



## **Responsive Application Design with Kendo UI Builder for OpenEdge**



Ricardo Perdigao, Solutions Architect  
Jean Munro, Principal Sales Engineer  
Carole Miller, Senior Sales Engineer

# Section 1: The Workshop Environment

---

In this workshop, you use the OpenEdge 11.6.2 **Progress Application Server for OpenEdge (PASOE)** with the **Kendo UI Builder for OpenEdge (KUIB)** to design a responsive web application.

Using **Progress Developer Studio for OpenEdge (PDSOE)**, you first create backend data services to expose OpenEdge data as REST services. You then create a responsive web application using the Kendo UI Builder for OpenEdge (KUIB). The web application will interface with the Progress Application Server (PASOE) and the OpenEdge database on the backend.

You will need to have access to:

**Progress Arcade.** Each workshop participant will be provided a unique IP address for use during the session. You will be using an OpenEdge 11.6.2 environment comprised of a complete installation of OpenEdge 11.6 PDSOE with Service Pack 02 added. Each Arcade image also includes the Google Chrome browser and a workshop folder containing some saved links and code snippets for convenience.

## Section 2: Configuring PDSOE

---

Virtual machines have been started on Progress Arcade for this workshop. The workshop host has provided you with a unique DNS address for a virtual windows server running on Progress Arcade. The first step is to connect to your workshop VM. Then you'll create a database and configure PDSOE for the workshop project.

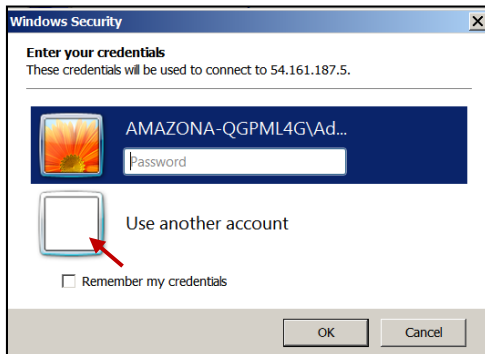
### Launching your PDSOE Demo machine

On your laptop, choose **Start** → [Windows] **Accessories** → **Remote Desktop Connection** and paste your unique DNS into the **Computer** field. Click the **Connect** button.

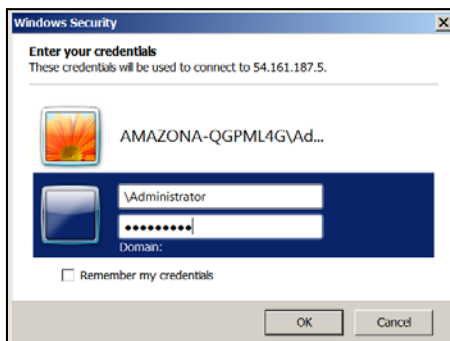
Note: You may need to click the button to the left of **Options** to change (erase) the domain name. Also, the screens may be slightly different for your version of Windows.



Choose the option to **Use another account**:



Enter **\Administrator** as the Username, and **Pug2016** as the Password, then click **OK**:

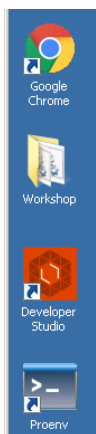


Click the **Yes** button to say you trust this remote connection:



You should now be connected to your personal workshop labs machine running in Progress Arcade.

The following shortcuts have been provided in your remote desktop:



Google Chrome web browser.


Workshop Files containing workshop sample code.

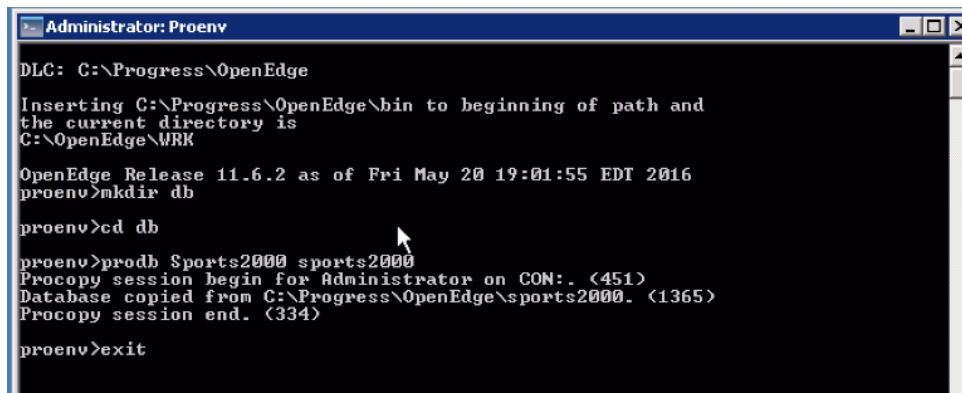
11.6.2 Progress Developer Studio for OpenEdge (PDSOE) – Eclipse-based IDE.

11.6.2 Proenv – sets OpenEdge environment variables and starts a command-line session in the OE working directory.

## Configuring the Development Environment

### Create Sports Database

1. Start a Proenv session by double-clicking on the Proenv desktop icon. 
2. Create a database named **Sports2000** (from sports2000). At the **proenv** prompt, enter the commands shown below:
  - **mkdir db** - create the “db” subdirectory
  - **cd db** - change directory into the new C:\OpenEdge\WRK\db subdirectory
  - **prodb Sports2000 sports2000** – create a new database named “Sports2000” as a copy of the sports2000 database provided with OpenEdge PDSOE
  - **exit** – close the proenv session



```
Administrator: Proenv
DLC: C:\Progress\OpenEdge
Inserting C:\Progress\OpenEdge\bin to beginning of path and
the current directory is
C:\OpenEdge\WRK
OpenEdge Release 11.6.2 as of Fri May 20 19:01:55 EDT 2016
proenv>mkdir db
proenv>cd db
proenv>prodb Sports2000 sports2000
Procopy session begin for Administrator on CON: (451)
Database copied from C:\Progress\OpenEdge\sports2000. (1365)
Procopy session end. (334)
proenv>exit
```

## Basic Configuration of PDSOE

3. Open Developer Studio:

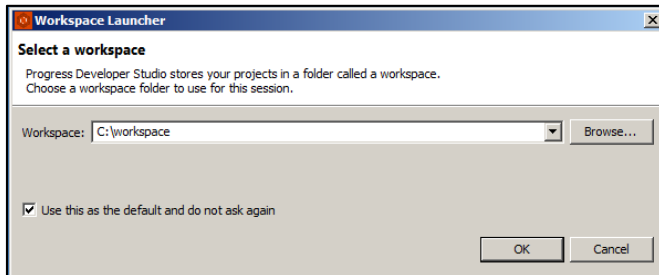
**Start->All Programs->Progress->OpenEdge 11.6 (64-bit)>Developer Studio**

or use the Desktop shortcut in your Arcade image

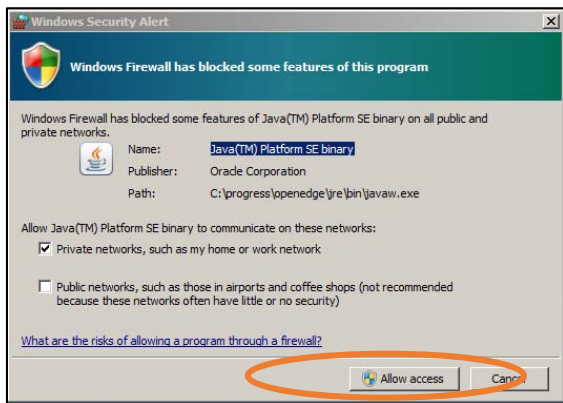


4. Modify the path for the workspace to use **C:\workspace**

Toggle on the option to **Use this as the default and do not ask again**, and click **OK**.



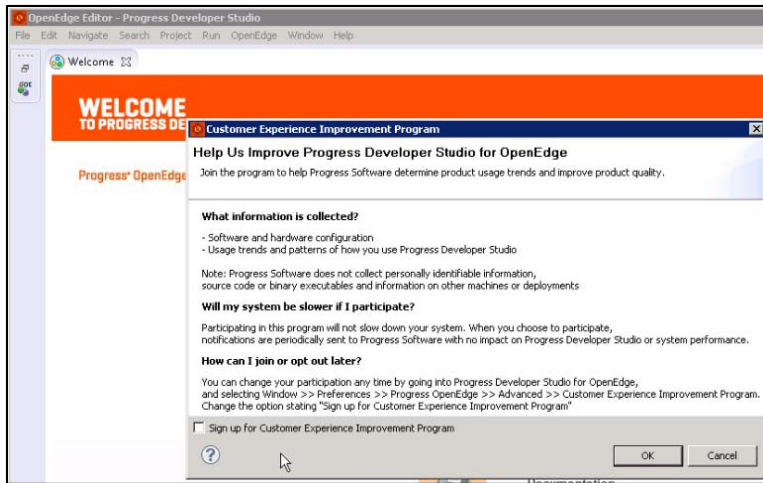
5. You **MAY** need to authorize Java by clicking **Allow Access**:



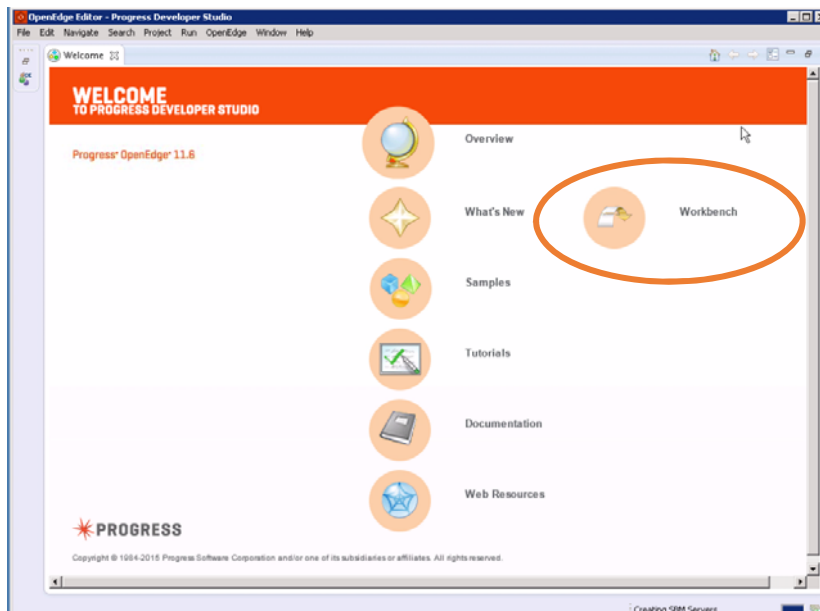
You will see the splash screen and Loading workbench... message:



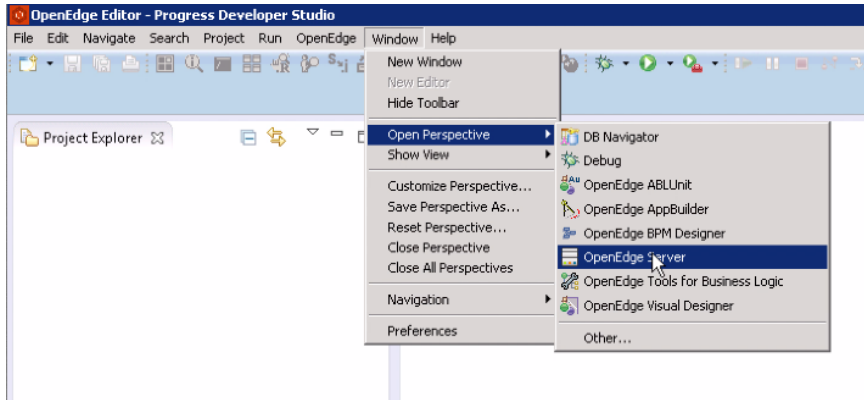
6. Progress offers participation in the Customer Experience Improvement Program. For this workshop session, **clear (uncheck) the Sign up** option and click **OK** to continue.



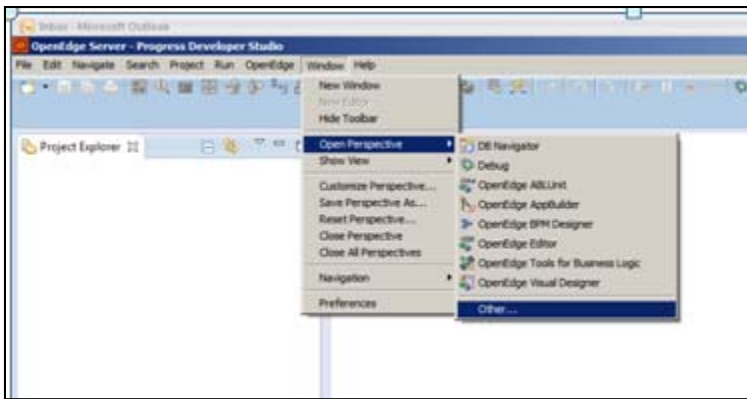
Close the Welcome Screen, or click **“Workbench”**



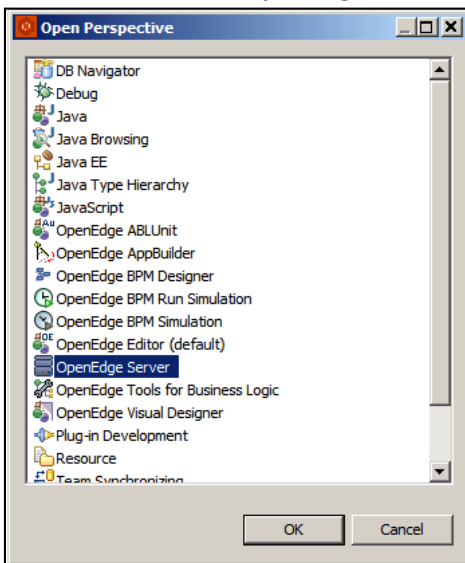
7. Click **Window -> Open Perspective** and select **OpenEdge Server**:



Note: If you don't see the OpenEdge Server perspective in the dropdown list, select **Other...** to expand the list:

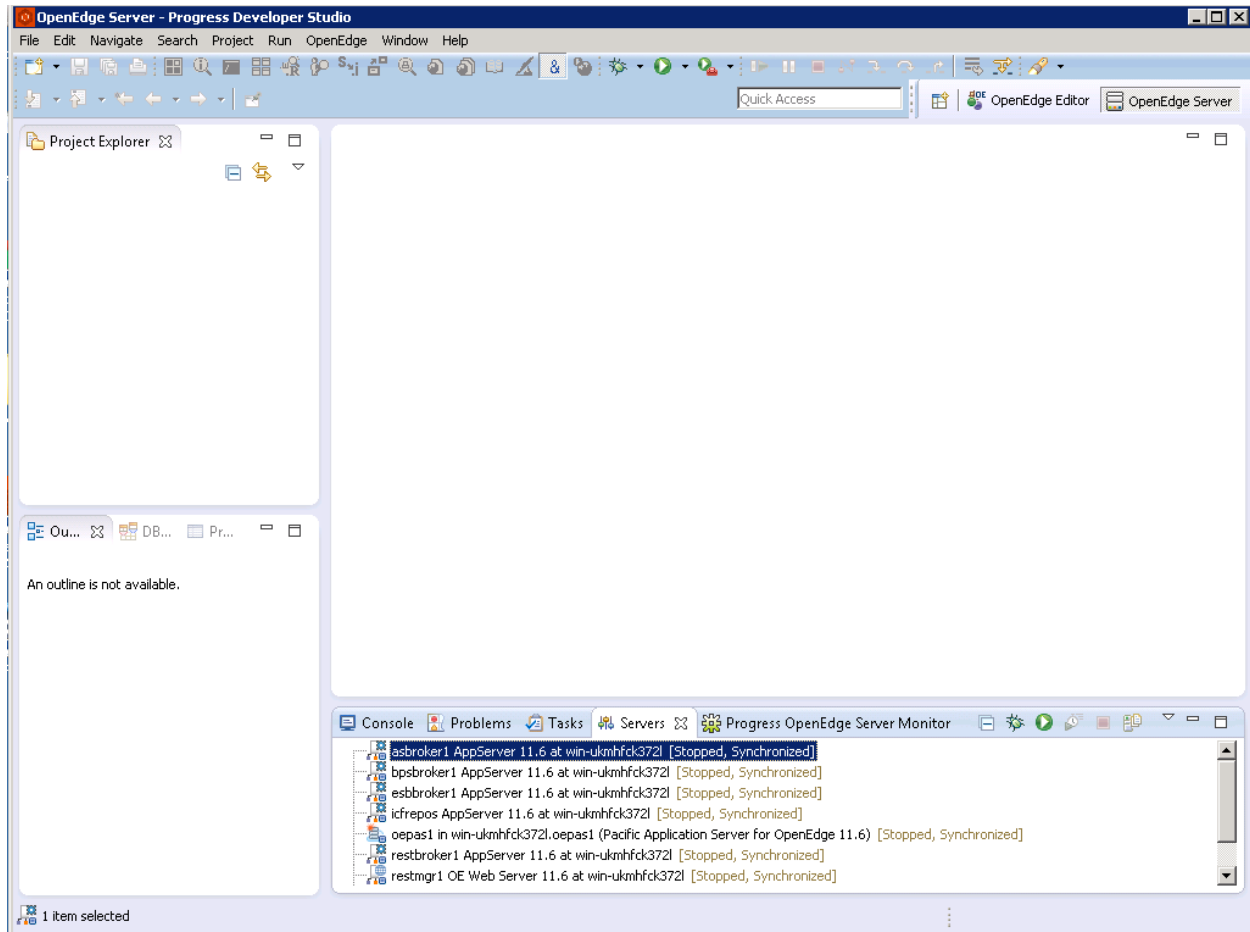


And then select **OpenEdge Server**:



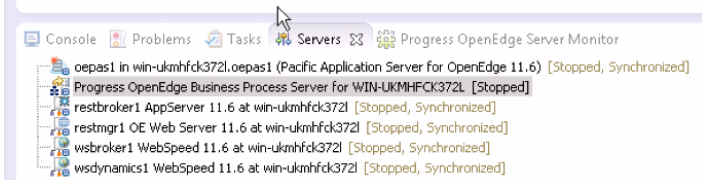
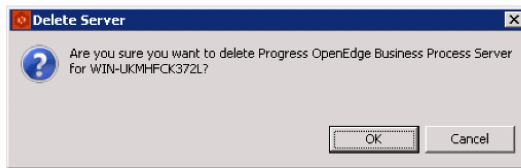
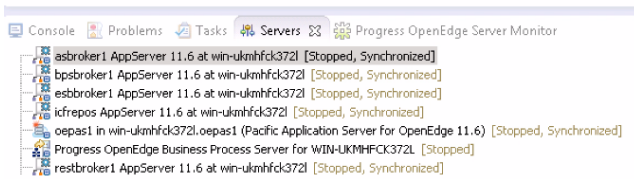
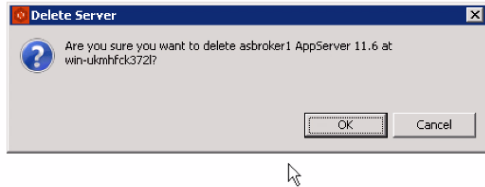


8. In the OpenEdge Server perspective, click on the **Servers** view at the bottom right. A list of links to preconfigured servers will appear in the pane:

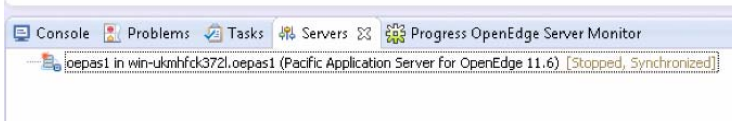


For the purposes of this workshop, you will use the link to the **oepas1** in the server list for your project.

9. **Optional:** For ease of use during the workshop, you may choose to delete the other server entries shown in the list by highlighting each entry **except oepas1** and deleting each of the other entries. **\*\*DO NOT delete the oepas1 entry.**



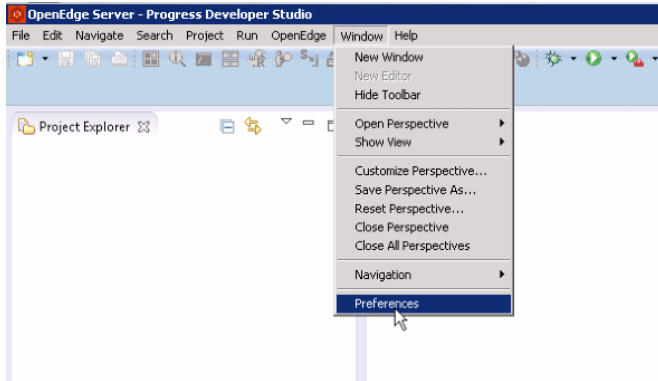
If you choose to remove the other server entries, you should see only **oepas1** in the Servers view when you are done with this step:



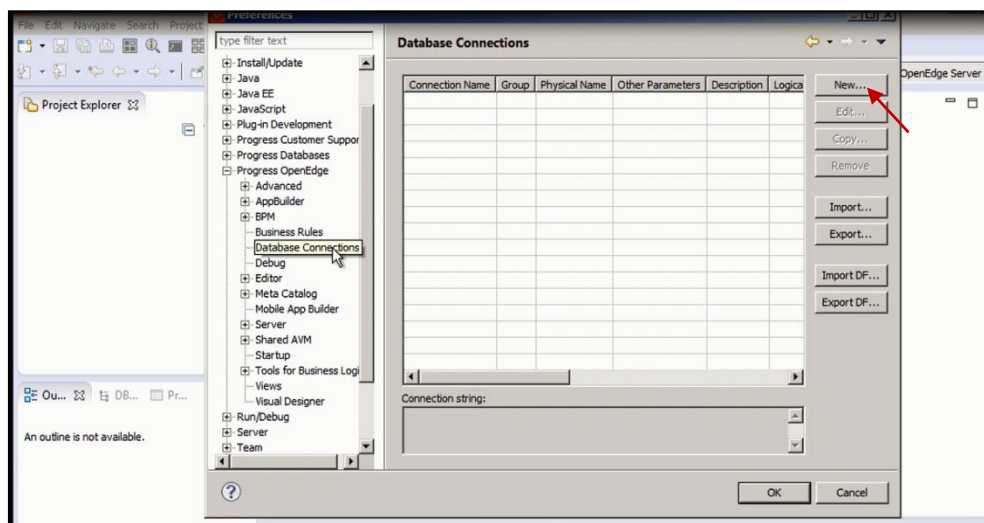
## Add Database Connection

In this step you configure a connection to the Sports2000 database you created earlier.

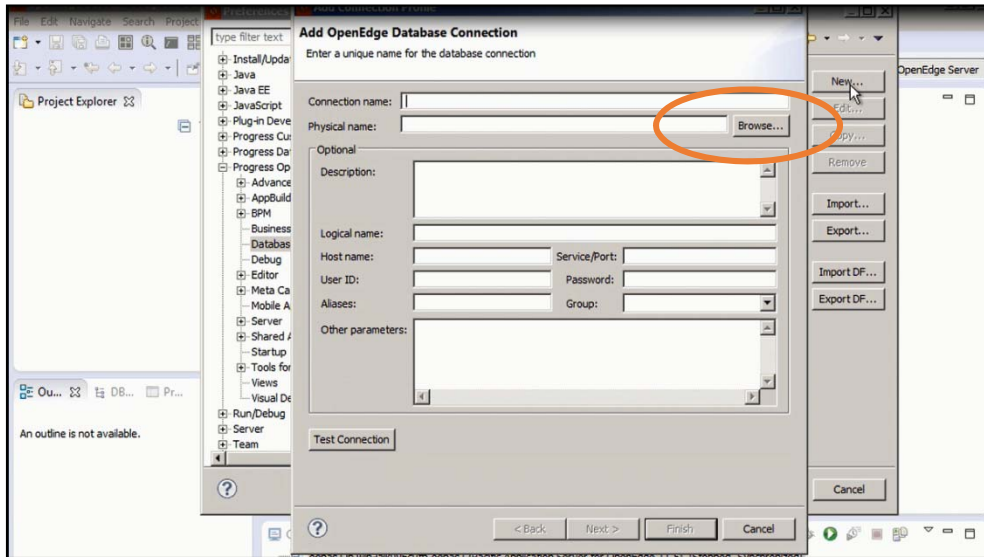
10. Select **Window -> Preferences:**



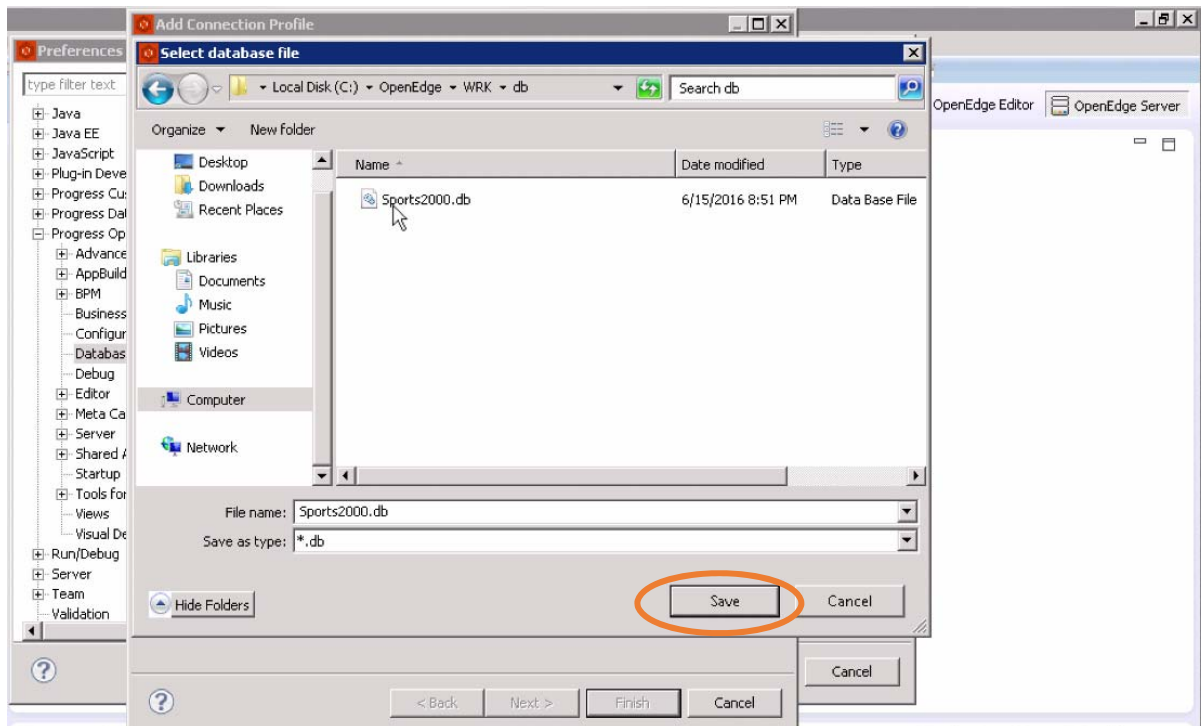
11. Click on the **Progress OpenEdge** node in the tree view to expand it, choose **Database Connections**, then click the **New...** button in the upper right:



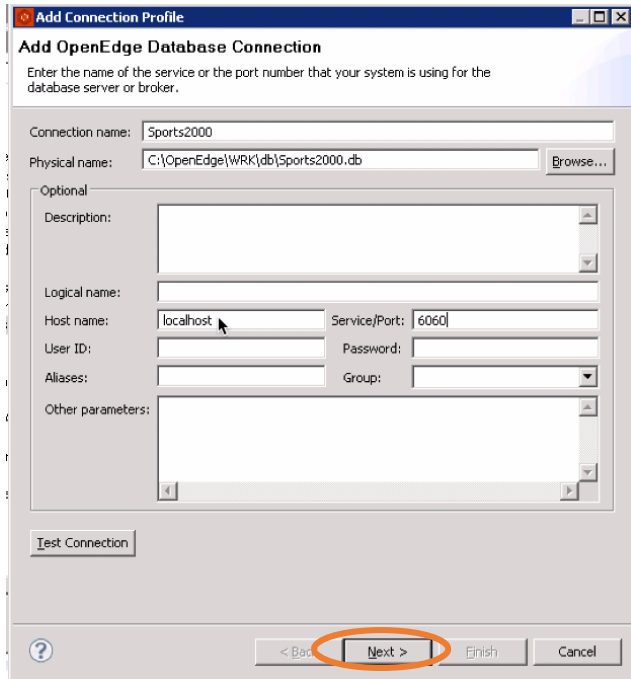
You will see a dialog that asks you to enter information about the database. Next to Physical Name click the **Browse** button:



12. If the Sports2000 database does not appear in the Select database file window, navigate to **C:\OpenEdge\Wrk\db** where you created the database earlier and choose the **Sports2000** database. Click **Save**:

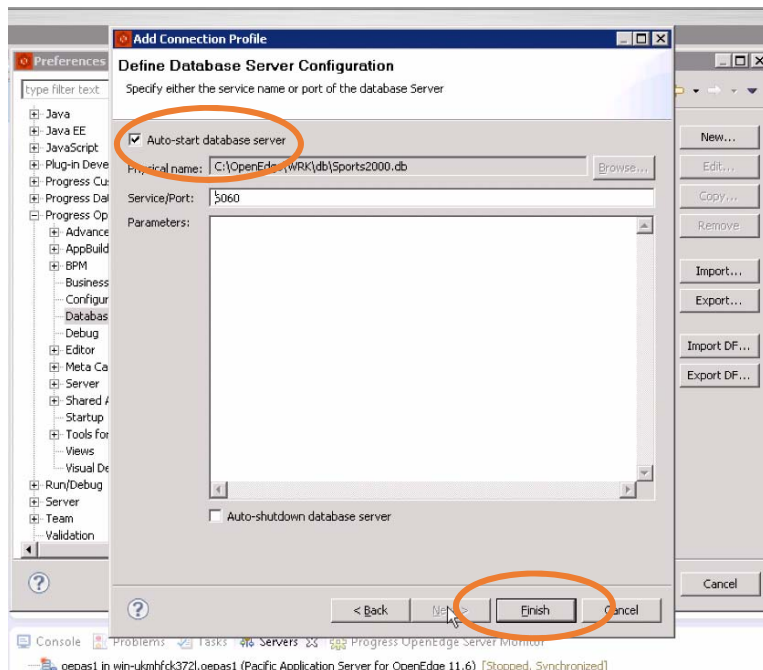


Enter the Connection name **Sports2000**, Host name **localhost** and Service/Port **6060** (any available TCP port could be used). Then click **Next**:

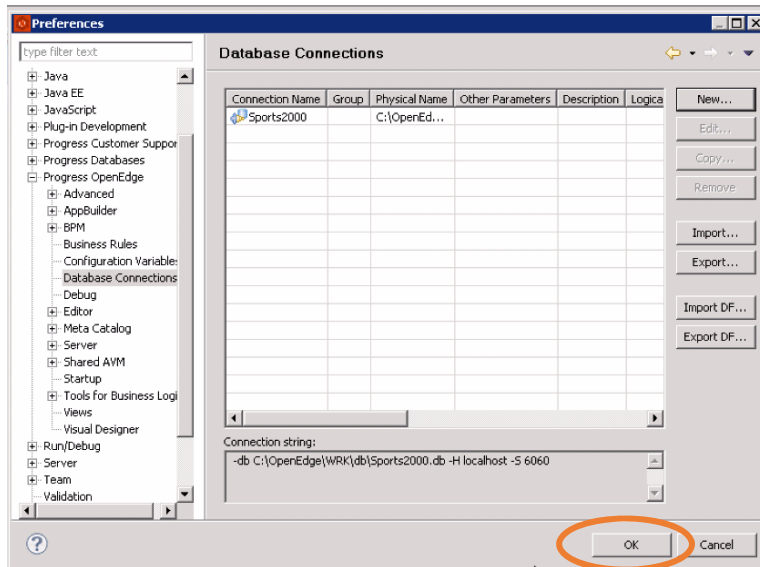


13. Click **Next** to take the default values on the next two screens (Define SQL Connection and Add SQL Connection Profile.)

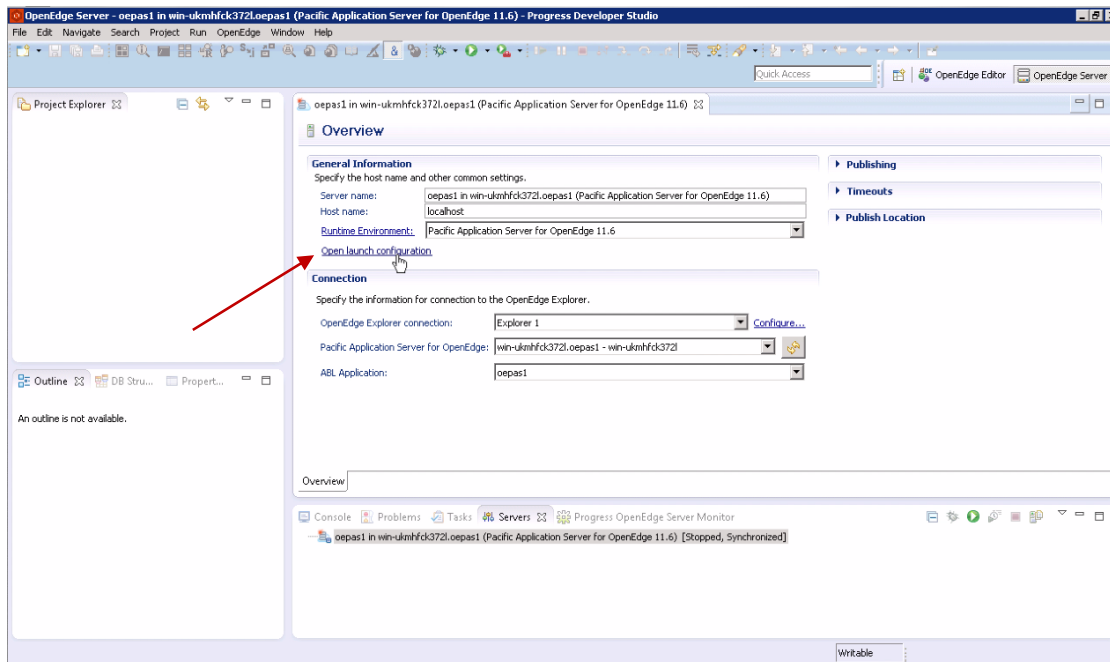
14. On the **Define Database Server Configuration** screen shown below, make sure that the **Auto-start database server** box is **checked**. Then click **Finish**.



