table which captures user input. Users can add as many columns to the table as required. For example, dependent's first names, last names, and ages.



Attachment

Allows users to attach files to the app, such as scanned images and other documents.



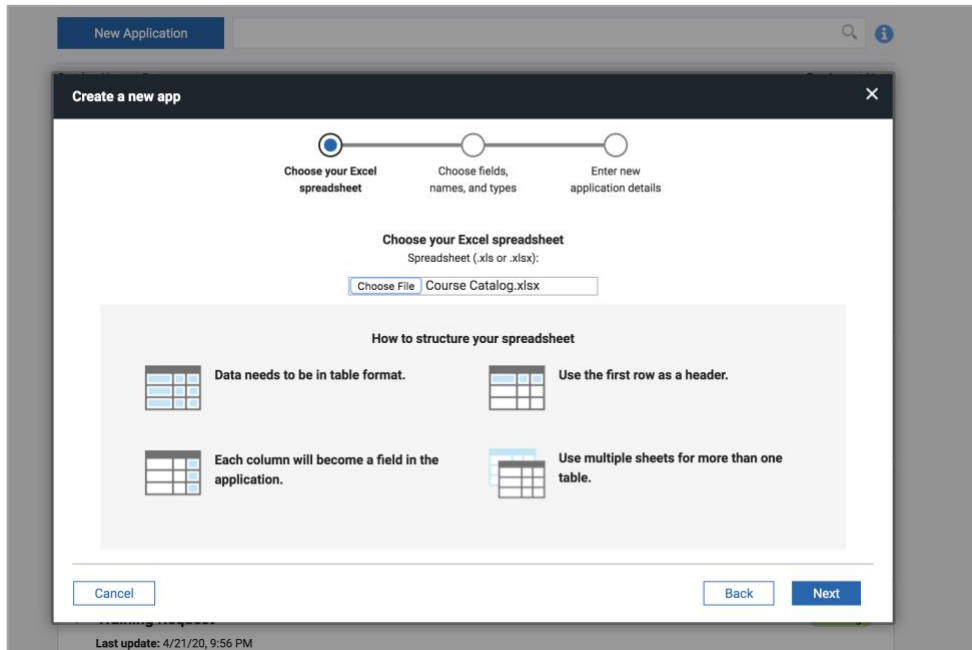
Data Grid

Creates a view of information from any form in the app. The grid can be sorted and filtered. Double clicking on an item in the grid opens to form document.

When we'll show a new control, concept, step, we'll endeavor to illustrate it. Once done, we'll assume the knowledge acquired and only provide the steps that need to be taken. This will help avoid this document being overly long.

2. Create the course catalog app from a spreadsheet

- Create a [New Application / From Spreadsheet](#) using *coursecatalog.xlsx*. Accept all the defaults.



- From the Manage page, click on [View Responses](#) for viewing the imported spreadsheet data.
- Customize the view by showing the form in right hand panel. Scroll down and select [View form in the right panel](#) in [Application viewing options](#).
- View records by clicking in them.

Responses Summary

Customize Create Filters Refresh Data Export Data Import Data Launch

Previous | Next

	Course #	Title	Date	Time	Cost	Location	Instructor
	1	Security Training	4/18/2020	9:00 AM	\$2,000.00	Chicago	Sylvia Woods
	2	New Hire Orientat...	5/23/2020	8:30 AM	\$3,000.00	Boston	Mary Jones
	3	Leadership	5/12/2020	8:30 AM	\$5,000.00	Boston	Ted Monson
	4	Advanced Domino...	4/26/2020	9:00 AM	\$1,000.00	New York	Marty Lechleider
	5	Intro to Domino	5/26/2020	8:30 AM	\$3,000.00	Chicago	Mary Jones
	6	JavaScript Basics	7/14/2020	9:00 AM	\$2,000.00	New York	Ted Monson
	7	JavaScript Advanc...	8/1/2020	9:00 AM	\$1,500.00	Boston	Marty Lechleider
	8	Node.JS	8/15/2020	9:00 AM	\$1,000.00	Miami	Sylvia Woods
	9	Styling with CSS	9/14/2020	8:30 AM	\$2,000.00	Raleigh	Marty Lechleider

Show 5 | 10 | 20 | 25 Items Previous | Next

Sheet1 - Su... Print Delete Record

Course #

Title

Date

Time

Cost

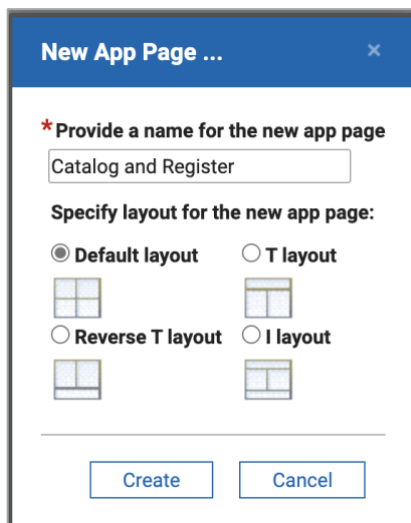
Location New York

3. Create an App Page for viewing the catalog data

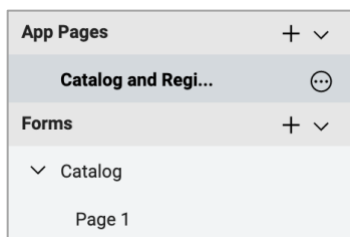
- Go back to the HCL Domino Volt management page.
- Edit the new Course Catalog application you just created.
- Add a new App Page by clicking on the + in the application outline.



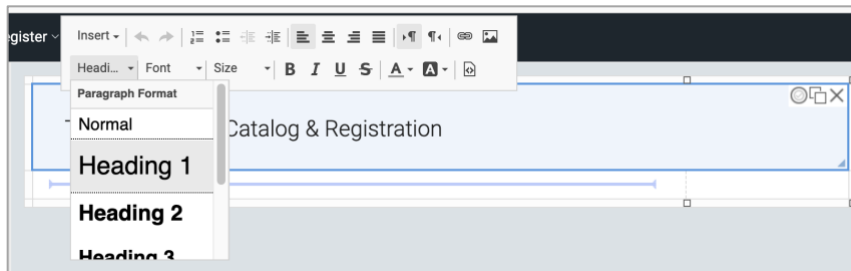
- Name the new App Page 'Catalog and Register' and accept Default Layout.



- Rename "Sheet1" to "Catalog" by clicking on the "Sheet1" text in the outline.

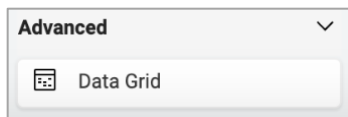


- With the App Page 'Catalog and Register' selected add a title the top of the canvas using the Text item on the palette. Hint: The Text item can be found un 'Display
- Drag the lower right handle of the Text item so it spans the entire page.
- Add 'Training Course Catalog & Registration' to the Text items and style it with Heading 1. Note: Text editing is done in place with 1.0.4

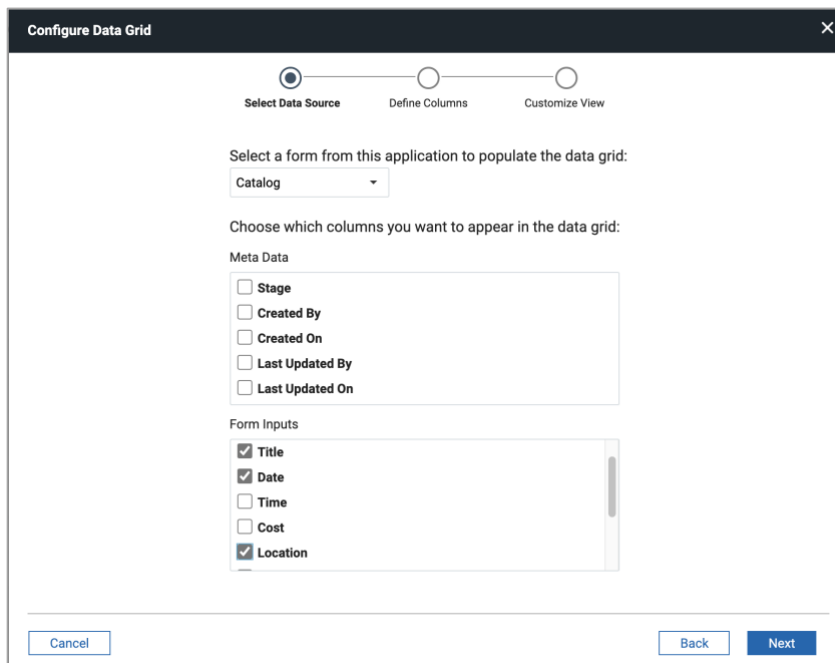


3.1. Adding a Data Grid

- Add a Data Grid item on to the canvas below the title. You will find the Data Grid under 'Advanced' at the bottom of the palette.



- Select Title, Date and Location fields to display from the Catalog Form. Note: The catalog is the only data option at this point since it is the only form in the app.



- Click 'Next' twice to skip setting a sort order and filters and then click 'Create'.
- Drag the lower left corner of the Data Grid so that it spans the entire width of the page. Note: The Data Grid is populated with dummy data in Design for preview purposes. Actual data from

the catalog is fetched and displayed when the app is deployed.

