Creating and Downloading an Employee Resume

Table of Contents

knowledge@outsystems.com

www.outsystems.com

Outline

In this tutorial, you will continue to extend the Employee Directory application, this time implementing a resource that will allow us to create a resume in a PDF file and download it.

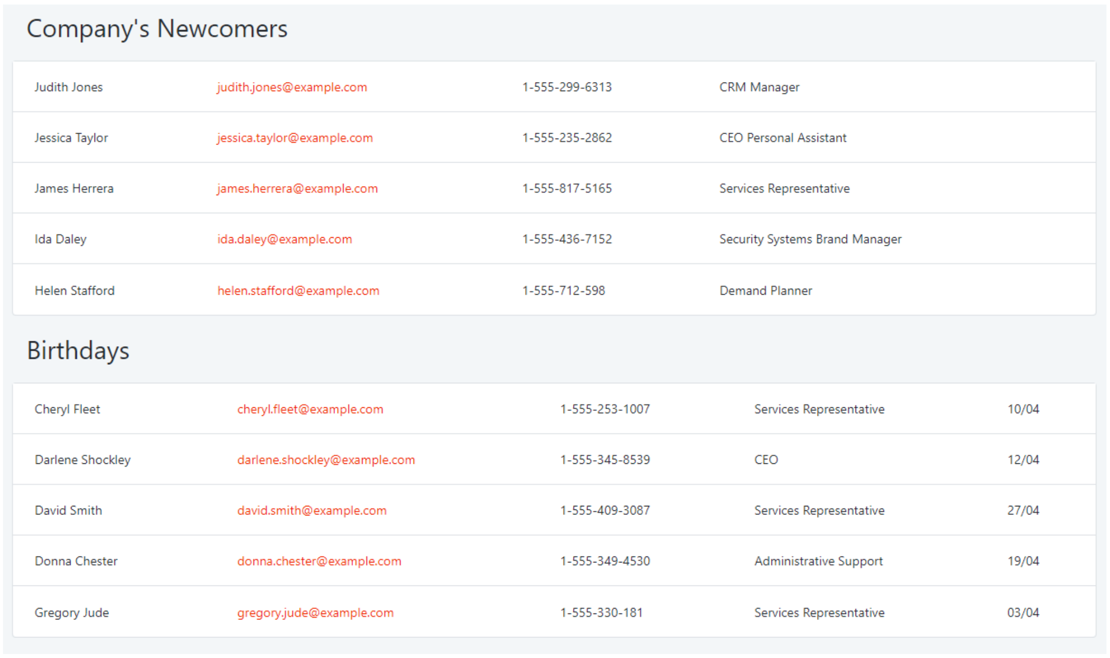
To implement this functionality we want to follow those requirements:

- Generate a resume based on the user's profile information.
- The resume should have the employee's personal information such as name, biography, and email.
- The resume should also include the projects led by the employee.

Scenario

The Employee Directory app at this point should have five Screens:

- The **Dashboard** Screen, which has a list of the newest five newcomers, and a list of the employee's birthdays of the current month. This is the current default Screen of the application.



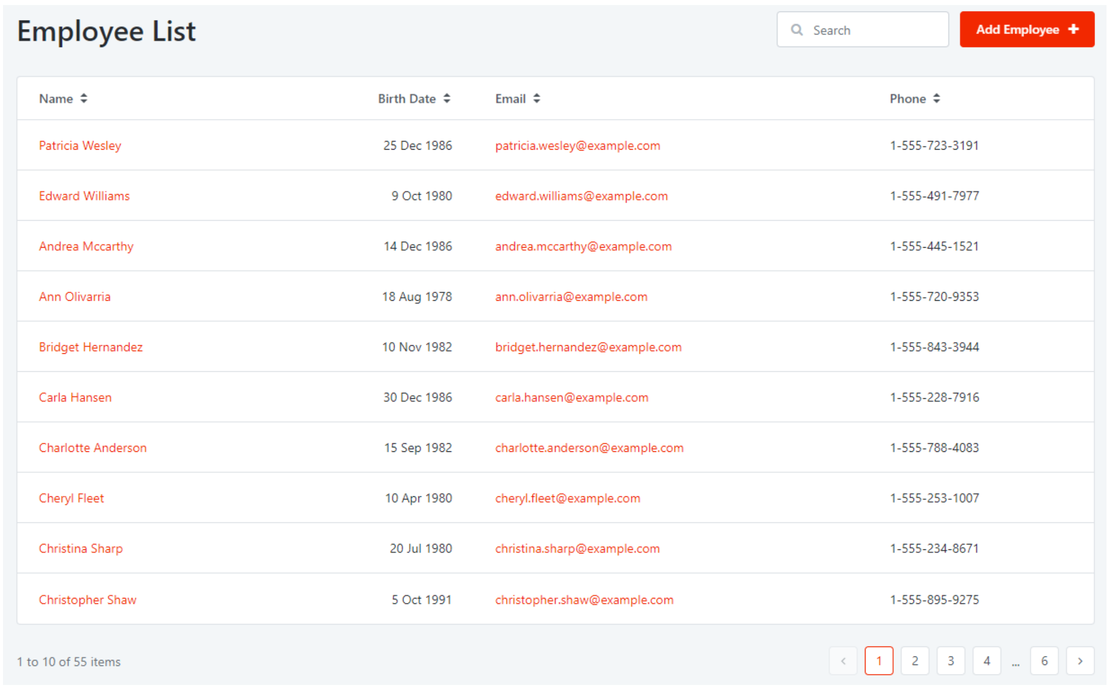
Company's Newcomers

Judith Jones	judith.jones@example.com	1-555-299-6313	CRM Manager
Jessica Taylor	jessica.taylor@example.com	1-555-235-2862	CEO Personal Assistant
James Herrera	james.herrera@example.com	1-555-817-5165	Services Representative
Ida Daley	ida.daley@example.com	1-555-436-7152	Security Systems Brand Manager
Helen Stafford	helen.stafford@example.com	1-555-712-598	Demand Planner

Birthdays

Cheryl Fleet	cheryl.fleet@example.com	1-555-253-1007	Services Representative	10/04
Darlene Shockley	darlene.shockley@example.com	1-555-345-8539	CEO	12/04
David Smith	david.smith@example.com	1-555-409-3087	Services Representative	27/04
Donna Chester	donna.chester@example.com	1-555-349-4530	Administrative Support	19/04
Gregory Jude	gregory.jude@example.com	1-555-330-181	Services Representative	03/04

- The **Employees List**, which has a list of employees with filters and pagination.

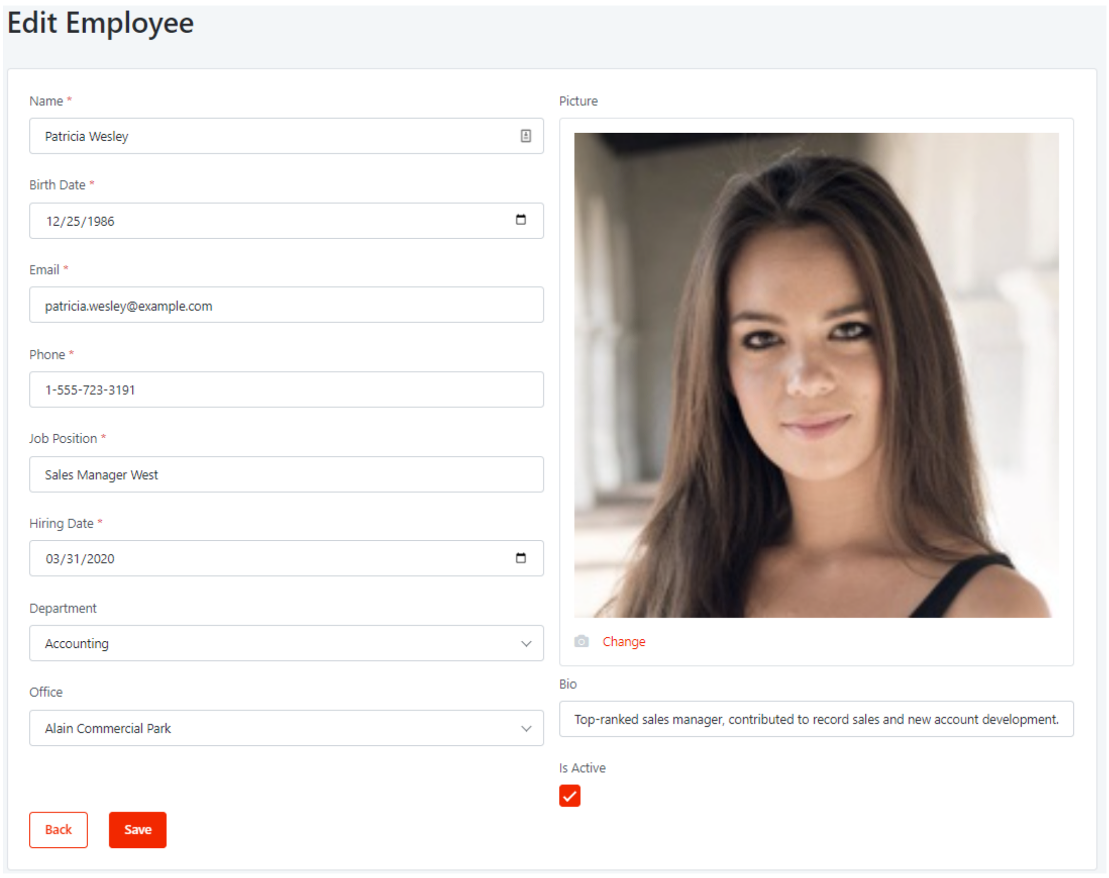


Employee List

Name	Birth Date	Email	Phone
Patricia Wesley	25 Dec 1986	patricia.wesley@example.com	1-555-723-3191
Edward Williams	9 Oct 1980	edward.williams@example.com	1-555-491-7977
Andrea Mccarthy	14 Dec 1986	andrea.mccarthy@example.com	1-555-445-1521
Ann Olivarria	18 Aug 1978	ann.olivarria@example.com	1-555-720-9353
Bridget Hernandez	10 Nov 1982	bridget.hernandez@example.com	1-555-843-3944
Carla Hansen	30 Dec 1986	carla.hansen@example.com	1-555-228-7916
Charlotte Anderson	15 Sep 1982	charlotte.anderson@example.com	1-555-788-4083
Cheryl Fleet	10 Apr 1980	cheryl.fleet@example.com	1-555-253-1007
Christina Sharp	20 Jul 1980	christina.sharp@example.com	1-555-234-8671
Christopher Shaw	5 Oct 1991	christopher.shaw@example.com	1-555-895-9275

1 to 10 of 55 items

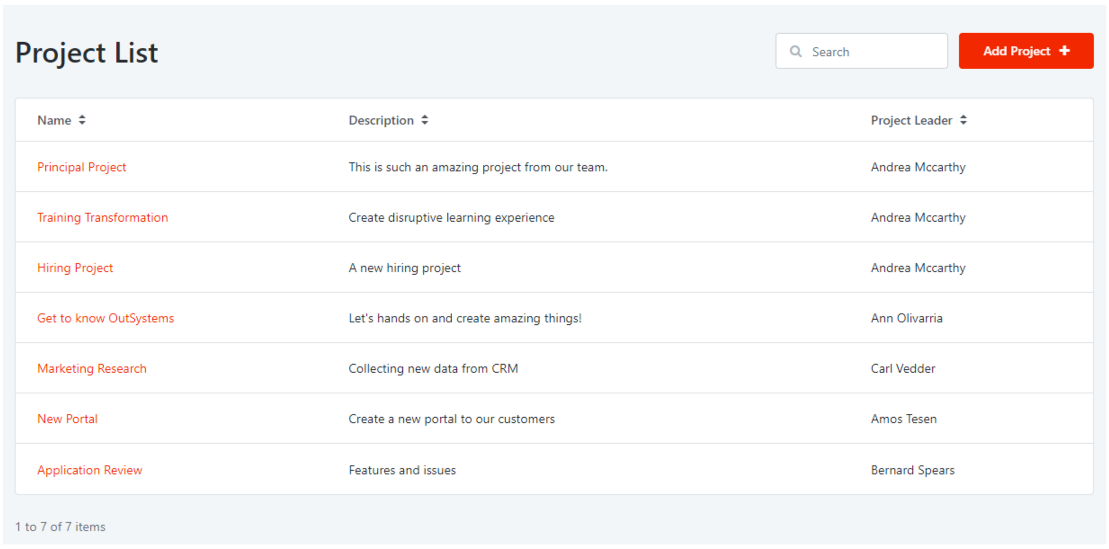
- The **Edit Employee** Screen, which allows us to edit the employee's data, including the employee picture.



Edit Employee

Name *	Patricia Wesley
Birth Date *	12/25/1986
Email *	patricia.wesley@example.com
Phone *	1-555-723-3191
Job Position *	Sales Manager West
Hiring Date *	03/31/2020
Department	Accounting
Office	Alain Commercial Park
Picture	
Bio	Top-ranked sales manager, contributed to record sales and new account development.
Is Active	✓

- The **Project List**, which has a list of projects with filters and pagination.

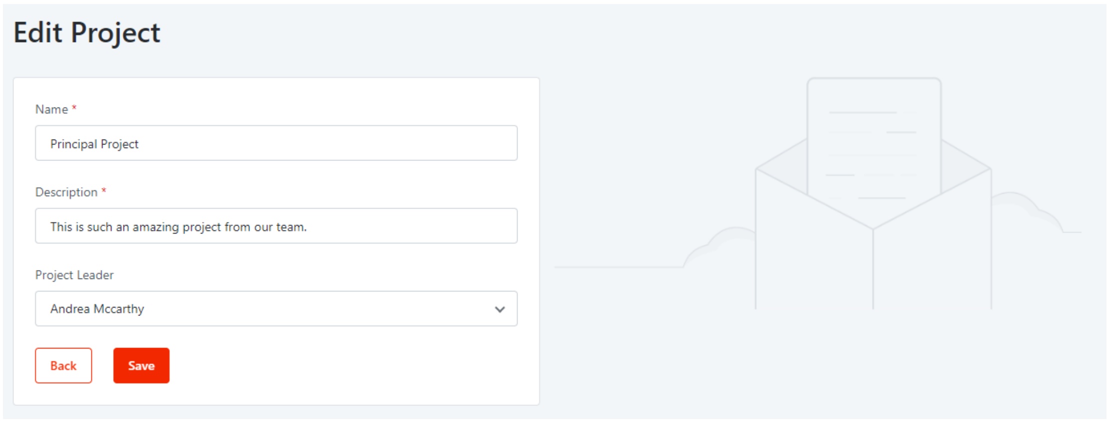


Project List

Name	Description	Project Leader
Principal Project	This is such an amazing project from our team.	Andrea Mccarthy
Training Transformation	Create disruptive learning experience	Andrea Mccarthy
Hiring Project	A new hiring project	Andrea Mccarthy
Get to know OutSystems	Let's hands on and create amazing things!	Ann Olivarria
Marketing Research	Collecting new data from CRM	Carl Vedder
New Portal	Create a new portal to our customers	Amos Tesen
Application Review	Features and issues	Bernard Spears

1 to 7 of 7 items

- The **Edit Project** Screen, which allow us to edit or create a project, defining the Project Leader associated.



Edit Project

Name *: Principal Project

Description *: This is such an amazing project from our team.

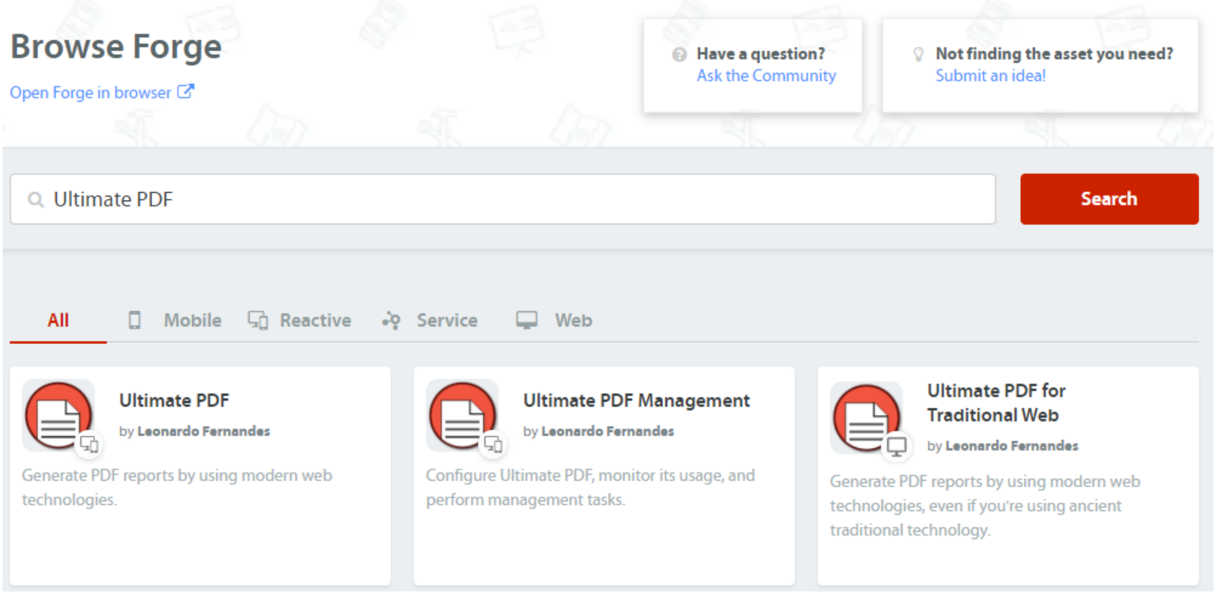
Project Leader: Andrea Mccarthy

You will now expand this app with an extra Screen and functionality.