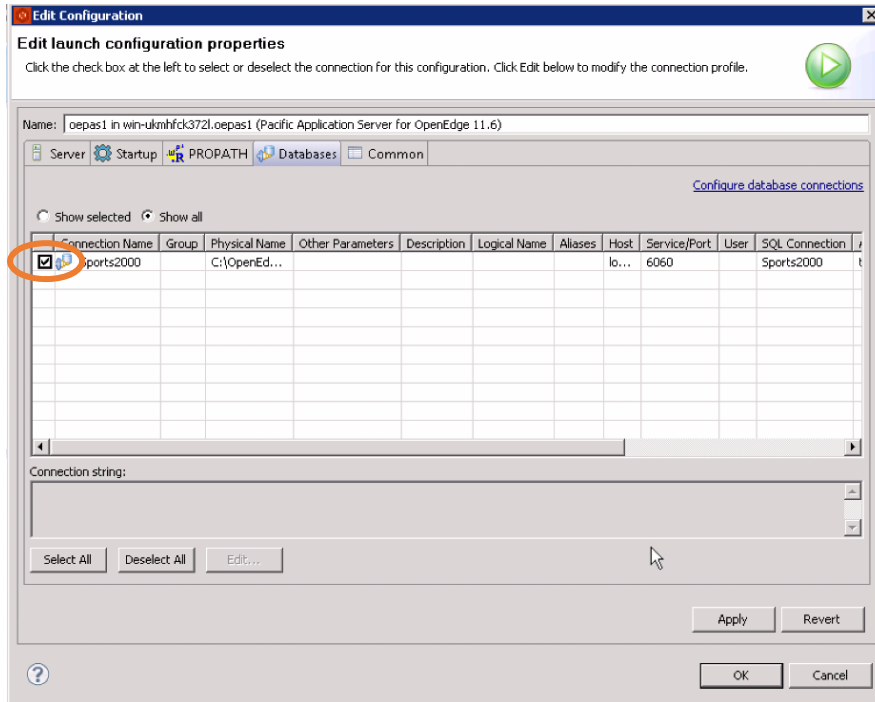
15. You will now see the Sports2000 database connection listed in the database connections screen. Click **OK** to complete the operation.



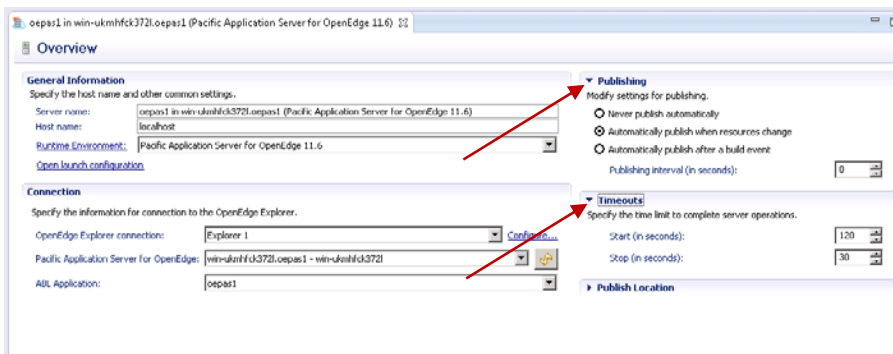
16. The last step is to configure the oepas1 app server to use the Sports2000 database. **Double-click** on the oepas1 app server in the servers view (or **right-click** on the oepas1 app server line and choose **Open**). You will see the Overview configuration screen. Click on the **Open launch configuration** link:



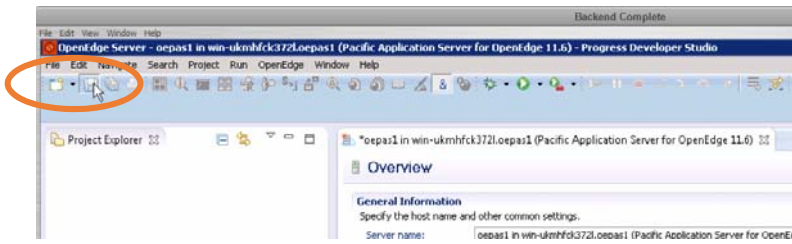
17. Select the **Databases** tab. Choose **Show all** and then check the box for the **Sports2000 database connection**. Click **Apply** and then click **OK**.



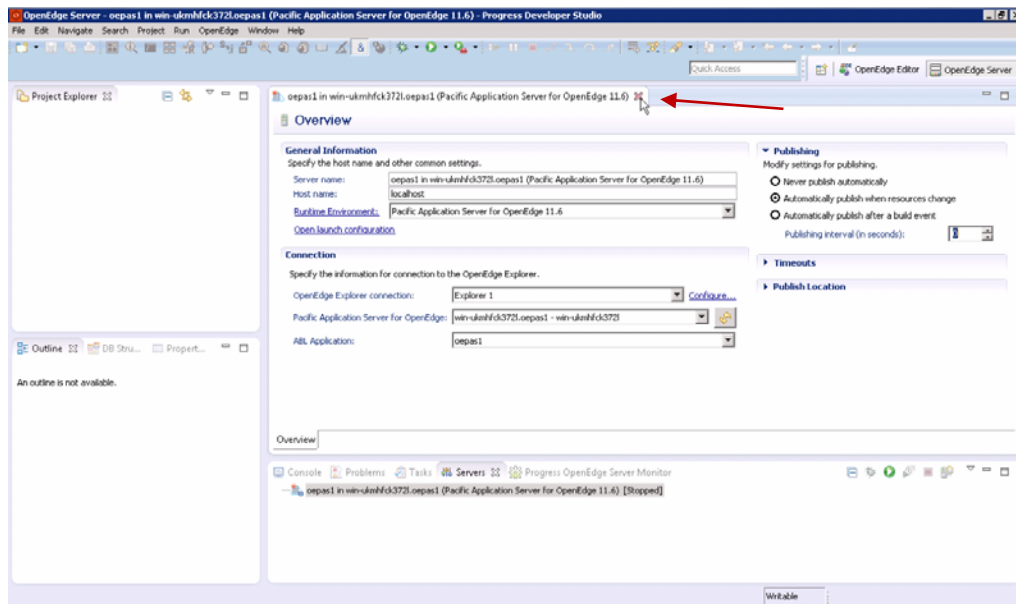
18. You are returned to the configuration Overview screen, still in the Server perspective. Select the **Publishing** settings as shown and modify the **Publishing interval** to set it to **0** seconds. Select the **Timeouts** settings and set the **Start** timeout value to **120** seconds. These adjusted values will work better in the Arcade instances for this workshop session.



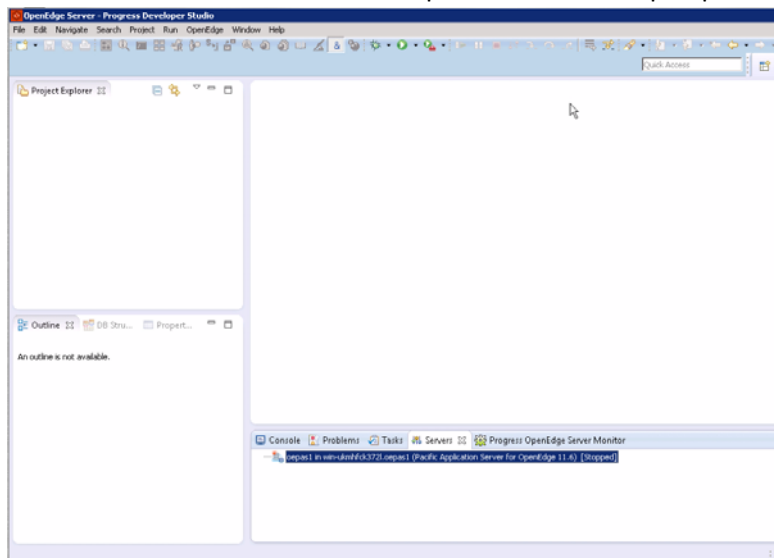
19. Save the Project configuration:



20. The development environment is now configured and ready to use. Close the configuration Overview pane:



21. You are returned to the Developer Studio Servers perspective:





# Section 3: Exposing OpenEdge Data as REST with PASOE

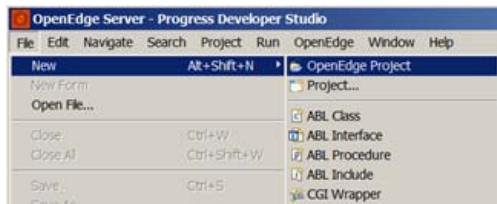
---

## Creating an ABL Web App Project

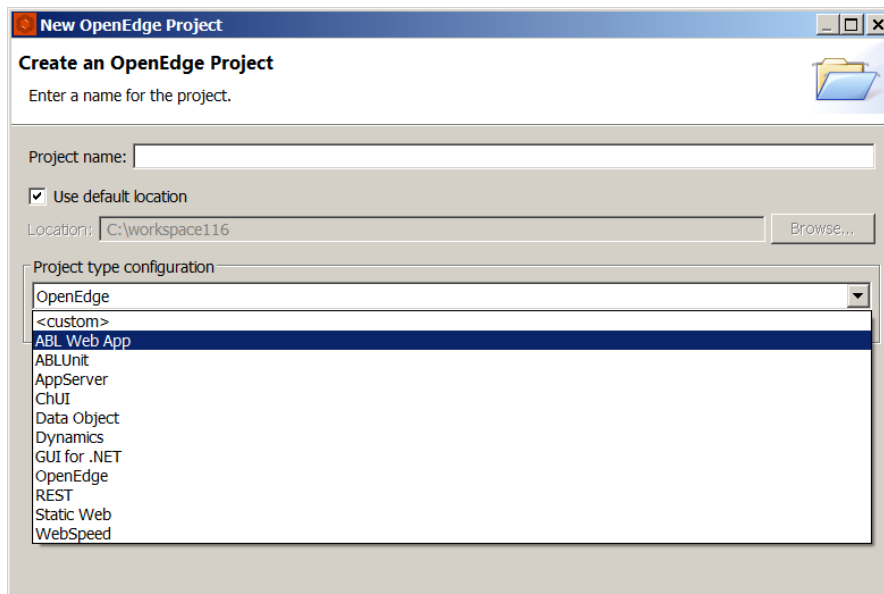
In this section you create a an ABL Web App Project. ABL Web App projects are configured to use JavaScript Data Objects (JSDO). The ABL Web App project type, with its built-in JSDO, simplifies the process of exposing OpenEdge services as REST services.

Create the sports project

1. Click on **File -> New -> OpenEdge Project**.



Create the new project named **Sports**, select the **ABL Web App** Project type configuration, and select **Next**.



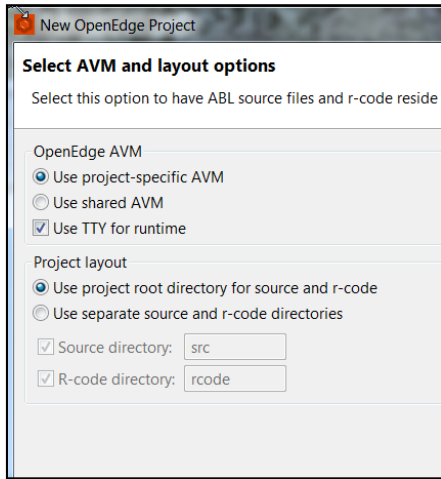
2. Choose to **Deploy as WebApp**. Select the **oepas1** server if it is not checked, then click **Next**:

The screenshot shows the 'New OpenEdge Project' dialog box with the title 'Provide ABL Web App deploy details'. Below the title is the instruction: 'Enter a name for the ABL AppServer module. Choose a name that clearly identifies this project.' The dialog is divided into three sections: 'Web Application', 'Business Logic', and 'Supported servers'. In the 'Web Application' section, the 'Deploy as WebApp' radio button is selected, and the text 'Sports' is entered in the adjacent text box. A red arrow points to this radio button. The 'Business Logic' section contains two text boxes: 'Module name:' with 'Sports' and 'ABL Source folder:' with 'AppServer'. The 'Supported servers' section is a table with a header 'Server Name' and one row containing a checked checkbox and the text 'oepas1'. At the bottom of the dialog are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

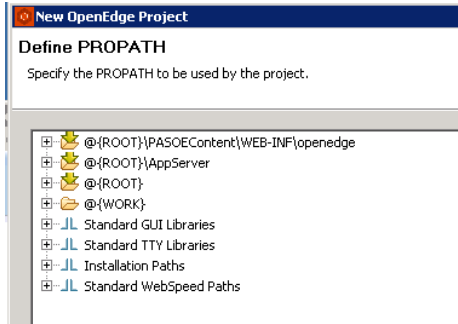
3. Select **"Data Object"** Service Type and click **Next**.

The screenshot shows the 'New OpenEdge Project' dialog box with the title 'Create an ABL Service'. Below the title is the instruction: 'Enter a name for the Data Object service.' The dialog is divided into two sections: 'Service type' and 'Data Object service details'. In the 'Service type' section, the 'Data Object (Annotated RPC)' radio button is selected, and a red arrow points to it. The 'Data Object service details' section contains three text boxes: 'Service name:' with 'SportsService', 'Service relative URI:' with '/SportsService', and 'Service description:' which is an empty text area.

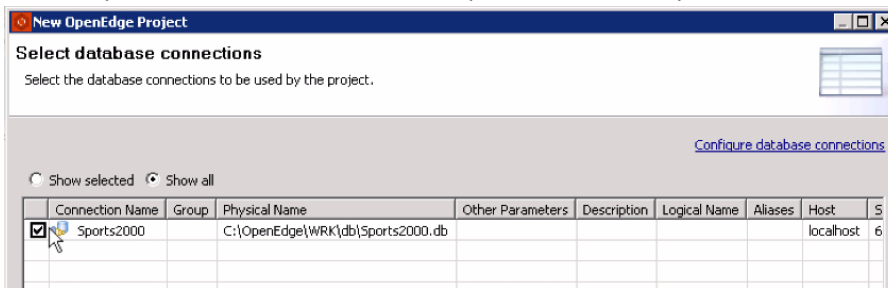
- Accept the default settings for AVM and layout options, and click **Next**:



- Accept the default settings for PROPATH and click **Next**:



- The project will be using the Sports2000 database that you added to the Developer Studio environment. **Check** the box on the **Sports2000** database connection line and click **Finish**. (You may need to **check** the **Show all** option to see the Sports2000 database connection.)

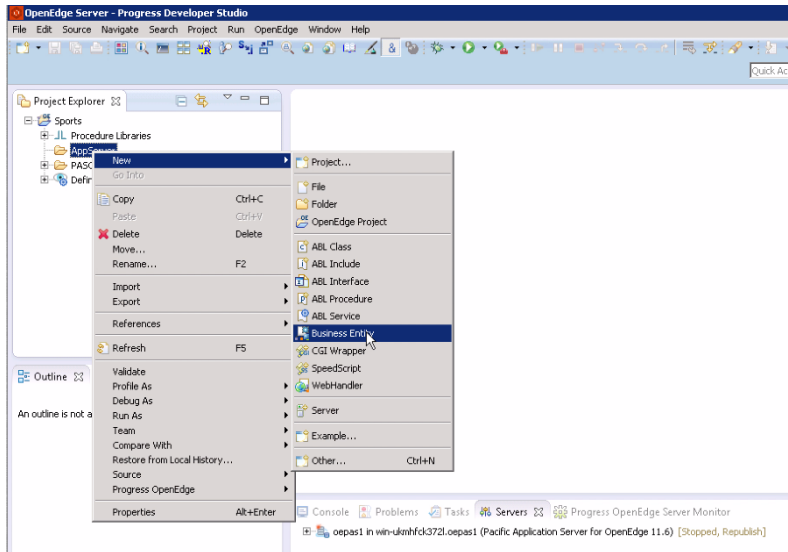


- Developer Studio creates your Sports project with everything you will need to create and deploy REST services on the Progress Application Server for OpenEdge (PASOE). You may be prompted to open the Servers perspective; if so, click **Yes**.