Title	Date	Location
Value 0	1/8/2021	Value 0
Value 1	2/12/2021	Value 1
Value 2	10/3/2020	Value 2
Value 3	2/20/2021	Value 3
Value 4	9/20/2020	Value 4

Rows per page: 5 1-5

- Select the Catalog Form in the outline that you created from a spreadsheet.
- Find the Title, Date and Location in the form and set the Sort Options in the properties panel to 'Both' for each of the 3 fields.

- Go back to the 'Catalog and Register' App Page by selecting it in the outline.
- Change the title of the Data Grid to 'Course List'.
- Click to unselect 'Allow user to open record' in the bottom part of the properties panel. We are doing this since we will be using the Data Grid for course selection and do not need to open and edit course catalog records.

- Click on the ellipse and then 'Properties' in the properties panel after each of the 3 columns to set 'allow users to sort'. This will let you sort the fields in the Data Grid by clicking on the column headings.

The screenshot displays the 'Training Course Catalog & Registration' application. A data grid is shown with the following data:

Title	Date	Location
Value 0	12/7/2020	Value 0
Value 1	8/5/2021	Value 1
Value 2	8/20/2021	Value 2
Value 3	8/2/2021	Value 3
Value 4	4/13/2021	Value 4

The properties panel on the right shows the 'Properties' tab for the 'Course List' data source. The 'Columns' section lists 'Title', 'Date', and 'Location'. The 'Date' column has a 'Properties' menu open, showing options to 'Move up', 'Move down', and 'Remove Column'. The 'Allow user to open record' checkbox is checked.

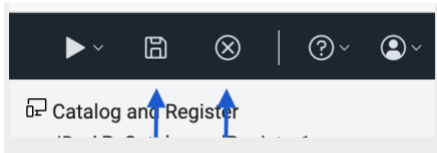
3.2. Setting the App Page to Home Page

- Click on 'Settings' (cog wheel) on the left most bar of the screen.
- On the Settings page at the bottom set the home page to the 'Catalog and Register' App Page

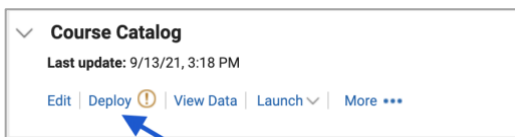
The screenshot shows the 'Settings' page for the 'Course Catalog' application. The 'General' section is selected, and the 'Home Page' is set to 'Catalog and Register'. The 'Application Language' is set to 'English' and the 'Application Regional Options' are set to 'United States'. The 'Application Name' is 'Course Catalog' and the 'Application Description' is empty. The 'Create full-text index' checkbox is unchecked.

3.3. Testing the App Page and Data Grid

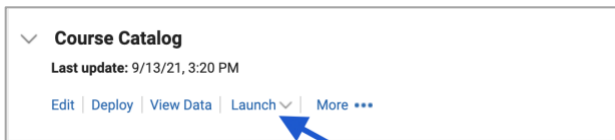
- Click the 'Save' icon and then the 'Close' icon on to top right part of your screen. Clicking the 'Close' icon will take you back to the Manage page.



- On the Manage page, click 'Deploy'. This will deploy saved changes to the runtime version of the app.



- Close the success confirmation dialog once deployed.
- Click 'Launch' under 'Course Catalog'. This will let you see the home page with data populated from the catalog form.



- Test to see if column sorting and paging work.

Training Course Catalog & Registration

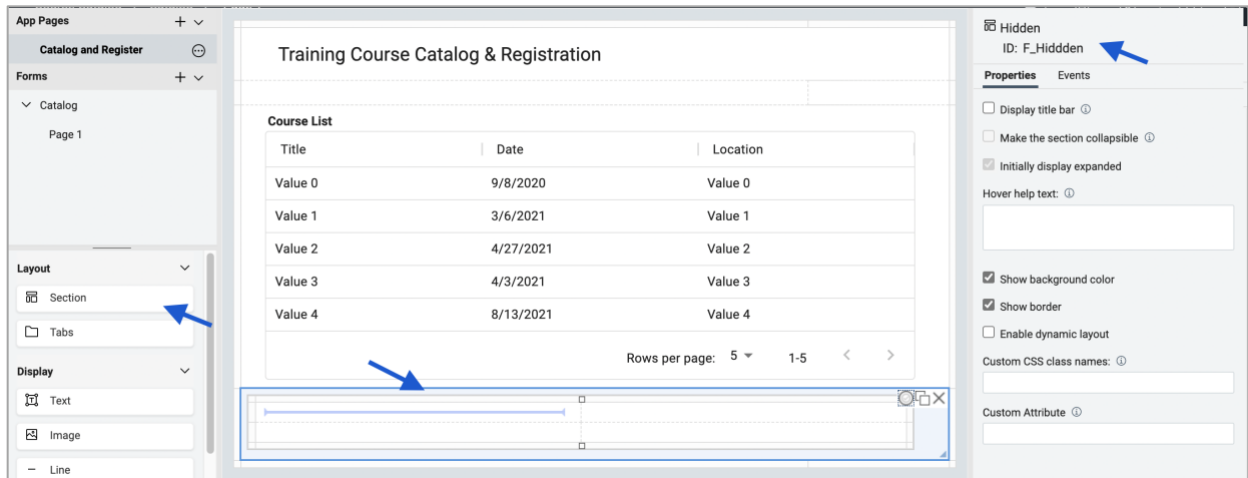
Course List

Title	Date ↑	Location
Security Training	4/18/2020	Chicago
Advanced Domino Volt	4/26/2020	New York
Leadership	5/12/2020	Boston
New Hire Orientation	5/23/2020	Boston
Intro to Domino	5/26/2020	Chicago

Rows per page: 5 1-5 < >

3.4. Customizing the App Page for course selection and registration

- Add a Section to the bottom of the 'Catalog and Register' App Page. The Section item can be found near the top of the palette under layout.
- Rename the Section to 'Hidden' and the Section ID to 'F_Hidden' in the properties panel.



- Add the follow Entry items to the section and name the IDs for each accordingly.

Item Type	Name	ID Name
Single Line Entry	Course	F_Course
Multi-Line Entry	Description	F_Description
Single Line Entry	Location	F_Location
Currency	Cost	F_Cost
Single Line Entry	Instructor	F_Instructor
Single Line Entry	Course Number	F_CourseNumber

- When done the App Page should look like below.

Training Course Catalog & Registration

Course List

Title	Date	Location
Value 0	9/8/2020	Value 0
Value 1	3/6/2021	Value 1
Value 2	4/27/2021	Value 2
Value 3	4/3/2021	Value 3
Value 4	8/13/2021	Value 4

Rows per page: 5 1-5 < >

Course

ID: F_Course

Properties Formula Events

Width: Medium

Maximum number of characters: 50

Hover help text:

Placeholder:

Change entry to uppercase

Format

Regular expression

Custom error message

Accessibility - Alternative label ID:

Custom CSS class names:

Custom Attribute

3.5. Add JavaScript to the Data Grid onRowSelect event

- Select the Data Grid and then the Event section of the properties panel. Here you will see the onRowSelect event.
- Click the onRowSelect event to open the event dialog.

Training Course Catalog & Registration

Course List

Title	Date	Location
Value 0	9/8/2020	Value 0
Value 1	3/6/2021	Value 1
Value 2	4/27/2021	Value 2
Value 3	4/3/2021	Value 3
Value 4	8/13/2021	Value 4

Rows per page: 5 1-5 < >

Course List

ID: F_DataGrid1

Properties Events

Run a formula, call a service, or execute custom JavaScript from an event.

Client Side:

- onDataChange
- onHide
- onRowDeselect
- onRowSelect**
- onShow

- Add the following JavaScript to the Data Grid onRowSelect event dialog below where it says `/* Add your own JavaScript here. */`.
- Note: All values from the catalog form are available via JavaScript even though only 3 fields are displayed in the Data Grid. This JavaScript will grab them to populate fields in the 'Hidden' section.

```

appPage.F_Course.setValue(rowData.F_Title1);
appPage.F_Cost.setValue(rowData.F_Cost1);
appPage.F_Description.setValue(rowData.F_Description1);
appPage.F_CourseNumber.setValue(rowData.F_Number1);
appPage.F_Instructor.setValue(rowData.F_Instructor1.savedValue);
appPage.F_Location.setValue(rowData.F_Location1.savedValue);

```

About: onSelect and rowData

- The onSelect event fires when a row in the Data Grid is selected.
- rowData is the variable tied to this even that represents all data for the selected row/record
- rowData includes all field and metadata data for the selected record.
- Field data is reference with rowData followed by '.' and the field ID. For example, `rowData.F_Cost1`
- Metadata is reference by rowData follow by '.' and the descriptor. For example `rowData.lastModifiedBy`
- See the appendix of this tutorial for full list of rowData descriptors.

- Click 'Save' and then 'Close'.
- From the Manage Page click 'Deploy' to update the runtime app.
- From the Manage Page click 'Launch' to test the JavaScript. If successful you should see the values in the section populate as you click different courses.