Generating a Resume

In this tutorial, you will create the logic and UI to allow the user to download a resume, based on the employee's information displayed in the Edit Employee screen, including the projects.

It will be your first contact with a **Forge** Component, and the process to define dependencies in a project.



Browse Forge

Open Forge in browser

Have a question? Ask the Community

Not finding the asset you need? Submit an idea!

Ultimate PDF

Search

All · Mobile · Reactive · Service · Web

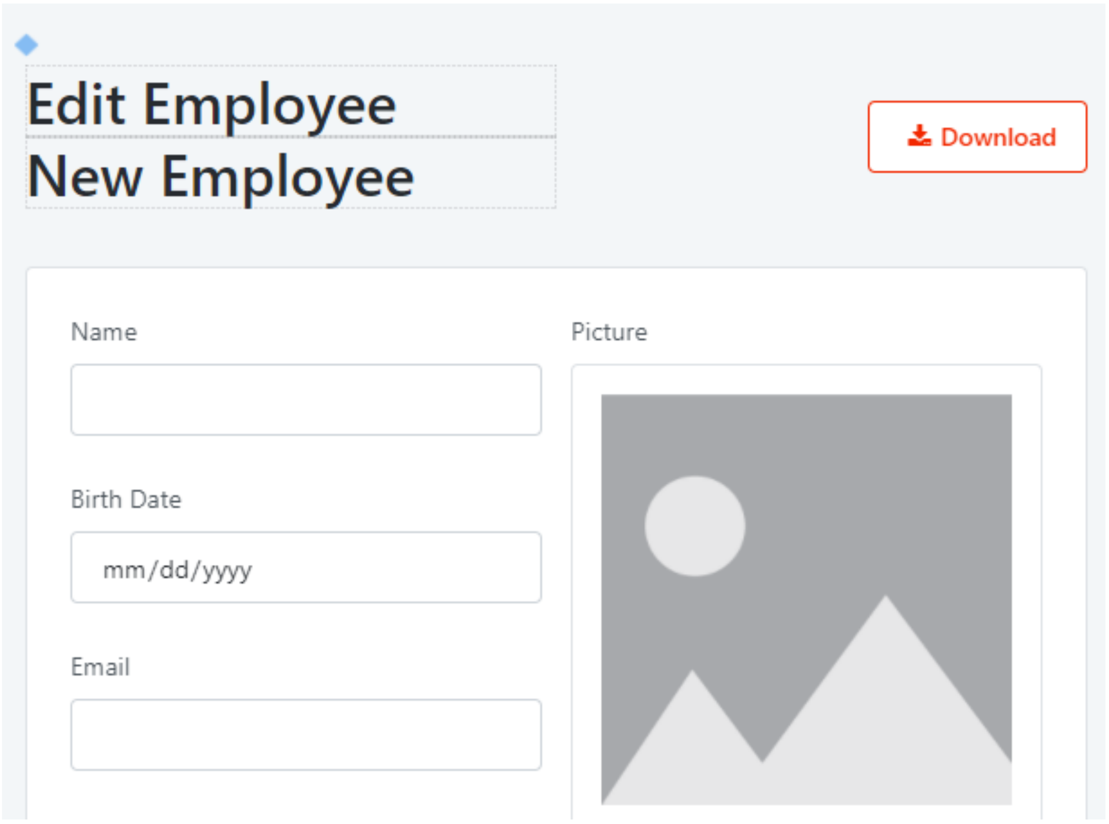
Ultimate PDF by Leonardo Fernandes — Generate PDF reports by using modern web technologies.

Ultimate PDF Management by Leonardo Fernandes — Configure Ultimate PDF, monitor its usage, and perform management tasks.

Ultimate PDF for Traditional Web by Leonardo Fernandes — Generate PDF reports by using modern web technologies, even if you're using ancient traditional technology.

In the Employee Directory application, we will need to do three major steps to achieve the objective:

- Add a Button in the EmployeeDetail Screen to download a pdf.



Edit Employee

New Employee

Download

Name

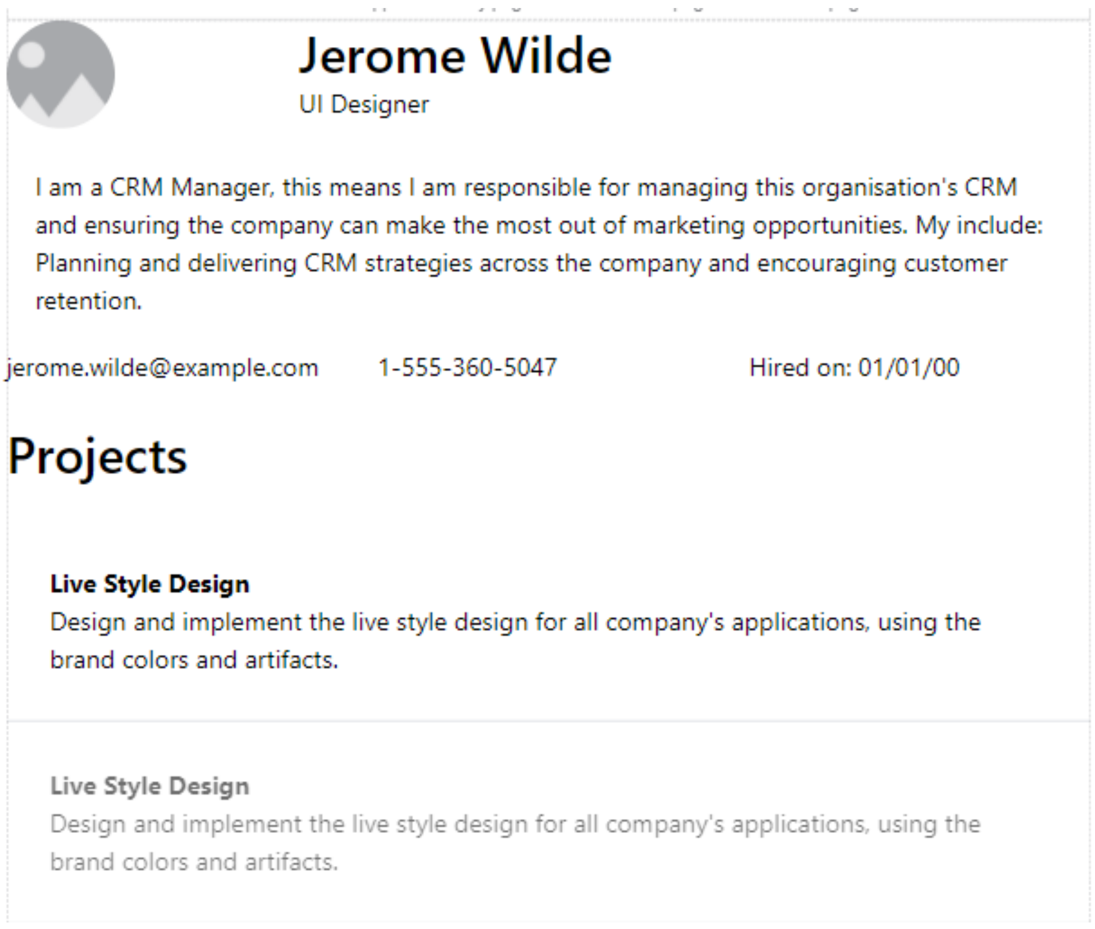
Picture

Birth Date

mm/dd/yyyy

Email

- Link the Button to a new Screen that will have the look and feel of the employee's resume.



Jerome Wilde
UI Designer

I am a CRM Manager, this means I am responsible for managing this organisation's CRM and ensuring the company can make the most out of marketing opportunities. My include: Planning and delivering CRM strategies across the company and encouraging customer retention.

jerome.wilde@example.com 1-555-360-5047 Hired on: 01/01/00

Projects

Live Style Design
Design and implement the live style design for all company's applications, using the brand colors and artifacts.

Live Style Design
Design and implement the live style design for all company's applications, using the brand colors and artifacts.

- Define the logic in the application to download the PDF with 2 simple drag-and-drops, leveraging the Ultimate PDF component.

How-to

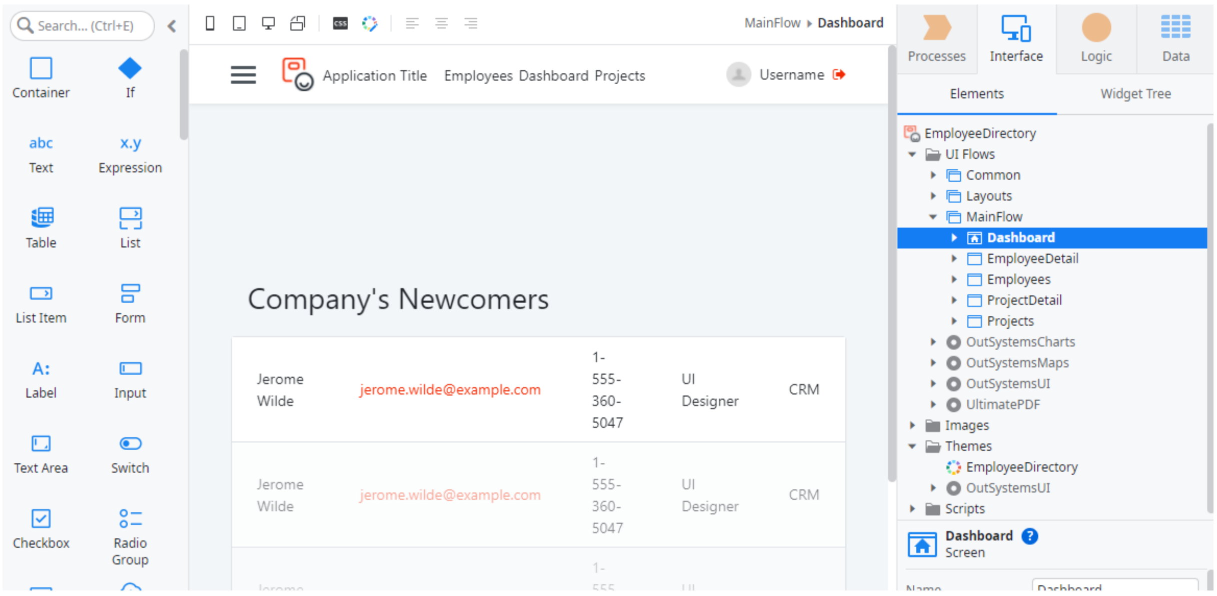
In this section, we'll show a thorough step-by-step description of how to implement the scenario described in the previous section.

Getting Started

In this tutorial we are assuming that you have already followed the previous tutorials, and have the five Screens and logic screens and logical.

In case you haven't yet created, it is important to go back to the previous tutorials, and create the application.

To start this exercise, we need the Service Studio with the module EmployeeDirectory opened. You should see the Screen below with the source of our application.



Company's Newcomers

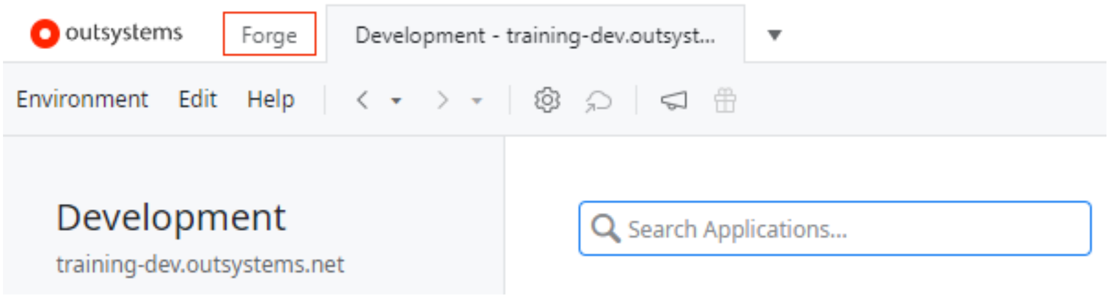
Jerome Wilde	jerome.wilde@example.com	1-555-360-5047	UI Designer	CRM
Jerome Wilde	jerome.wilde@example.com	1-555-360-5047	UI Designer	CRM

Installing a Component from the Forge

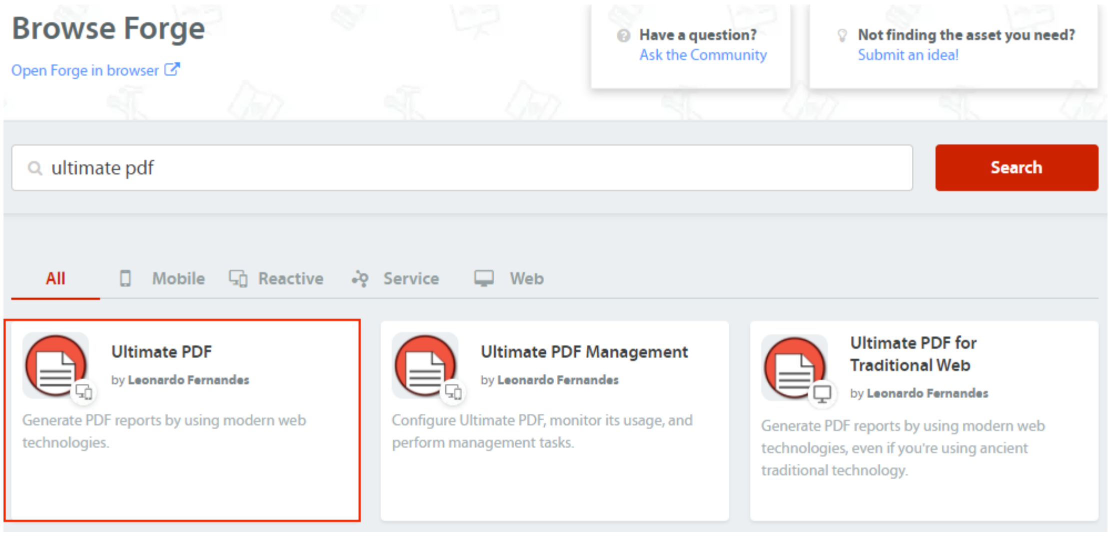
The OutSystems Forge is a repository of reusable, open code modules, connectors, UI components, and business solutions to help speed up app delivery time. To generate and download a PDF file, we will leverage the **Ultimate PDF** Forge component.

The first step is to then install it in your environment, so we can then use it in your app.

1) In Service Studio, click on the **Forge** tab located on the left-top.



2) On the search box type **Ultimate PDF** and click on **Search**. Click on the first option to open the details of this component.

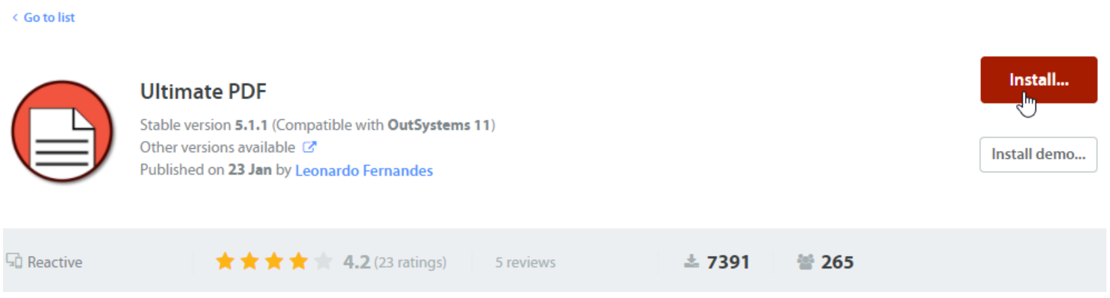


Ultimate PDF
by Leonardo Fernandes
Generate PDF reports by using modern web technologies.

Ultimate PDF Management
by Leonardo Fernandes
Configure Ultimate PDF, monitor its usage, and perform management tasks.

Ultimate PDF for Traditional Web
by Leonardo Fernandes
Generate PDF reports by using modern web technologies, even if you're using ancient traditional technology.

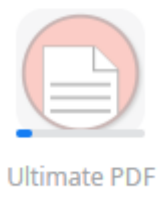
3) Click on **Install...** to start the installation process.



Ultimate PDF
Stable version 5.1.1 (Compatible with OutSystems 11)
Other versions available
Published on 23 Jan by Leonardo Fernandes

If by any chance you find some compatibility warnings at this stage, do not worry. Sometimes we may have different versions from related components, which may cause this incompatibility. So, assuming that it is an environment created for learn purpose, you can click on **Force Install**.

This will take you back to the main applications' area in Service Studio. You will see the app being installed. It takes a while to finish, be patient.

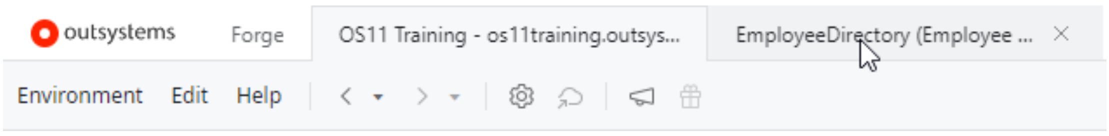


4) After the installation is finished, you will see the app available in the environment.