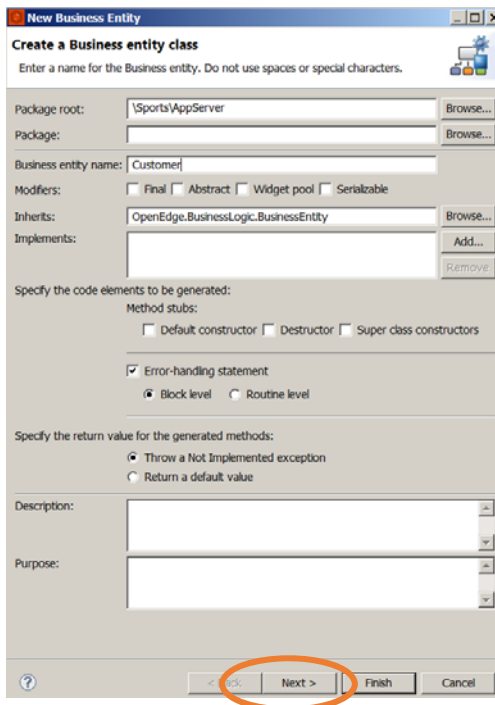
## Creating a Business Entity to Expose Data

In this section you will create a Business Entity for the Customer table of the Sports2000 database. Later you will expose this Business Entity via your REST service.

8. **Right-click** on the AppServer folder in the Sports project and select **New** and then **Business Entity**:



9. Enter the Business entity name **Customer**. Accept all of the other suggested default settings and click **Next**.



10. In the top section of the Select schema file screen, you decide which operations to make available to the REST services. **Select CRUD and Submit** (CRUD “Create, Read, Update, and Delete” with option for Submit support. )

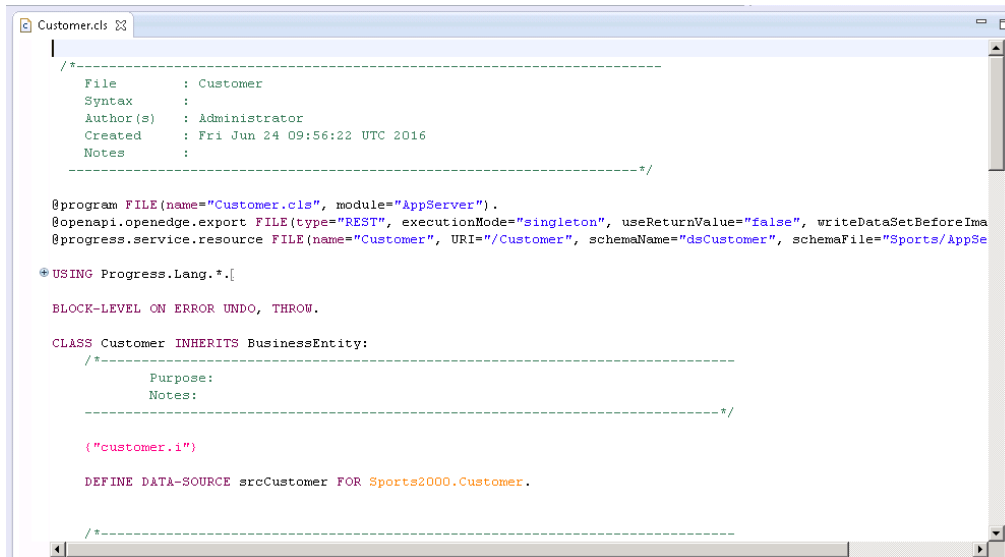
In the second section, you select either a database table or an include file that represents the ProDataSet that you wish to enable for REST. For this workshop example, **check Select database table** and **choose** the table named **Customer**, then click **Finish**.

The screenshot shows a Windows-style dialog box titled "New Business Entity" with a sub-header "Select a schema file". Below the sub-header is a small icon of a server and a gear. The main content area is divided into several sections:

- Resource name:** A text box containing "Customer".
- Operations:** Three radio buttons: "Read-only", "CRUD", and "CRUD and Submit" (which is selected). Below them is a checked checkbox labeled "Write dataset before image".
- Select database table:** A radio button that is selected. Below it are two dropdown menus: "Connection" (set to "Sports2000") and "Table" (set to "Customer").
- Select schema from file:** A radio button that is not selected. Below it are fields for "Schema file:" (with "Browse..." and "Clear" buttons) and "Schema:" (an empty list box).
- Using:** Three radio buttons: "Include file", "Schema definition", and "Class Definition".
- Expose as Data Object service:** A checked checkbox.

At the bottom of the dialog, there is a question mark icon on the left and four buttons: "< Back", "Next >", "Finish", and "Cancel".

- Developer Studio creates a new class file called **Customer.cls** in the AppServer node. This file includes methods for the CRUD and Submit operations that will be available to REST. These methods can be modified by inserting custom ABL code if desired. For now, just accept the default code.



```
Customer.cls
/*-----*/
File       : Customer
Syntax     :
Author(s)  : Administrator
Created    : Fri Jun 24 09:56:22 UTC 2016
Notes     :
/*-----*/

@program FILE(name="Customer.cls", module="AppServer").
@openapi.openedge.export FILE(type="REST", executionMode="singleton", useReturnValue="false", writeDataSetBeforeIma
@progress.service.resource FILE(name="Customer", URI="/Customer", schemaName="dsCustomer", schemaFile="Sports/AppSe

@USING Progress.Lang.*.[

BLOCK-LEVEL ON ERROR UNDO, THROW.

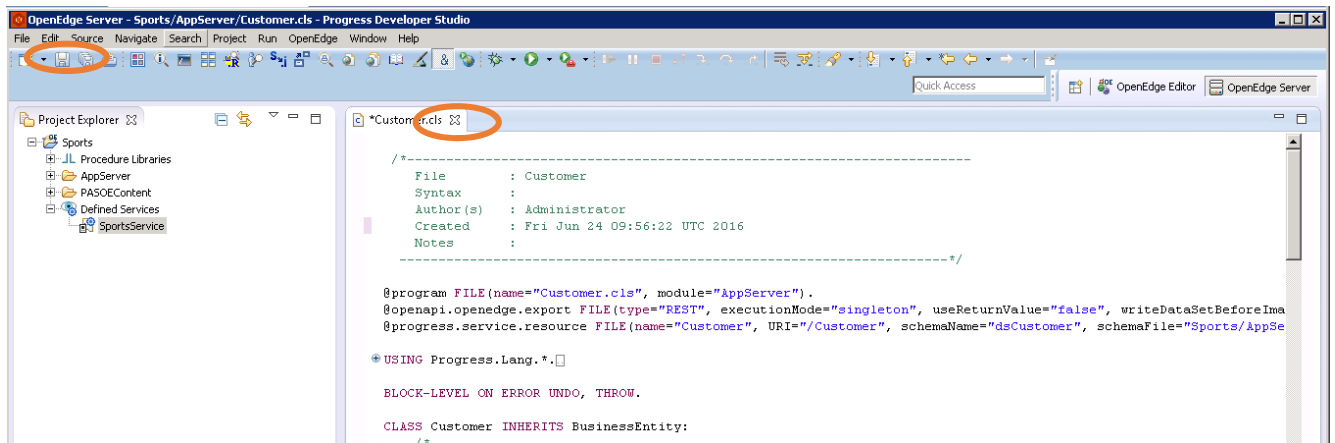
CLASS Customer INHERITS BusinessEntity:
/*-----*/
Purpose:
Notes:
/*-----*/

("customer.1")

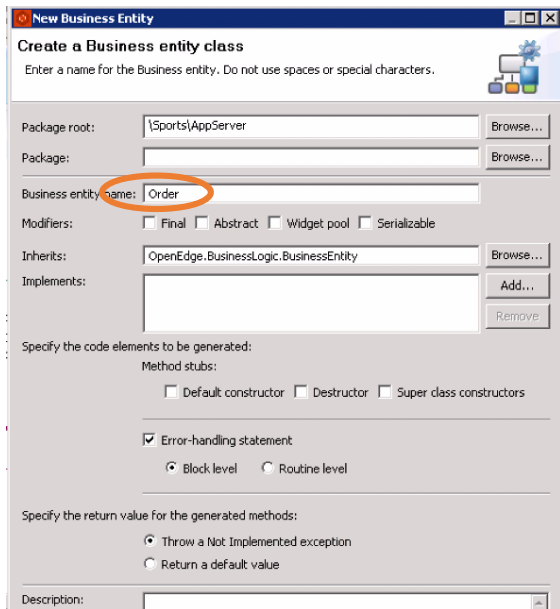
DEFINE DATA-SOURCE srcCustomer FOR Sports2000.Customer.

/*-----*/
```

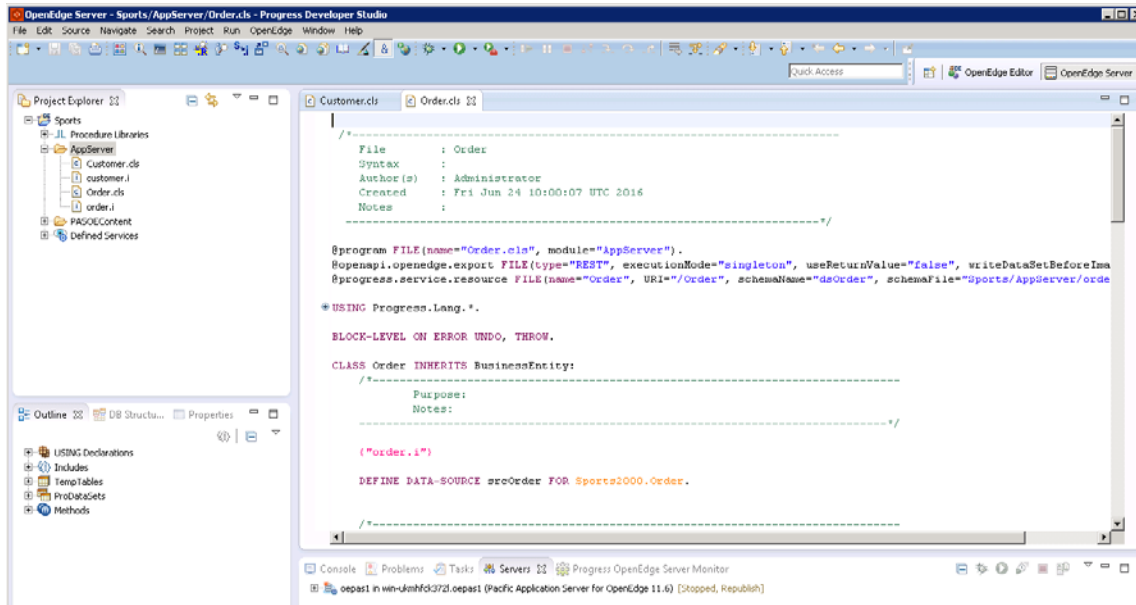
Save and **Close** the file **Customer.cls** file.



12. Repeat steps 8 through 11 for the **Order** table of the Sports2000 database:



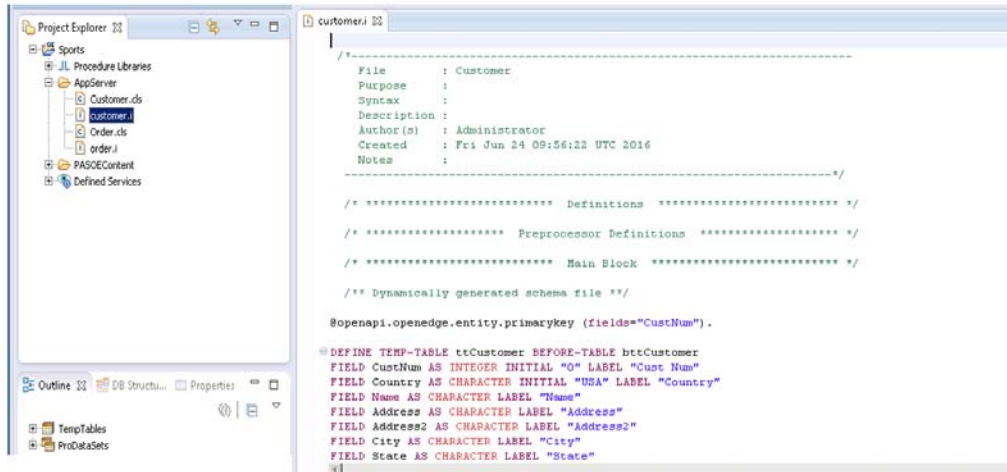
When you have completed the Business Entity wizards, you should have two .cls files in your Project: **Customer.cls** and **Order.cls**. Be sure to **save and close the Order.cls** file.



## Specify Semantic Type for a Field

In the next steps, you specify a semantic type for the Customer.EmailAddress field within the customer.i ABL source file. The semantic type will enforce valid formatting of the email address entered at runtime. The ABL syntax to specify the semantic type has been included in a text.

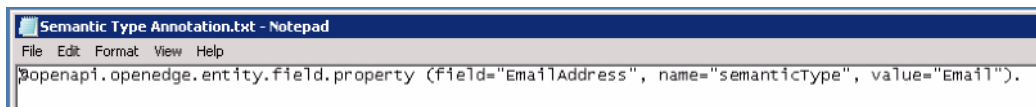
13. In your Sports project directory, expand the **AppServer** node and open the **customer.i** file:



14. Open the **Workshop** folder on your Arcade Desktop:



15. Open the **Semantic Type Annotation.txt** file in Notepad and select/copy the entire line:

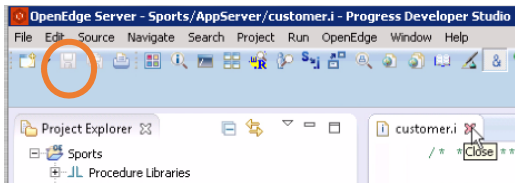


16. Paste it into the **Main Block** of **customer.i** as shown below:

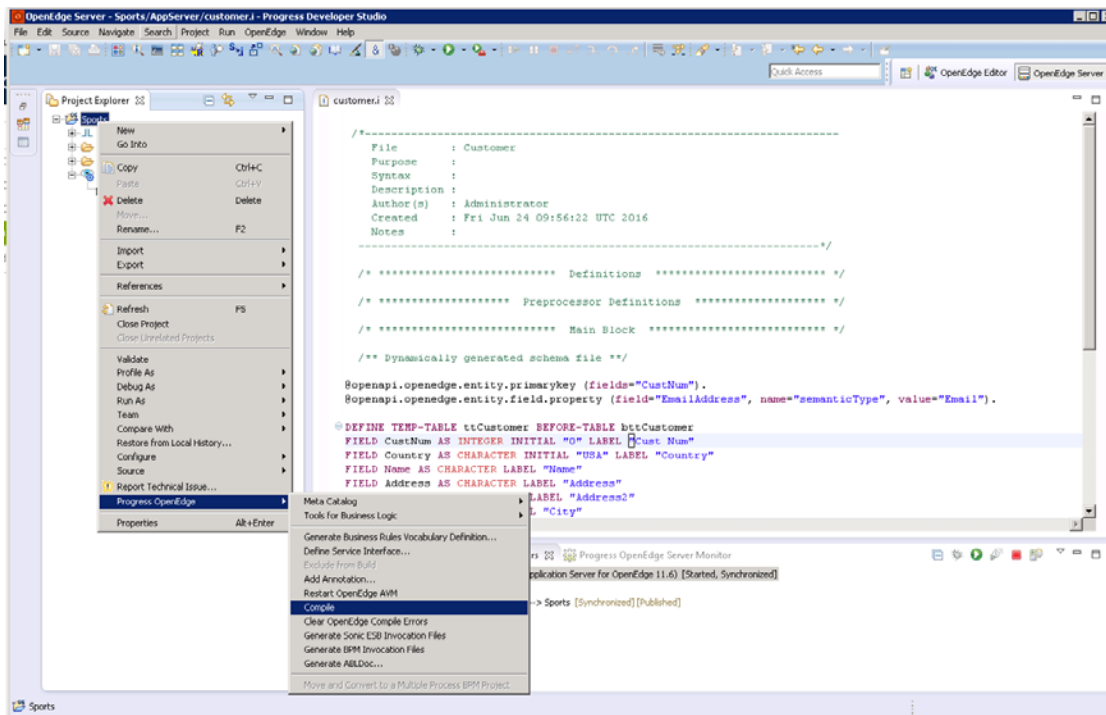




17. Save the customer.i file.

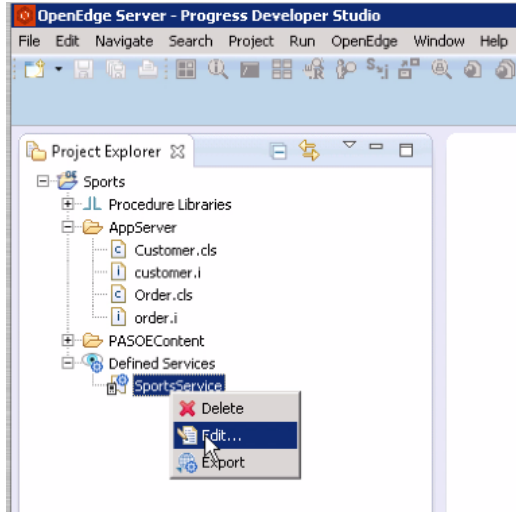


18. Recompile the project to incorporate the changes you just made to the customer.i contents.  
**Right-click on the Sports project in the Project Explorer view. Select Progress OpenEdge -> Compile.**