Training Course Catalog & Registration

Course List

Title	Date	Location
Security Training	4/18/2020	Chicago
New Hire Orientation	5/23/2020	Boston
Leadership	5/12/2020	Boston
Advanced Domino Volt	4/26/2020	New York
Intro to Domino	5/26/2020	Chicago

1 row selected Rows per page: 5 ▾ 1-5 < >

Course <input type="text" value="Advanced Domino Volt"/>	Description <input type="text" value="Learn how to build Leap applications that can transform your organization. Topics covered includes using"/>
Location <input type="text" value="New York"/>	Cost <input type="text" value="\$1,000.00"/>
Instructor <input type="text" value="Marty Lechleider"/>	Course Number <input type="text" value="4"/>

3.6. Improving the presentation of selected course information

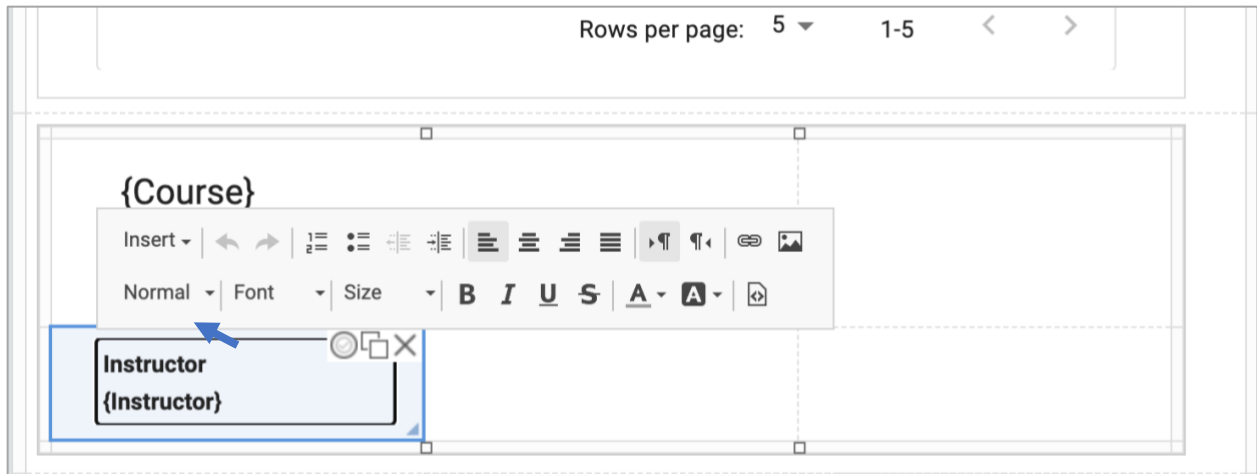
- Add a Section item below the Data Grid. The Section item can be found under Layout.
- Drag the low right handle so the Section spans the entire width of the page.
- Click the top edge of the Section so the 'row and column pop-up' appears.
- Add a new column to the Section so that there are 3.
- With the Section selected, set the properties so that the 'Show background color' and 'Show border' are turned off.

The screenshot shows the Domino Designer interface. On the left, a Data Grid titled "Training Course Catalog & Registration" is displayed. The grid has the following data:

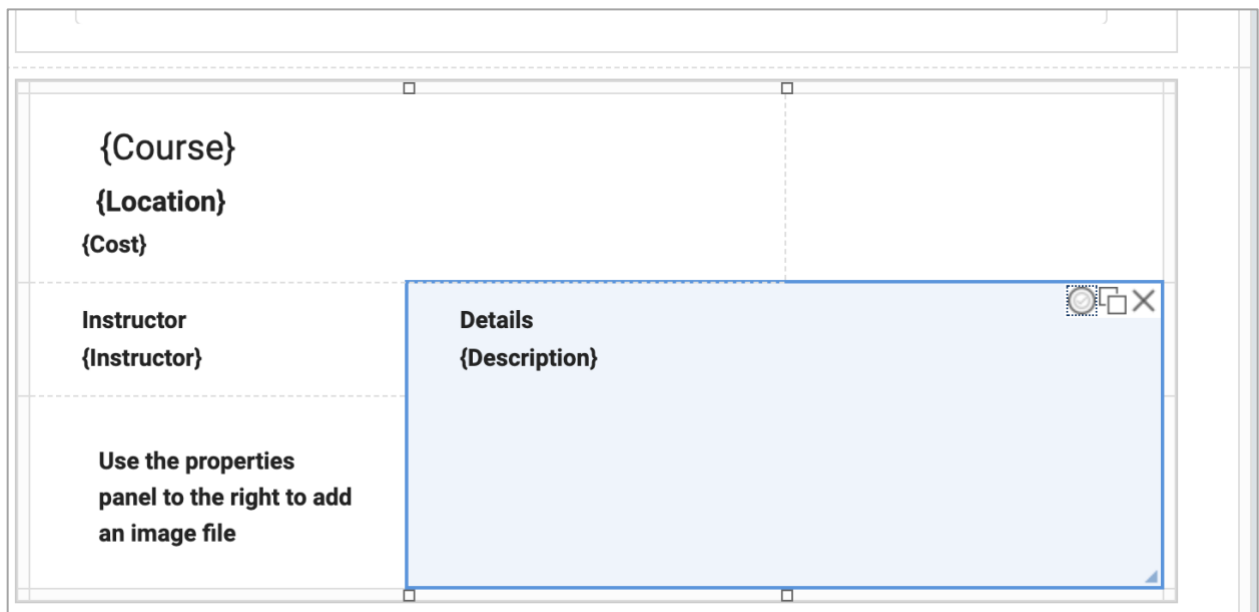
Stage ↑	Create...	Last U...	Last U...	Date	Course
Value 0	Aug 10, 2021	Value 0	Jun 17, 2021	7/14/2021	Value 0
Value 1	May 19, 2021	Value 1	Aug 16, 2021	2/11/2021	Value 1
Value 2	Feb 24, 2021	Value 2	Oct 5, 2020	2/21/2021	Value 2
Value 3	Apr 29, 2021	Value 3	Jul 1, 2021	8/27/2021	Value 3
Value 4	Aug 5, 2021	Value 4	Jun 10, 2021	1/29/2021	Value 4

Below the table, the "Rows per page" is set to 5. A Section item is positioned below the Data Grid, and its properties panel is open on the right. The "Show background color" and "Show border" checkboxes are highlighted with blue arrows, indicating they are to be turned off.

- Add a Text item into the top left of the Section. Text items will be used to echo values from the fields in the 'Hidden' section.
- Use the 'Insert' dropdown to select the echo values. Add 'Course' with Location on the next line and 'Cost' on the 3rd line. Format 'Course' to Headline1, 'Location' to Headline3 and 'Cost' to Normal.
- Grab the lower right handle and drag the Text item so it spans the first 2 columns.
- Add a second Text item below it in the first column. And add 'Instructor' to it with the Instructor echo value below it.



- Add an Image item below the Instructor. This can be found in the Display section of the palette.
- In the properties panel set the height for the Image item to 100.
- Add a Text item to the right of Instructor.
- Set enter 'Details'. Below details add the echo field value for Description. Make the word 'Details' bold.
- Drag Description so it spans the to the right and down a row.



- Select the Data Grid and add the follow line of JavaScript to the onRowSelect event. This line will get the picture URL from the Catalog and set it to the Image item.

```
appPage.F_Image1.setURL(rowData.F_Picture1);
```

- Add the following line of JavaScript to the `onDataChange` event of the Data Grid. This will prime the display by setting the selection to the first row when the app is opened.

```
appPage.F_DataGrid1.selectFirstRow();
```

- Select the section 'Hidden' and add the following line of JavaScript to the `onShow` event. This will hide the section at runtime.

```
apItem.setVisible(false);
```

- Save your work, close, deploy and then launch the app to preview the updates.
- Click the column heading to sort and page icons. Notice how the first row is selected when the data changes.

Training Course Catalog & Registration

Course List


Title	Date	Location
Security Training	4/18/2020	Chicago
New Hire Orientation	5/23/2020	Boston
Leadership	5/12/2020	Boston
Advanced Domino Volt	4/26/2020	New York
Intro to Domino	5/26/2020	Chicago

1 row selected Rows per page: 5 ▾ 1-5 < >

Security Training

Chicago
\$2,000.00

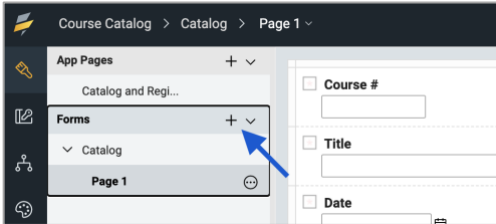
Instructor
Sylvia Woods



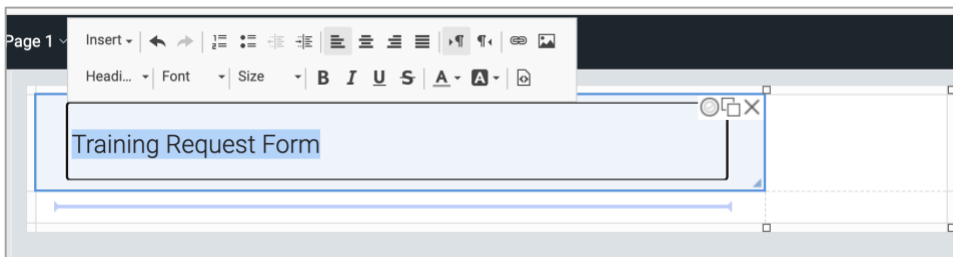
Details
GLBA, PCI, HIPAA, SarBox) normally require formal ISAT for all employees, usually once or twice a year. Many Small and Medium Enterprises (SME's) do not require ISAT for regulatory compliance, but train their employees to prevent a cyberheist. Internet Security Awareness Training at this point in time is usually provided via online courses. ISAT is a subset of general security awareness Training.