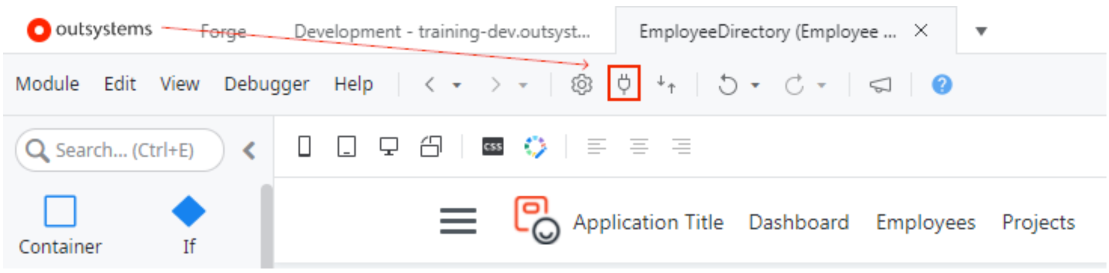
Referencing the Component

Moving back to our application, the first thing that you want to do is to reference the elements from the Ultimate PDF component that you will need in the Employee Directory app. bring the references from the component to the application module.

1) Open the EmployeeDirectory module to go back to our application.

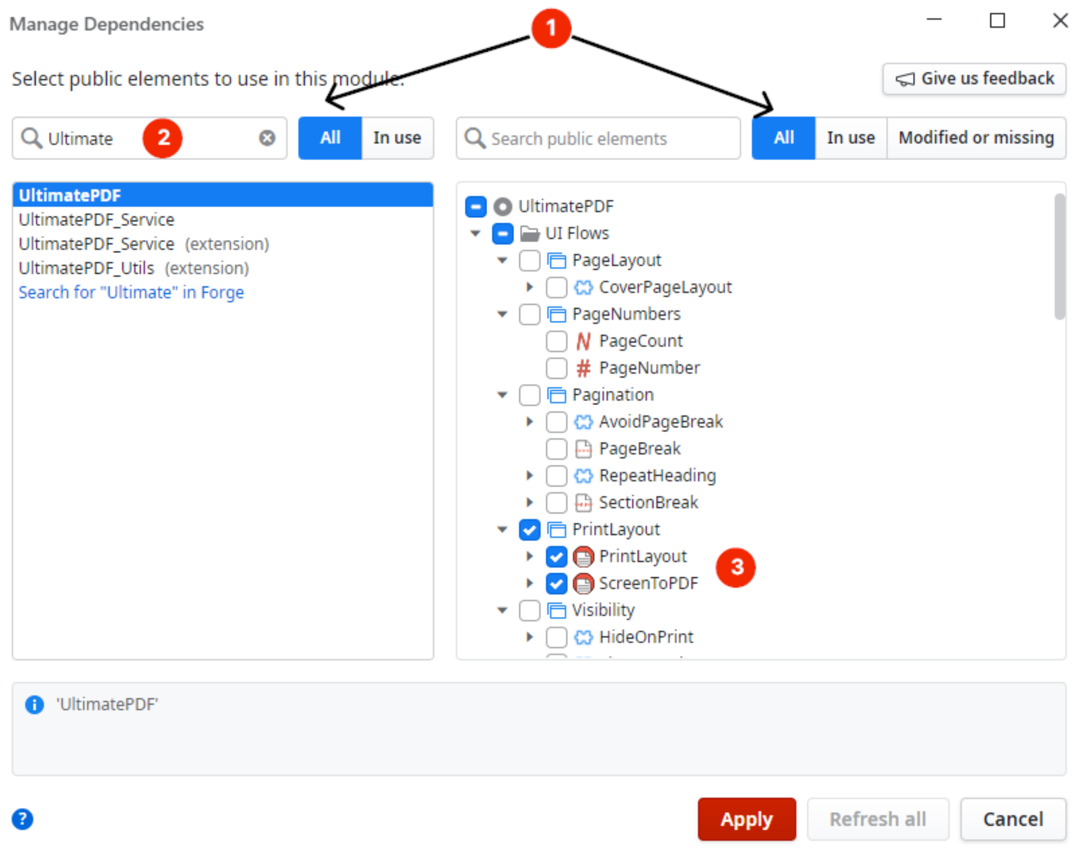

outsystems
Forge
OS11 Training - os11training.outsys...
EmployeeDirectory (Employee ...
Environment
Edit
Help

2) Click on the socket icon to open the **Manage Dependencies** box.



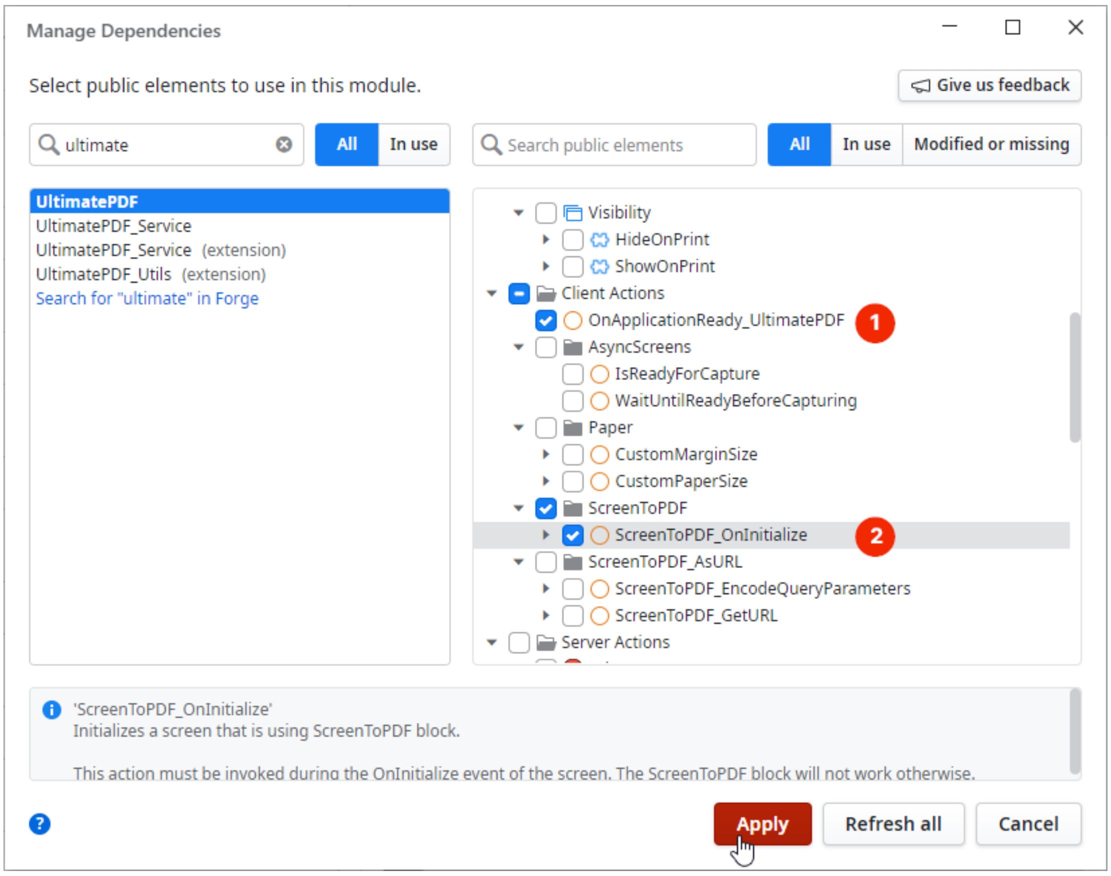
This option allows that we can reference any public elements from other applications that exist in the environment. In this case, we will search and reference the elements we want from the Ultimate PDF application.

3) With the toggle **All** selected (1), type **UltimatePDF** on the search box (2), and click to open the dependencies options. Select **PrintLayout** and **ScreenToPDF** (3).



These are visual elements that we will use to generate the PDF.

4) Still in the same dialog, roll down the mouse wheel, select **OnApplicationReady_UltimatePDF** (1) and **ScreenToPDF_OnInitialize** (2). Click **Apply** when it's done.



'ScreenToPDF_OnInitialize'
Initializes a screen that is using ScreenToPDF block.

This action must be invoked during the OnInitialize event of the screen. The ScreenToPDF block will not work otherwise.

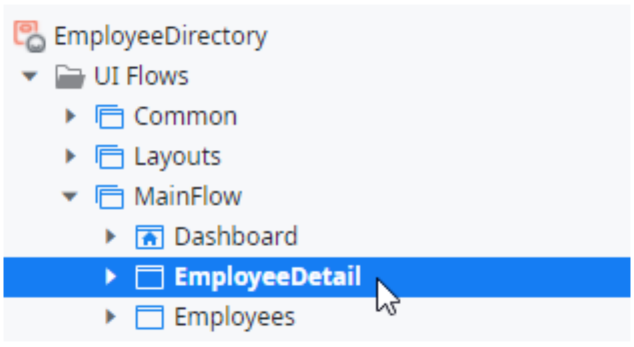
5) Publish the module to save the changes.

Implementing the Download PDF Feature

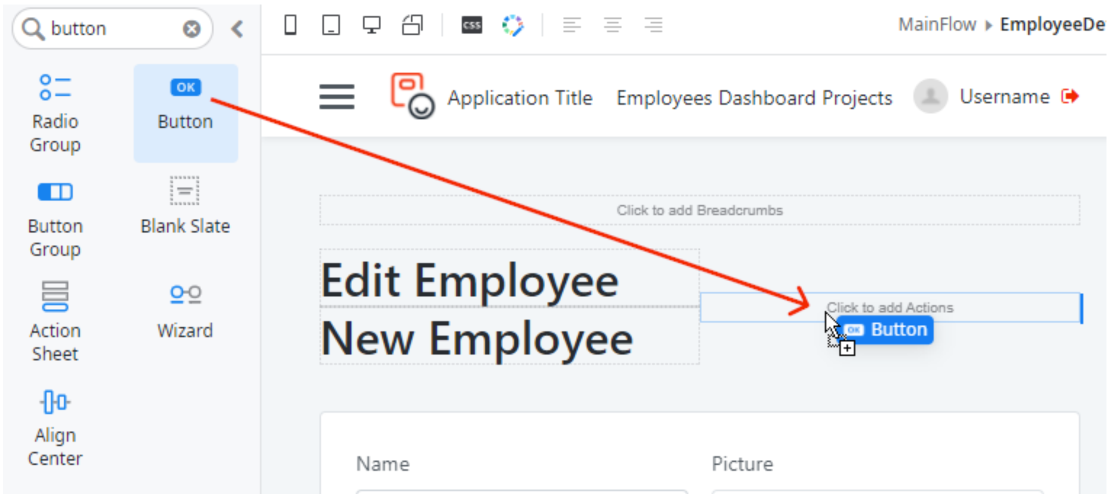
Now that we have everything that we need, it's time to implement the feature to download the PDF.

Download Button

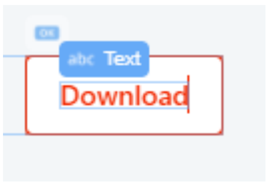
1) Double click on **EmployeeDetail** Screen to open the edition mode.



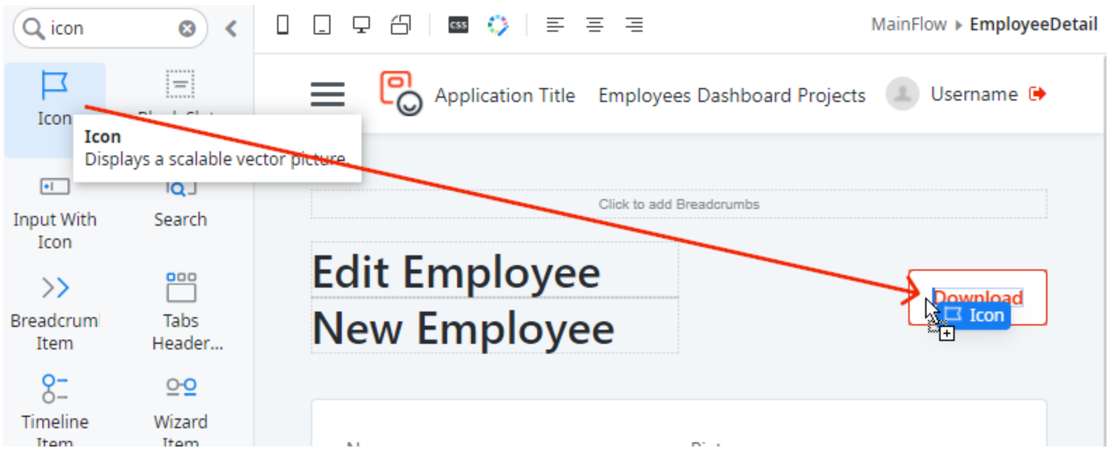
2) Drag a Button element from the left sidebar and drop it into the **Action** placeholder of the EmployeeDetail Screen.



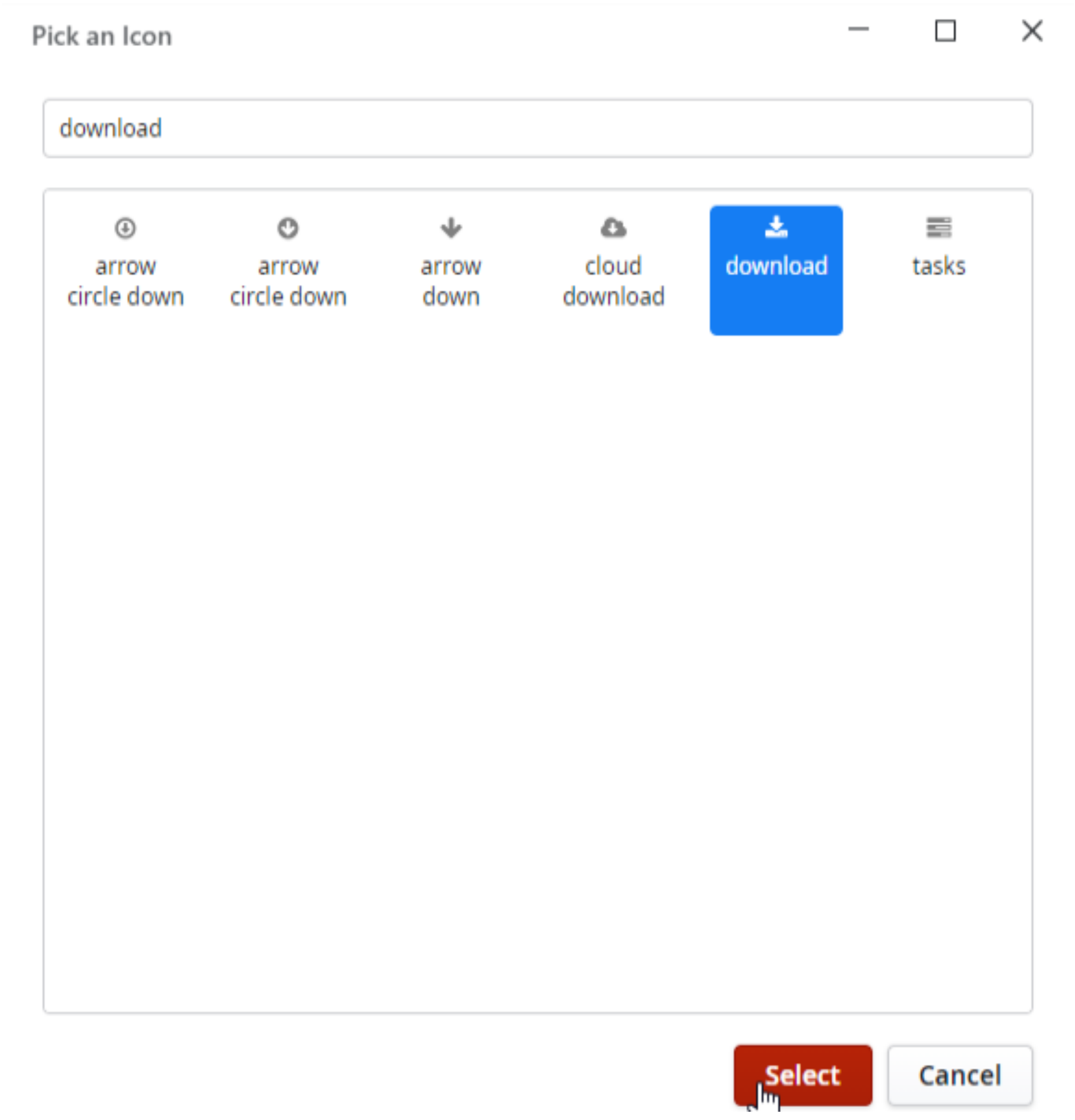
3) Select the Text inside the Button and replace it by *Download*.



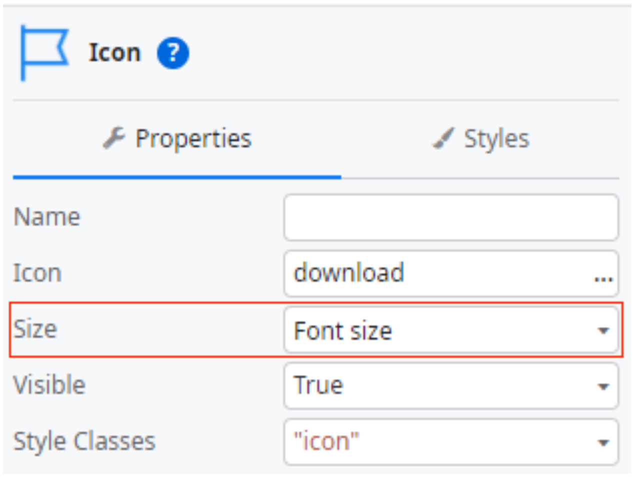
4) Drag an **Icon** from the left sidebar and drop it right before the Download text, inside the Button.