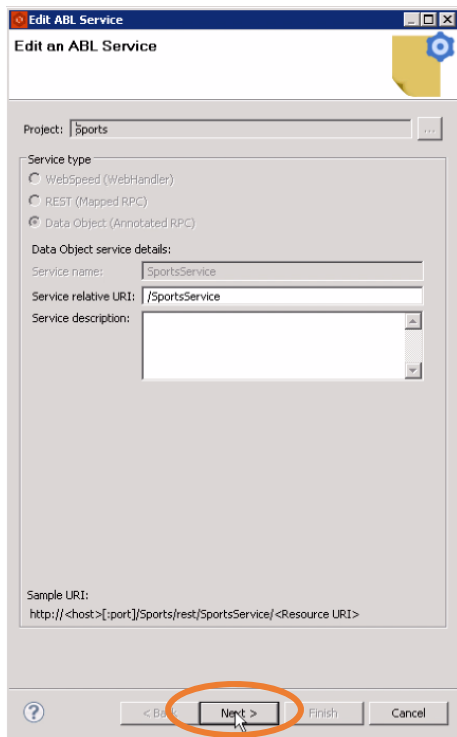


## Select the Resources to Make Available

19. In the next steps, you will make the new class file available to the REST server in PAS for OE. In the **Project Explorer**, expand the **Defined Services** node. **Right-click** on **SportsService** and choose **Edit**:



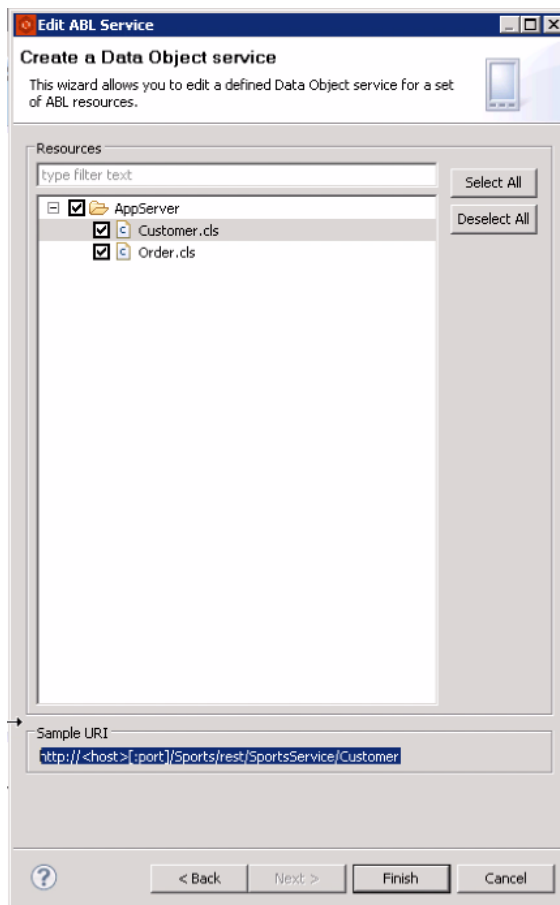
20. Accept the default of **/SportsService** and click **Next**:



21. In the **Resources** section you can choose which procedures to make available to REST. **Check both Customer.cls and Order.cls** by clicking on the boxes next to both items.

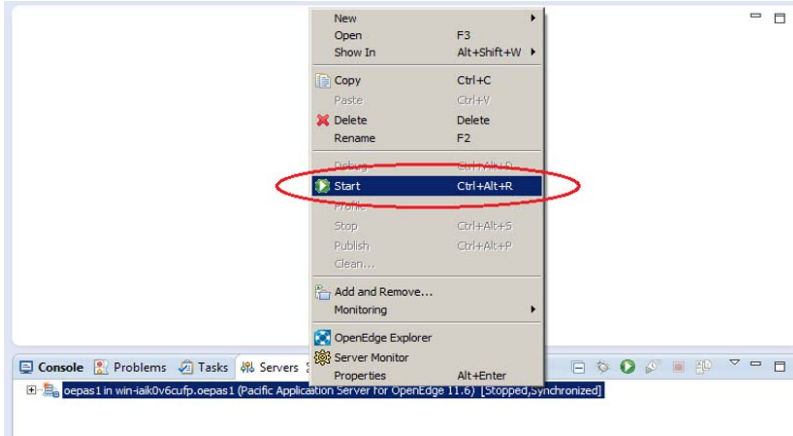
Also on this screen, note that under **Sample URI**, you will see the actual URI that will be used to call each of the two REST services that you are publishing. These URIs are already saved as bookmarks in your Chrome browser instance.

Click **Finish**.



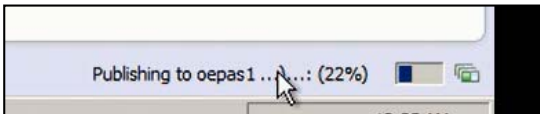
## Deploying the REST service

22. Start the oepas1 server by **right-clicking** on **oepas1** in the Servers view and choosing **Start**:

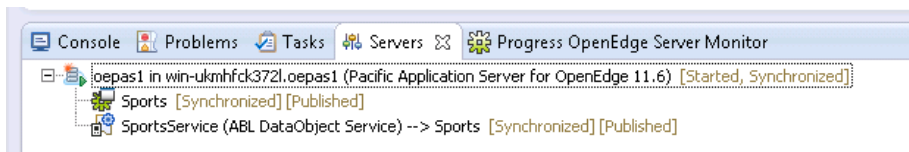


23. At the bottom right corner of the window, the appserver status is displayed.

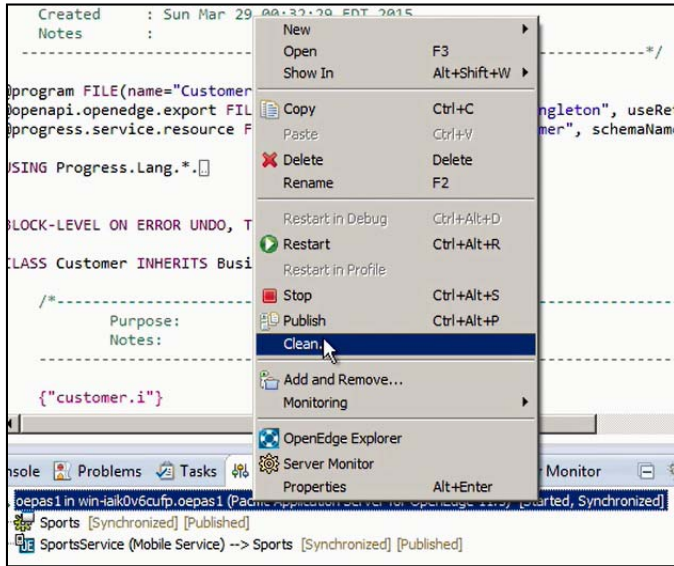
**Wait for the process to complete. This may take a few minutes.**



24. Wait until the oepas1 server status is **Started** and **Synchronized** and the Service is **Published**:



25. NOTE: If your system has trouble completing the synchronization/publish after a few minutes, right-click on **oepas1** and choose **Clean** to clean the environment and re-deploy the services. This process will take a little while to complete.



## Testing the Service in the Browser (JSDO Service)

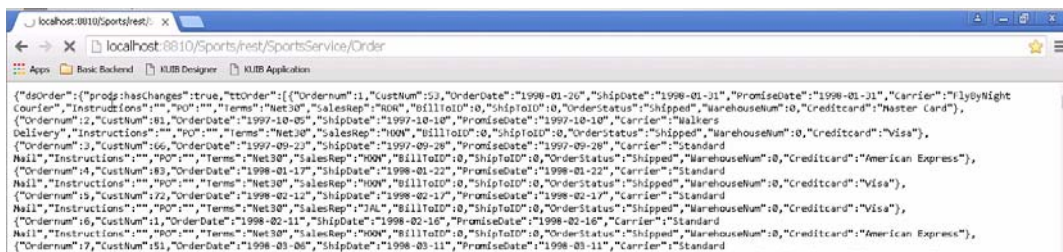
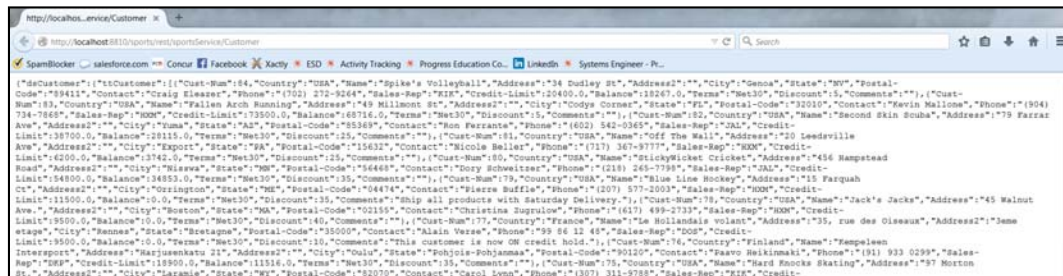
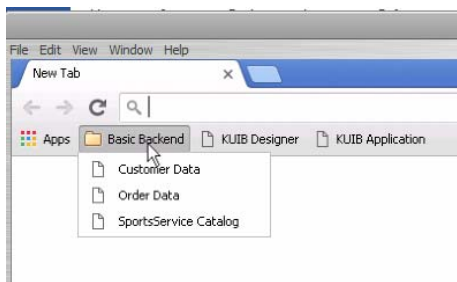


26. Open the **Google Chrome** browser

27. To view all customers enter this URL: <http://localhost:8810/Sports/rest/SportsService/Customer>

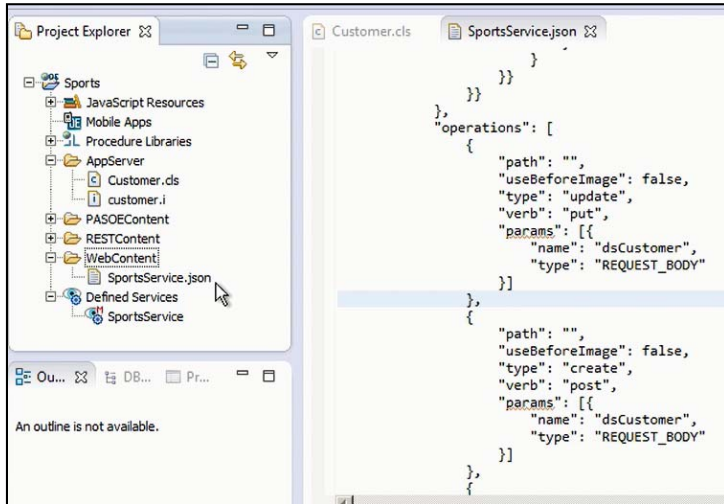
To view all orders enter this URL: <http://localhost:8810/Sports/rest/SportsService/Order>

Or, you can use the **Customer Data** and **Order Data** links that were already saved for you under the **Basic Backend** browser bookmark tab.

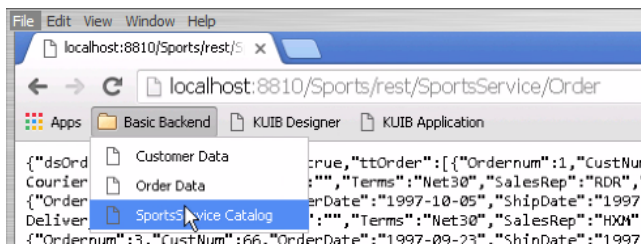


If you see the above screens after you press enter on the URL, you have successfully made the Customer and Order tables available as a REST service.

28. In Developer Studio, publishing the REST service includes the creation of a JSDO catalog. This catalog documents the service in JSON format. To view the catalog, **expand the PASOEContent /static folder** in your project and open **SportsService.json**. The JSDO catalog provides the schema of the data that is being passed during REST operations.



The catalog can be accessed by using this URL in a browser: <http://localhost:8810/Sports/static/SportsService.json> or by using the saved link to SportsServiceCatalog in the **Basic Backend** browser bookmark tab.



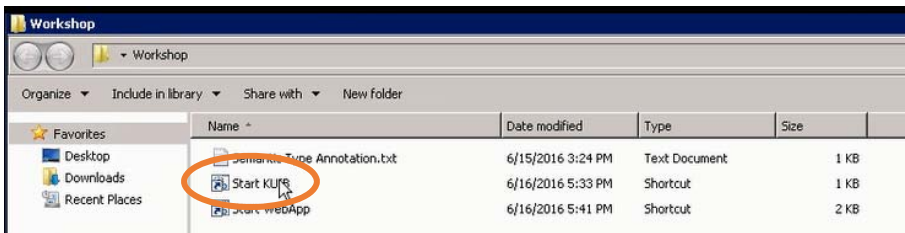
The JSDO catalog will need to be retrieved by the calling services. You reference the catalog URI when you create KUIB samples later in this workshop.

# Section 4: Using OpenEdge Data with Kendo UI Builder for OpenEdge

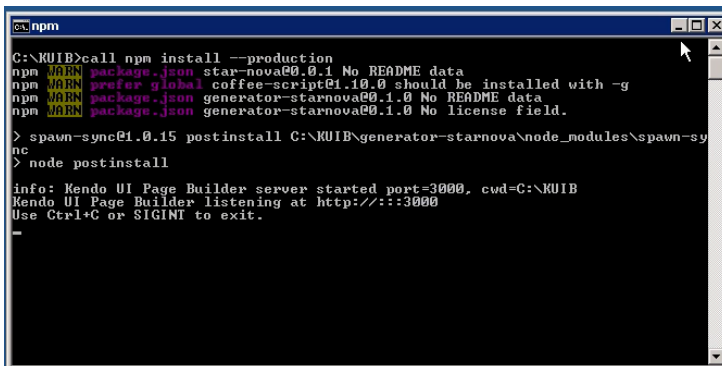
In this section you use the exposed REST data from the preceding section as the datasource for a Kendo UI responsive web application. You build the web application using the Kendo UI Builder for OpenEdge (KUIB) that has been pre-installed in your Arcade image.

## Starting the Kendo UI Builder for OpenEdge (KUIB)

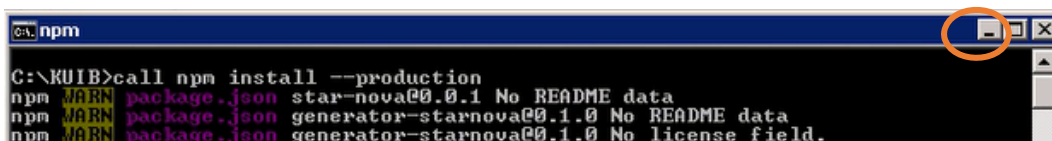
1. The command to start the KUIB has been saved as a Windows Shortcut in the Workshop folder in your Arcade image. Open the Workshop folder on your Arcade desktop, and **double-click** the **Start KUIB** Shortcut icon to start the KUIB as shown:



The command will run in a separate Command window. It will take a little while to start up, and you may see some warning messages which you can disregard. **Wait** until you see the message “Kendo UI Page Builder listening at <http://:::3000>” before continuing:

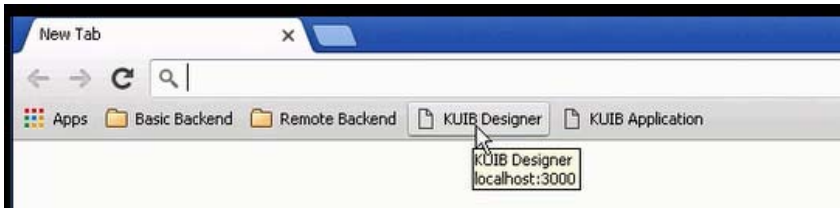


2. **Minimize** the Command Window to leave the KUIB running – do not close it.