4. Create the Request Form

- From the manage page click 'edit' to open the app back up in design.
- Add a new Form by clicking on the + in the application outline.



- Name the new form 'Requests' and accept Default Layout.
- Add a Text item to the top of the form and enter 'Training Request Form'.
- Set the style to Heading 1



- Add a Section below the title. Drag the lower right corner so it spans 2 columns.
- Add the following Entry items to the section.
- Name each item and change its ID to those listed in the table. You can do this in the properties panel when the item is selected.

Item	Name	ID
Single Line	Name	F_Name
Number	Course Number	F_CourseNumber
Single Line	Manager Name	F_ManagerName
Email	Email	F_Email
Email	Manager Email	F_MgrEmail
Date	Date	F_Date1
Single Line	Course	F_Course
Single Line	Location	F_Location
Currency	Cost	F_Cost

- When you are finished your form should look like this.

The screenshot shows the HCL Domino Volt form editor interface. On the left, there is a sidebar with 'App Pages' and 'Forms' sections. The main area displays a 'Training Request Form' with several input fields: Name, Course Number, Manager Name, Email, Manager Email, Date, Course, Location, and Cost. The 'Cost' field is highlighted with a blue border, and its properties panel is open on the right. The properties panel for 'Cost' shows 'ID: F_Cost', 'Properties' tab selected, 'Width: Short', 'Currency type: US Dollar', 'Placeholder', and 'Range' with 'Minimum' and 'Maximum' fields.

4.1. Improving the presentation of the form items

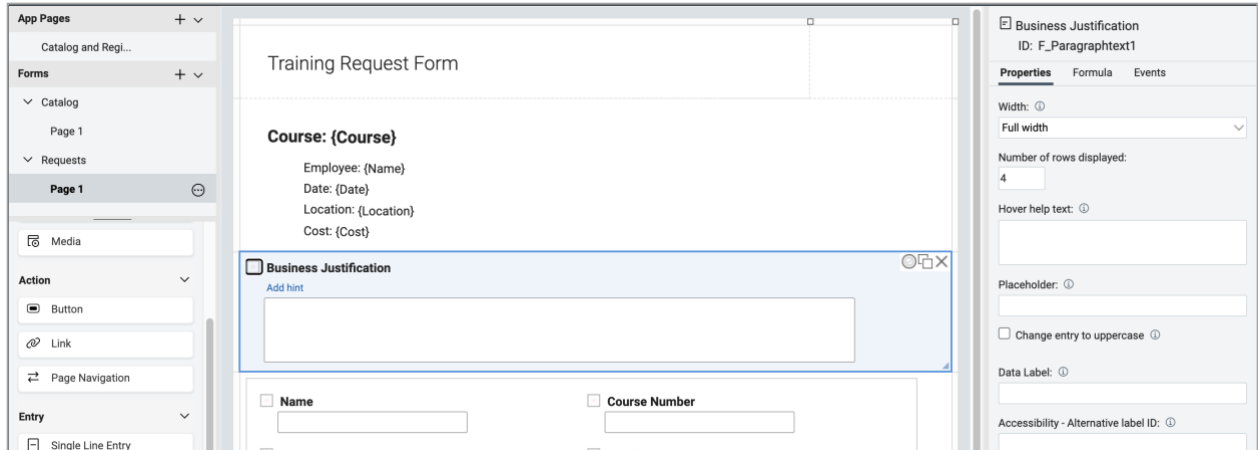
- Add a Text item below the title and above the section.
- Enter the following text, each on its own line into the item. Course, Employee, Date, Location, Cost.
- Use the Insert dropdown to add the echo field values to the right of each.
- Set the style of Course to Headline 2
- Indent the rest of the text using the indent icon on the editor toolbar

The screenshot shows a close-up of the text editor in the HCL Domino Volt form editor. The toolbar at the top includes an 'Insert' dropdown, navigation arrows, list and table icons, bold, italic, underline, strikethrough, text color, background color, and link icons. Below the toolbar, the text area contains the following text:

Course: {Course}
 Employee: {Name}
 Date: {Date}
 Location: {Location}
 Cost: {Cost}

Below the text area, the 'Name' and 'Course Number' input fields are visible, with the 'Name' field having a blue border.

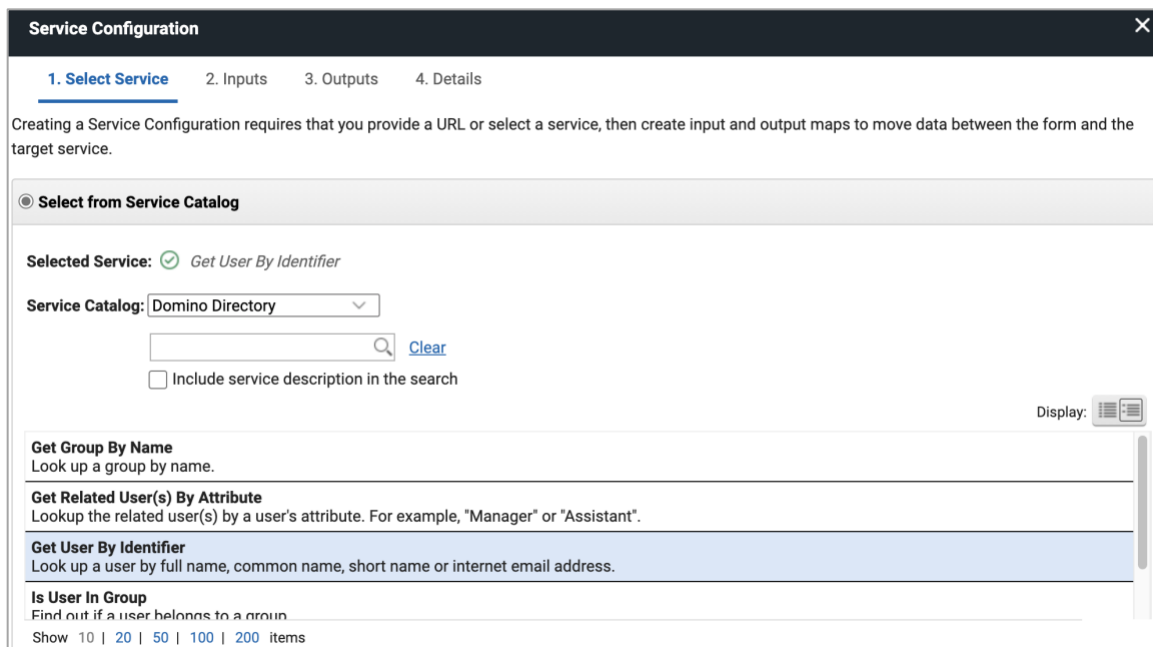
- Add a Multi-Line item below the Text and name it 'Business Justification'.
- Set the width in the properties panel to full
- Set rows displayed in the in the properties panel to 2.