5) In the new dialog that appears we can select the type of icon we want. Type *download* and select the **download** icon that appears.



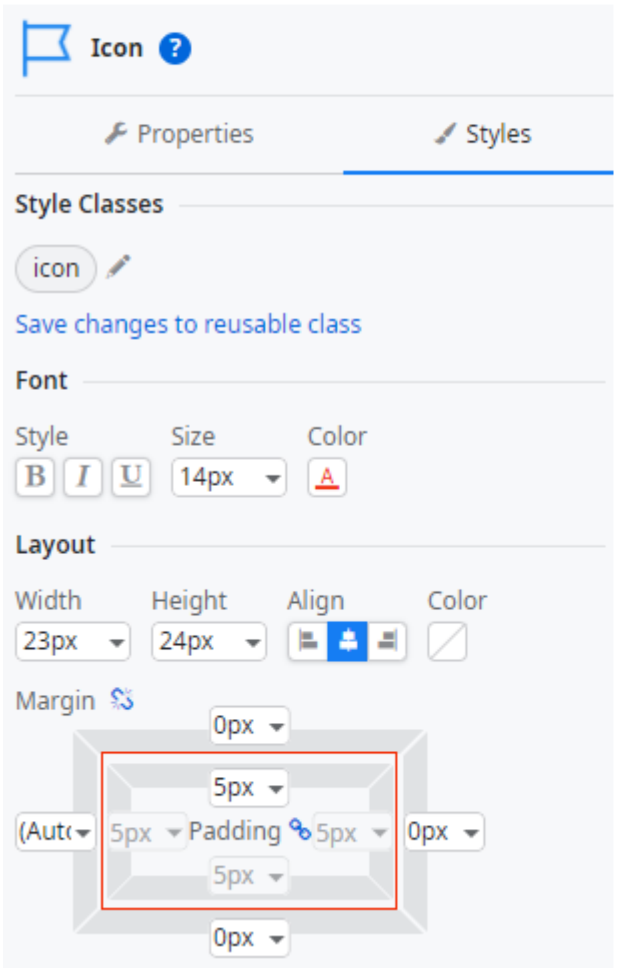
6) In the right sidebar, look at the Icon properties and change the **Size** to *Font size*


Icon
Properties | Styles

Name	
Icon	download
Size	Font size
Visible	True
Style Classes	"icon"

This property controls the size of the icon, and the *Font size* option puts the icon with the same size of the text.

7) Still on the Icon's properties, switch to the **Styles** tab, and define the padding value to *5px*, to create some margin around the icon.


Icon
Properties | Styles

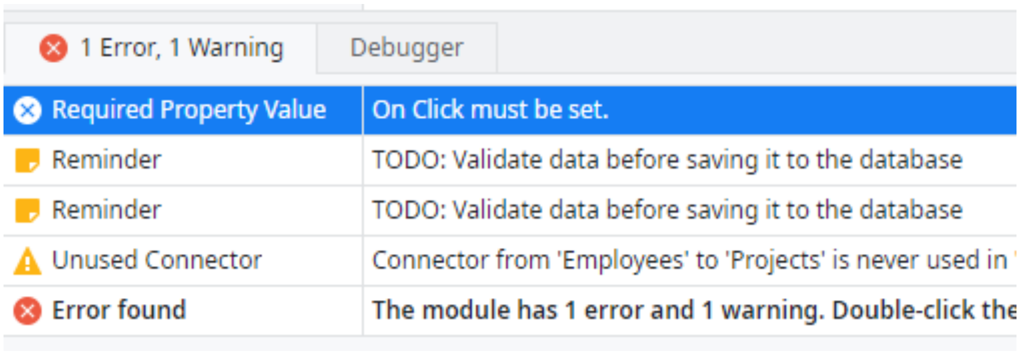
Style Classes
icon
Save changes to reusable class

Font
Style: B I U
Size: 14px
Color

Layout
Width: 23px
Height: 24px
Align
Color

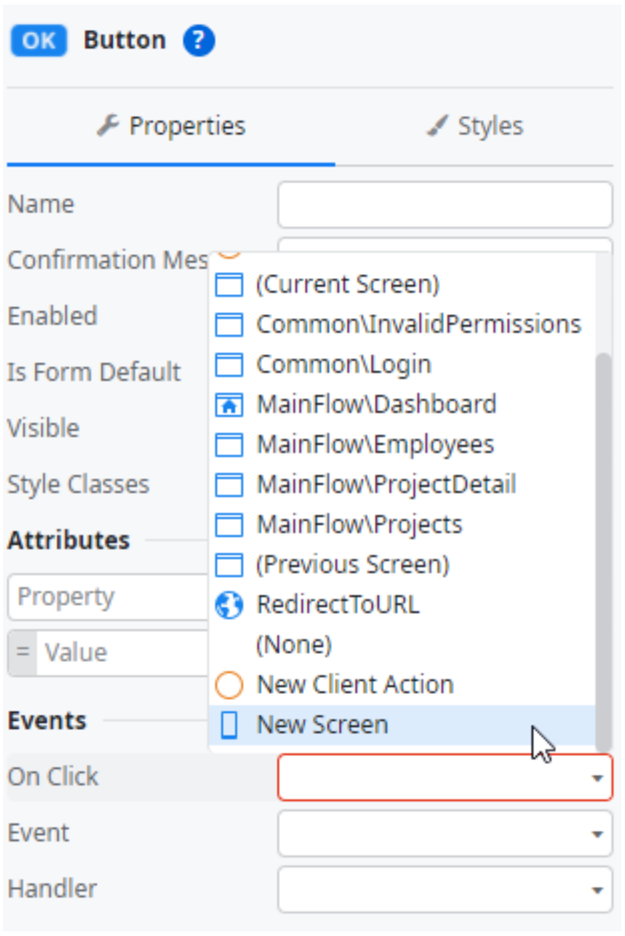
Margin: top 0px, left (Auto), right 0px, bottom 0px
Padding: top 5px, left 5px, right 5px, bottom 5px

As you may notice, Service Studio is displaying an error. If you double-click on the error message, you will be guided to the missing property that you still need to define.



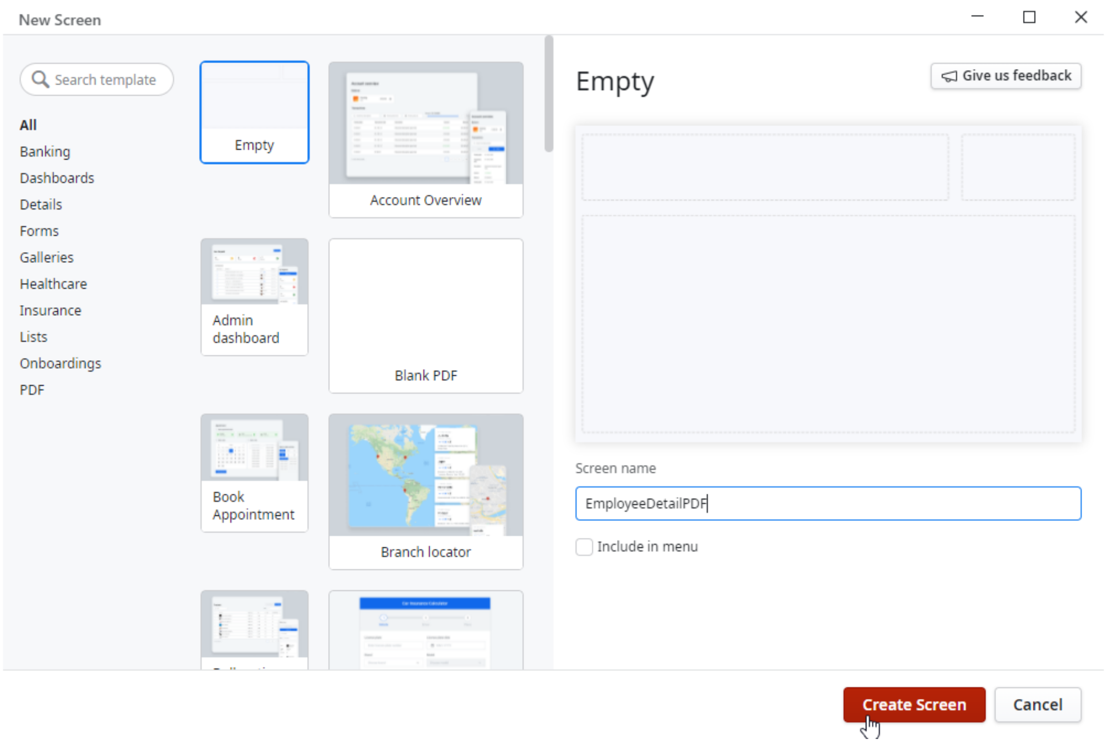
Required Property Value	On Click must be set.
Reminder	TODO: Validate data before saving it to the database
Reminder	TODO: Validate data before saving it to the database
Unused Connector	Connector from 'Employees' to 'Projects' is never used in
Error found	The module has 1 error and 1 warning. Double-click the

8) We are now on the Button we just added before. On the right sidebar, let's switch to the Properties of the Button. The **On Click** property is missing. Set the **On Click** property to **New Screen**.



The On Click property of the button is mandatory, since the button is a clickable element, and defines the behavior the button will have when the button is clicked by the user. In this case, we will navigate to a new Screen, which will have the layout we want for our PDF.

9) Create an **Empty Screen** with the name *EmployeeDetailPDF*.

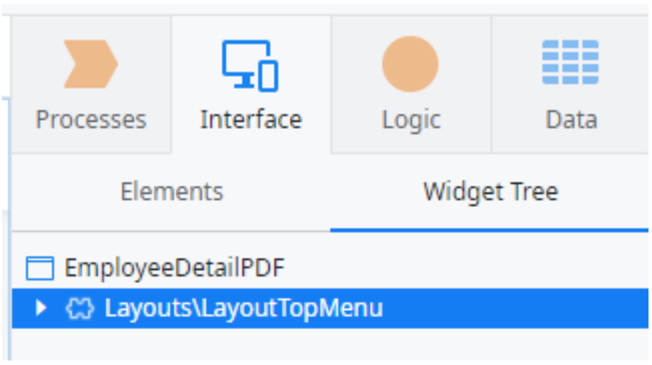


Screen name: EmployeeDetailPDF

Resume Screen

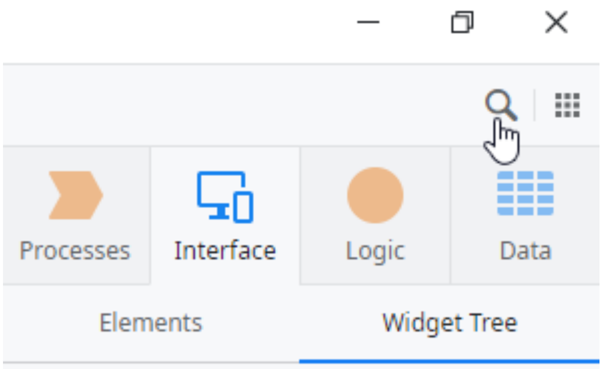
You just created a new empty Screen that will have the information that we want to have in the PDF file. So, we will now build the Screen with a layout that is PDF friendly, with the help of the elements referenced from the Ultimate PDF component.

1) Open the widget tree of the new Screen and delete **Layouts\LayoutsTopMenu**

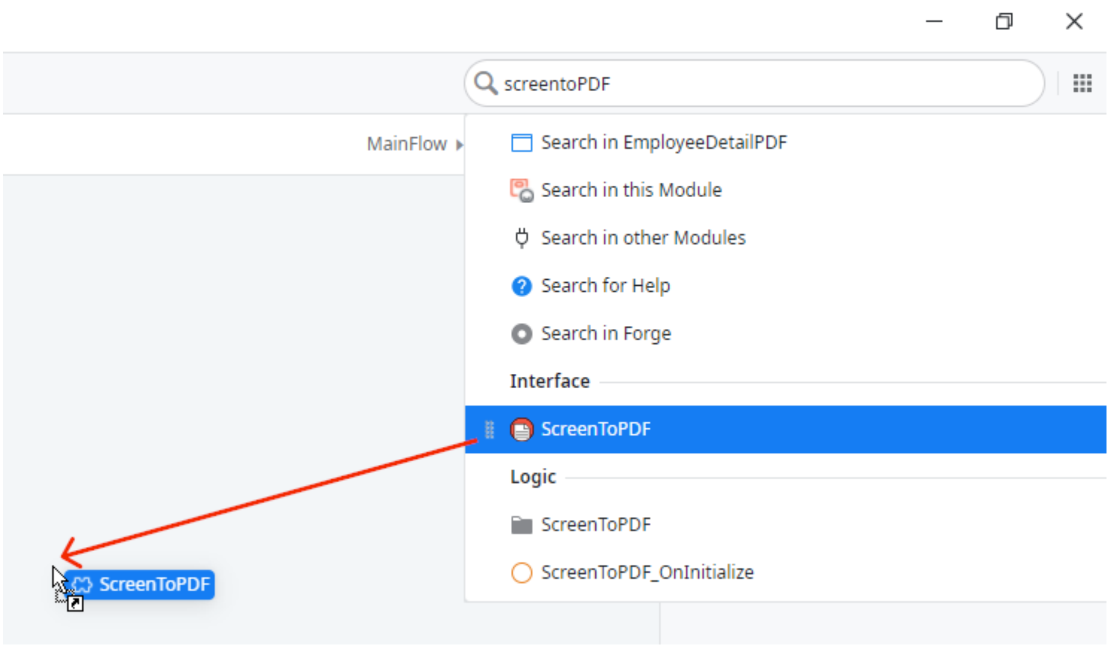


Every Screen is created with a default layout, that we can always change or delete. In this case, we will use a layout from the Ultimate PDF component.

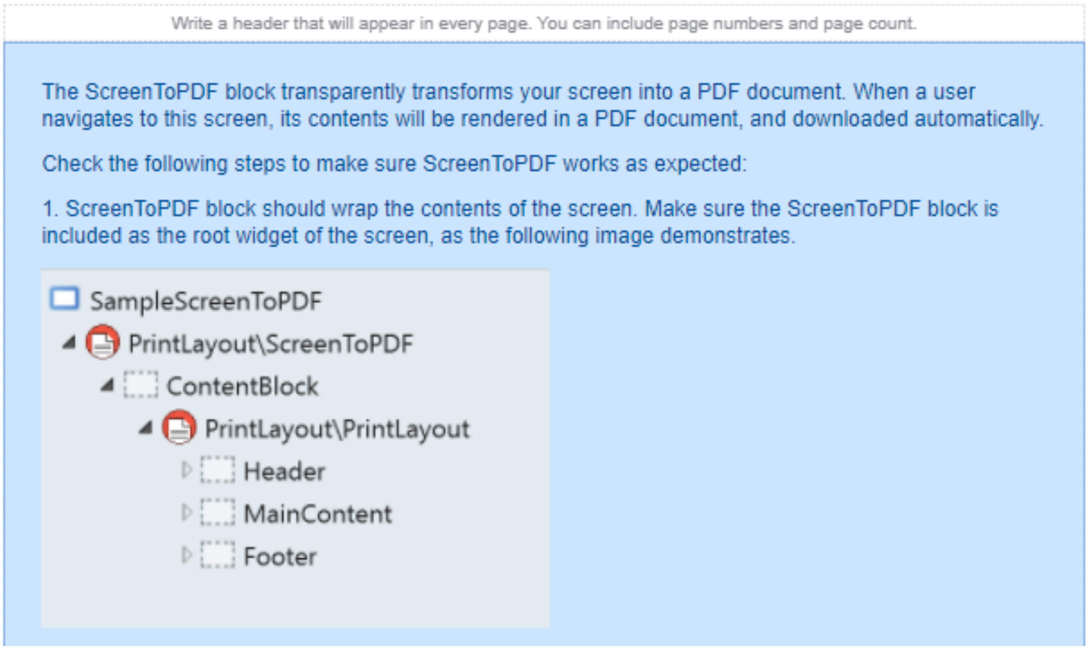
2) On the top right section of Service Studio, you will find a Search option.



3) Type *ScreenToPDF* in the search, find the element under the Interface section, drag it and drop it on the Screen.



You'll see something like the image below

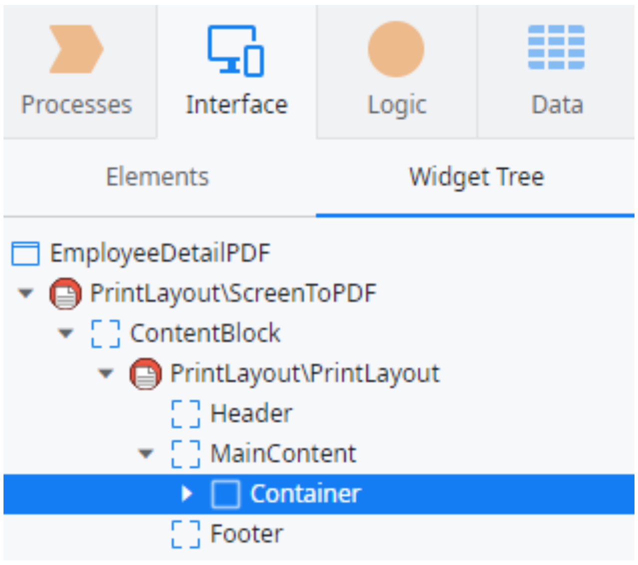

Write a header that will appear in every page. You can include page numbers and page count.

