



SAP Cloud Platform - Technical Workshop

Exercise 1 - Create your UI using SAP BUILD

Summary of Exercise:

In this exercise you will create a new project in SAP BUILD. You will create the user interface (UI) for a Single-Page-Application (SPA) that will be used in later exercises. This SPA will later be a translation application, leveraging REST services from the SAP API Business Hub and SAP Cloud Platform.

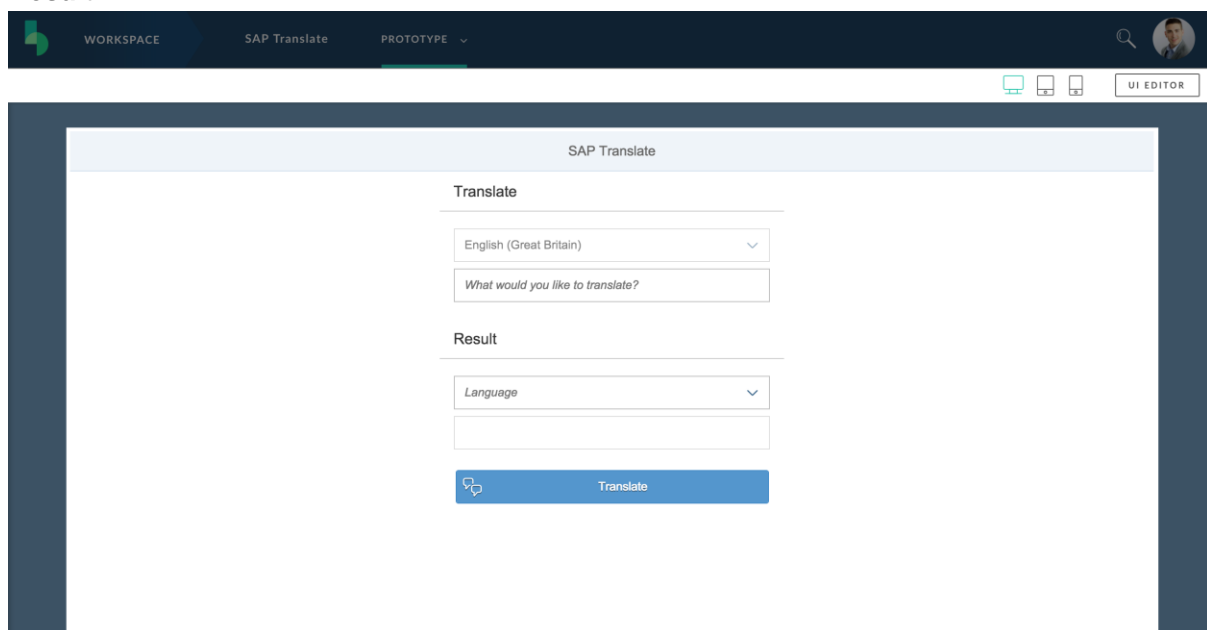
What will you use?

- [SAP Cloud Platform](#)
- [SAP BUILD](#)

What will you learn?

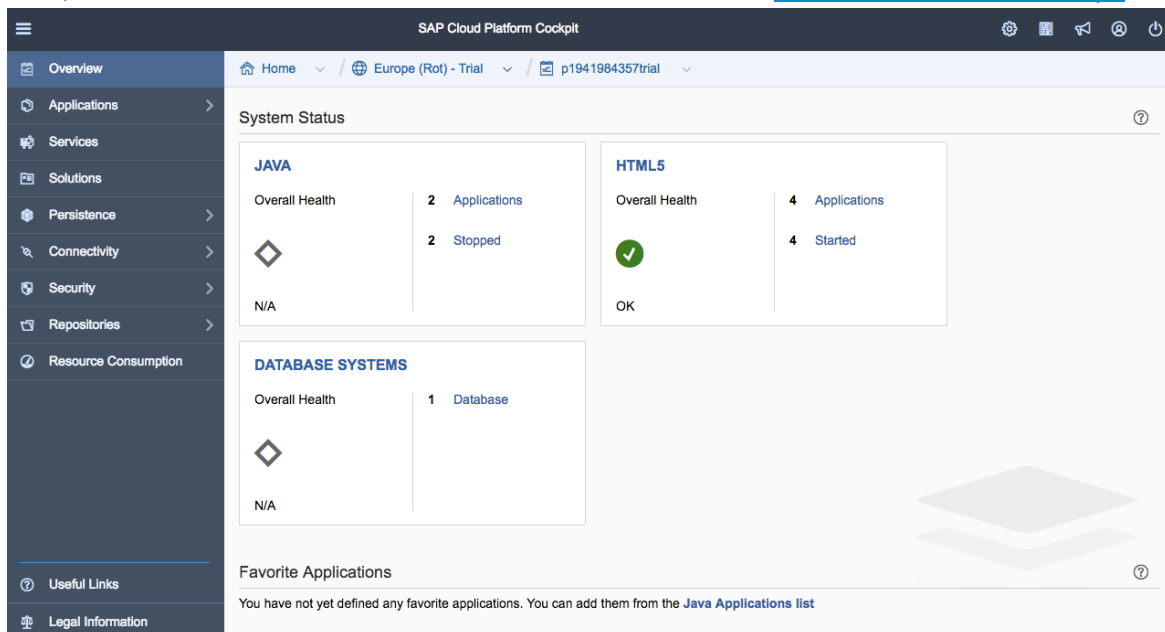
- Build a responsive high-fidelity user interface in SAP BUILD
- Learn about the Dropdown List, Input, and Button controls
- Preview a prototype and test the interactions
- Publish a study
- Export the underlying code of your Project

Result

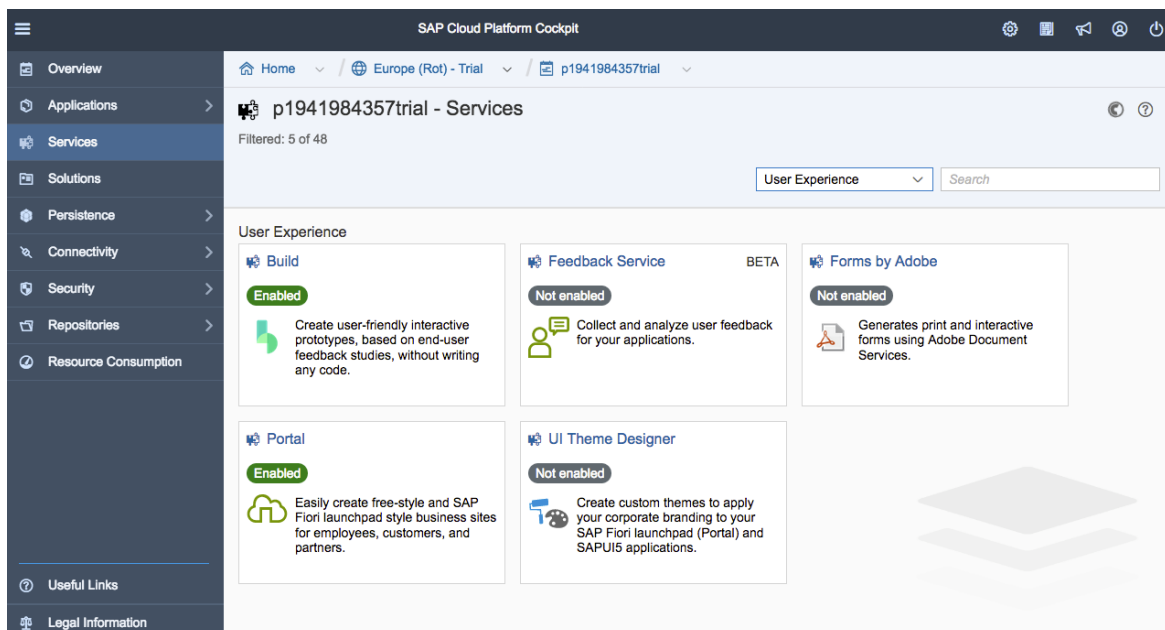


Let's get started!

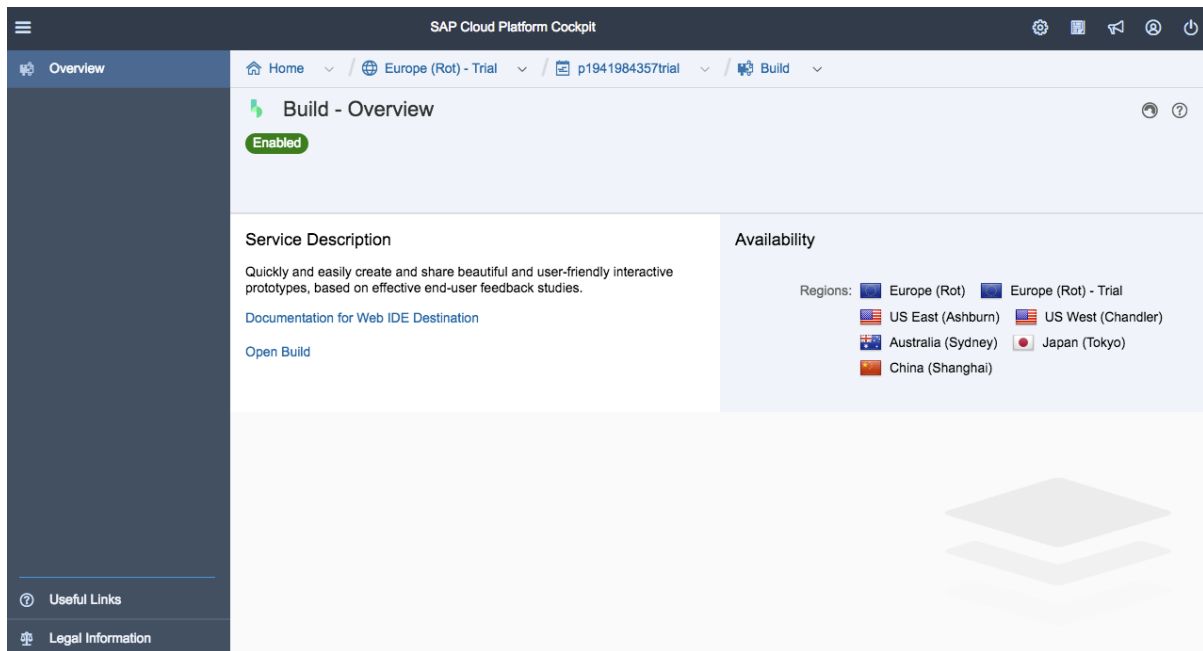
First, we need to access SAP BUILD from our Services in the [SAP Cloud Platform Cockpit](#).