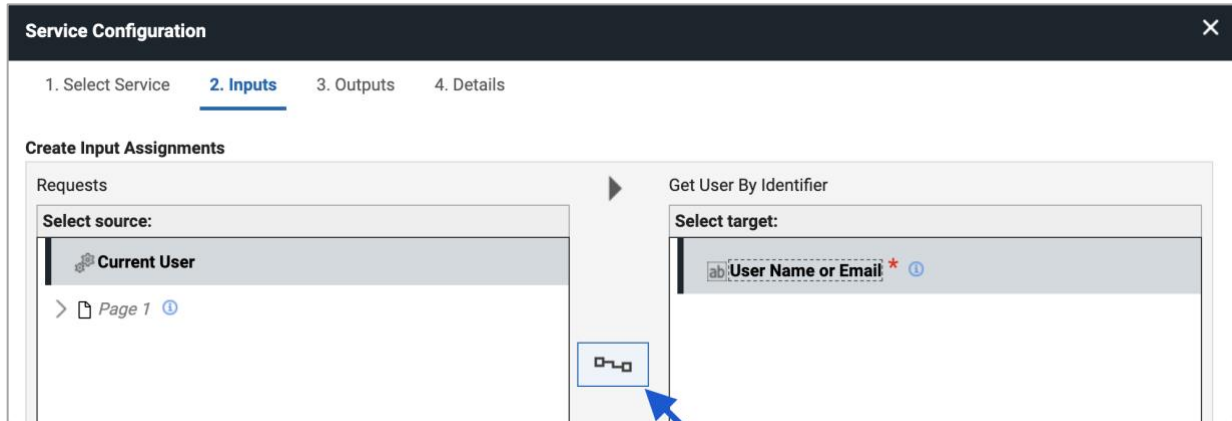
4.2. Prepopulating values in the form

4.2.1. User and manager information

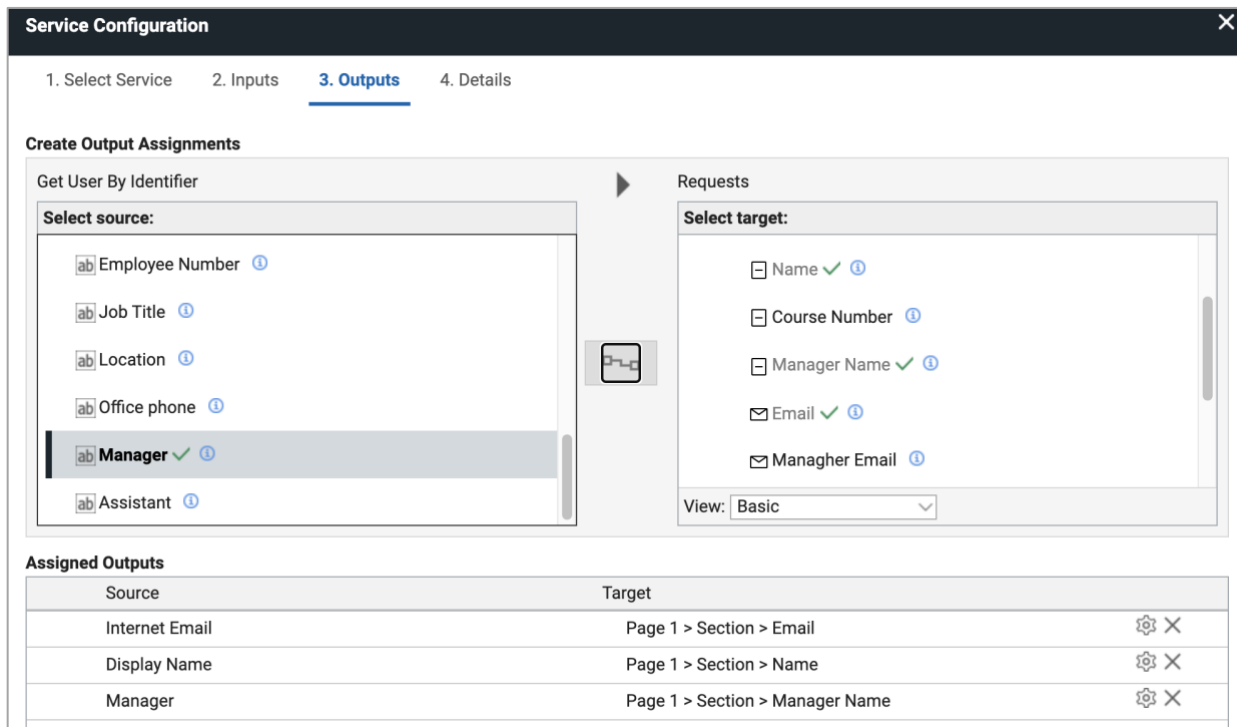
- Click on the Requests form in the outline so that the properties for the form are displayed in the properties panel.
- In the properties panel for the Requests form, click on Events and then on New.
- Note: The onNew event fires when a new request is created.
- In the onNew event dialog select 'Call a Service' and then click 'Add/Edit Service Configuration'. This will take you to the service configuration dialog.
- In the service configuration dialog select 'Select from Service Catalog'.
- Select 'Domino Directory' under the service catalog dropdown.
- Select the 'Get User by Identifier' service.



- Click 'Next' or '2. Inputs' and then map 'Current User' on the left (source) to 'Email or User Name' on the right (target)
- To map the items, select each one and then click the 'Assign' icon on the middle.



- Click 'Next' or '3. Outputs'
- Map 'Internet Email' on the left (source) to 'Email' on the right (target). Also map 'Display Name' to 'Name' and 'Manager' to 'Manager Name'.
- Note: Mapping 'Manager' assumes that manager information is stored in the Domino directory. If that is not the case, you can skip doing so.



- Click 'Ok' twice to exit out of the dialogs.

4.2.2. Course information

- Click on the Request form onNew event again and enter the following JavaScript.

```
BO.F_CourseNumber.setValue(app.getUrlParameter('CourseNumber'));
```

- Note: Later in this tutorial you will add a link to the 'Catalog and Request' App Page that will open the 'Requests' form with a parameter (CourseNumber) that identifies the course the user intends to register for. The JavaScript above gets this parameter and puts it in the 'Course Number' field in the form.
- Click 'OK' to exit the dialog.
- Click on the 'Course Number' field in the section.
- Click on 'Events' in the properties panel for 'Course Number' and then on the 'onItemChange' event.
- Click on 'Call Service' and then 'Add/Edit Service Configuration' in the dialog.
- Select 'Current Application' from the service catalog dropdown and then 'Course Catalog / Catalog / Retrieve'.
- Click 'Next' or '2. Inputs'.
- Map 'Course Number' on the left (source) to 'Retrieve by (F_Number1) "Course #"'.

Service Configuration

1. Select Service **2. Inputs** 3. Outputs 4. Details

Create Input Assignments

Requests

Select source:

- Name
- Course Number** ✓
- Manager Name
- Email
- Managher Email

View: Basic

Course Catalog / Catalog / Retrieve

Select target:

- Retrieve by Unique Identifier
- Retrieve by (F_Number1) "Course #"** ✓
- Retrieve by (F_Title1) "Title"
- Retrieve by (F_Date1) "Date"
- Retrieve by (F_Time1) "Time"

View: Basic

Assigned Inputs

Source	Target
Page 1 > Section > Course Number	Retrieve by (F_Number1) "Course #"

- Click 'Next' or '3. Outputs'
- Map 'Title' on the left (source) to 'Course' on the right (target).

- Do the same for Date, Location and Cost by mapping them to their counterparts in the Requests form.

Service Configuration

1. Select Service 2. Inputs **3. Outputs** 4. Details

Create Output Assignments

Course Catalog / Catalog / Retrieve

Select source:

- Record Unique Identifier
- Line ID
- Stage ID
- Last Updated
- Updater ID
- Updater Name

Requests

Select target:

- Page 1
 - Business Justification
 - Section

View: Basic

Assigned Outputs

Source	Target	
Result (F_Title1) "Title"	Page 1 > Section > Course > Value	⚙️ ✕
Result (F_Date1) "Date"	Page 1 > Section > Date > Value	⚙️ ✕
Result (F_Location1) "Location"	Page 1 > Section > Location > Value	⚙️ ✕
Result (F_Cost1) "Cost"	Page 1 > Section > Cost > Value	⚙️ ✕

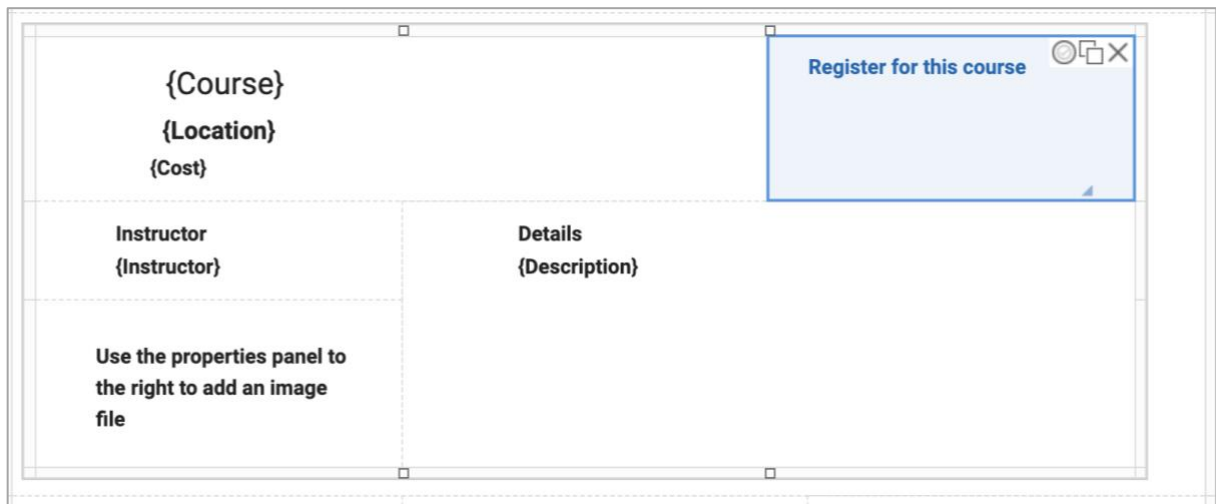
- Click 'OK' twice to exit out of the dialogs.
- Select the section and then click 'Events' in the properties panel.
- Click the onShow event and then enter the line of JavaScript below. This will hide the section at runtime.

```
item.setVisible(false);
```

- Click the 'Save' icon and then 'Close' icon and then 'Deploy' the updates.

