



Developing a Progress OpenEdge Web App Using Kendo UI Builder

Kendo UI Builder is a tool for creating responsive client web apps for OpenEdge applications that can be accessed from a tablet or desktop browser. Kendo UI Builder supports OpenEdge Data Object Services so that your web app can easily access data from ABL applications. The web app that you create can be deployed to a PAS for OpenEdge instance, a classic OpenEdge AppServer, or any other web server.

In this video, we'll create a web app, called **SportsIncWeb**, that gives users read-only access to customer data. We'll use an existing Data Object Service called **CustomerService** that, when invoked, connects to the sample sports3000 database and fetches data from the Customer table.

Let's begin by creating the web app in Kendo UI Builder. We click **Create App** to opens the **Create App** dialog box. Here, we enter **SportsIncWeb** as the name of the web app, specify a location for the web app, enter the description, and then click **Create App**. The web app is created.

Next, we'll create a Web UI project for the web app in Progress Developer Studio. Notice that there already is a Data Object Service project called **SportsInc**. To create our Web UI project, we launch the New OpenEdge Project dialog box.

We specify **SportsIncWeb** as the project name, **Web UI** as the project specialization, and then click **Next**.

On the Provide Web UI deploy details page, we uncheck **oepas1**, since we do not want to deploy it to a PAS for OpenEdge instance at this time. Then we click **Finish**. The Web UI project is created. We right-click the project and select **Refresh** to synchronize the files in the file system with the files in the Developer Studio workspace.

Next, we'll start developing the **SportsIncWeb** web app by adding a module. As you can see, the web app that we created already has a default module called **Application**. Let's create a module and name it **CustomerService** to denote its business function. To do this, we:

- Click **Add Module**.
- Enter **CustomerService** as the module name.
- Select and modify the icon that best represents the module.
- Enter the description.
- And click **Add Module**.

The **CustomerService** module is created.

Next, we need to add a data provider. In our example, since we want to consume data from an OpenEdge ABL application, we must add the Data Object Service as the data provider. To add an OpenEdge Data Object Service as a data provider, we need to know the service's Service and Catalog URI. Let's click **Add Data Provider**. The Add Data Provider dialog box opens. Here, we enter **CustomerData** as the name of the data provider, enter the service and catalog URIs, and then click **Add Data Provider**. As you can see, the **CustomerData** data provider is now added.

By default, **CustomerData** has a data source called **Customer**, which maps to the **Customer** table in the **sports3000** database and provides access to all of its fields. You can create an additional data sources to map to a subset of the fields. For this video, we'll use the default data source.

Now, we'll add a view to our module that will enable users to view customer data. To add a view, we need to edit the module by clicking on it. Then, we click the **Add View** icon. The Add View dialog box, which displays the different built-in view templates that are available, appears. We name the view **ViewCustomers**, select the **DATA GRID** view template (which provides read and write access), and then click **Add View**. The view is created.



Next, we'll configure some key properties for the new view.

- First, we specify **Customers** as the grid title.
- Next, we select **CustomerData** as the data provider. Since our data provider has only one data source, it is automatically set as the data source here.
- Then, we edit the Grid Columns property to include customer number, customer name, country, Sales Rep, and balance.
- Next, we specify **10** as the page size for this view. This will ensure that the grid displays 10 rows at a time.
- We then enable filtering, grouping, and resizing for the group.

Finally, we save our configuration.

Now, we are ready to preview our web app. We click the **Preview** button. After a few seconds, the web app is launched. We select the **CustomerService** module. The ViewCustomers view opens. This view displays customer number, name, country, sales rep, and balance. Also, we see that 10 customers are listed at a time in the non-editable grid. Remember that we enabled grouping for this view. Let's group data by Country. We can see that the grouping works.

You have now seen how to develop and preview an OpenEdge web app using Kendo UI Builder. To learn more about developing OpenEdge web apps, take the *Developing a Progress OpenEdge Web App Using Kendo UI Builder* course.