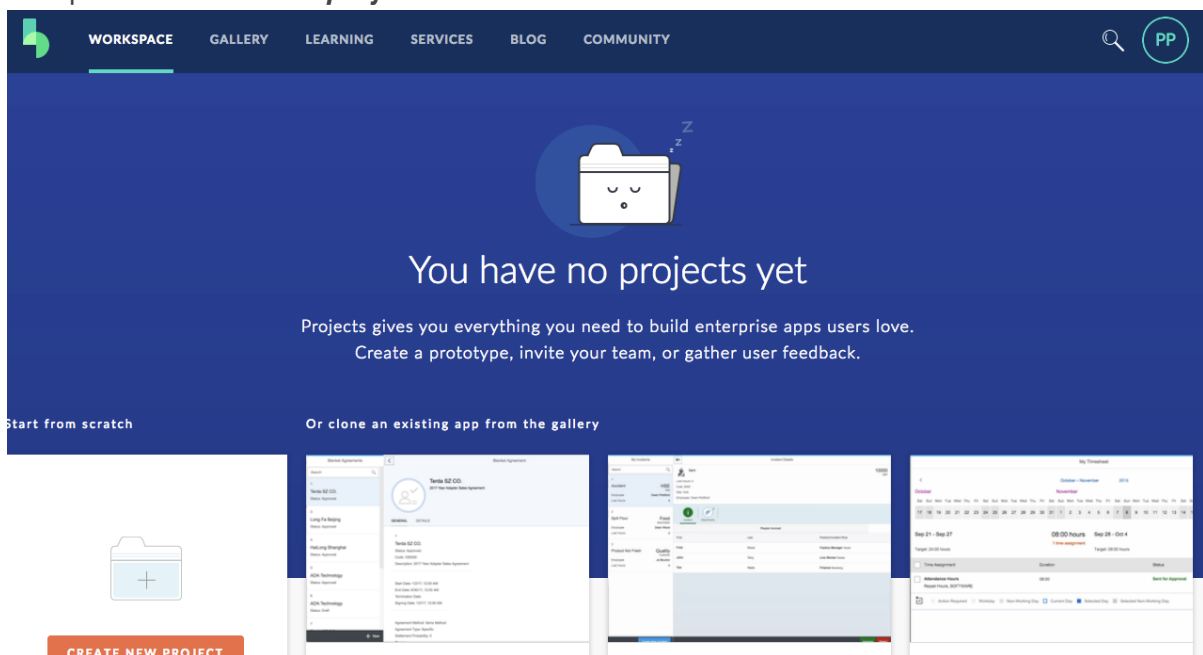
From the left-hand side menu, Select **Services** and filter to the **User Experience** category. Select **Build**.



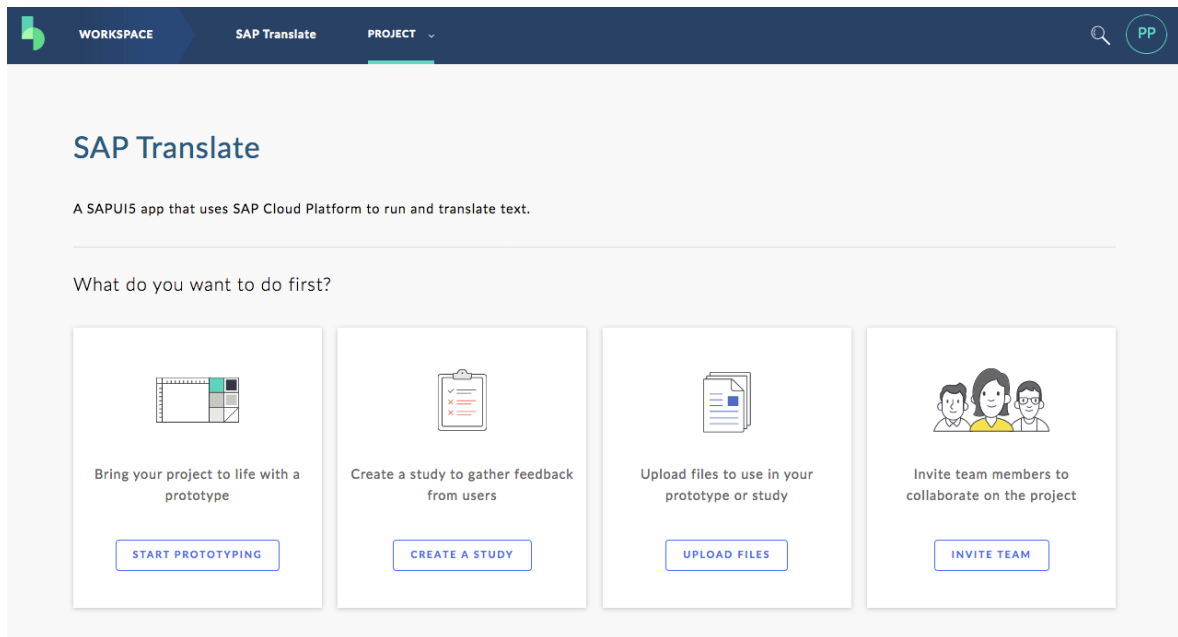
If your service is not enabled, select the big Enable button. Once your Build service is enabled, select **Open Build**. This will launch a new window with the SAP BUILD tool.



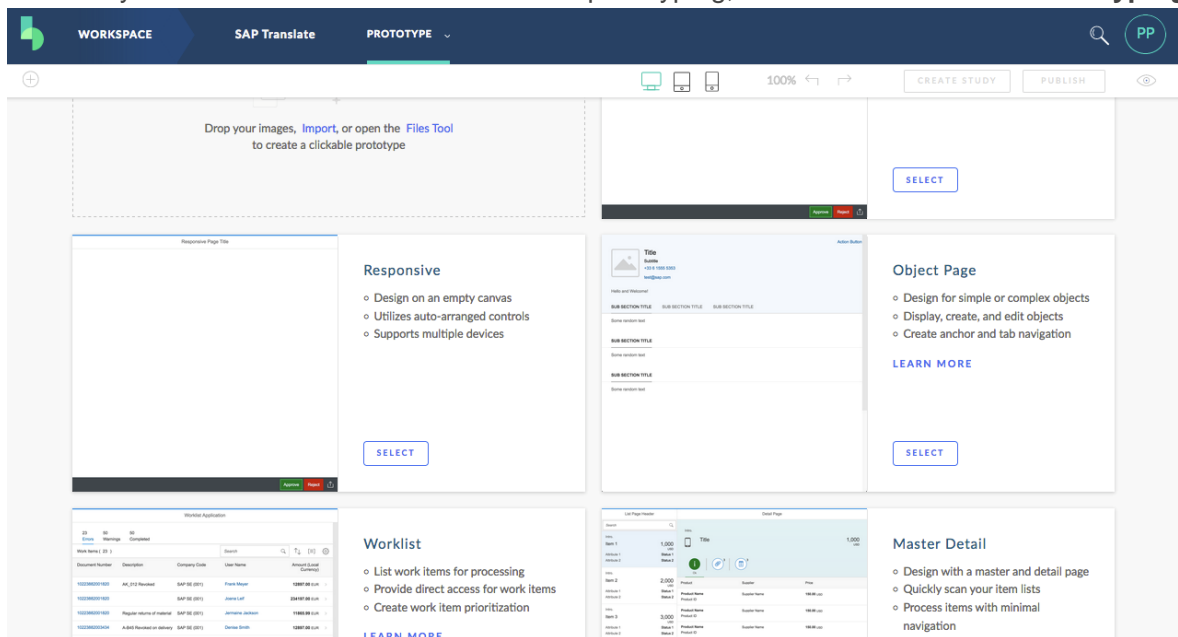
Once open and signed in, you will see your empty workspace; similar to what you see below. Select the option to **Create new project**.



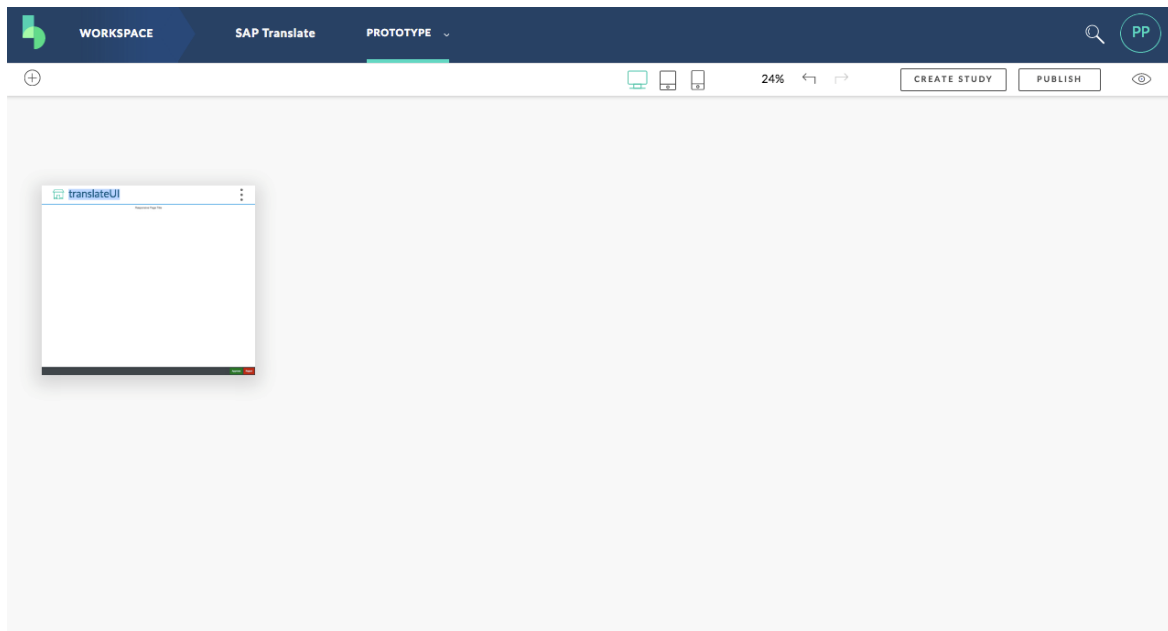
You will be asked to setup the basic information for this project. **Name** your project “SAP Translate” and the **description** as follows: “A SAPUI5 app that uses SAP Cloud Platform to run and translate text”. Once completed, select **Create Project** to get started.