5. Creating a registration link on the App Page

- Go back in to 'Edit' the app from the Manage page.
- Select the 'Catalog and Register' App Page in the outline.
- Add a Link item to the right of the Text item in the Section below the Data Grid. The Link item can be found under Action.
- Name the Link 'Register for this course'.



5.1. Adding a dynamic parameter to pass the course number

- Select the Link onClick event in the properties panel under Events.
- Add the following JavaScript. This will dynamically set the Link value to include the course number as a parameter to be passed to the request form.

```
var CourseNumber = appPage.F_CourseNumber.getValue();
var url = app.getFormLaunchURL("F_Requests1") + "&CourseNumber="
+ CourseNumber;
appPage.F_StaticWebLink1.setLinkValue(url);
```

- Click 'OK' to exit out of the event dialog
- Click the 'Save' icon and then 'Close' icon and then 'Deploy' the updates.
- Launch the app to test the 'Register for this course' link. It should open the request form and prepopulate it with the course and user information.
- The link should open the form and pass it the course number. The course number field in the form then should look up the course info and populate the fields.

Training Request Form

Course: Advanced Domino Volt

Employee: Admin

Date: 4/26/2020

Location: New York

Cost: \$1,000.00

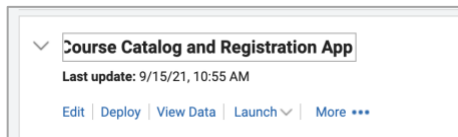
Business Justification

Submit

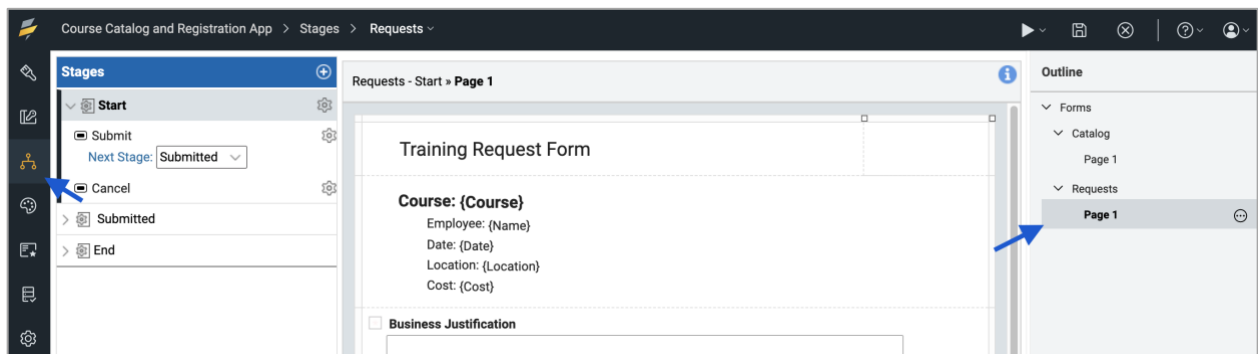
Cancel

6. Add workflow to the request form

- While on the Manage page, click on the name of the app 'Course Catalog' and change it to 'Course Catalog and Registration'. This is more descriptive given its modification since being initially created from a spreadsheet.



- Edit the app and select the workflow design view icon (stages) in the left and then select the Requests form on the right.



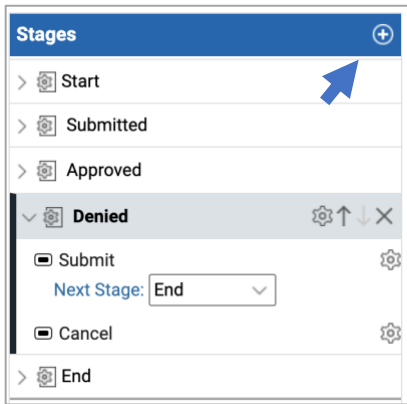
- Delete the Cancel button for the Start stage. You can do this by selecting 'Start' clicking the 'X' next to 'Cancel' button at the bottom of the screen.
- Click the cogwheel icon next to Submit button for the Start stage. This will bring up the action properties dialog.
- Scroll down in the dialog to the field 'Action Completion Redirect URL'.
- Enter the URL for the home page that you saved from an earlier step. Adding this will make the app return to the home page when a request is submitted. Only enter the portion of the URL that starts with /volt-apps/. This way the redirect will work on any domain. For example:

/volt-apps/secure/org/app/90efa93d-28d4-481d-8125-0f92d0e50d12/main/index.html#AP_CatalogandRegister1

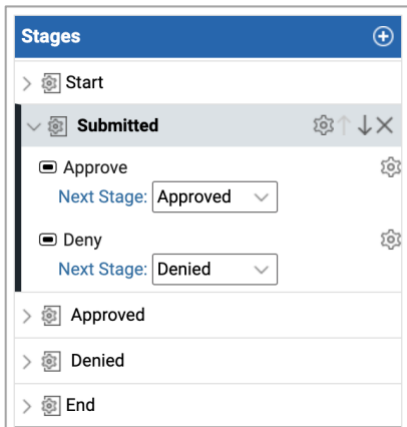
Action Completion Redirect URL (optional):

`/volt-apps/secure/org/app/90efa93d-28d4-481d-8125-0f92d0e50d12/main/index.html#AP_CatalogandRegister1`

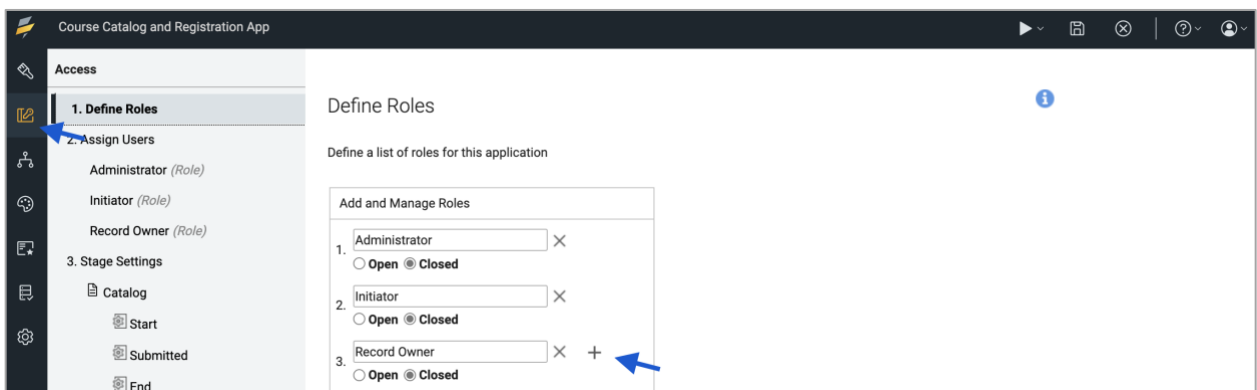
- Select the 'Submitted' stage and then click the + to add 2 additional stages after it.
- Name the stages 'Approved' and 'Denied'.



- Select the 'Submitted' stage again.
- Delete the 'Cancel' button for the 'Submitted' stage.
- Rename 'Update' to 'Approve' and select 'Approved' as the next stage.
- Use the + add icon on the bottom of the screen to add another submit button to the 'Submitted' stage.
- Name the new submit button to 'Deny' and set the next stage to 'Denied'.



- Select the access view on the left and click the + to add a new role.



- Name the new role 'Manager'
- Set the Manager role to 'Open'. This will allow us to dynamically set the membership of this role base on who is filing out the request.

4. × +
 Open Closed

- Select the 'Submitted' stage under the 'Requests' form on the left. This brings up the permission for this stage in the request form.
- Make sure the manager role has 'Read' and 'Update' privileges so that managers can review and approve requests
- Make sure the record owner role can only 'Read' in this stage. This way those making request can see the status of their request and the form but cannot update it.

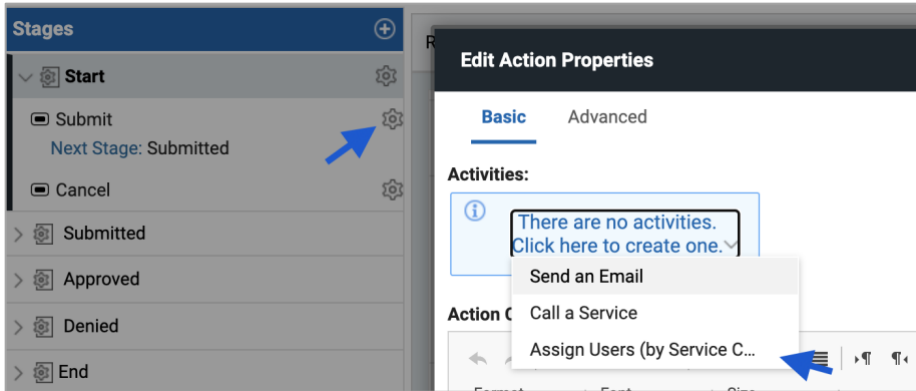
The screenshot shows the 'Requests - Submitted' stage configuration. On the left, the 'Submitted' stage is selected under the 'Requests' form. The main area displays a table of permissions for various roles.