The ScreenToPDF block transparently transforms your screen into a PDF document. When a user navigates to this screen, its contents will be rendered in a PDF document, and downloaded automatically.

Check the following steps to make sure ScreenToPDF works as expected:

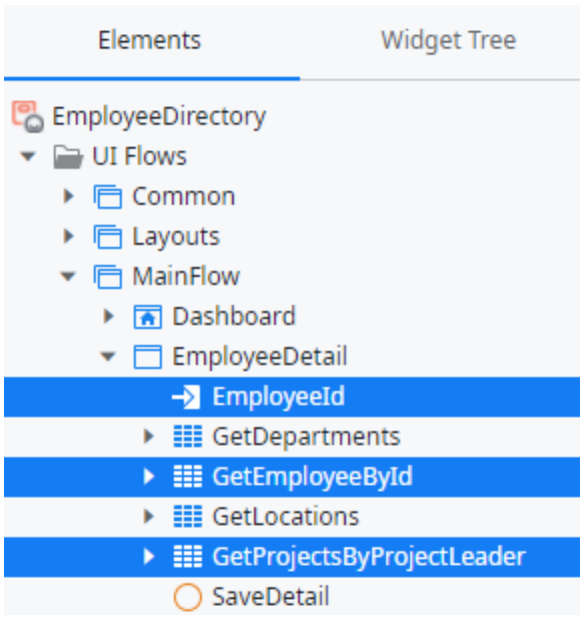
1. ScreenToPDF block should wrap the contents of the screen. Make sure the ScreenToPDF block is included as the root widget of the screen, as the following image demonstrates.

4) We can see now the widget tree of the new Screen with the ScreentoPDF element. Expand it until you find a **Container** and delete it.

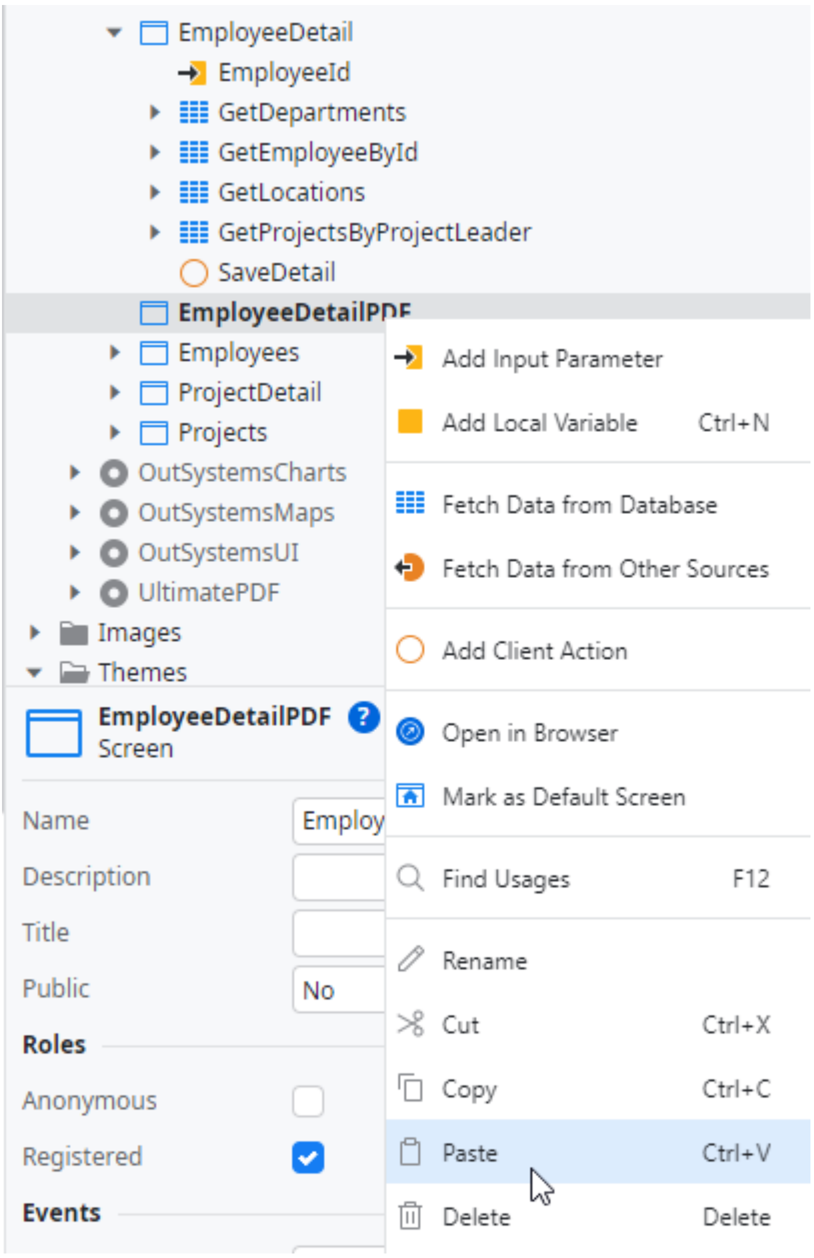


This Container has some instructions of usage, so we can delete it to clean up the Screen. And now you have the layout you need to generate the PDF.

5) Back to the elements tree on the right, expand the EmployeeDetail Screen and copy the input parameter **EmployeeId** and the **GetEmployeeById** and **GetProjectsByProjectLeader** Aggregates.

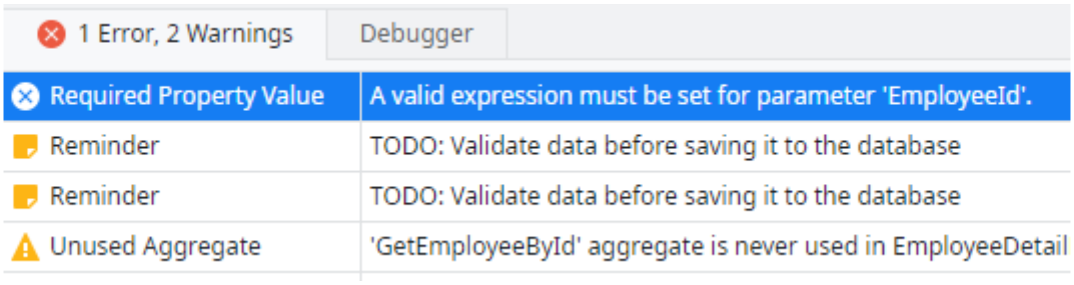


6) Paste the elements on the **EmployeeDetailPDF** Screen since they can be reused.



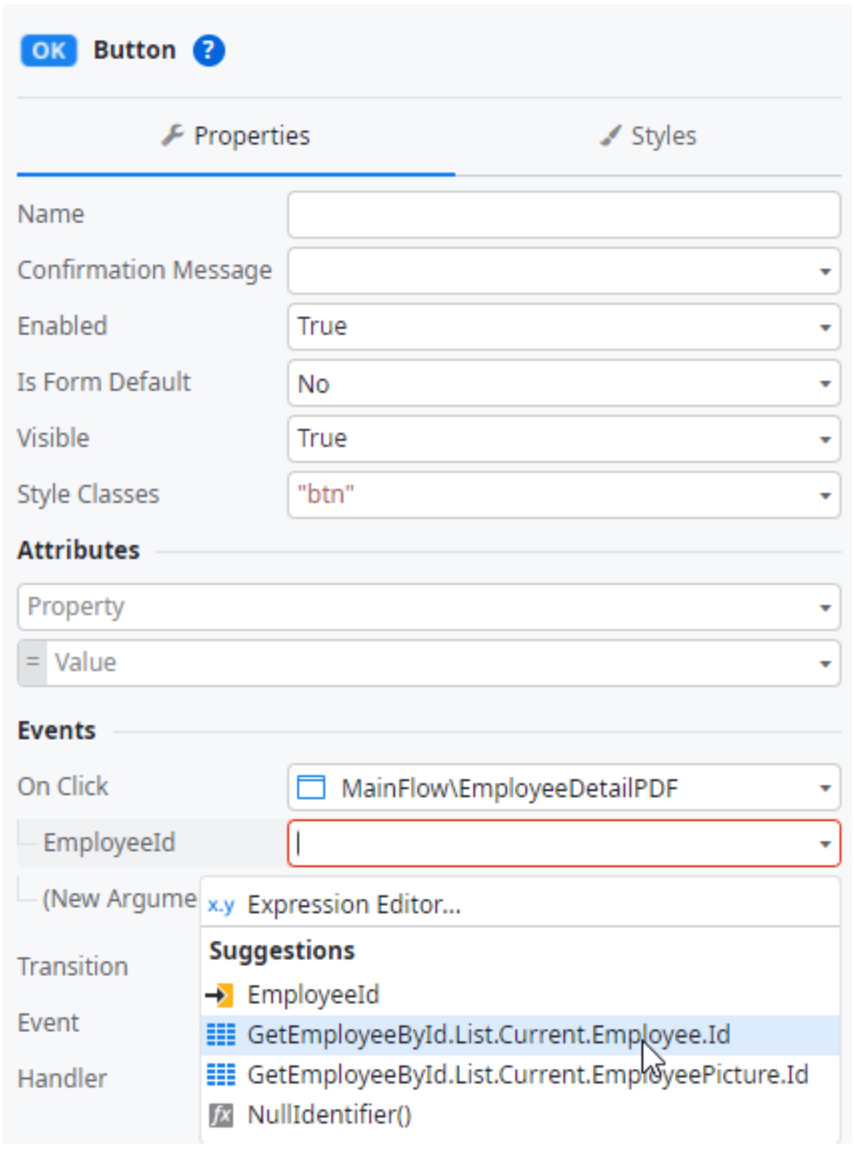
After this step, Service Studio will display an error. Since the new Screen now has an Input parameter, the Download button must pass a value for that parameter. Let's fix it!

7) On the bottom left in Service Studio, click on the tab that indicates that an error exists. Double-click the error message to open the Button properties, in the EmployeeDetail Screen.



Required Property Value	A valid expression must be set for parameter 'EmployeeId'.
Reminder	TODO: Validate data before saving it to the database
Reminder	TODO: Validate data before saving it to the database
Unused Aggregate	'GetEmployeeById' aggregate is never used in EmployeeDetail

8) On the right sidebar, in the Button properties, set the **EmployeeId** value to *GetEmployeeById.List.Current.Employee.Id*. If you don't see the property, scroll down in the area and you will find it.



Name	
Confirmation Message	
Enabled	True
Is Form Default	No
Visible	True
Style Classes	"btn"
Property	
= Value	
On Click	MainFlow\EmployeeDetailPDF
EmployeeId	|
(New Argume	
Transition	
Event	
Handler	

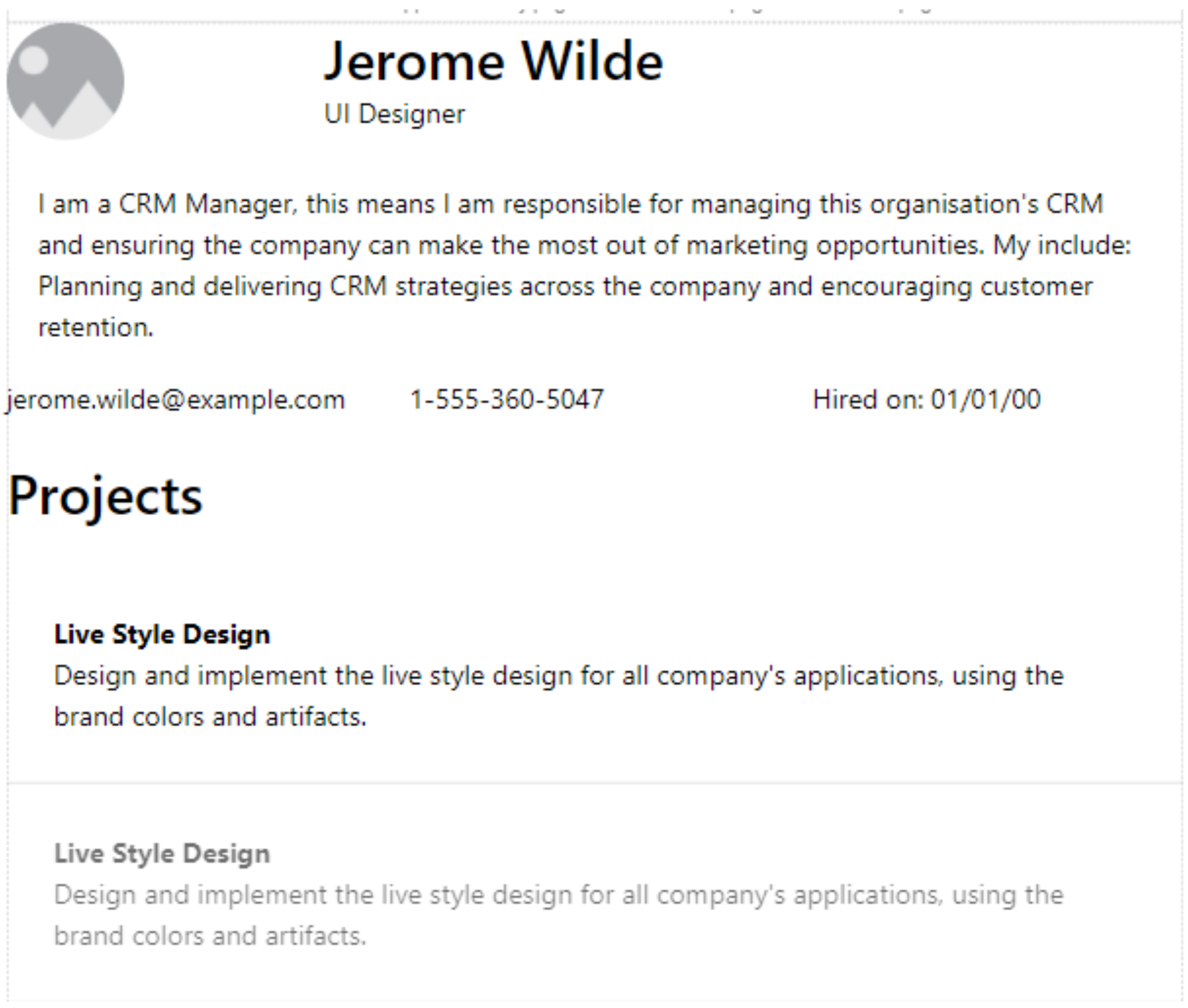
Suggestions:
- EmployeeId
- GetEmployeeById.List.Current.Employee.Id
- GetEmployeeById.List.Current.EmployeePicture.Id
- NullIdentifier()

This way, we make sure that the Id of the employee whose information is being displayed on the Screen, is actually the Id passed to the new Screen for the resume.

9) Publish the module to save the latest changes.

Design the UI for the Employee's Details

Now it's time to actually design the UI for the resume. Don't forget how the resume should look like.



Jerome Wilde

UI Designer

I am a CRM Manager, this means I am responsible for managing this organisation's CRM and ensuring the company can make the most out of marketing opportunities. My include: Planning and delivering CRM strategies across the company and encouraging customer retention.

jerome.wilde@example.com 1-555-360-5047 Hired on: 01/01/00

Projects

Live Style Design

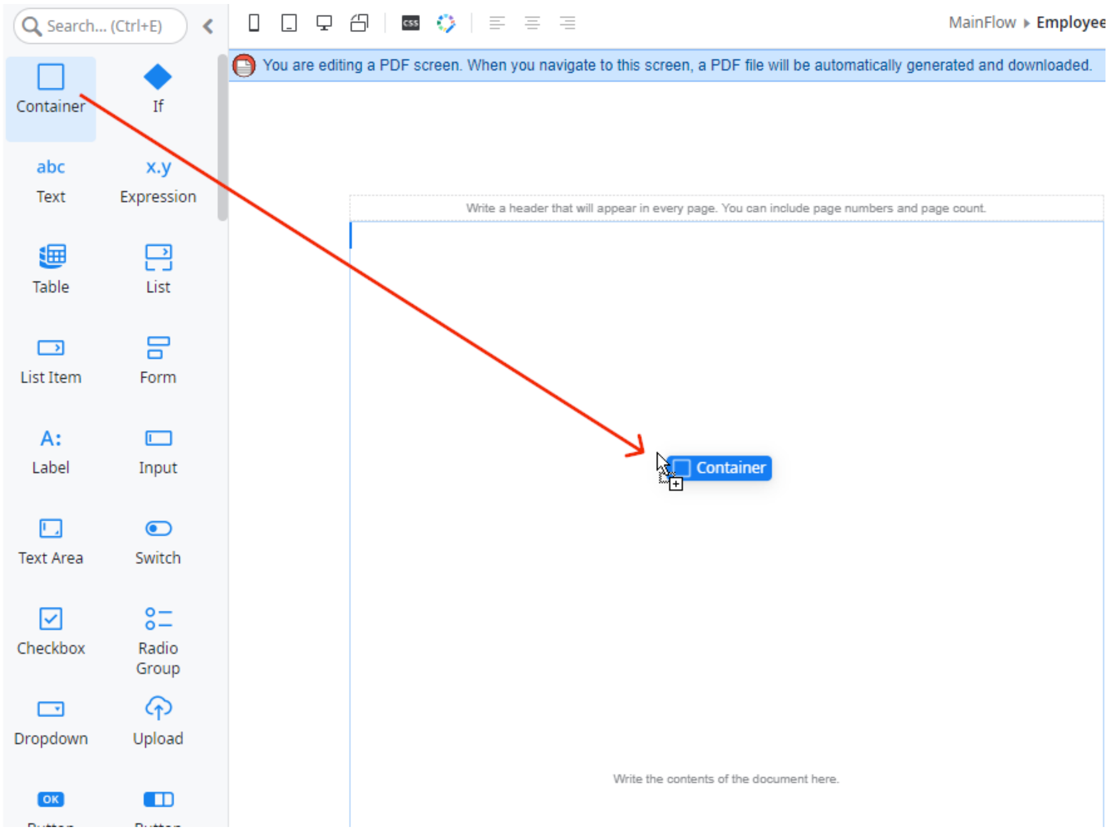
Design and implement the live style design for all company's applications, using the brand colors and artifacts.