What do you want to do first? We want to start prototyping, of course! Select **Start Prototyping**.

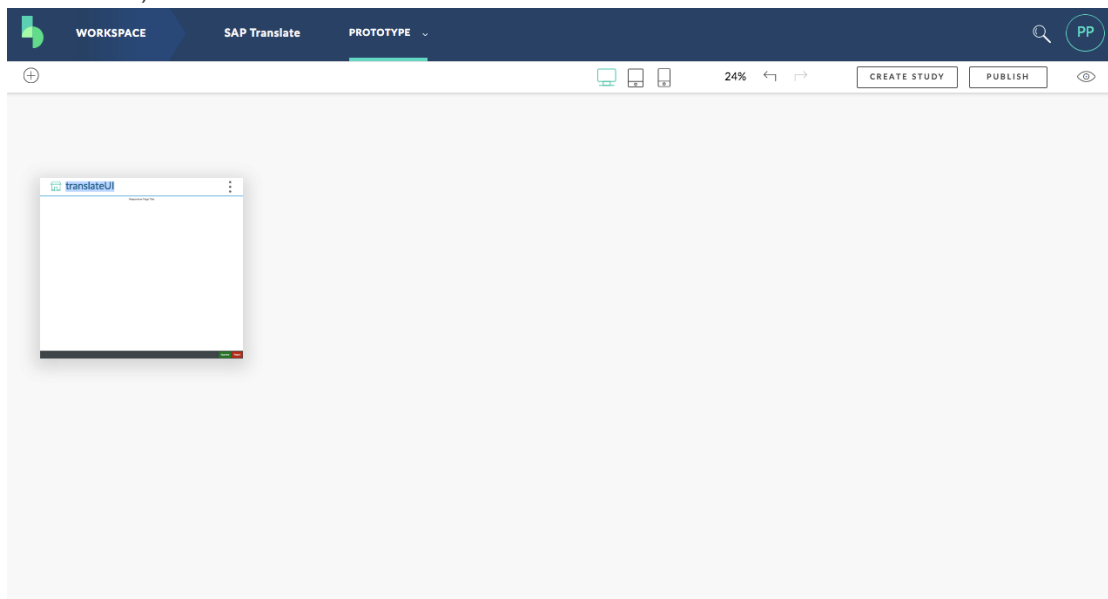


As I mentioned above, we want to build a responsive application UI. We have many other options including Freestyle. We also have application templates that include Object Page, Worklist, and Master Detail. For our application, we are going to select **Responsive** because we want this to work on a mobile device later on.



This will take us to the BUILD workspace. We have a single responsive page ready for us to prototype with. Simply double-click on the default *Page 1* text and change it to “**translateUI**”. **This is very important. Although it is a very minor change for us, this changes the name of the applications view and controller behind the scenes!**

Next, hover over the translateUI page and select **Edit Page** (or select the 3 vertical dots and select **Edit**).



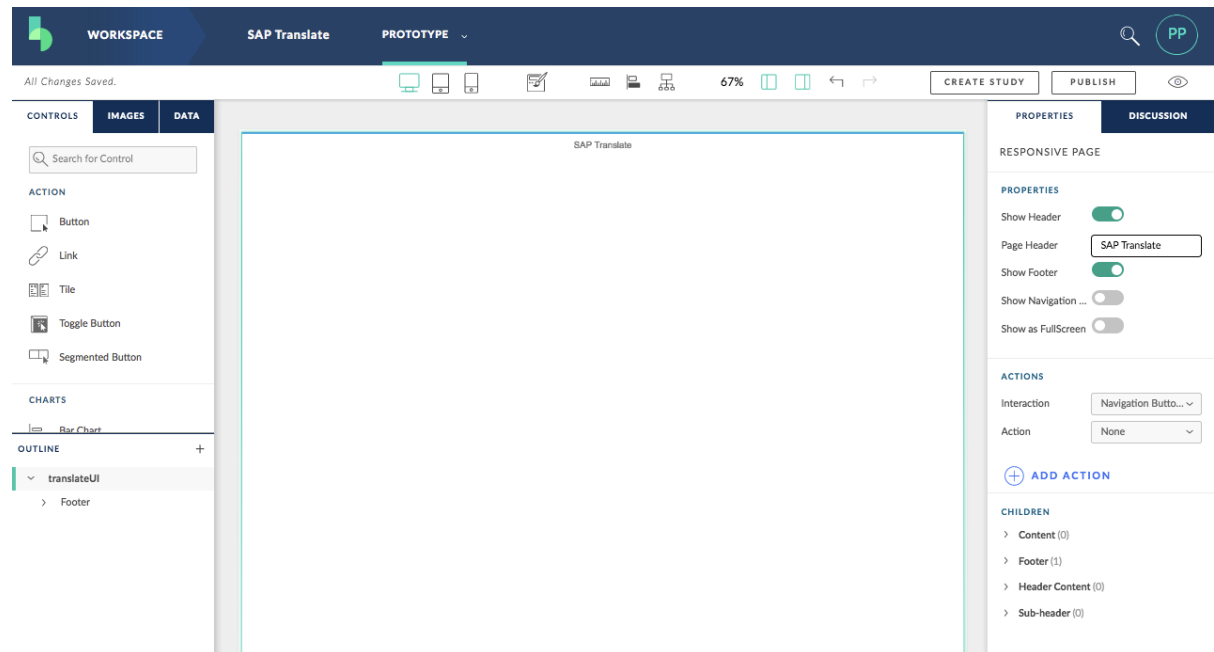
Add Controls to the Application

Now that we're in the empty shell of our application, let's start by changing the name of the application by changing the **Page Header**. Change it on the right-hand side properties; Responsive Page Title to **SAP Translate**. We will also **turn off our Footer** by turning off the switch on the right-hand side.

You will notice in the UI Editor, we have a lot of options. On the left, we can add many controls (from Controls), upload and add images (from Images) and even add mock data (from Data). Because of the user-friendliness of SAP BUILD, you can easily drag-and-drop any of these onto

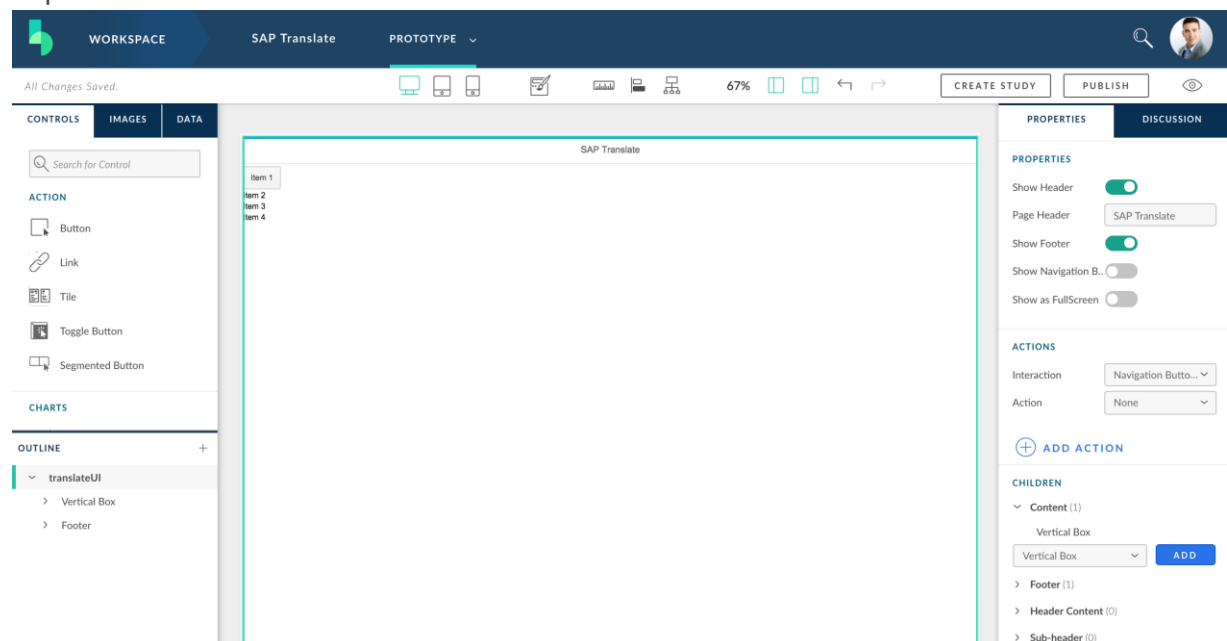
your application. But because the application we are building requires many details, we will add many of our controls from the **Properties** on the right-hand side of our screen.

In the UI Editor, each **Control** has **properties** defined in the right panel that you can customize for your prototype. **Properties** define the look and behavior of each UI control. When you select a control on your prototype page, the control properties and actions display on the Properties tab on the right-panel. Some controls can include additional properties, such as page children, size, color, and more.