Roles	Create	Read	Update	Delete
Administrator	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Initiator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Record Owner	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manager	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

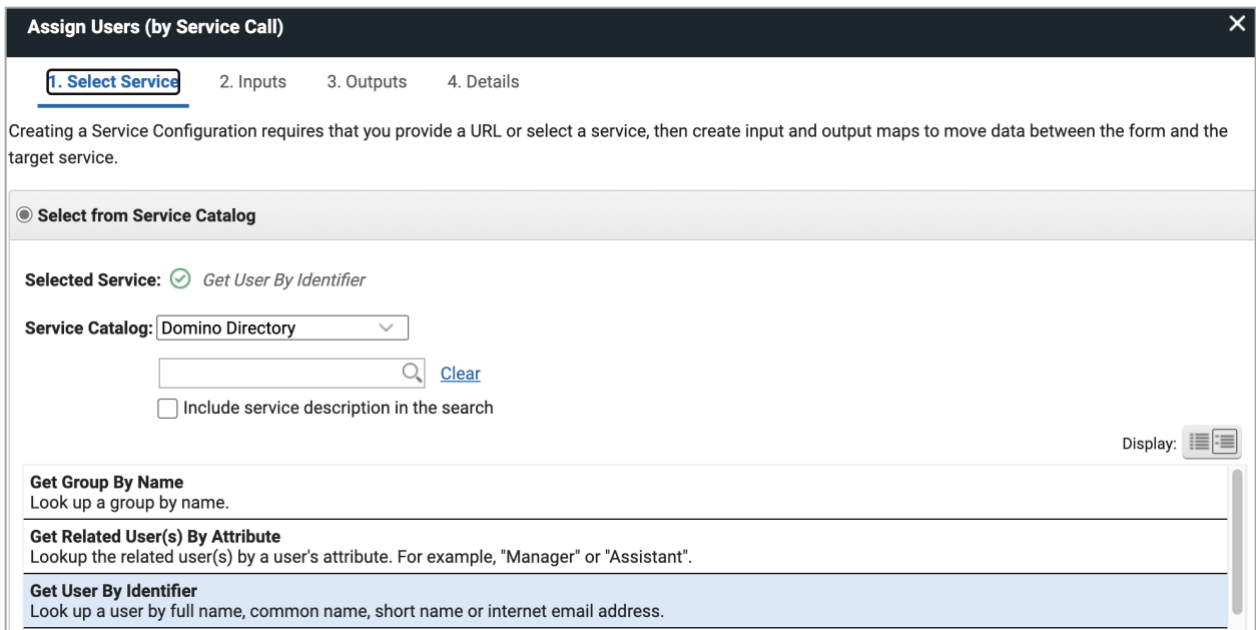
6.1. Dynamically assignment to the manager role via service call

This part of the tutorial assumes that manager information is stored in the Domino directory. If it is not, then services call can be made to other sources to determine who the requestor's manager is.

- Go back to the workflow design view (stages).
- Select the 'Start' stage.
- Click on the cogwheel icon next to the 'Submit' button to bring up its action properties.
- Click the activities dropdown and select 'Assign User by Service Call'.



- Select 'Domino Directory' from the service catalog dropdown and then 'Select Get User Identifier'



- Click 'Next' or '2. Inputs'.
- On inputs map the 'Current User' to 'User Name or Email'.
- Click 'Next' or '3. Outputs'.
- On outputs map 'Manager' to 'Members'

Assign Users (by Service Call)

1. Select Service 2. Inputs **3. Outputs** 4. Details

Create Output Assignments

Get User By Identifier

Select source:

- ab Employee Number ⓘ
- ab Job Title ⓘ
- ab Location ⓘ
- ab Office phone ⓘ
- ab Manager ✓ ⓘ
- ab Assistant ⓘ

Requests

Select target:

- ▼ Manager
- Members ✓

Assigned Outputs

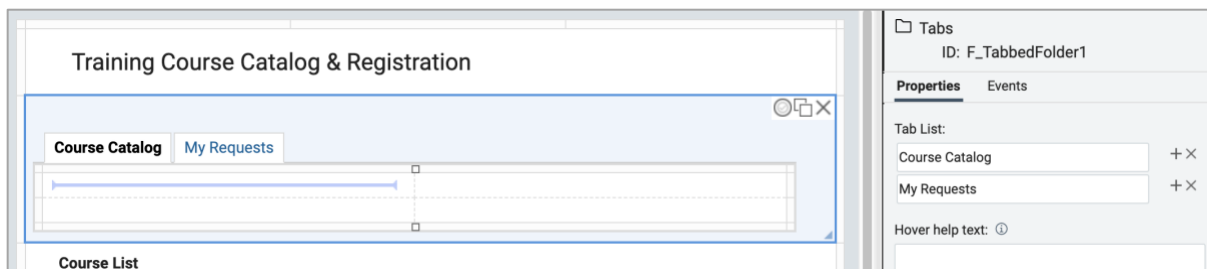
Source	Target
Manager	Role: Manager > Members

- Click 'OK' twice to exit the dialogs.
- Click the save icon and close icon and then deploy the app updates.

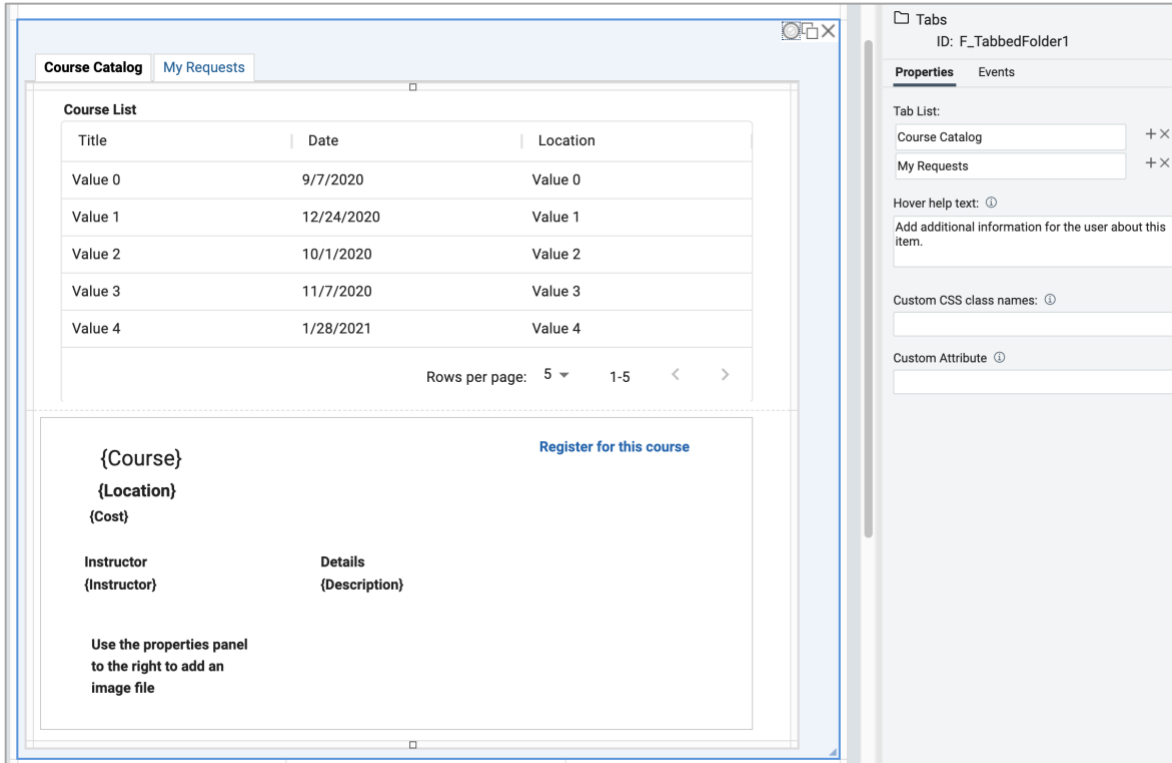
7. Add 'My Requests' view to the home page

7.1. Adding a Tab Section

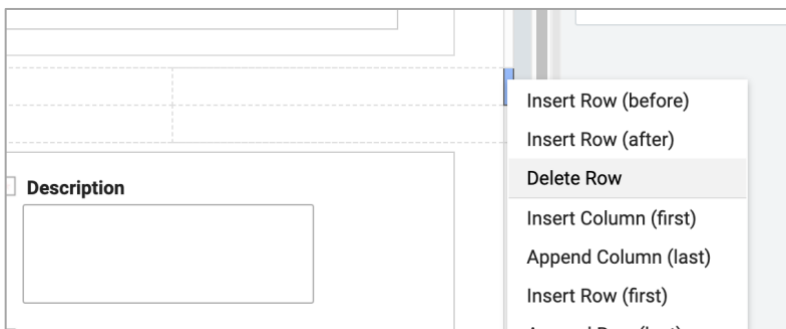
- Note: The Tabs section will let the user click between sections to see the catalog or my requests view.
- Edit the app and select the App Page.
- Add a Tabs item to the top of the page under the title. The Tabs item can be found under layout.
- Drag the lower right handle so it spans the width of the page.
- In the properties panel for the Tab item under tabs list, rename the first tab to 'Course Catalog'.
- Add a second tab and name it 'My Requests'



- Select the 'Course Catalog' tab on the canvas.
- Select the 'Course List' Data Grid and drag and drop it into the 'Course Catalog' tab section.
- Drag the lower right handle so the grid spans the width of the tab section.
- Drag the Section with the echo text, image and registration link into the 'Course Catalog' tab section below the Data Grid.
- Drag the lower right handle so the grid spans the width of the tab section.
- The 'Course Catalog' tab section should now look like below.