Live Style Design

Design and implement the live style design for all company's applications, using the brand colors and artifacts.

1) Double-click the **EmployeeDetailPDF** Screen to open it.

2) Drag a **Container** from the left sidebar and drop it on the Screen.

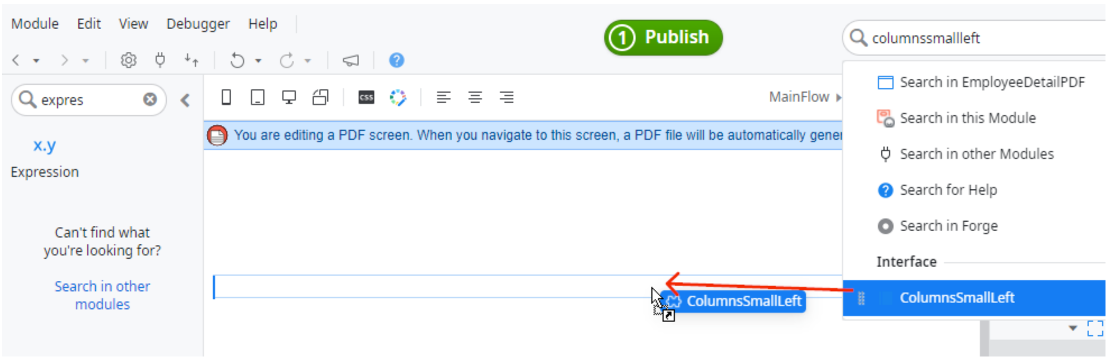


You are editing a PDF screen. When you navigate to this screen, a PDF file will be automatically generated and downloaded.

Write a header that will appear in every page. You can include page numbers and page count.

Write the contents of the document here.

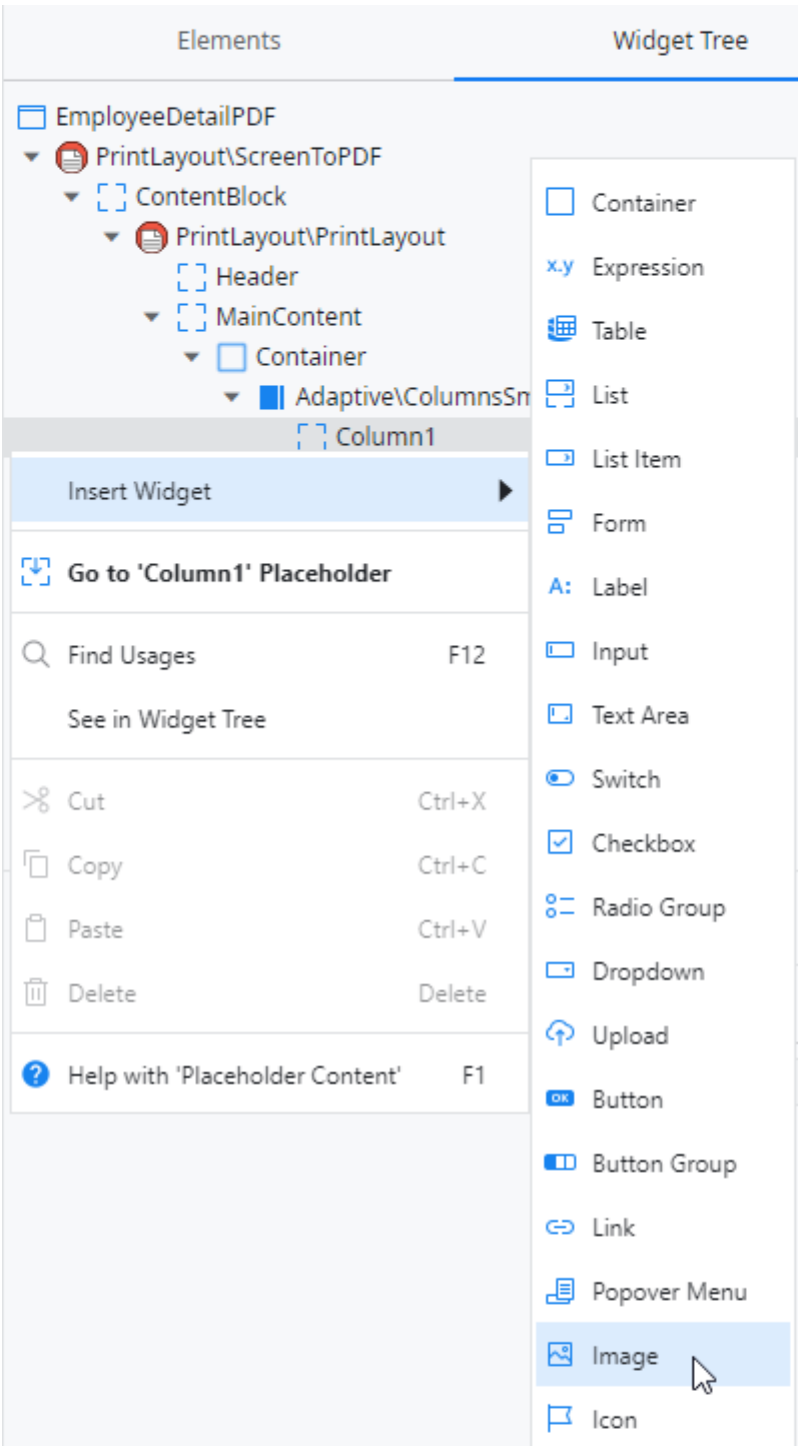
3) Go back to the Search menu on the top right of Service Studio, search for *ColumnSmallLeft*, drag it, and drop it on the Container created on the previous step.



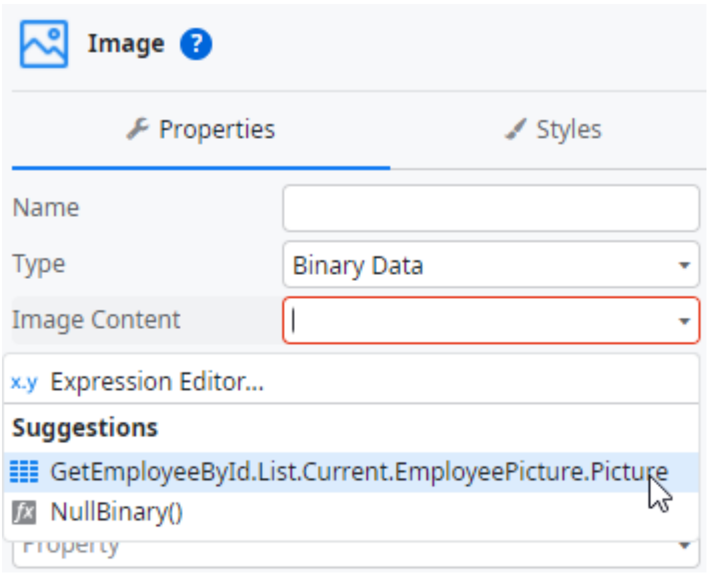
You are editing a PDF screen. When you navigate to this screen, a PDF file will be automatically gener

This element automatically divides the Screen in two columns, with a smaller column on the left.

4) On the widget tree, right-click on the **Column1** element. Click on **Insert Widget** and then in **Image**.

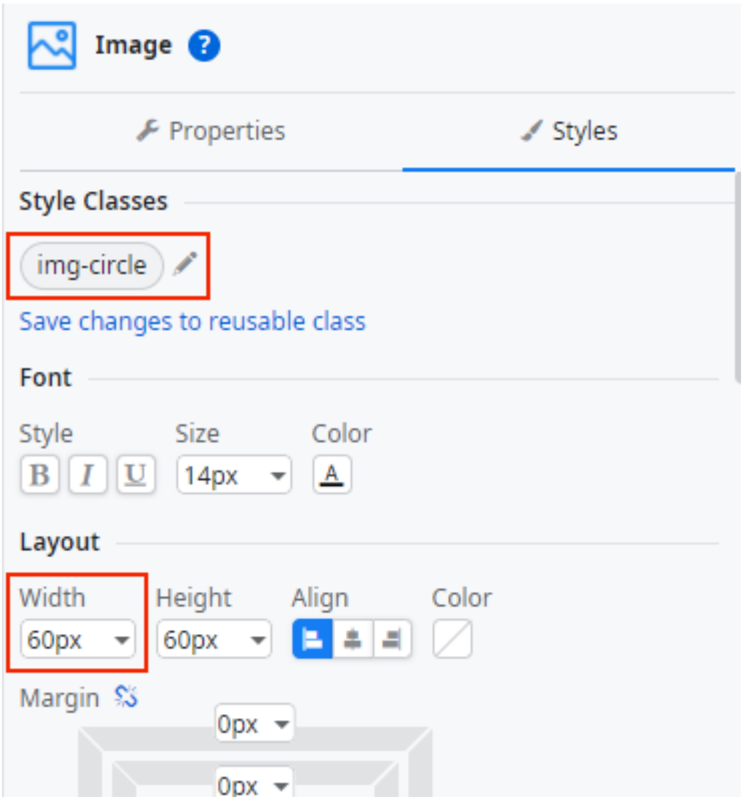


5) On the properties section of the Image, change the **Type** to *Binary Data* and the **Image Content** to *GetEmployeeById.List.Current.EmployeePicture.Picture*.

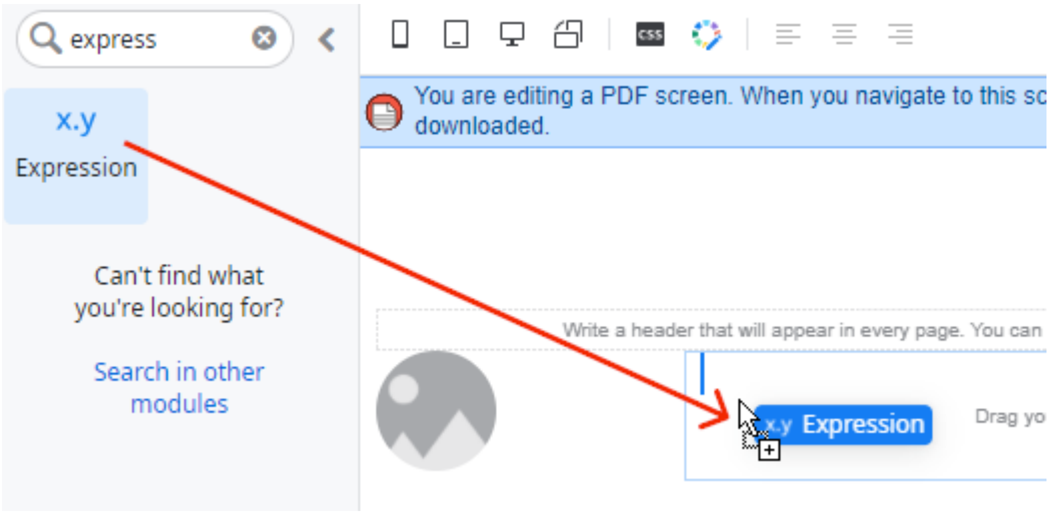


This ties the Image visual element with the employee's picture saved in the database.

6) Still on the Image properties, switch to the **Styles** tab and change the image width to *60px* and the image classes to `img-circle`.

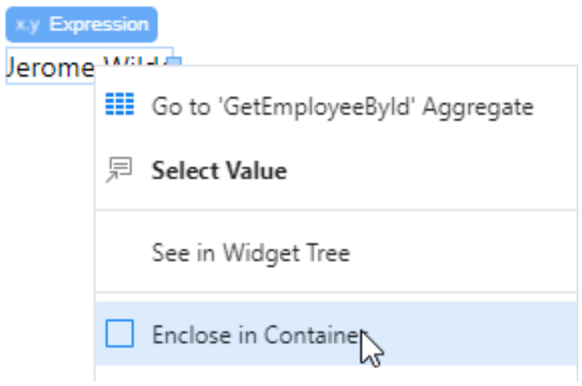


7) Back on the Screen, drag an Expression element from the left sidebar and drop it into the Column2 section of the ColumnSmallLeft element. Set the expression's **Value** to *GetEmployeeById.List.Current.Employee.Name*.

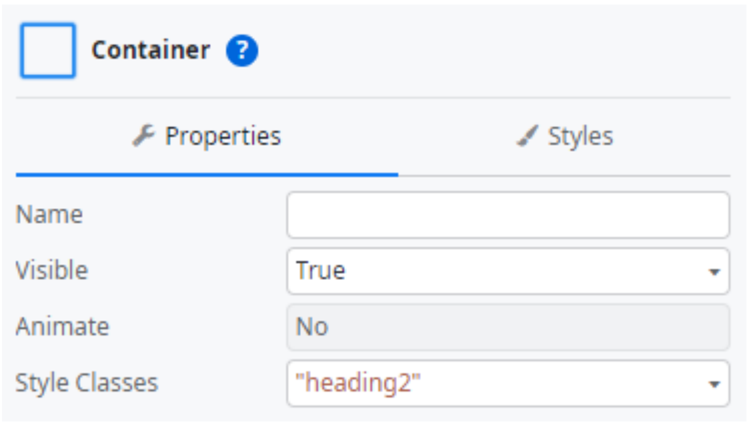

express
x.y
Expression
Can't find what you're looking for?
Search in other modules
You are editing a PDF screen. When you navigate to this sc
downloaded.
Write a header that will appear in every page. You can
x.y Expression
Drag yo

This expression will have the employee's name.

8) Right-click on the expression and click on **Enclose in Container**.

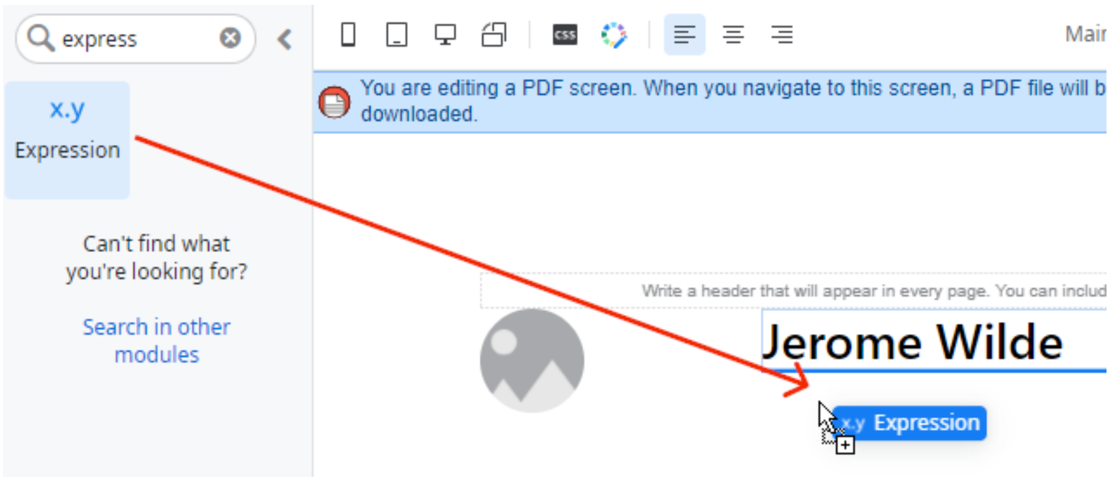


9) In the properties of the Container, **Style Classes** to `"heading2"`.



Name	
Visible	True
Animate	No
Style Classes	"heading2"

10) Drag a new Expression and drop it under the Container. Set its **Value** to *GetEmployeeById.List.Current.Employee.JobPosition*. Enclose it in a Container as well.

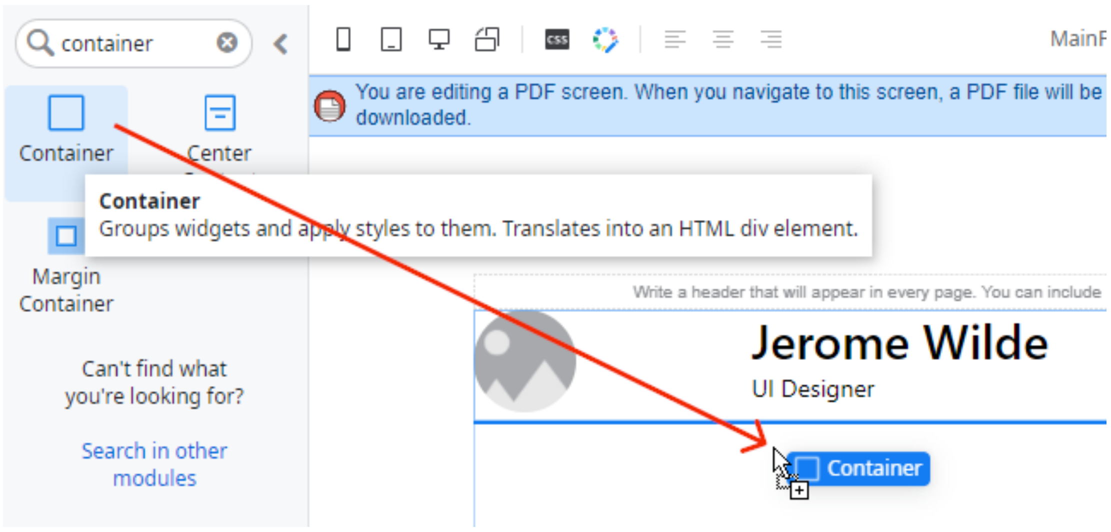


You are editing a PDF screen. When you navigate to this screen, a PDF file will b downloaded.