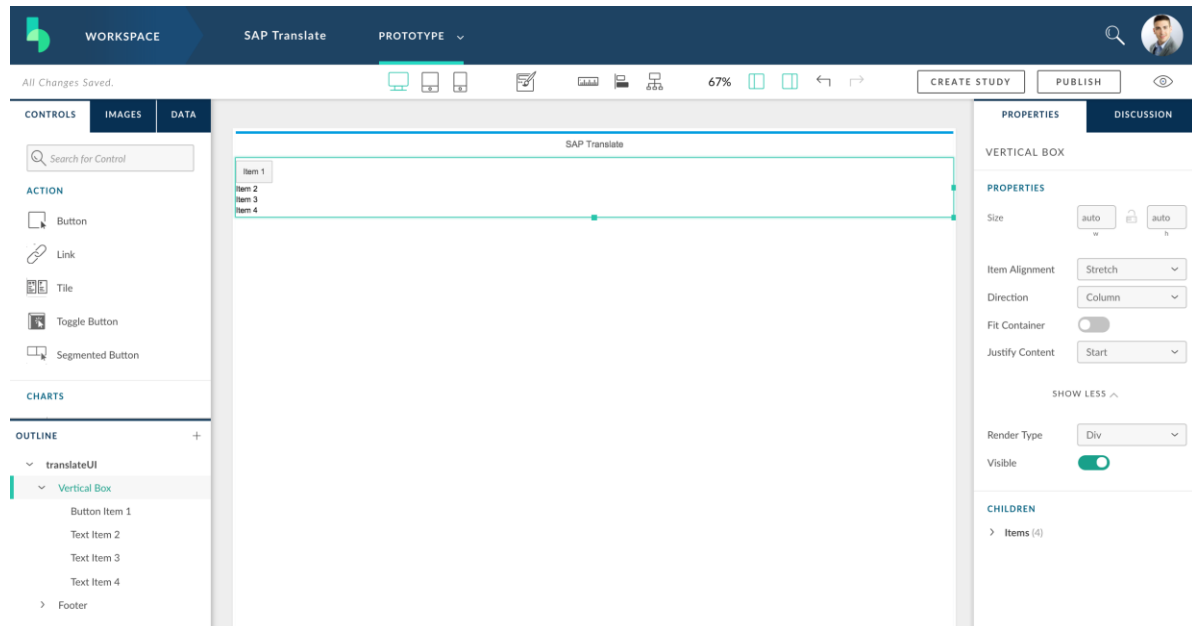
You should also notice, many of the controls have relationships. If you select the object and its relationships, you can modify the object and its children. For example, our page that is selected has 4 (four) children, including Content, Footer, Header Content, and Sub-header.

We'll add controls to our page in this exercise, so that falls under the category of **Content**. Expand **Content** and add the **Vertical Box** control from the selection list.

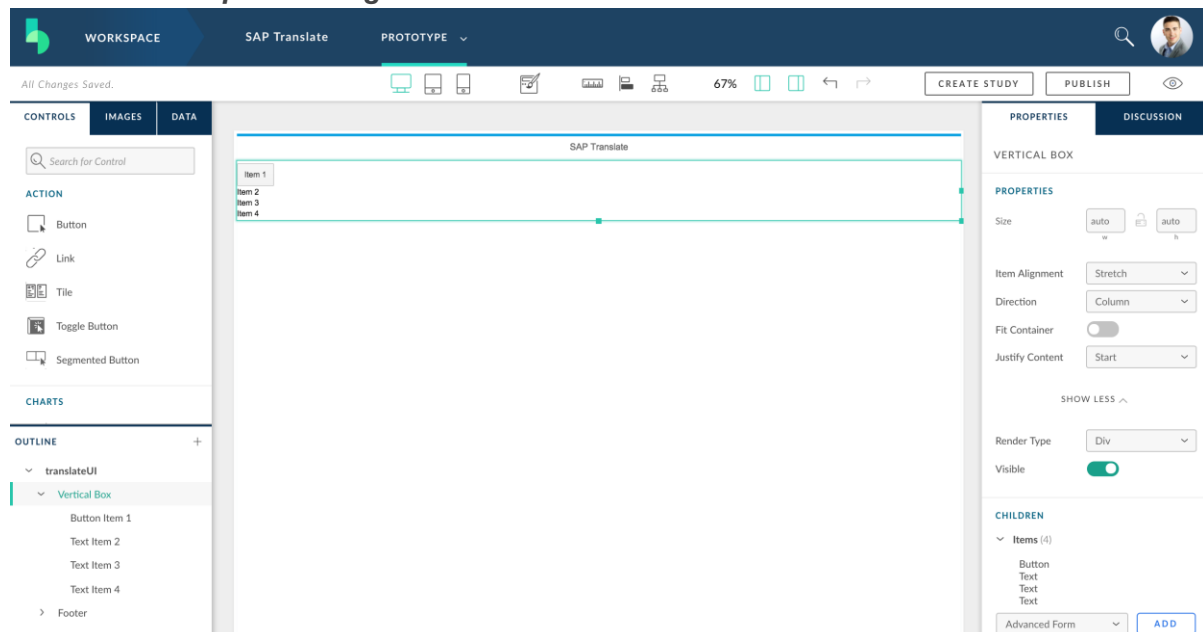


The Vertical Box control builds the container for a vertical flexible box layout. The Vertical Box is a convenience control, as it is just a specialized FlexBox control. This means that we can add Controls and objects inside of it.

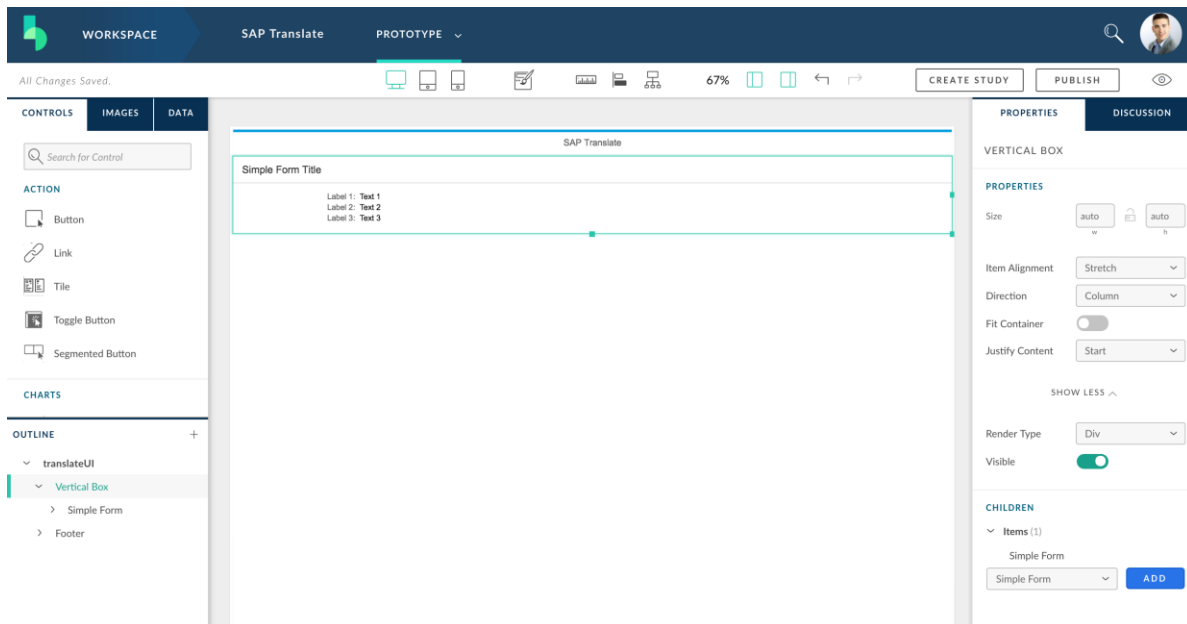
Now, Select “**Vertical Box**” from below “Content”.



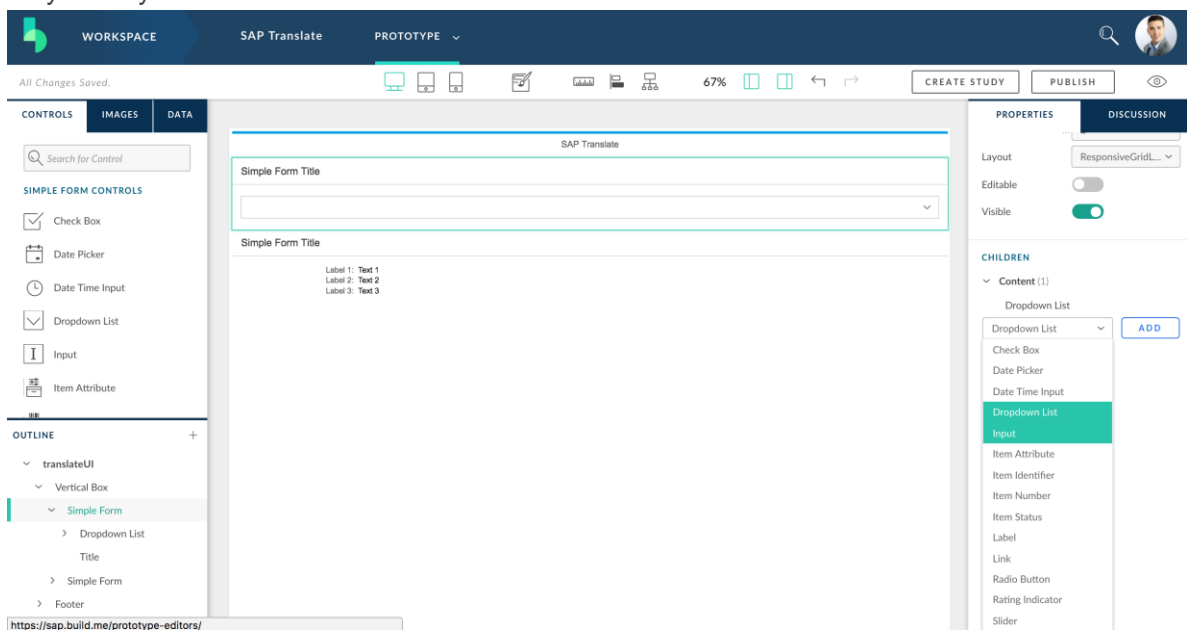
The Vertical Box had 4 (four) children by default. We don't need these so we're going to delete them. Expand the Items list below Children to view the children in more detail. If you hover your cursor over each individual control (i.e. “Button” or “Text”), a trash can will appear. Select the trash can to **delete all pre-existing children** in the Vertical Box control.



Now that we've made room in our Vertical Box, we can begin adding Controls inside of it. From the selection list, select and “Add” 2 (two) **Simple Form** controls into the Vertical Box (under Children>Items).