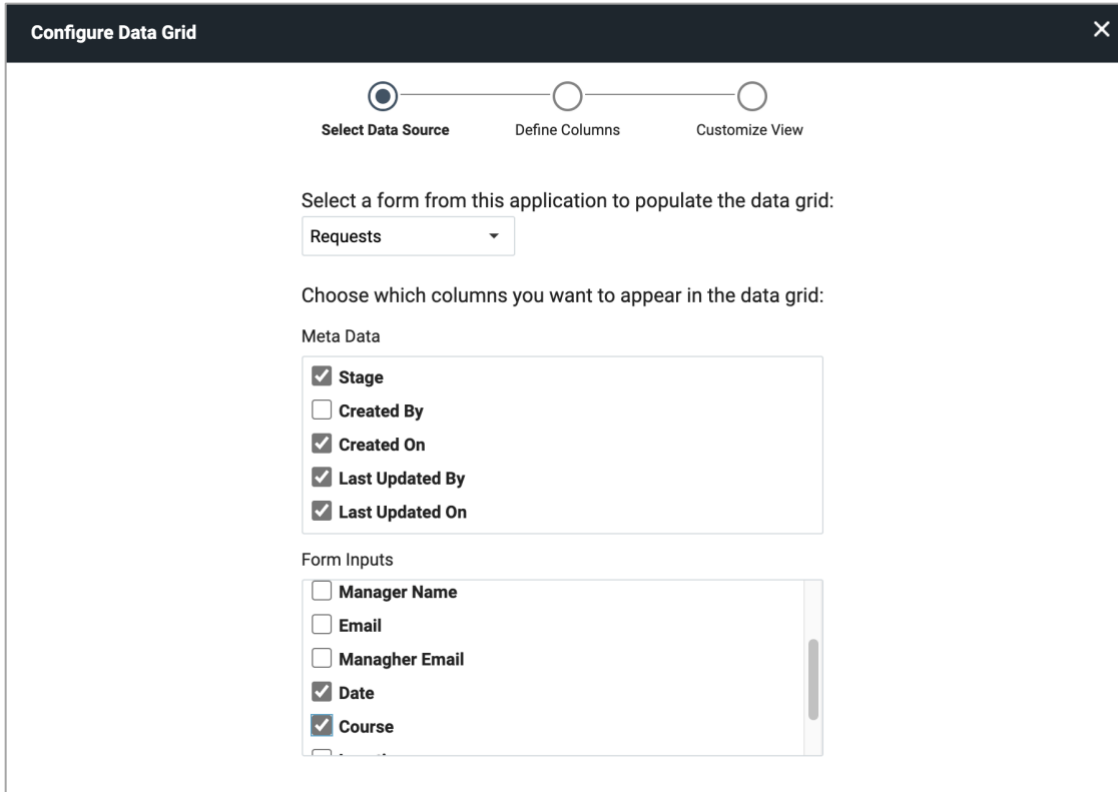


- Delete any empty rows left behind after moving things around by clicking on the row edge to reveal to 'row and column' pop-up

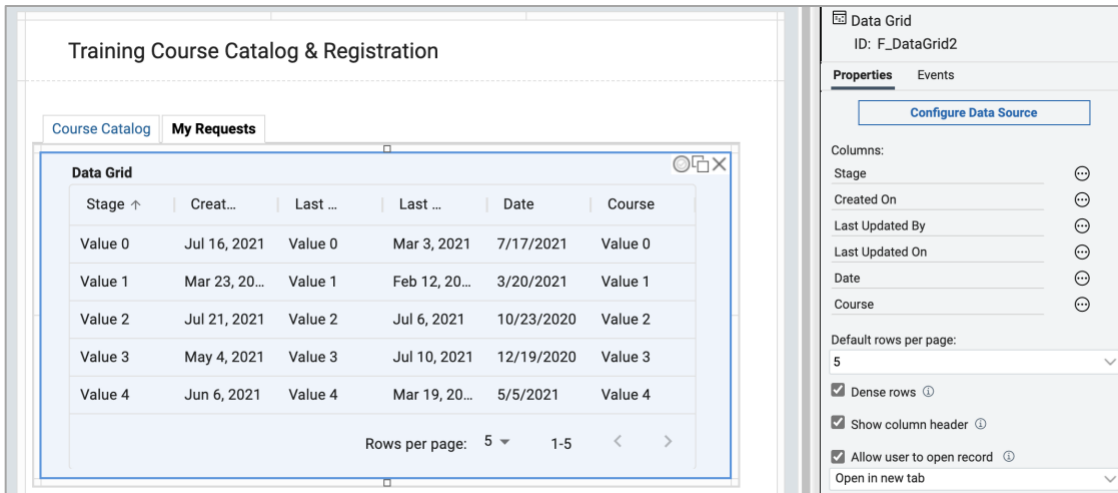


7.2. Adding a 2nd Data Grid for 'My Requests'

- Select the 'My Requests' tab.
- Add a new Data Grid item into the 'My Requests' tab section.
- Select the 'Requests' form as the form to populate the grid.
- Select the following columns to appear: Stage, Created On, Last Updated by, Last Updated On, Date and Course.



- Click 'Next' and then 'Create'.
- Drag the lower right handle so that the Data Grid spans the width of the Tab section.



- Click 'Save' and 'Close' and 'Deploy' the updates.

8. Test the finished app

- Launch the app, select a course and click 'Register for this course'.
- Add a 'Business Justification'.
- Click 'Submit'
- Click on 'My Requests' in the Tab folder.
- Your request should appear showing it is in the 'Submitted' stage.

Training Course Catalog & Registration

Course Catalog My Requests

Course List

Title	Date	Location
Security Training	4/18/2020	Chicago
New Hire Orientation	5/23/2020	Boston
Leadership	5/12/2020	Boston
Advanced Domino Volt	4/26/2020	New York
Intro to Domino	5/26/2020	Chicago


1 row selected Rows per page: 5 ▾ 1-5 < >

Advanced Domino Volt

New York

\$1,000.00

Instructor
Marty Lechleider



Details
Learn how to build Leap applications that can transform your organization. Topics covered includes using service calls, javascript and events, reporting, using css and more. The session will also cover on integration of Leap applications into DX.

[Register for this course](#)

Training Request Form

Course: Advanced Domino Volt
Employee: Admin
Date: 4/26/2020
Location: New York
Cost: \$1,000.00

Business Justification

This is my business justification.

Submit

Training Course Catalog & Registration

Course Catalog**My Requests**

Data Grid

Stage ↑	Create...	Last U...	Last U...	Date	Course
Submitted	Sep 16, 2021	Admin/Renovation	Sep 16, 2021	4/26/2020	Advanced Dom...

Rows per page: 5 ▾ 1-1 < >

9. Appendix: rowData Schema

The JSON Schema below indicates how meta data and field data can be referenced in the Data Grid onRowSelect event.

For example:

- `rowData.lastModified` is how you would get the last modified date and time for the selected record.
- `rowData.F_SingleLine1` is how you would get the value of the field `F_SingleLine1` for the selected record.

```
{
  editUrl: "https://...",
  viewUrl: "https://...",
  printUrl: "https://...",
  uid: "f308d85b-9bd6-449f-87b7-b09ea27fa23c",
  id: "F054B58209264E008825872700748692",
  stageId: "ST_Submitted",
  stageTitle: "Submitted",
  lastModified: Date obj,
  lastModifiedBy: {
    displayName: "Daisy Dixon/Acme",
    email: "Daisy Dixon/Acme",
    login: "CN=Daisy Dixon/O=Acme"
  }
},
created: Date obj,
createdBy: {
  displayName: "Daisy Dixon/Acme",
  email: "Daisy Dixon/Acme",
  login: "CN=Daisy Dixon/O=Acme"
},
F_SingleLine1: "hello world",
F_EmailAddress1: "test@example.com",
F_Number1: 56.89,
F_Currency1: 12.34,
F_CheckBox1: true,
F_RichText1: "<p>this is some <span style=\"color:#2980b9\"><span
style=\"font-size:18px\"><strong>rich </strong></span></span>text</p>",
F_DropDown1: {savedValue: "opt1", displayValue: "Option 1"},
F_SelectMany1: [
  {savedValue: "opt1", displayValue: "Option 1"},
  {savedValue: "opt2", displayValue: "Option 2"}
],
F_Attachment1: {
  uid: "abc123",
  fileName: "MyFile.png",
  viewUrl: "https://...",
  downloadUrl: "https://..."
},
F_Name1 {
  displayName: "Doug D'Antonio",
  id: "CN=Doug D'Antonio/O=Acme",
```

```
        email: "Doug D'Antonio/Acme",  
        type: "user"  
    }  
};
```