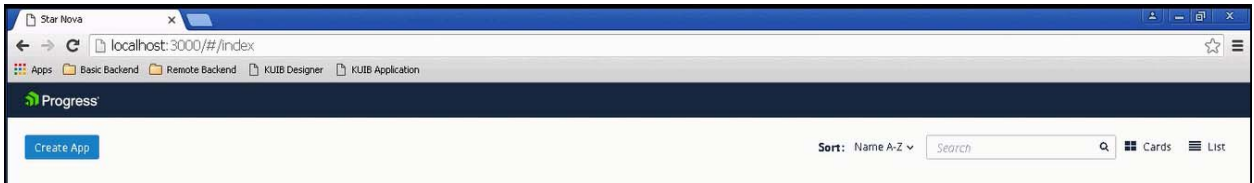




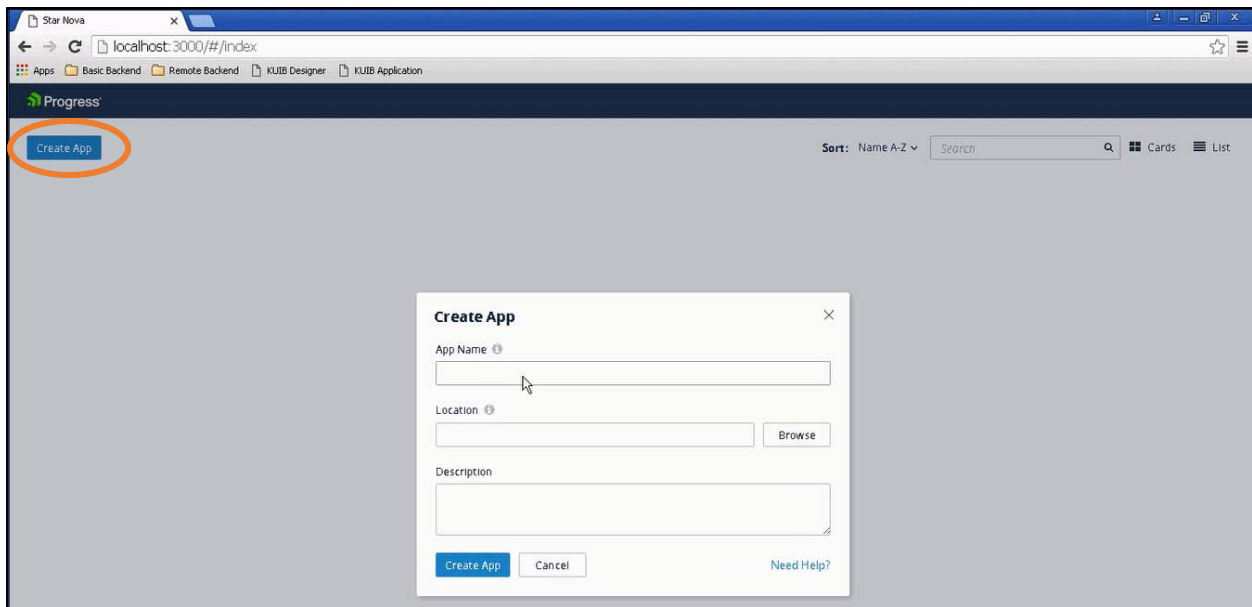
3. **Open Chrome** in your Demo machine window. **Click on the KUIB Designer** bookmark to connect to the KUIB running on your Arcade machine:



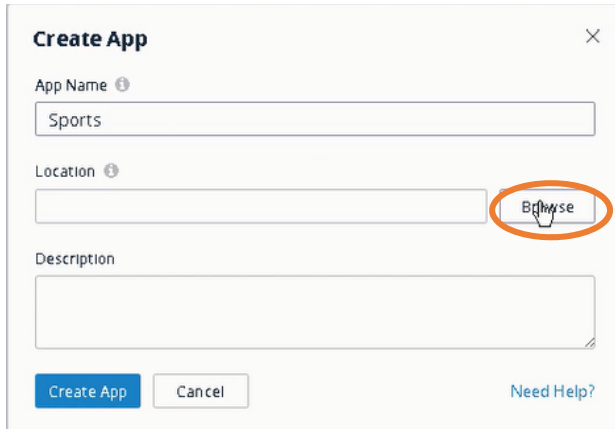
The Kendo UI Builder will open in your browser window:



4. **Click on the Create App** button and the Create App window appears:



5. Specify the App Name **Sports**. Then use the **Browse** button to set the Location:



**Create App** [X]

App Name ⓘ  
Sports

Location ⓘ  
[Empty field] **Browse**

Description  
[Empty text area]

**Create App** Cancel [Need Help?](#)

Navigate using the treeview to **C:\Workspace\SportsKUIB** directory (you must use this directory during this workshop).

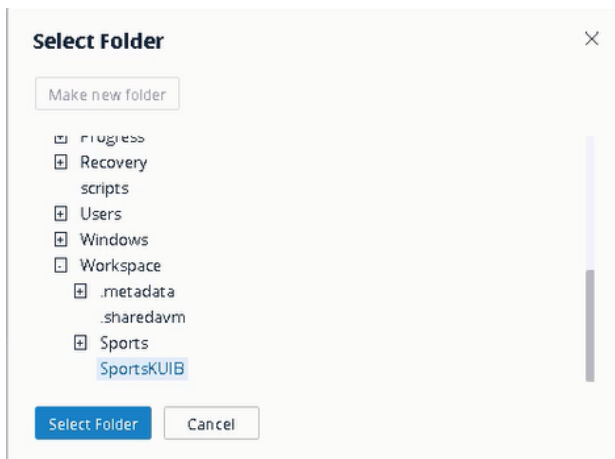


**Select Folder** [X]

Make new folder

**C:**

**Select Folder** Cancel



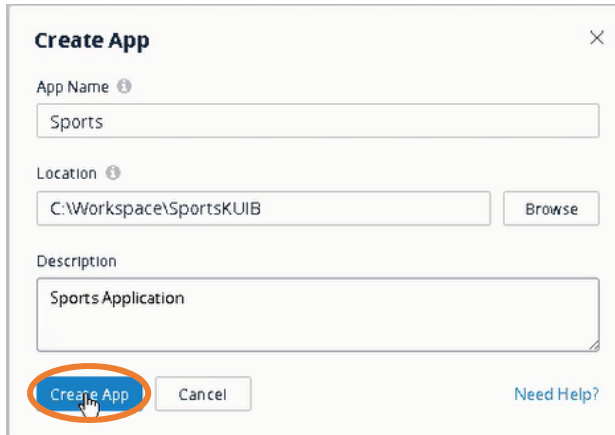
**Select Folder** [X]

Make new folder

- Progress
- Recovery
- scripts
- Users
- Windows
- Workspace
  - .metadata
  - .sharedavm
  - Sports
    - SportsKUIB**

**Select Folder** Cancel

Provide a description of your App, then click the **Create App** button:



**Create App** [Close]

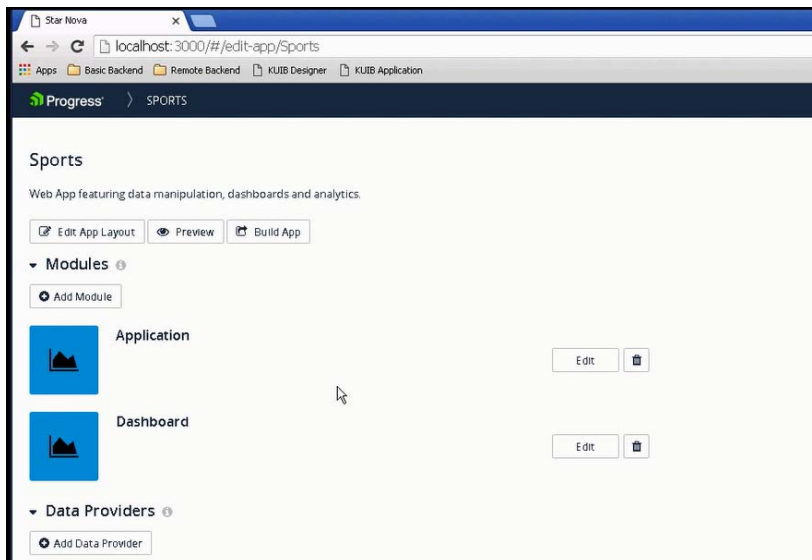
App Name ⓘ  
Sports

Location ⓘ  
C:\Workspace\SportsKUIB [Browse]

Description  
Sports Application

[Create App] [Cancel] [Need Help?](#)

The KUIB will display the Sports app page:



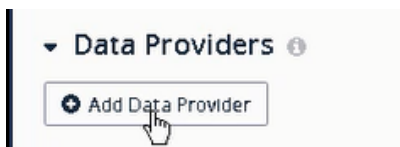
## Adding a Data Provider

In the next steps you configure the data services to be used by your application. You do this by specifying the URIs for the REST services that have been published on the AppServer.

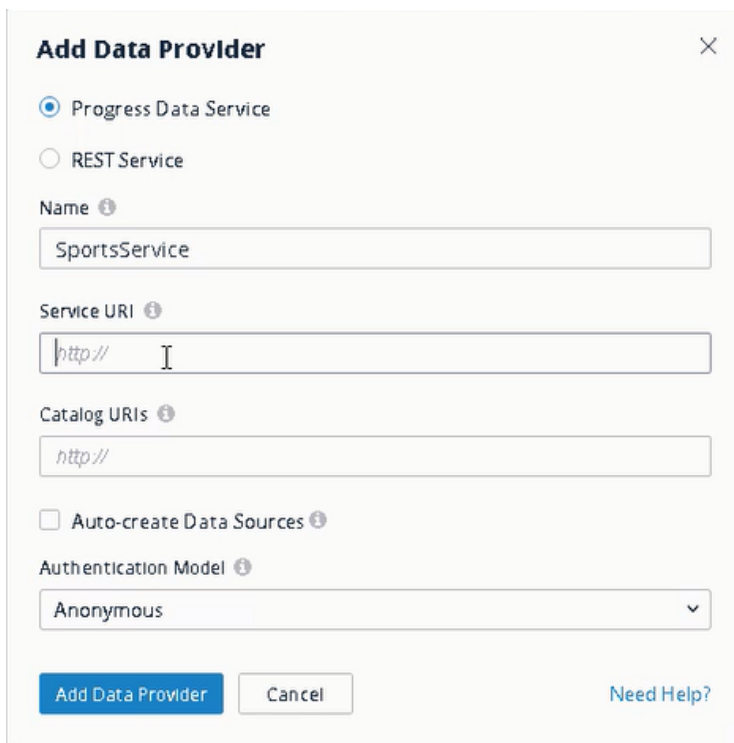
If you were successful in creating both the Customer and Order services earlier in this workshop, use those URIs. They have been saved for you in the “**Basic Backend**” links in the Chrome browser.

If you have not created the Customer and Order services in your Arcade instance, you can use the links provided in the “**Remote Backend**” bookmarks to access a remote Arcade resource instead.

6. Click on the **Add Data Provider** button:



The Add Data Provider window appears in KUIB. Specify the Name as **SportsService**:

A screenshot of a dialog box titled 'Add Data Provider' with a close button (X) in the top right corner. The dialog contains the following elements:

- Two radio buttons: 'Progress Data Service' (selected) and 'REST Service'.
- A text input field labeled 'Name' containing the text 'SportsService'.
- A text input field labeled 'Service URI' containing the text 'http/'.
- A text input field labeled 'Catalog URIs' containing the text 'http/'.
- A checkbox labeled 'Auto-create Data Sources' which is unchecked.
- A dropdown menu labeled 'Authentication Model' with 'Anonymous' selected.
- At the bottom, there are three buttons: 'Add Data Provider' (blue), 'Cancel' (white), and 'Need Help?' (blue text).

7. Enter the **Service URI** and **Catalog URI** for your OpenEdge data services. The data service links have been saved in the Chrome browser bookmark tabs. Use either the **Basic Backend** links to your local data services -or- the **Remote Backend** links if you did not create the services earlier in the workshop. **Note:** The **Service URI** in the saved bookmarks needs to be adjusted as shown.

Check the **Auto-create Data Sources** box. Keep the Authentication Model as **Anonymous** for this example.

#### Local data service:

**Add Data Provider**

Progress Data Service  
 REST Service

Name <sup>?</sup>  
SportsService

Service URI <sup>?</sup>  
http://localhost:8810/Sports

Catalog URIs <sup>?</sup>  
http://localhost:8810/Sports/static/SportsService.json

Auto-create Data Sources <sup>?</sup>

Authentication Model <sup>?</sup>  
Anonymous

Add Data Provider Cancel Need Help?

#### Remote data service:

Service URI:

<http://ec2-54-172-150-158.compute-1.amazonaws.com:8810/Sports>

Catalog URI:

<http://ec2-54-172-150-158.compute-1.amazonaws.com:8810/Sports/static/SportsService.json>

**Add Data Provider**

Progress Data Service  
 REST Service

Name <sup>?</sup>  
SportsService

Service URI <sup>?</sup>  
http://ec2-54-172-150-158.compute-1.amazonaws.com:8810/Sports

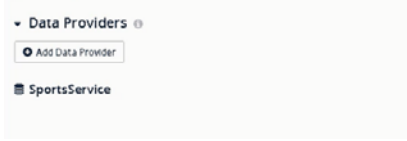
Catalog URIs <sup>?</sup>  
http://ec2-54-172-150-158.compute-1.amazonaws.com:8810/Sports

Auto-create Data Sources <sup>?</sup>

Authentication Model <sup>?</sup>  
Anonymous

Add Data Provider Cancel Need Help?

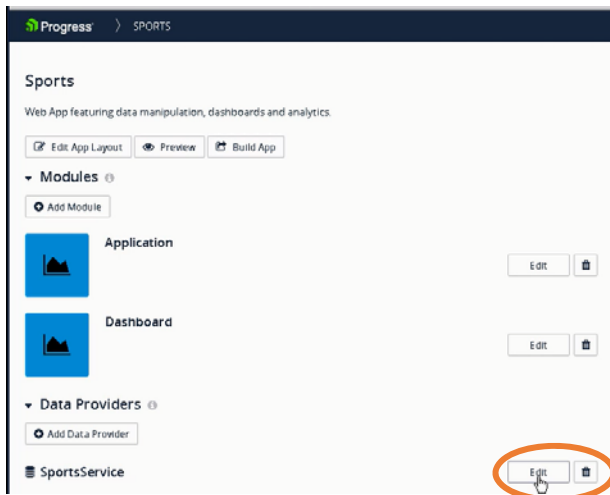
8. Click **Add Data Provider**. You will see that your SportsService Data Provider is added:



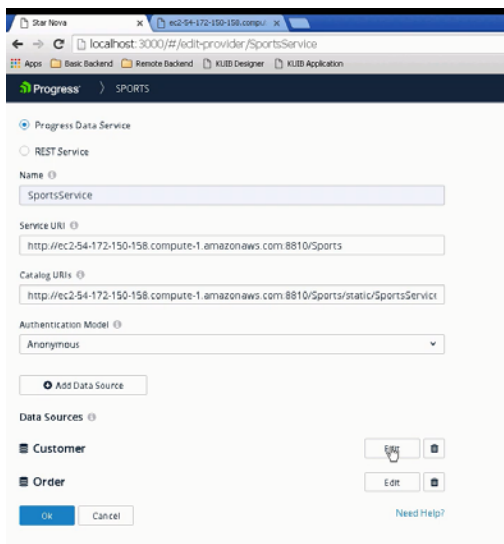
## Editing Data Sources

The next steps will configure the fields available to the web application for the data sources.

9. Click on the **Edit** button for the SportsService provider:

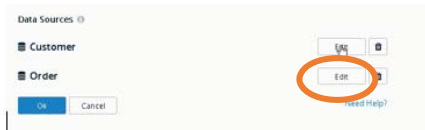


The KUIB will display the properties and Data Sources (Customer and Order) for the SportsService Data Provider:

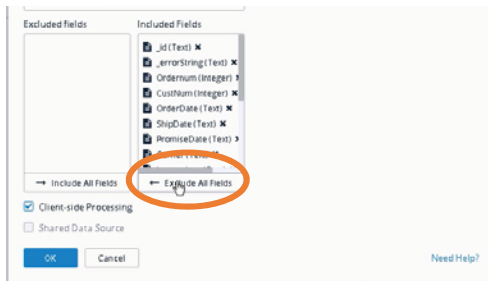


By default all the fields defined for your data source will be available to the application. For this workshop, you will limit the fields that can be used by the Sports application.

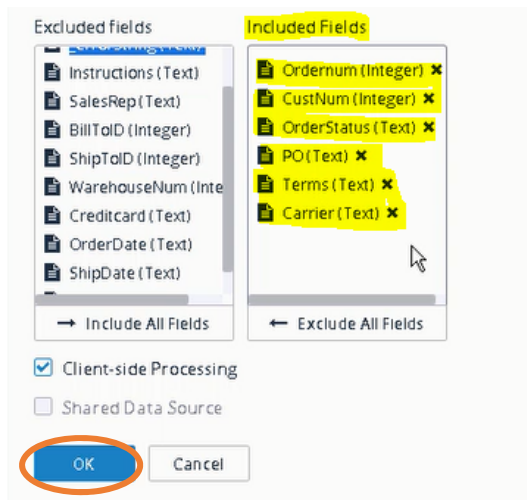
10. Click on the **Edit** button for the **Order** data source to display the available fields and properties of the Order table:



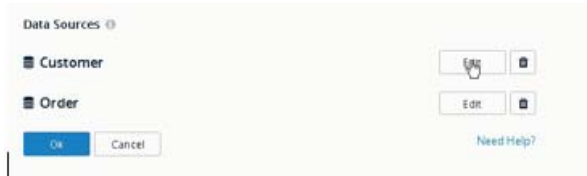
To change the fields available to the application, click on **Exclude All Fields** button to remove all Included Field items.