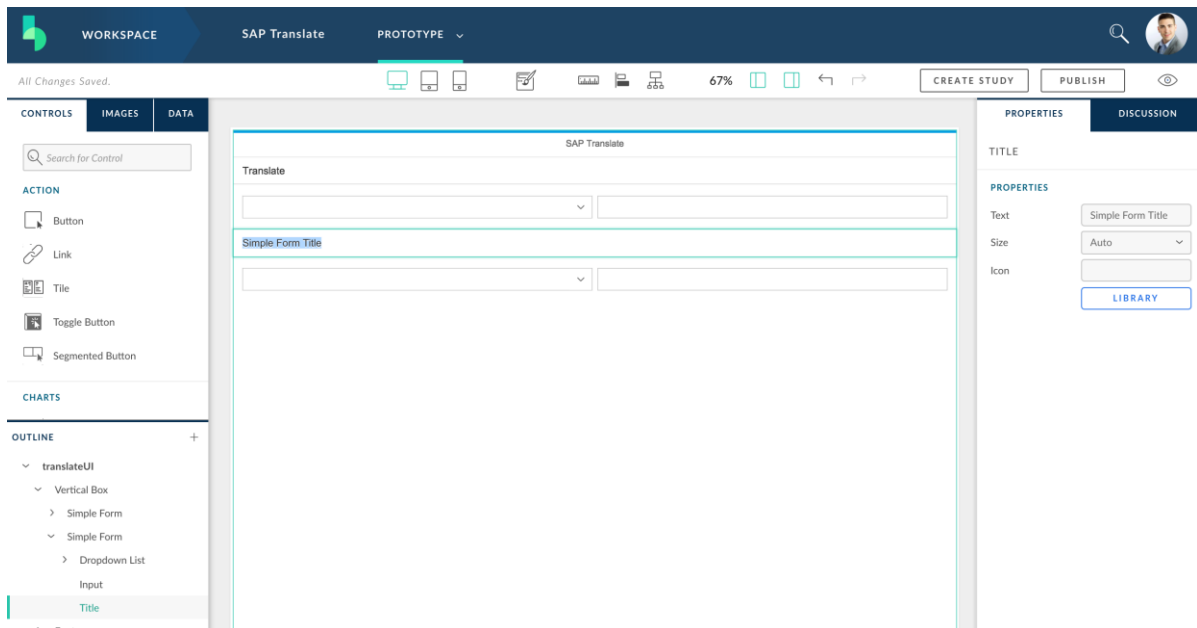


As we saw with the Vertical Box, the BUILD tool likes to add content into the controls. For example, the Simple Form has 3 (three) labels and 3 (three) text controls that we don't need. You may delete with the method used previously or you may select them with your cursor and use the “Delete” key on your keyboard to delete them.

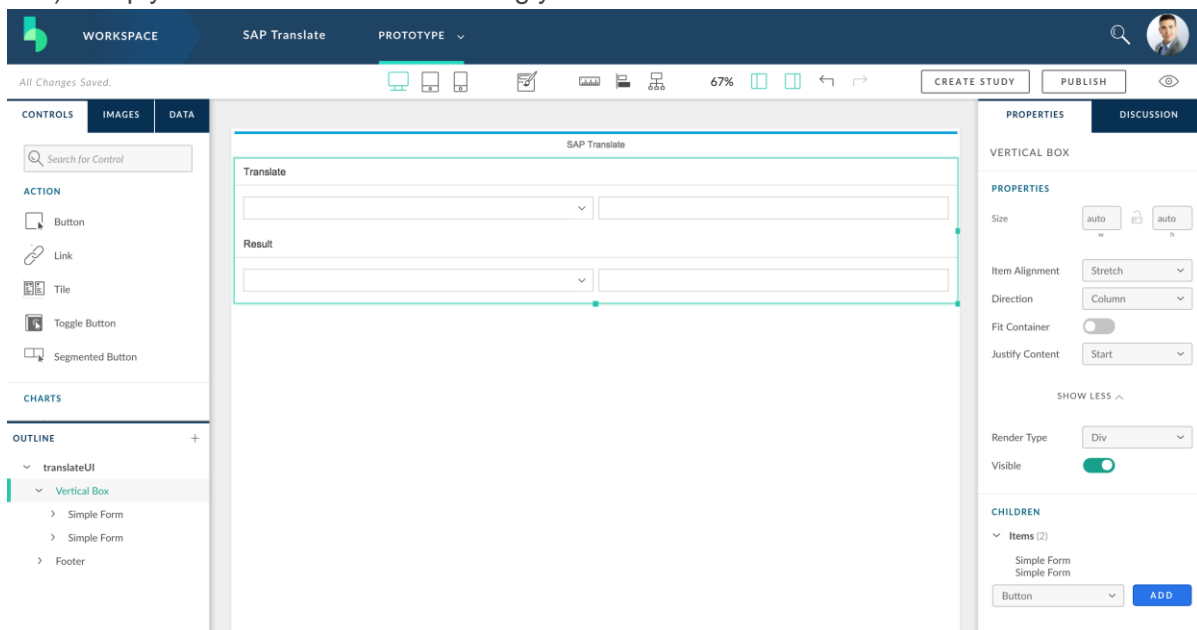


Once deleted, you can add the **Dropdown List** control and the **Input** control from the selection list. Simply open the selection list, find/select the control, then click **Add**.

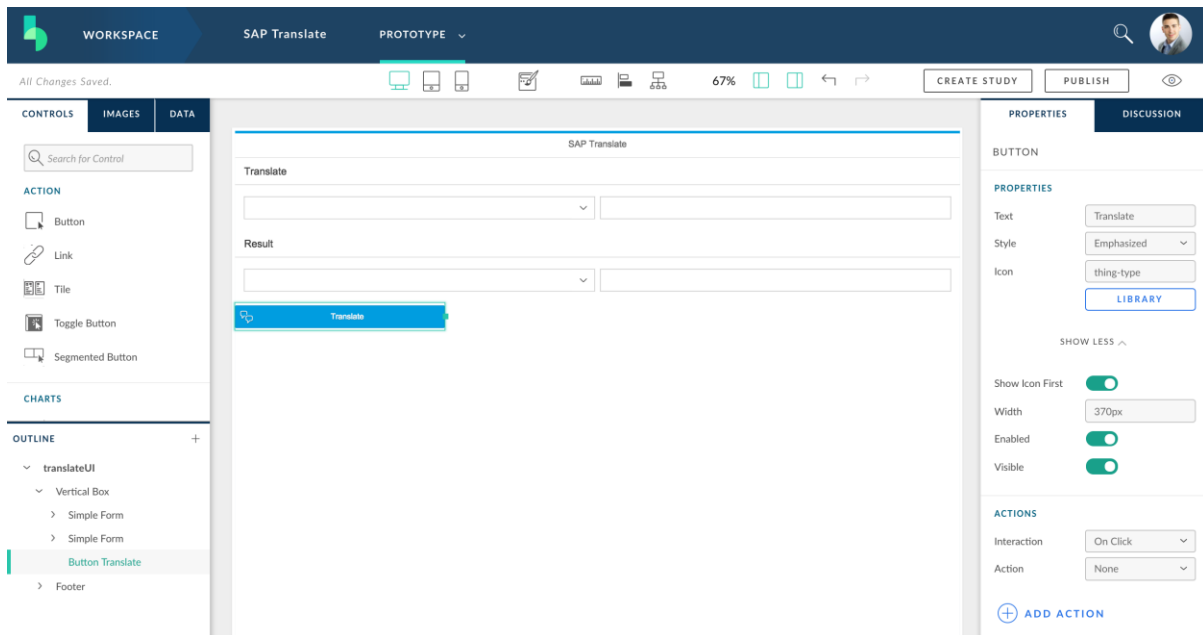
(Important) Please Note: As there is no back-button to navigate to specific depths of the application's UI, use your cursor and left-click on the control you want to edit. For example, if you want to add these controls to the Vertical Box, you select the Vertical Box with your cursor to view the Properties and Item/Children. If you want to edit the Dropdown List control, you select the Dropdown List with your cursor to view the Properties and Items/Children.



Change the Titles of our Simple Forms to **Translate** (on the top one) and **Result** (on the bottom one). Simply double-click on the title using your cursor.



Once modified, it should look like the above screenshot. Next, select the Vertical Box control using your cursor and left-click. You will notice that our Vertical Box has Items (Children) on the left-hand Properties pane. From the selection list, let's add a **Button** control by selecting **Button** from the list and clicking "**Add**".



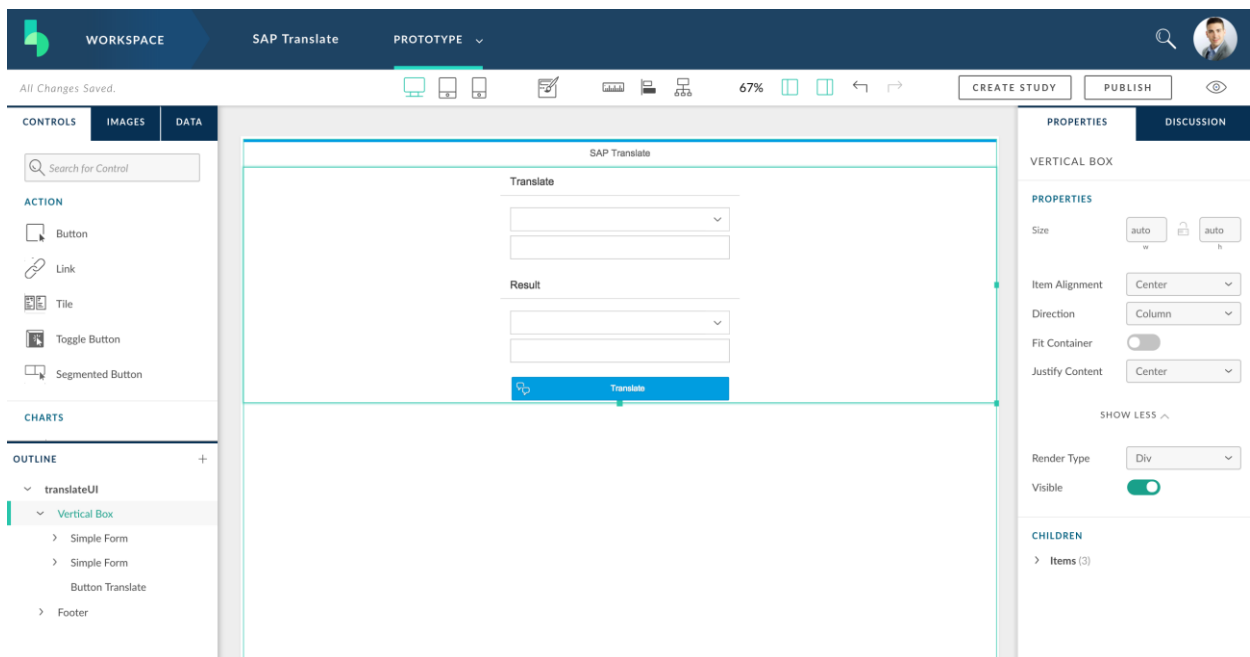
Select the Button with your cursor, and modify the Properties to the following:

Text: **Translate**

Style: **Emphasized**

Icon: (**locate the “thing-type” icon from Library**)

Width: **370px**



Now, select your Vertical Box using your cursor and modify the Properties to the following:

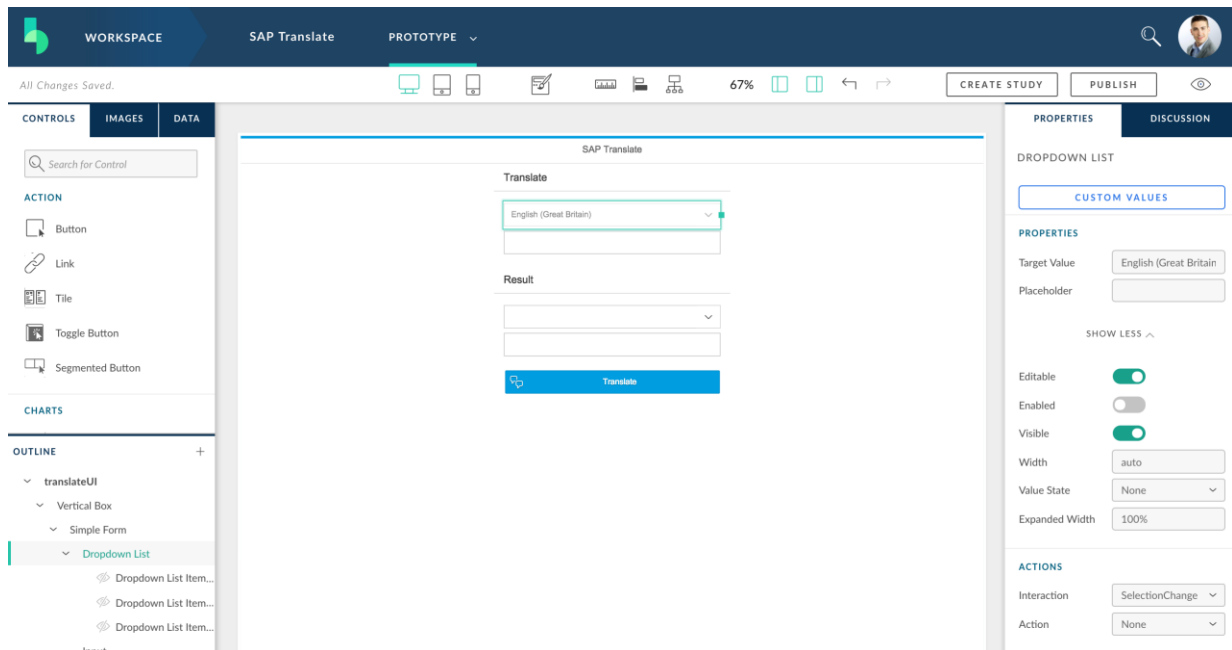
Item Alignment: **Center**

Direction: **Column**

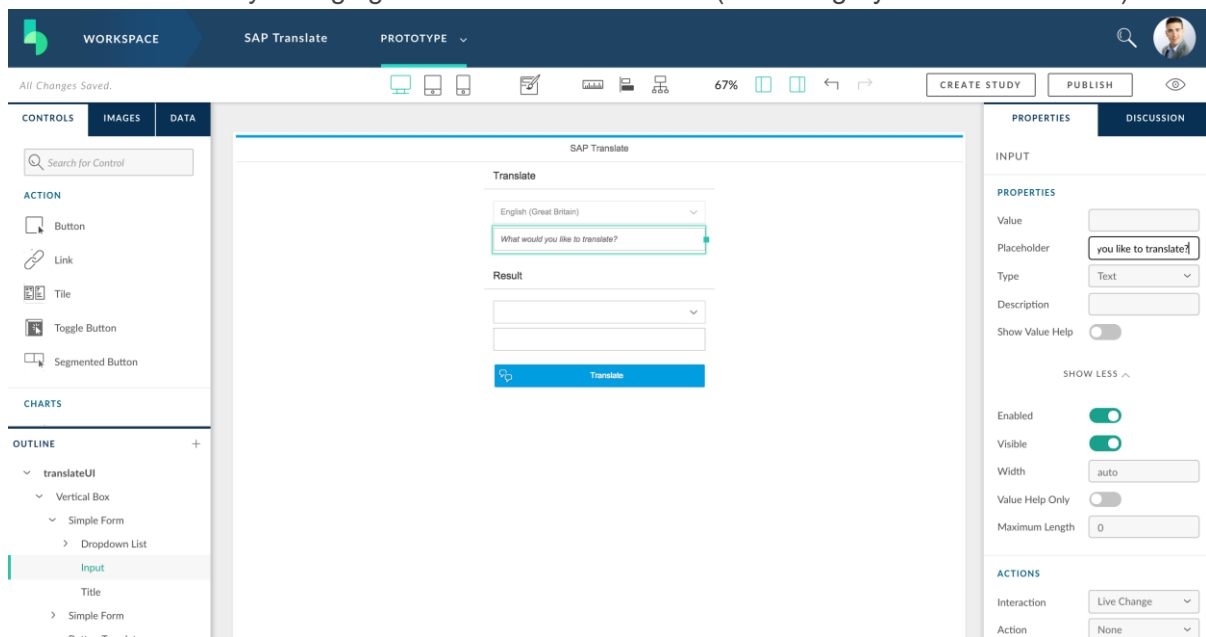
Justify Content: **Center**

Configure Details of the Application

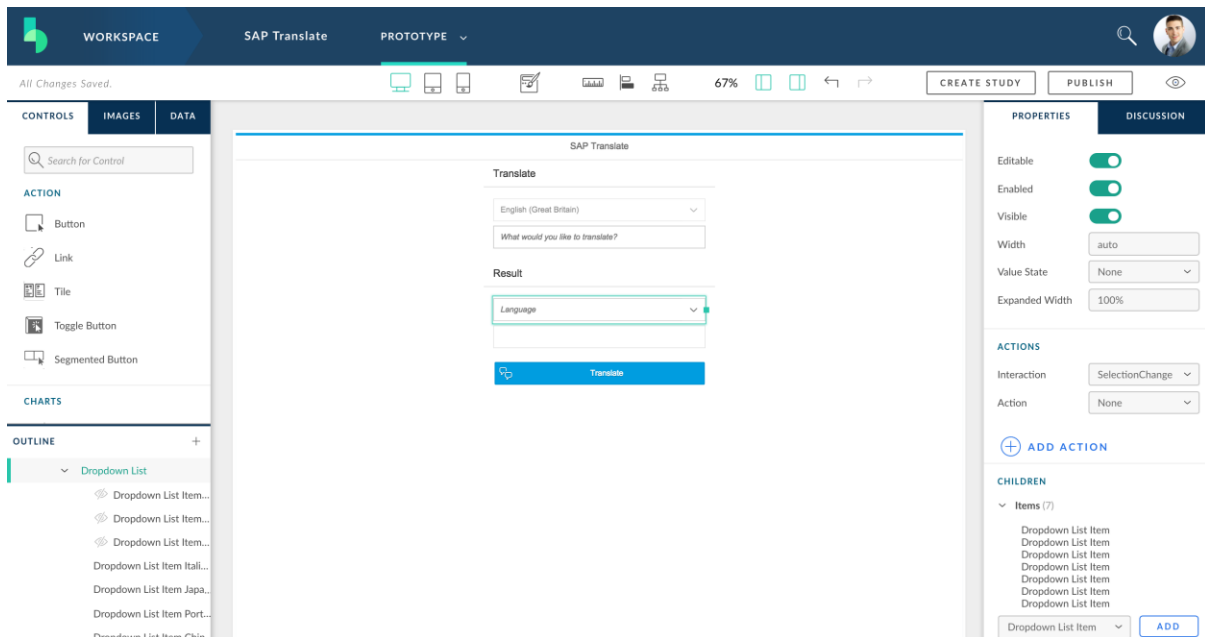
You'll now notice that we've have most of our UI finished. We have only a few remaining details!



To begin our detailed modifications, select the top Dropdown List control (below *Translate*). In the Properties pane, change the Target Value to **English (Great Britain)**. Also, verify this control is also **not enabled** by changing the **Enabled** switch to **off** (it will be greyed out when it's off).



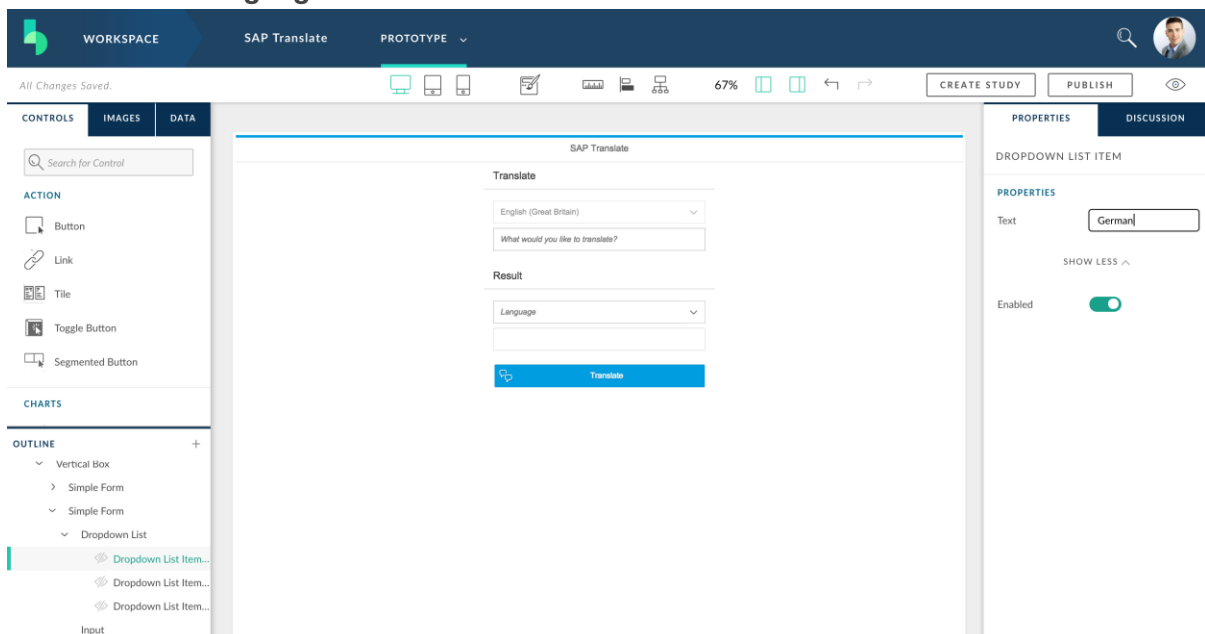
Next, select the Input control (in the *Translate* Simple Form). In the Properties pane, change the Placeholder to **What would you like to translate?**.