Write a header that will appear in every page. You can includ

We now have an expression for the job position.

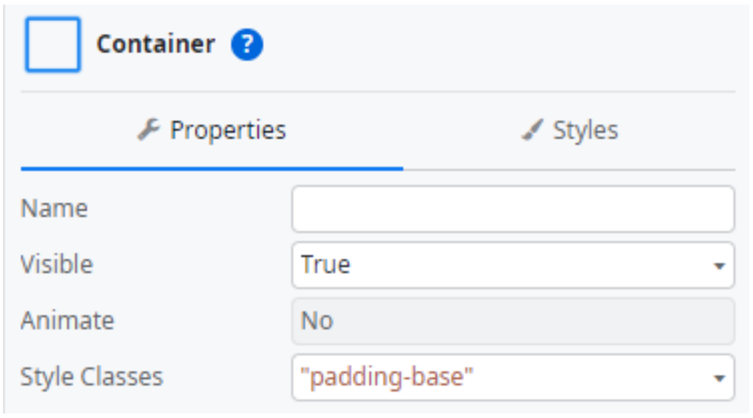
11) Drag a **Container** from the left sidebar and drop it under the elements created before. Set the **Style Classes** of this container to `"padding-base"`.



Container
Groups widgets and apply styles to them. Translates into an HTML div element.

You are editing a PDF screen. When you navigate to this screen, a PDF file will be downloaded.

Write a header that will appear in every page. You can include

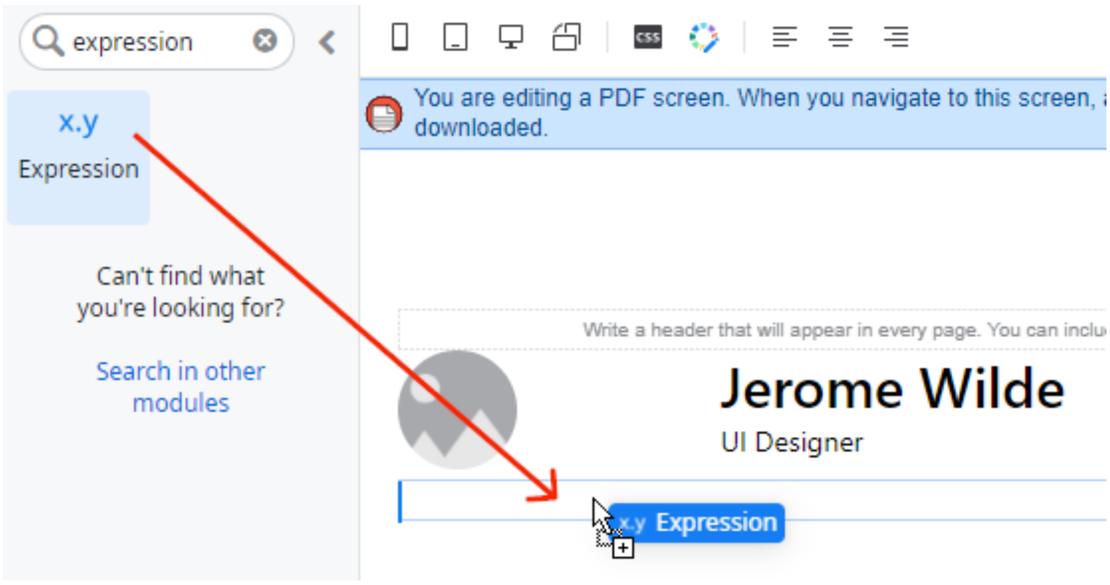


Container

Properties | Styles

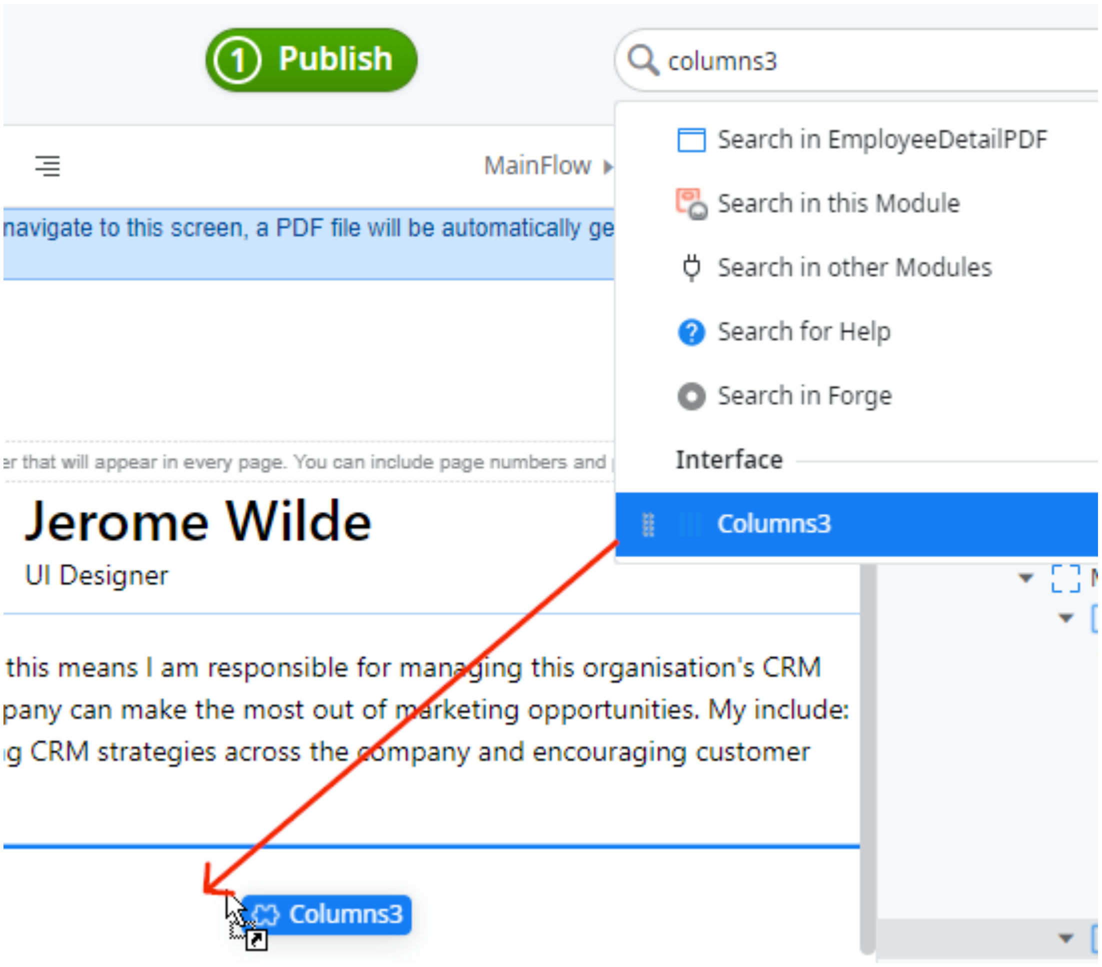
Name	
Visible	True
Animate	No
Style Classes	"padding-base"

12) Drag a new **Expression** and drop it in the new Container. Set its **Value** to *GetEmployeeById.List.Current.Employee.Bio*.


expression
x.y
Expression
Can't find what you're looking for?
Search in other modules
You are editing a PDF screen. When you navigate to this screen, a
downloaded.
Write a header that will appear in every page. You can inclu
Jerome Wilde
UI Designer
x.y Expression

And with this, the resume will have an expression to display the employee's biography.

13) Go again to the Search tool on the top right of Service Studio, search for the **Columns3** element, drag it, and drop into the main page under all the elements created before.



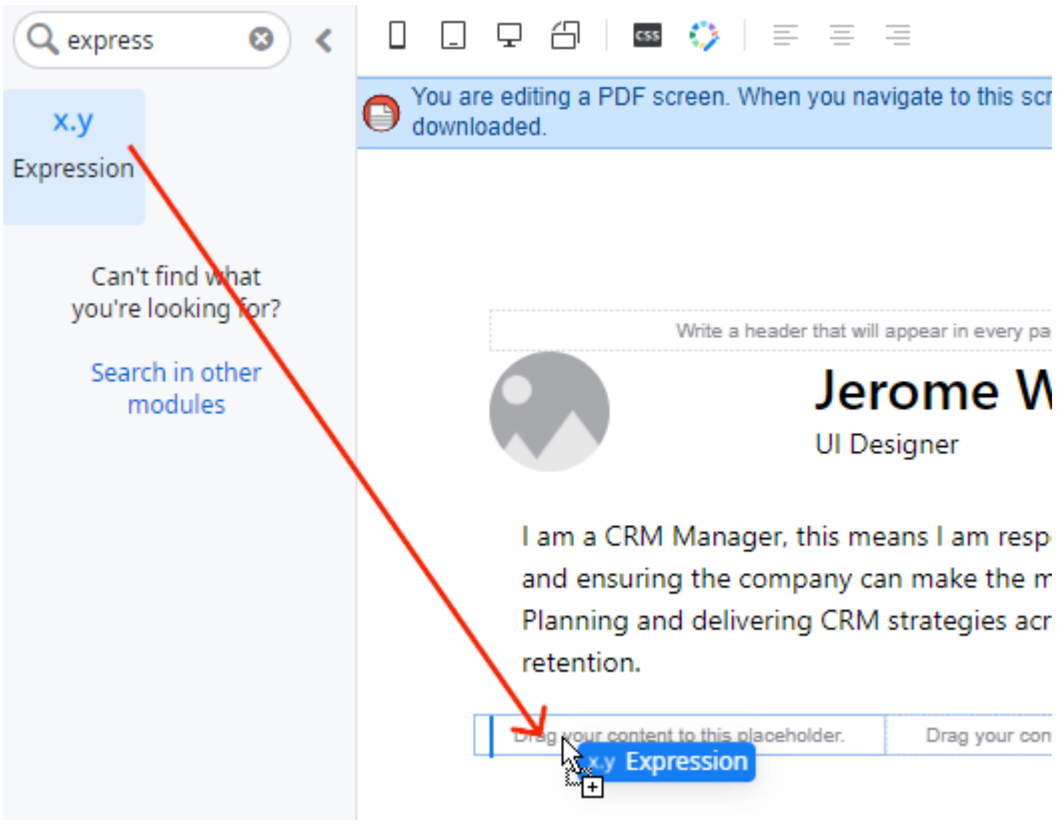
navigate to this screen, a PDF file will be automatically ge

er that will appear in every page. You can include page numbers and

Jerome Wilde
UI Designer

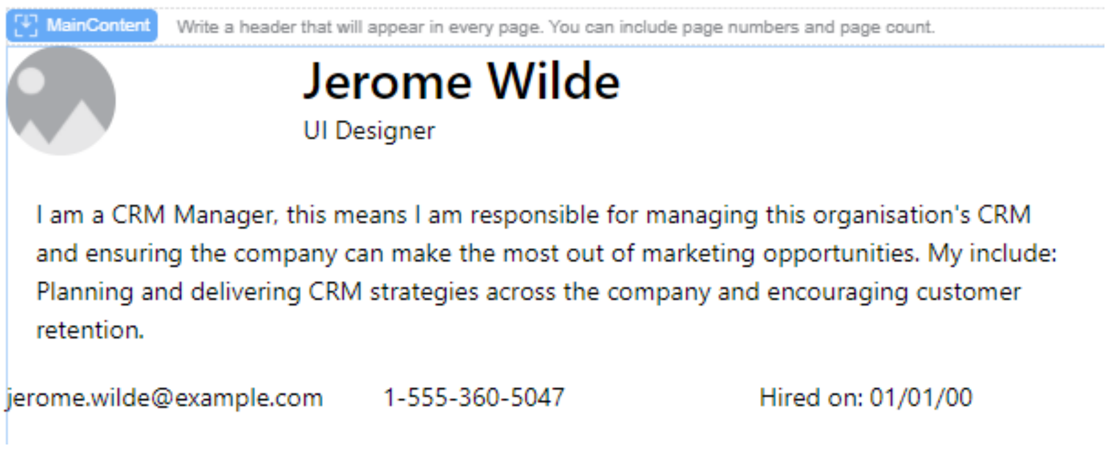
this means I am responsible for managing this organisation's CRM
pany can make the most out of marketing opportunities. My include:
g CRM strategies across the company and encouraging customer

14) Drag an **Expression** and drop it in the first column of the Columns3 element. Set its **Value** to *GetEmployeeById.List.Current.Employee.Email*.


express
x.y
Expression
Can't find what you're looking for?
Search in other modules
CSS
You are editing a PDF screen. When you navigate to this scr
downloaded.
Write a header that will appear in every pa
Jerome W
UI Designer
I am a CRM Manager, this means I am resp
and ensuring the company can make the m
Planning and delivering CRM strategies acr
retention.
Drag your content to this placeholder.
Drag your con
x.y Expression

15) Repeat the previous step with a new Expression and drop it on the second column. Set its **Value** to *GetEmployeeById.List.Current.Employee.Phone*.

16) For the last column, use a new Expression and set its **Value** to `"Hired on: " + FormatDateTime(GetEmployeeById.List.Current.Employee.HiringDate,"dd/MM/yy")`

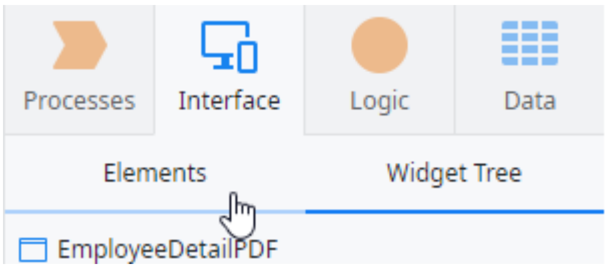

MainContent
Write a header that will appear in every page. You can include page numbers and page count.
Jerome Wilde
UI Designer
I am a CRM Manager, this means I am responsible for managing this organisation's CRM and ensuring the company can make the most out of marketing opportunities. My include: Planning and delivering CRM strategies across the company and encouraging customer retention.
jerome.wilde@example.com
1-555-360-5047
Hired on: 01/01/00

Now we have three columns, with the email, phone and hiring date. The *FormatDateTime* Function allows defining the format we want for a date, in this case the format day/month/year.

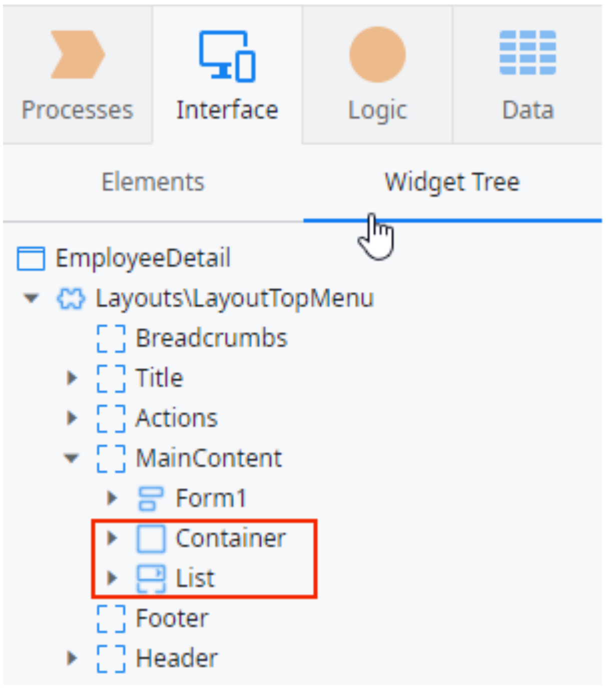
Add Projects to the Resume

At this point, you're only missing the projects. To add that part, you will leverage the UI from the EmployeeDetail Screen.

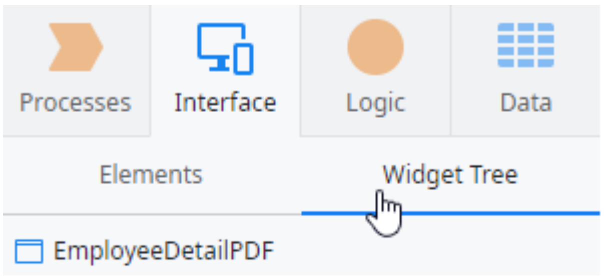
1) In the right sidebar, switch to the Elements tab.