Now, we need to **edit the Children/Items** in the *Result* Dropdown List. By default, this comes with 3 (three) Dropdown List Items. We will need to add 4 (four) more to have **7 (seven) total** (notice the screenshot). Simply select **Dropdown List Item** from the selection List and click **“Add”**.

Alternatively, after you add the 7 (seven) total items to the Dropdown List, you may select **Custom Values** from the top of the left-hand **Properties** pane to mass edit the items.

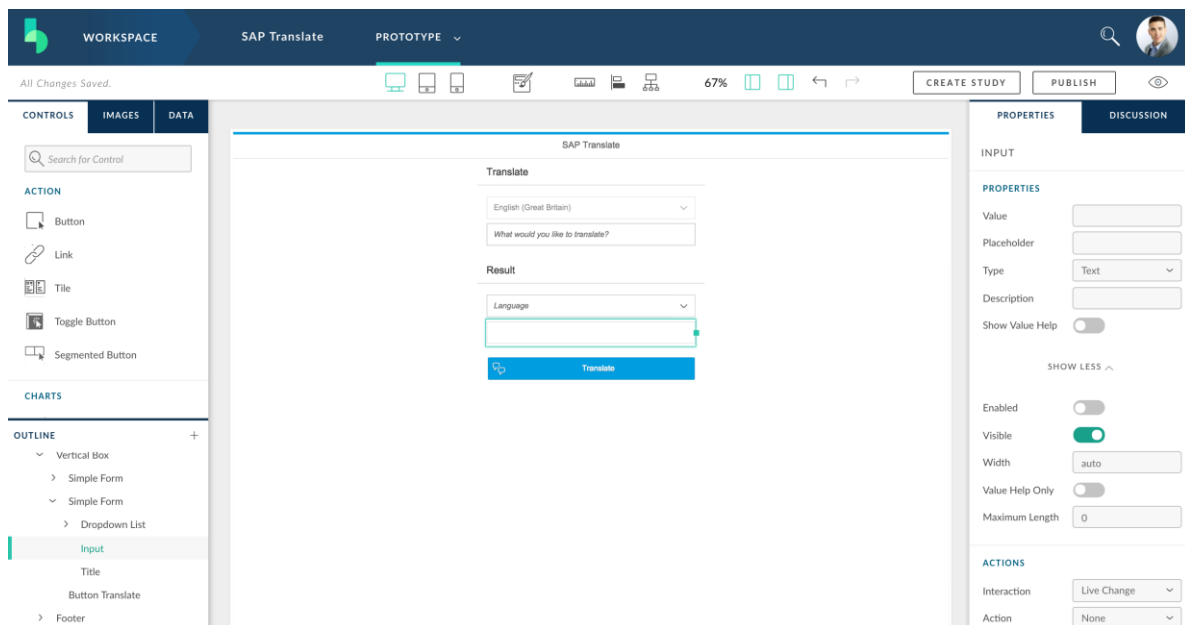
Lastly, select the Dropdown List (in the *Result* Simple Form). In the Properties pane, change the Placeholder to **Language**.



(Important) Please Note: This process may be unpleasant as this control is not made for mass editing in SAP BUILD. You will need to repeatedly select your Dropdown List control and navigate to each individual Dropdown List item to edit. Also, they must be in the same order as listed below. This will help setup the next exercise. Don't worry, your hard work will pay off!

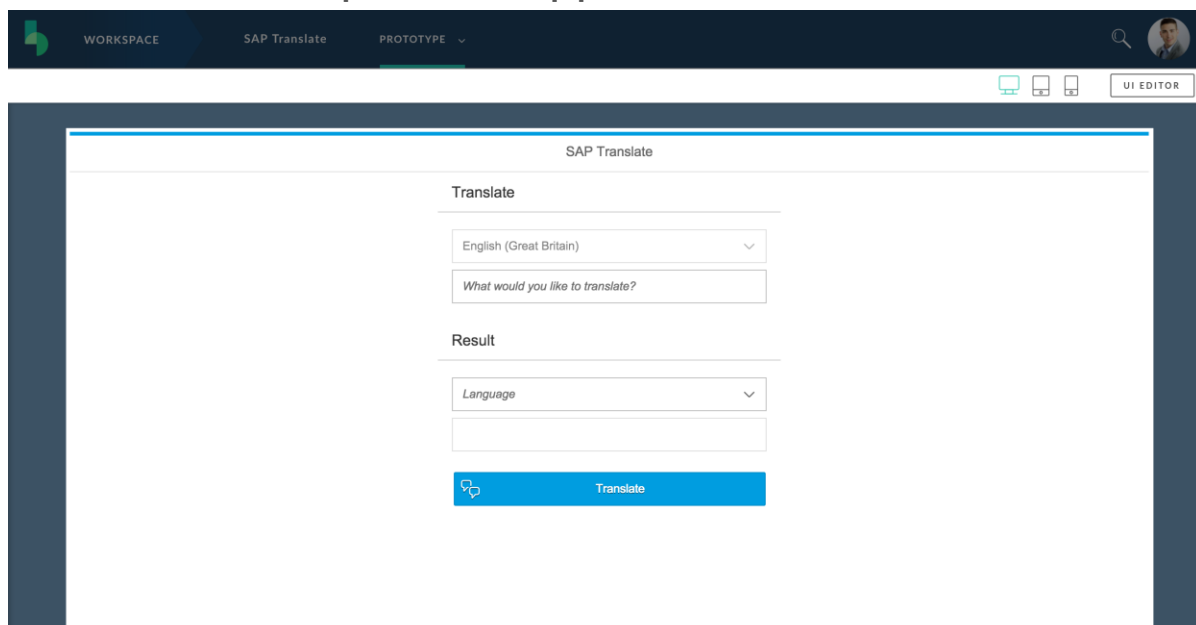
Your items (Dropdown List Items) should be as follows (and in this order). Simply **select** the **Dropdown List Item** and change the **“Text”** to what is described below. If they are not in this order, please re-arrange them.

1. German
2. Spanish (Spain)
3. French (France)
4. Italian
5. Japanese
6. Portuguese (Brazil)
7. Chinese (Simplified)

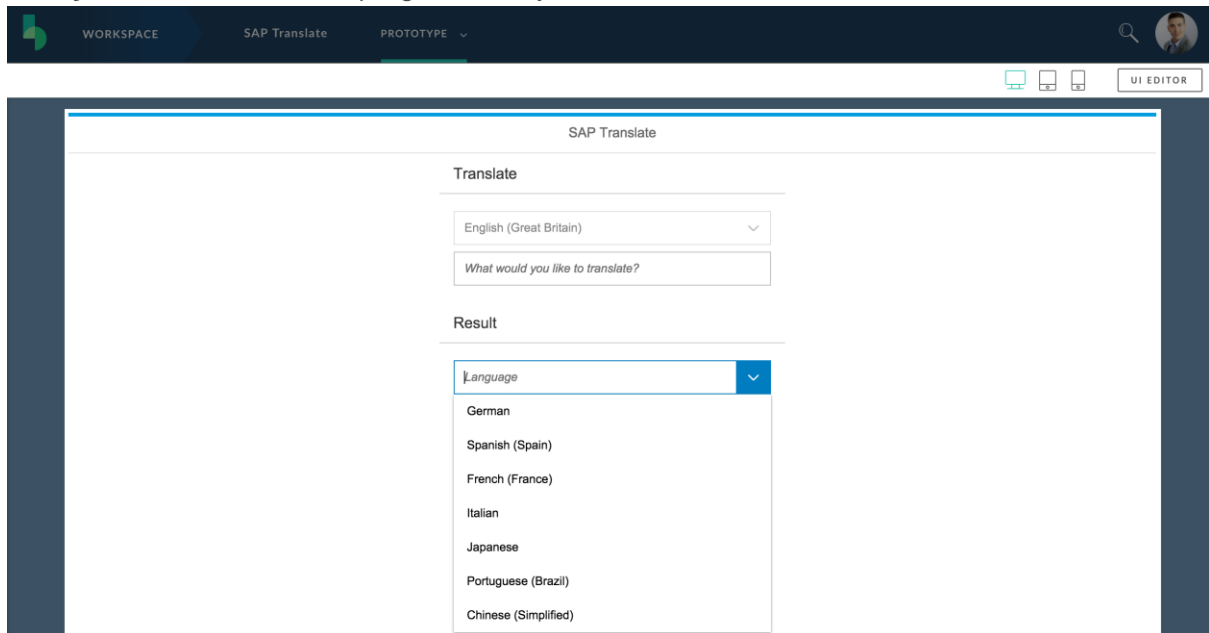


Lastly, we will modify the **Input** control (in the *Result* Simple Form). In the Properties pane, change the **Enabled** switch to **off**.