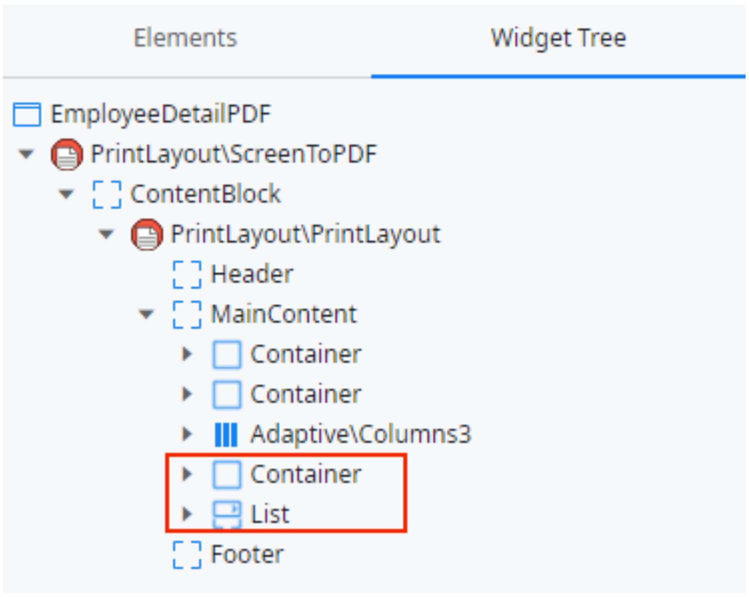
2) Double-click the EmployeeDetail Screen and switch to the **Widget Tree** tab. Expand the MainContent, select the **Container** and **List** elements and copy them.



3) Switch back to the Elements tab, double-click the EmployeeDetailPDF Screen to open it and switch back again to the Widget Tree of that Screen.



4) Paste the elements in the MainContent placeholder. You should have the new elements right after the other elements added in the previous steps.



5) Publish the module to save the latest changes. You're almost done!

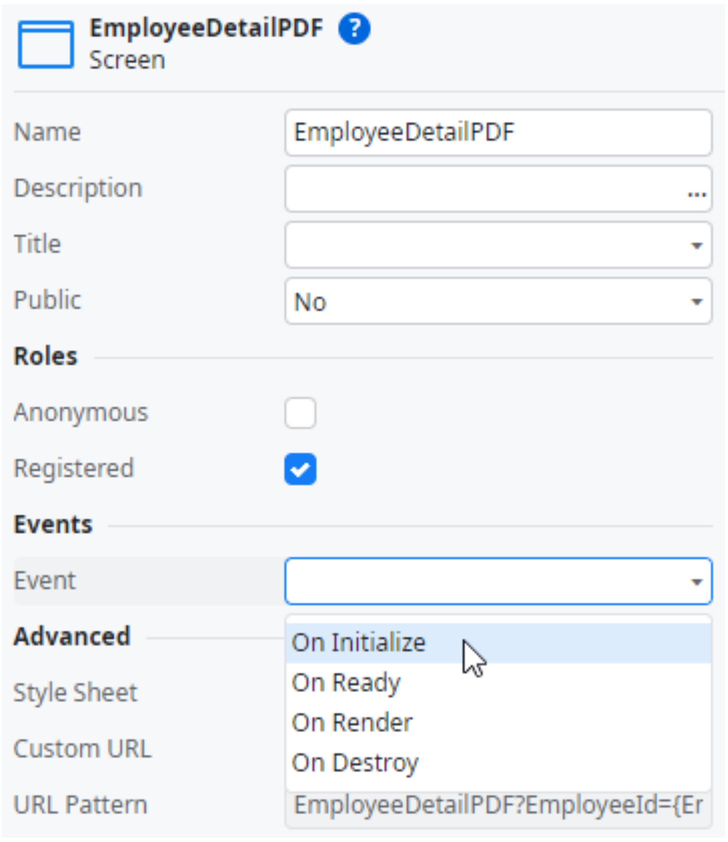
Logic to Create and Download the PDF

Now that you defined the UI for the resume, you now need to define some logic to make sure the user is able to create and downloads the file.

The functionality for downloading the PDF has a small trick. When the user clicks on the Download button, we don't want to have the user navigating to a new Screen, even if right now that's what would happen. The functionality we want is that when the user clicks on the Download button, the PDF is generated and downloaded. Let's build that functionality.

1) Exit from the Widget Tree and switch to the Elements tab.

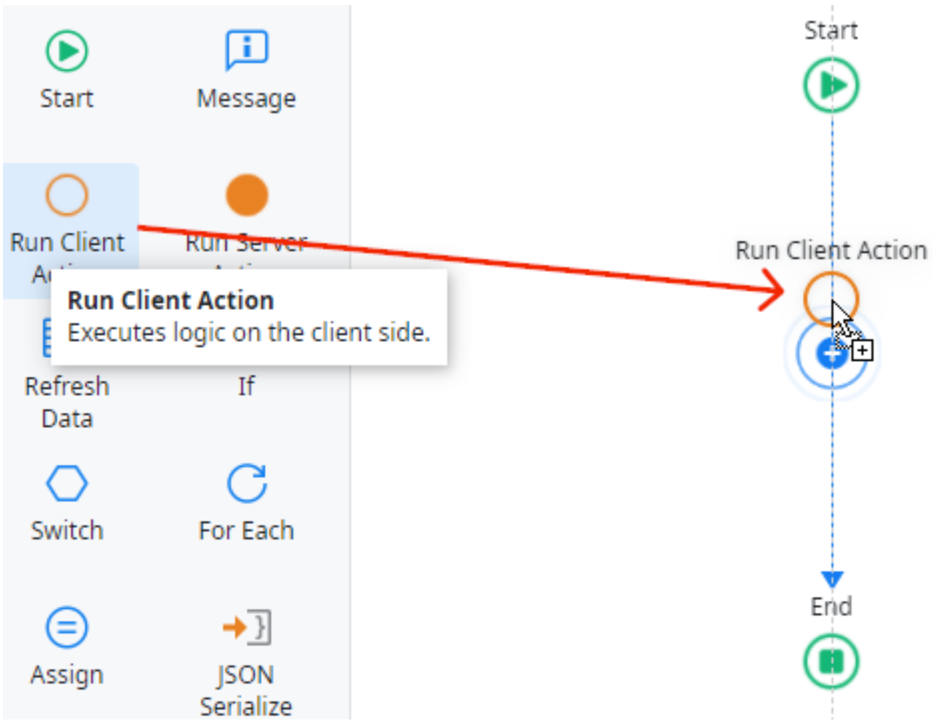
2) On the properties tab of the **EmployeeDetailPDF** Screen, select the **OnInitialize** event.



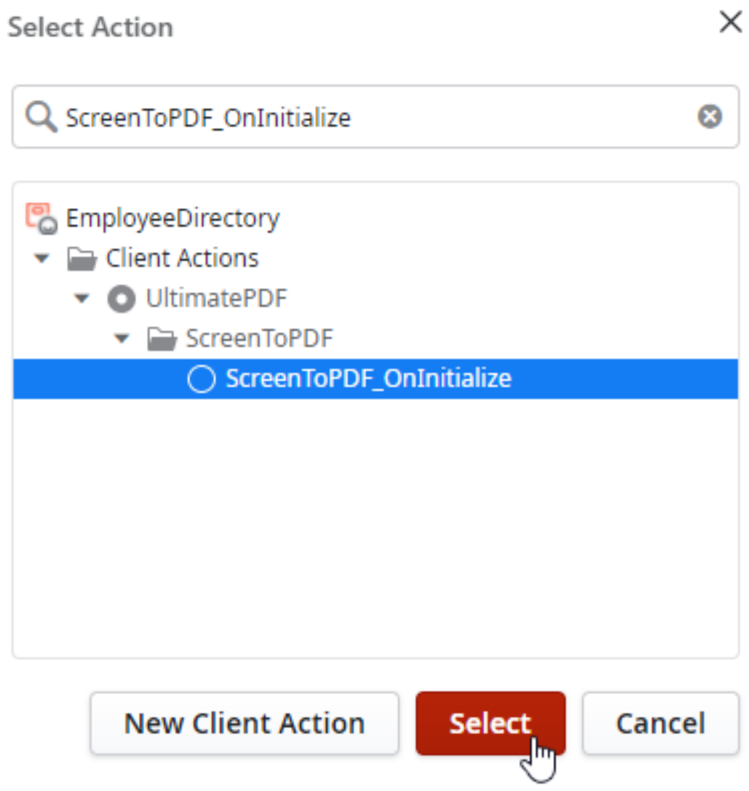
EmployeeDetailPDF	
Screen	
Name	EmployeeDetailPDF
Description	
Title	
Public	No
Roles	
Anonymous	☐
Registered	✓
Events	
Event	
Advanced	
Style Sheet	
Custom URL	
URL Pattern	EmployeeDetailPDF?EmployeeId={Er

Every Screen has a couple of events that trigger some logic automatically. In this case, when the Screen is being initialized, this Action will run. In this case, we want to leverage one of the Ultimate PDF's Actions, to generate and download the PDF. Let's implement that logic.

3) Inside the Action flow, drag a Run Client Action from the left sidebar and drop it to the Action flow.

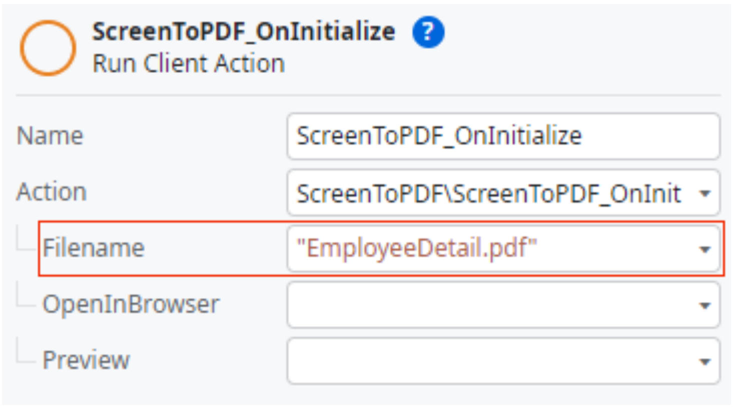


4) Type *ScreenToPDF_OnInitialize* on the search box and select the Action that appears.



This is the Action that will do the heavy lifting and it will allow you to download the PDF file, with the layout defined on the Screen, when you click on the Download button. But, we have some properties that we need to fill.

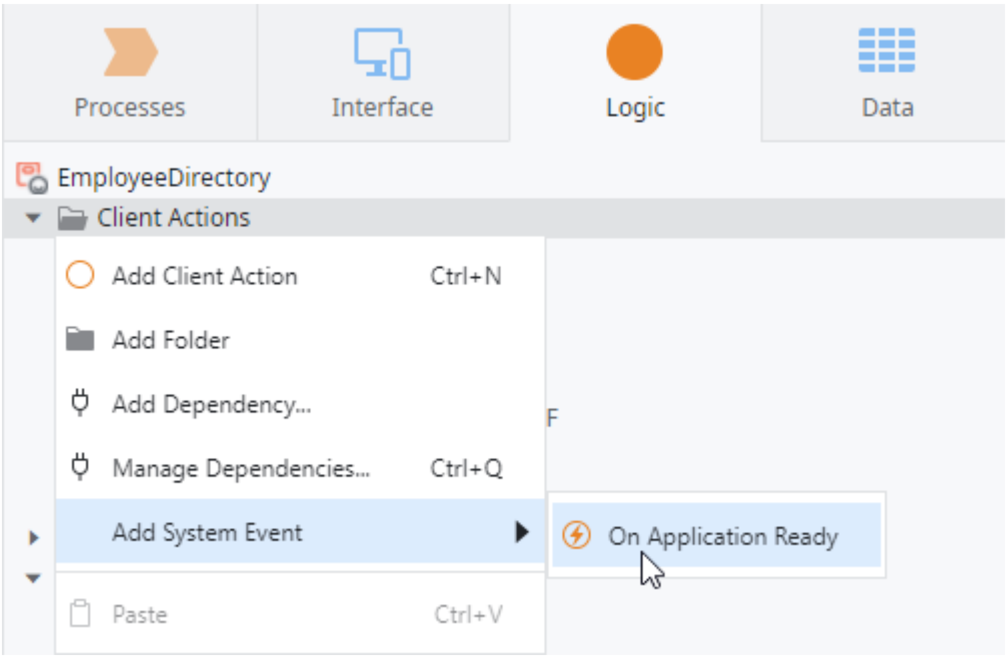
5) In the properties of the recently dragged Client Action, we have an error in the **Filename**. To download a file, we need to define a name for it. Let's set the **Filename** to `"EmployeeDetail.pdf"`.



ScreenToPDF_OnInitialize
Run Client Action

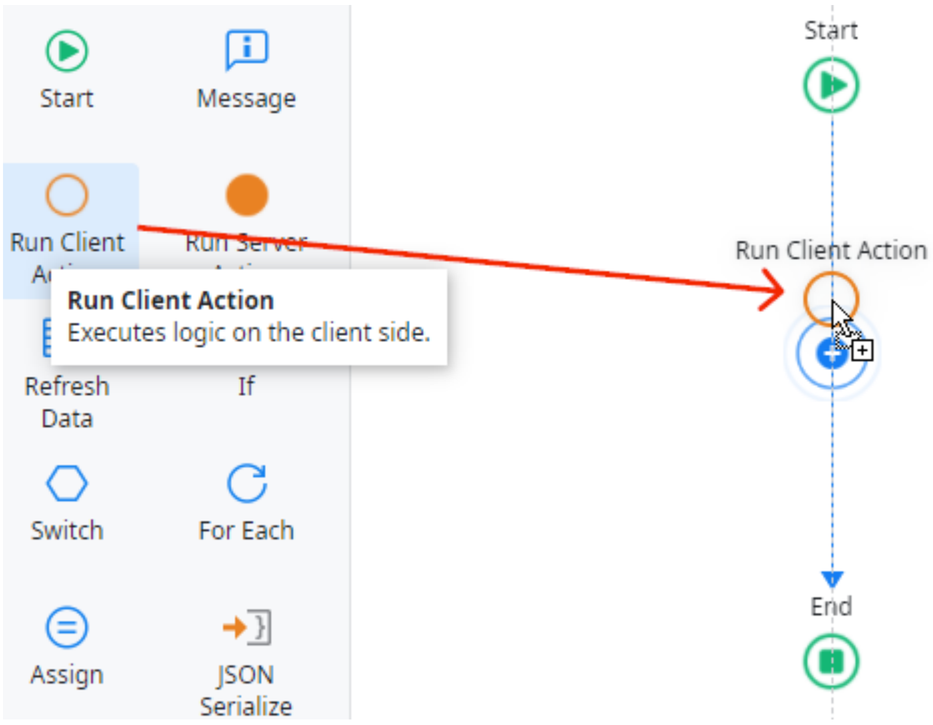
Name	ScreenToPDF_OnInitialize
Action	ScreenToPDF\ScreenToPDF_OnInit
Filename	"EmployeeDetail.pdf"
OpenInBrowser	
Preview	

6) To close this logic, switch to the **Logic** tab on the right sidebar of Service Studio. Right-click on the **Client Action** folder and click on **Add Systems Event**. Select **On Application Ready**.



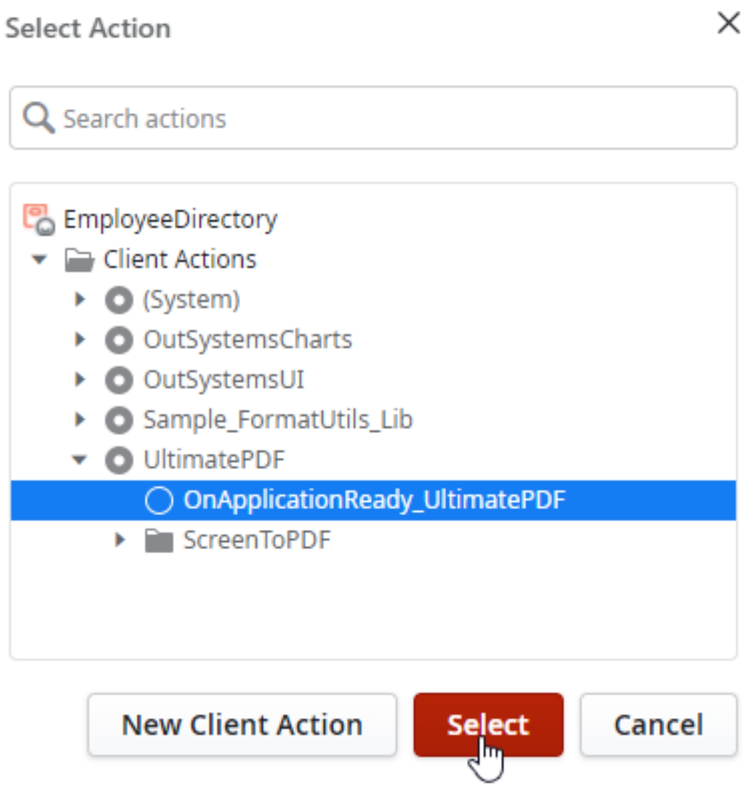
This is a system event that runs automatically when the applications gets ready for the user to use it in the browser. We can leverage to run some logic. In this case, we will again leverage an Action from the Ultimate PDF component that "activates" the Ultimate PDF component to be used in the app.

7) Drag and drop a **Run Client Action** to the flow of the Action that was just created.



Run Client Action
Executes logic on the client side.

8) Select the *OnApplicationReady_UltimatePDF* Action.



9) And you're done. Publish the application and test it in the browser. Navigate to the details of an Employee and try to download the resume.

Wrapping up

Congratulations on finishing this tutorial. With this exercise, you had the chance to see a component from the Forge in action, as well as building some of the UI on your own.

References

If you want to deep dive into the subjects that we have seen in this exercise, we have prepared some useful links for you:

1) Understand Strong and Weak Dependencies

2) How to install or update dependencies

3) OutSystems UI and all its UI components

4) Ultimate PDF

You're done! Thanks for following this first tutorial! See you around!