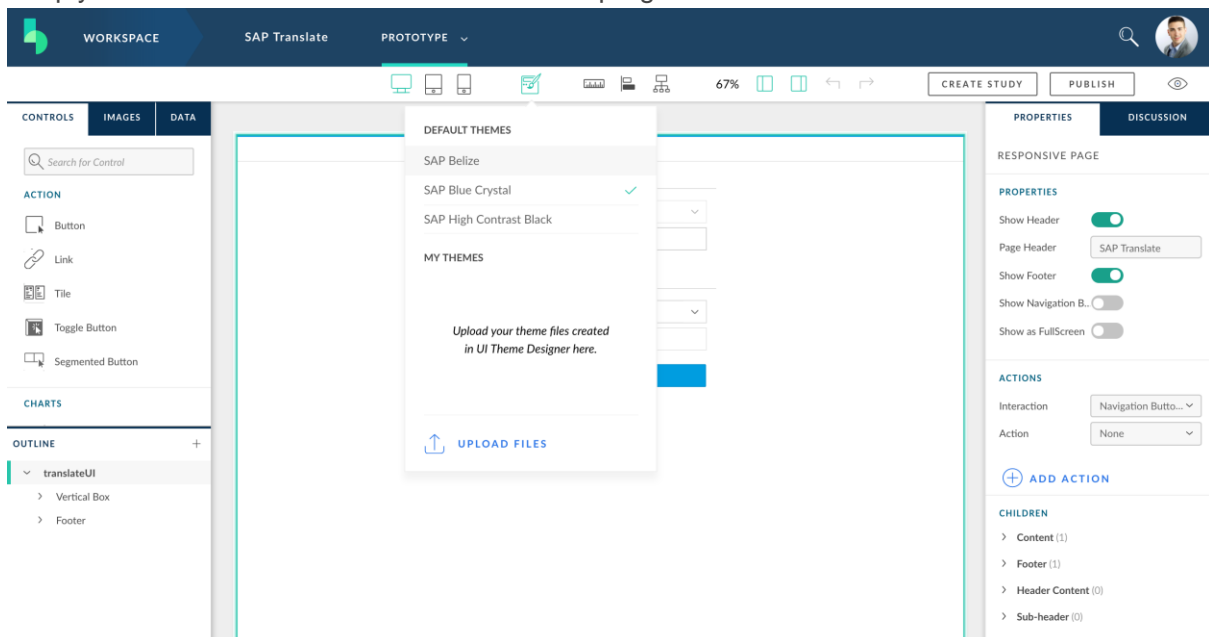
Preview and Export the Application



We will now test our SAP Translate UI in **Preview Mode**. With SAP BUILD, we're able to preview our prototype pages and test the interactions before we publish or share with others. Simply **select the eyeball icon** from the top-right side of your UI Editor.

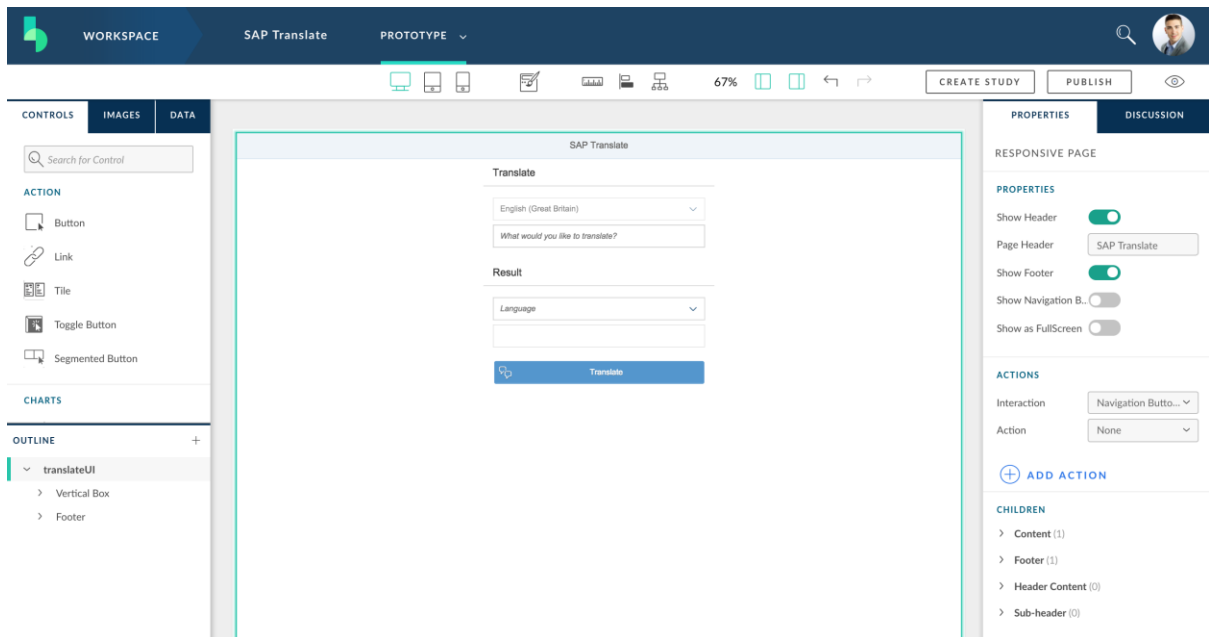


Select your **Dropdown List** control from the **Language** field to verify your item properly appear. Simply select the **“UI Editor”** button from the top-right of the screen to return to the UI Editor.

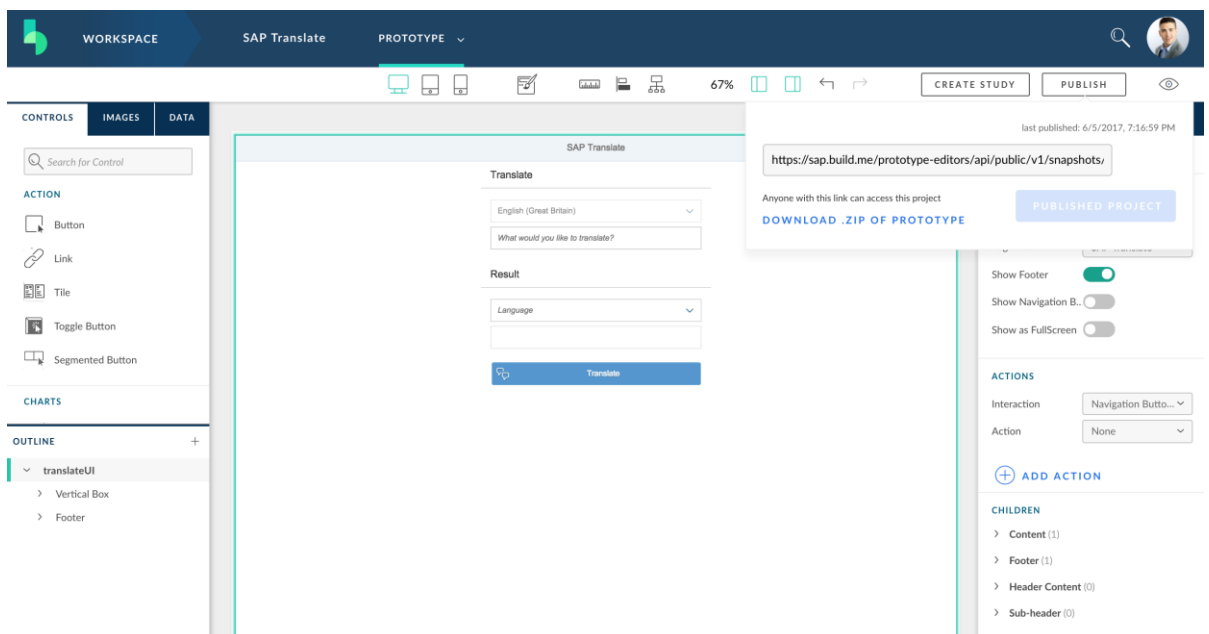


Once you return to the UI Editor, select the **“Theme”** button, at the top of your toolbar (as seen in the screenshot above). Select the **SAP Belize** theme.

With the introduction of SAP Fiori 2.0, a new visual theme called “Belize” was introduced as the successor to the Blue Crystal theme.



Following the theme change, select the **“Publish”** button, in between Create Study and the Preview button.



Click **“Publish”** again from the popup. Then select **“Download .ZIP of Prototype”** to download the shell of the application.

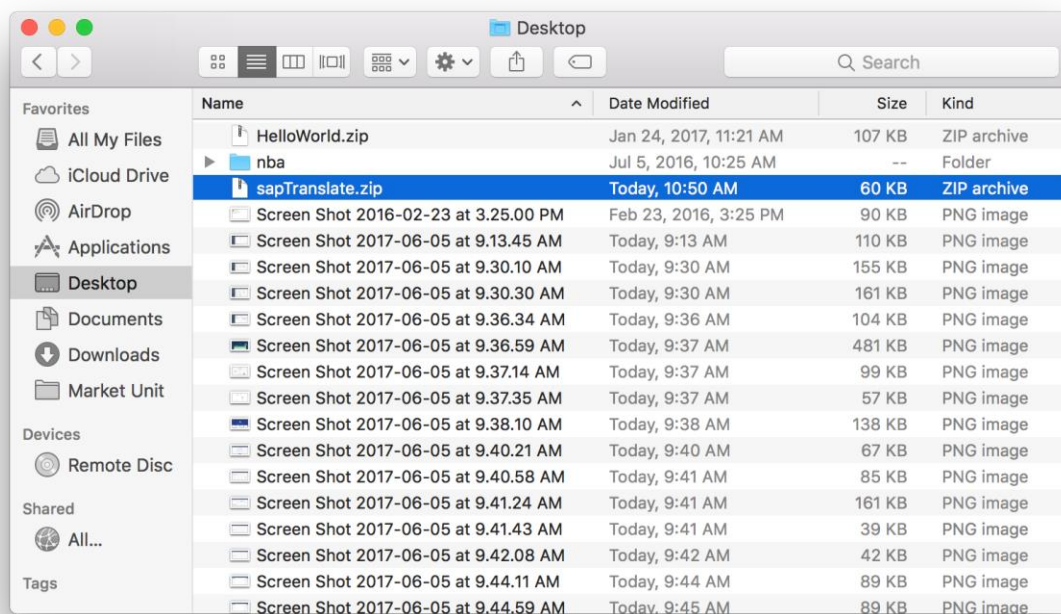
With SAP BUILD, you can publish a prototype at any time and share it with others. This creates a snapshot of the prototype design that you can share with others using a shareable link.

We’re manually exporting our prototype as a .ZIP file to import into our development environment in the next exercise. We’re doing this because SAP BUILD generates the code that you can import into the development environment as a starting point for developers. The same code that is generated for your BUILD prototype, used in previewed and published applications, is the code that you will import.

The preferred method is using the SAP BUILD plug-in. However, based on the time constraints of this in-person workshop, we will use the manual import to save time for the following exercises. If you want to learn more about setting up these plug-ins, you can do so [here](#).

Change the .ZIP file name

We will need to rename the .ZIP file. Navigate to the downloaded .ZIP file on your local directory and change the name to “**sapTranslate.zip**”.



Ready to make this a real, full-functioning application that will actually translate texts?