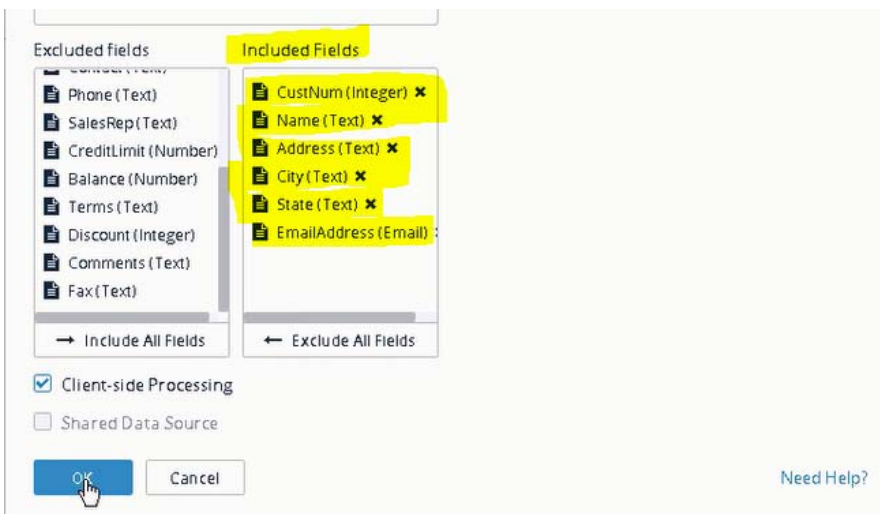
Then **drag** only the desired fields as shown from the Excluded Fields list over to the **Included Fields**, and press **OK**:



11. Click on the **Edit** button for the **Customer** data source to display the available fields and properties of the Customer table:

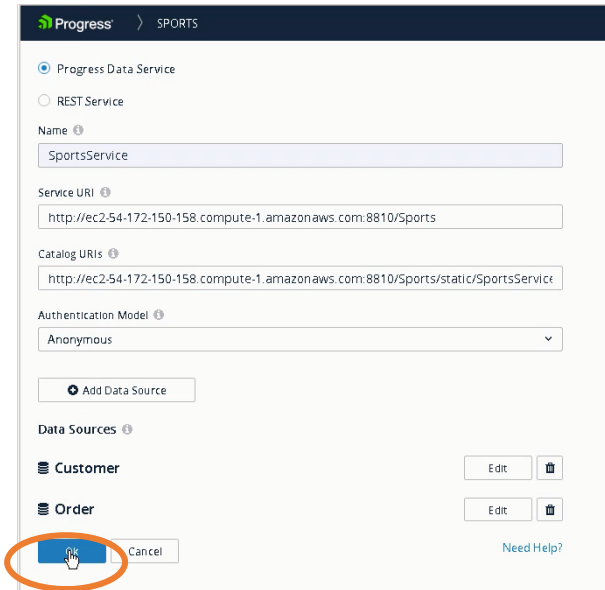


Change the Included Fields as shown, and press **OK**:





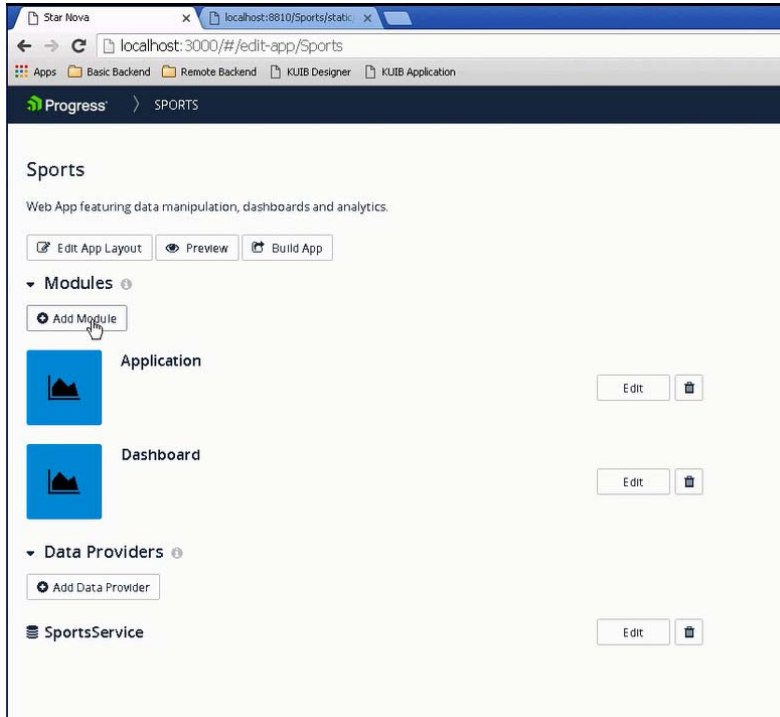
12. The data services are completed. Press **OK** at the Progress Data Service page:



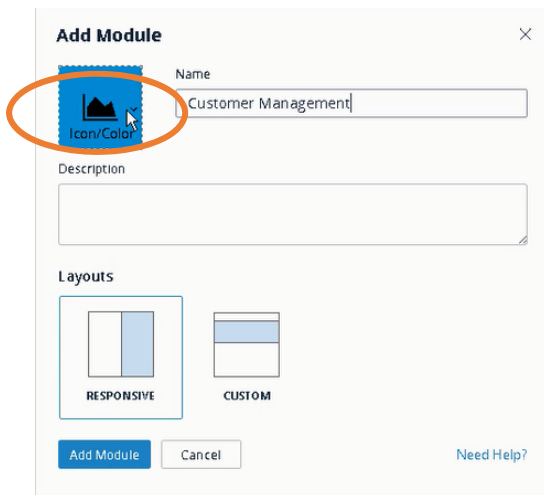
## Creating a DataGrid View

Now that the Data Provider has been created and configured, you can begin building out your web application modules.

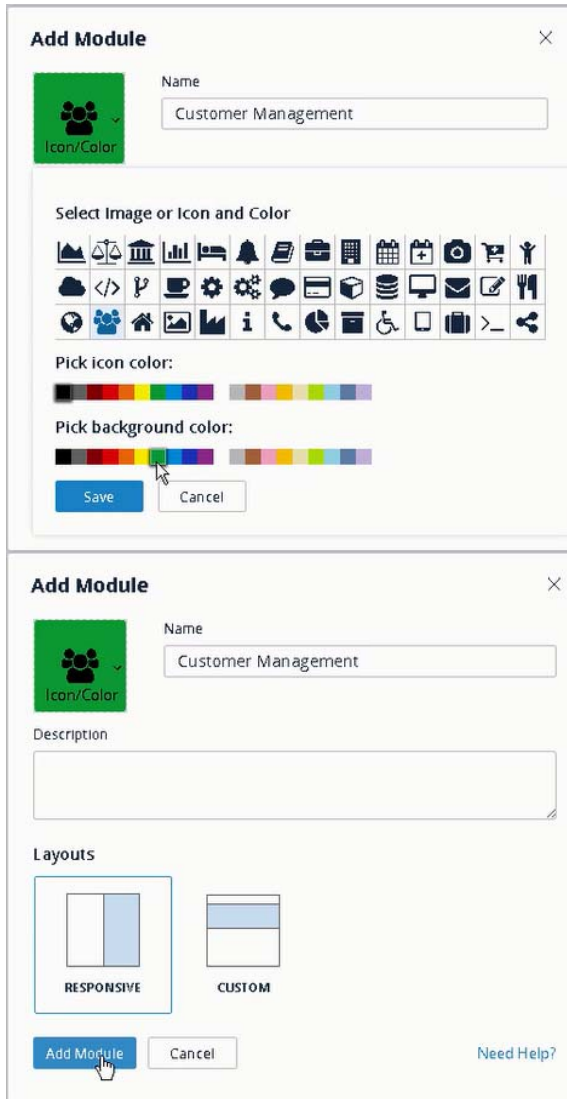
13. In the KUIB, click the **Add Module** button:



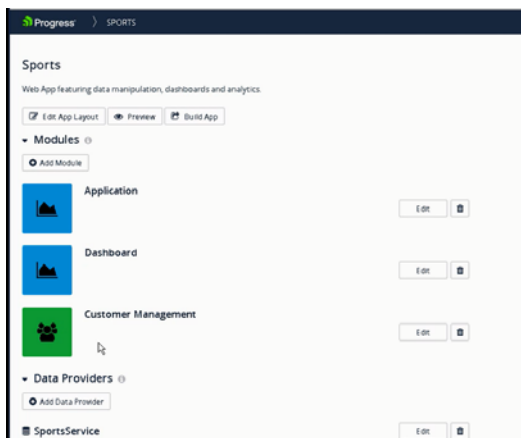
14. Enter the Module Name of **Customer Management** and click on the **Icon/Color** button:



15. Select the desired icon and color, and press **Save**, and then **Add Module**:



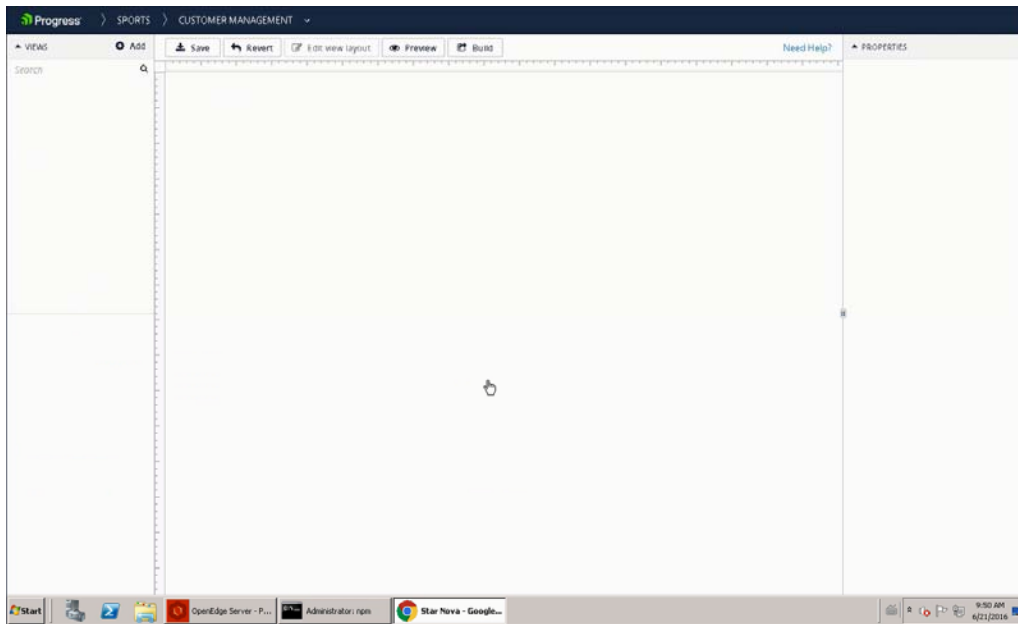
The new module appears in the Sports app page:



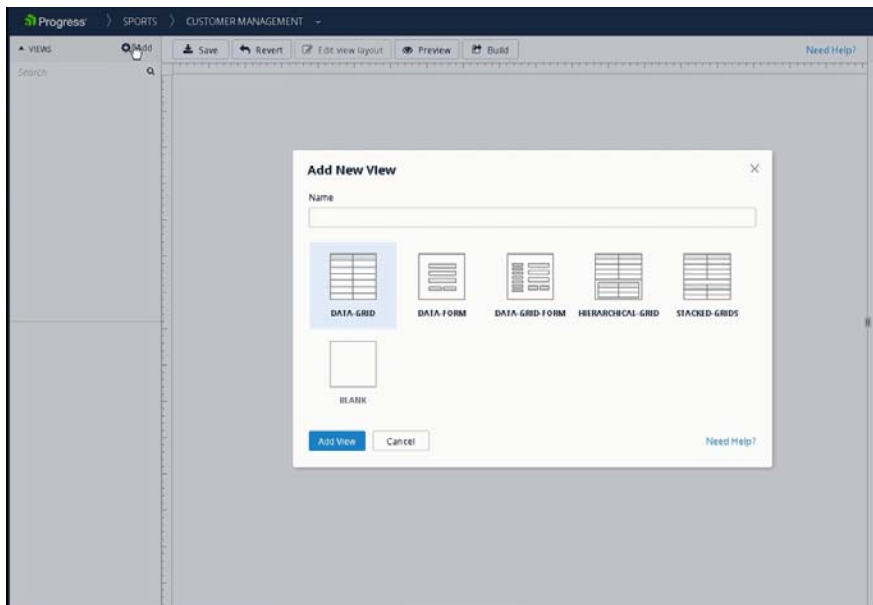
16. Click the **Edit** button for the Customer Management module:



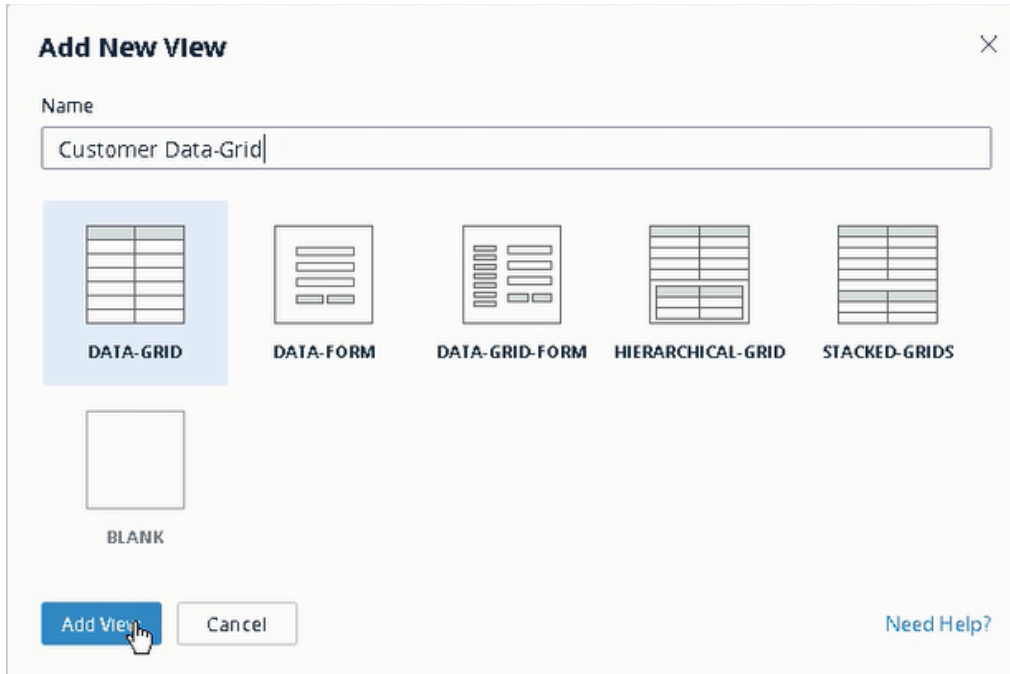
The View Editor appears:



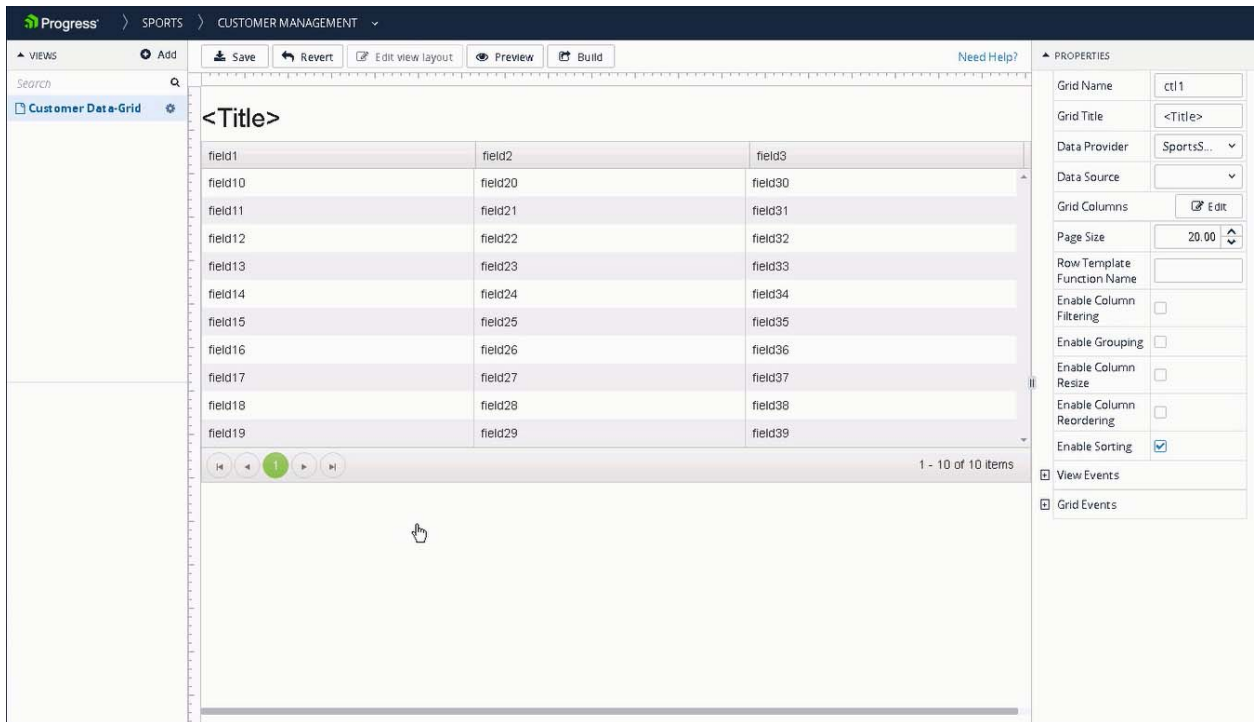
17. In the KUIB View Editor, click on the **+Add** button to add a new view:



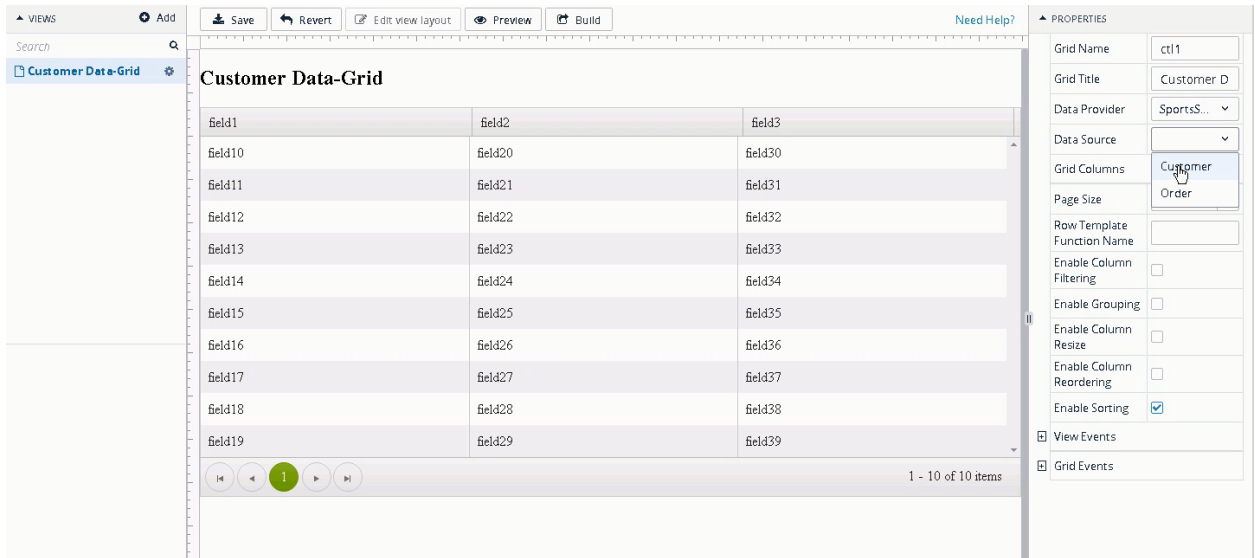
18. Type the name of your new View **Customer Data-Grid**, select the **DATA-GRID** view template, and press the **Add View** button:



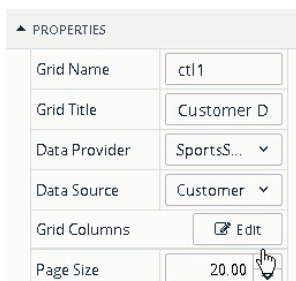
19. The KUIB adds the View to your application and displays the View Properties page:



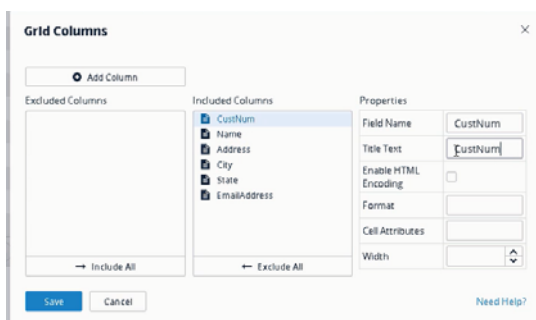
20. In the Properties pane on the right, give the view a title such as **Customer Data-Grid**. Notice that the title is visualized as you enter the value. Specify the **SportsService** as the Data Provider for this View, and select **Customer** as the Data Source:



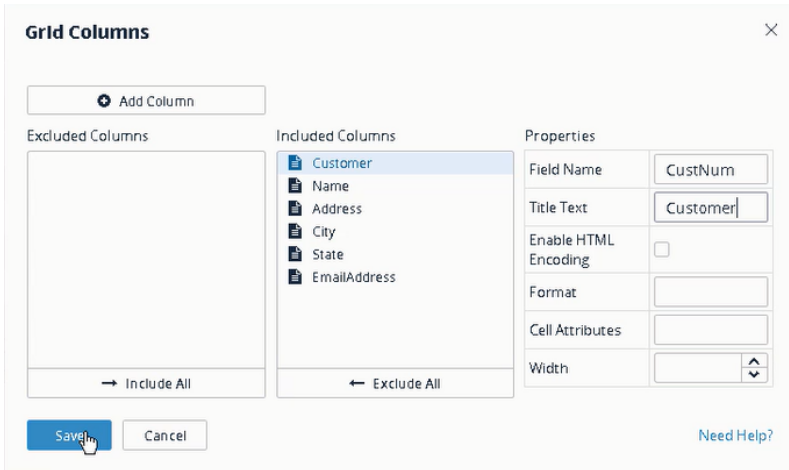
21. Choose **Edit** in the Grid Columns to modify columns in the view:



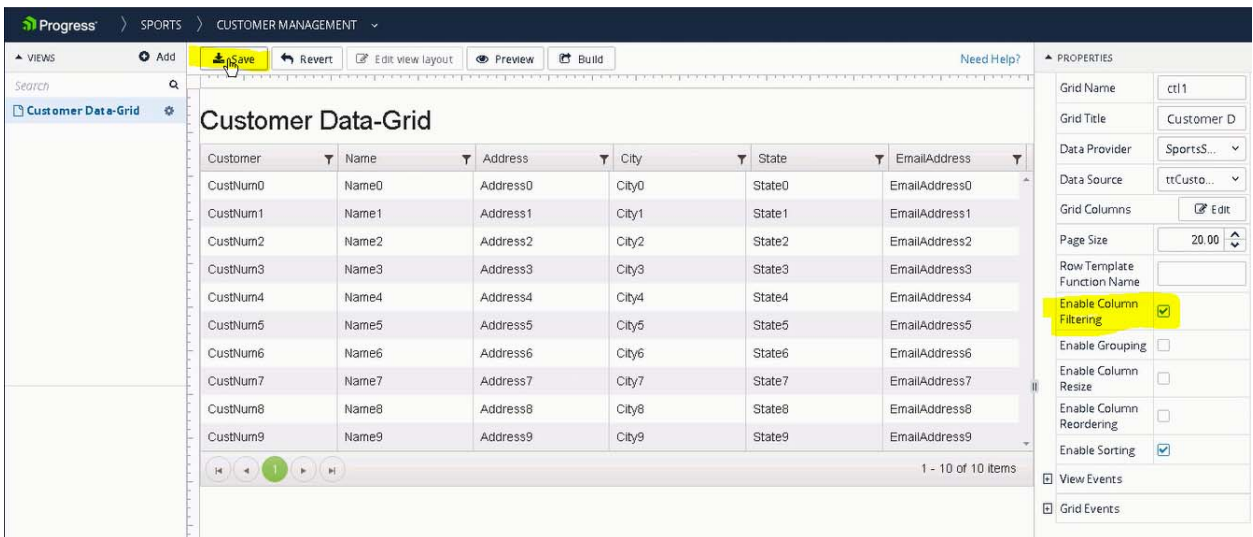
22. Select **Cust-Num** in the Included Columns list:



Modify the Cust-Num Title Text to **Customer**. Notice that the changed field title is visualized in the list of Included Columns for the data grid. Press **Save**:



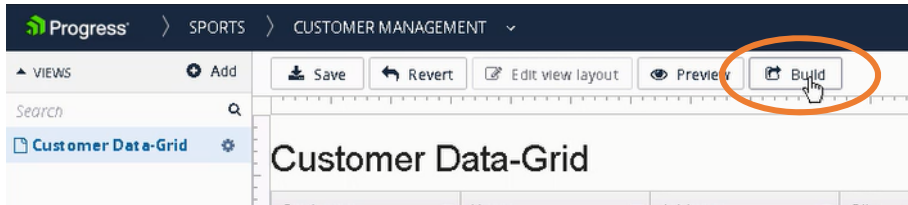
23. In the View Editor, check the box for **Enable Column Filtering**, then press **Save**:



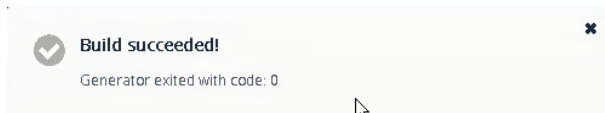
Wait until you see the **Save Succeeded** message box. Dismiss the message.



24. To create your application with the first View, press the **Build** button. This step will take a few moments.



The Build process is complete when you see the message shown below. Dismiss the message.





## Testing the Web Application

25. Minimize the Chrome browser window to access your Arcade desktop.

In the Workshop folder, click on the **StartWebApp** shortcut icon. Disregard warnings and messages that appear.

```
npm
Checking for node_modules...

C:\Workspace\SportsKUIB\Sports>call cd webapp

C:\Workspace\SportsKUIB\Sports\webapp>call npm install
npm WARN deprecated minimatch@0.2.14: Please update to minimatch 3.0.2 or higher
to avoid a RegExp DoS issue
npm WARN deprecated graceful-fs@1.2.3: graceful-fs v3.0.0 and before will fail on
node releases >= v7.0. Please update to graceful-fs@^4.0.0 as soon as possible
. Use 'npm ls graceful-fs' to find it in the tree.
npm WARN deprecated minimatch@0.3.0: Please update to minimatch 3.0.2 or higher
to avoid a RegExp DoS issue
```

When the process completes, you will see the sample Application Desktop displayed in your Chrome browser:

The screenshot shows a web browser window displaying a dashboard for 'Progress'. The browser address bar shows 'localhost:9000/#/dashboard'. The dashboard includes a sidebar with 'Customer Management' and 'Dashboard' options. The main content area features several widgets:

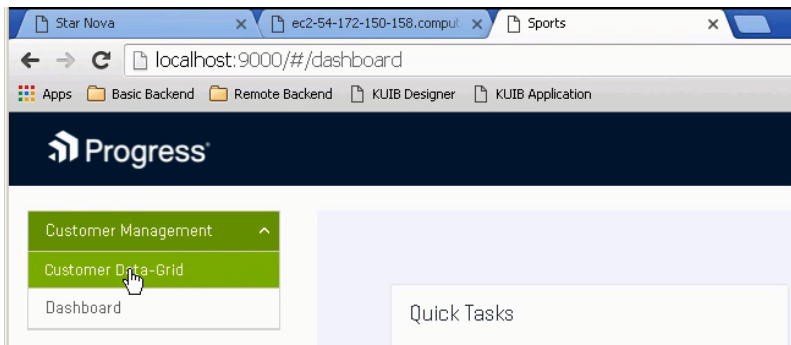
- Quick Tasks:** Buttons for 'New Order', 'New Quote', 'New Product', and 'New Employee'.
- Sales Pipeline:** A funnel chart showing stages: Needs Analysis (\$225,000), Prospecting (\$160,000), Perception Analysis (\$85,000), and Qualification (\$45,000). Summary: Total Opportunities: \$515k, New Opportunities: \$96.5k.
- Sales Actuals vs. Quota:** A gauge chart showing Monthly Actual (\$26.8k), YTD Actual (\$335.5k), and YTD Goal (\$1M).
- Top Open Opportunities (Current Month):** Table with columns: Company, Probability, Amount.
- Top Open Activities (This Week):** Table with columns: Company, Count.
- Top Closed Opportunities (Current Month):** Table with columns: Company, Amount.

| Company           | Probability | Amount    |
|-------------------|-------------|-----------|
| Quinu             | 10%         | \$150,000 |
| Eadel Enterprises | 61%         | \$110,000 |
| Alvee Inc         | 42%         | \$85,000  |
| Talane Sports     | 60%         | \$24,229  |
| Wikizz Resources  | 20%         | \$12,000  |
| Total             |             | \$515,000 |

| Company           | Count |
|-------------------|-------|
| Quinu             | 5     |
| Eadel Enterprises | 5     |
| Alvee Inc         | 4     |
| Talane Sports     | 4     |
| Wikizz Resources  | 2     |
| Total             | 20    |

| Company           | Amount   |
|-------------------|----------|
| Quinu             | \$12,000 |
| Eadel Enterprises | \$5,000  |
| Alvee Inc         | \$4,465  |
| Talane Sports     | \$2,300  |
| Wikizz Resources  | \$2,000  |
| Total             | \$26,865 |

26. In addition to the sample application, you also see a link in the upper left for the Customer Management application you are building. **Expand the Customer Management link and select the Customer Data-Grid view.**



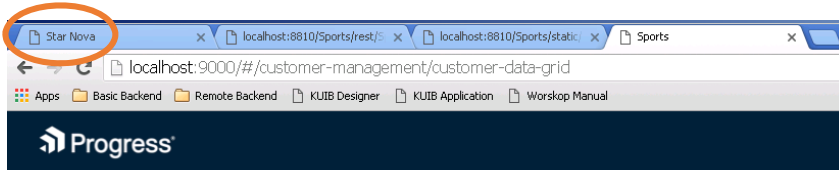
27. Your Customer Data-Grid view is displayed with data from your OpenEdge database:

| Customer | Name                 | Address              | City        | State           | EmailAddress |
|----------|----------------------|----------------------|-------------|-----------------|--------------|
| 1        | Lift Tours           | 276 North Drive      | Burlington  | MA              |              |
| 2        | Urpon Frisbee        | Rattipolku 3         | Oslo        | Uusima          |              |
| 3        | Hoops                | Suite 415            | Atlanta     | GA              |              |
| 4        | Go Fishing Ltd       | Unit 2               | Harrow      | Middlesex       |              |
| 5        | Match Point Tennis   | 66 Homer Pl          | Boston      | MA              |              |
| 6        | Fanatical Athletes   | 20 Bicep Bridge Rd   | Montgomery  | AL              |              |
| 7        | Aerobics valine Ky   | Peltolantie 3        | Tikkurila   | Uusimaa         |              |
| 8        | Game Set Match       | Box 60               | Deatsville  | AL              |              |
| 9        | Pihtiputaan Pyora    | Putikentie 2         | Pihtipudas  | Etela-Savo      |              |
| 10       | Just Joggers Limited | Fairwind Trading Est | Ramsbottom  | Sussex          |              |
| 11       | Kellailu ja Biljardi | Vattuniemenkuja 8 A  | Helsinki    | Uusimaa         |              |
| 12       | Surf Lautaveikkoset  | Venemestarinkatu 35  | Salo        | Varsinais-Suomi |              |
| 13       | Biljardi ja tennis   | Urhellutie 1         | Mantsala    | Uusimaa         |              |
| 14       | Paris St Germain     | 113, avenue du Stade | Paris       | Seine           |              |
| 15       | Hoopla Basketball    | 87 Calumnet St       | Egg Harbor  | NJ              |              |
| 16       | Thundering Surf Inc. | 354 Market St.       | Coffee City | TX              |              |
| 17       | High Tide Sailing    | 178 Schooner Hill    | Immingham   | Humberside      |              |
| 18       | Antin Metsastysse    | Vanhainkodikuja 1    | Sysma       | Keski-Suomi     |              |
| 19       | Buffalo Shuffleboard | 155 Carolina Ave     | Buffalo     | NY              |              |
| 20       | Espoon Pallokeskus   | Sinikalliontie 18    | Espoo       | Uusimaa         |              |

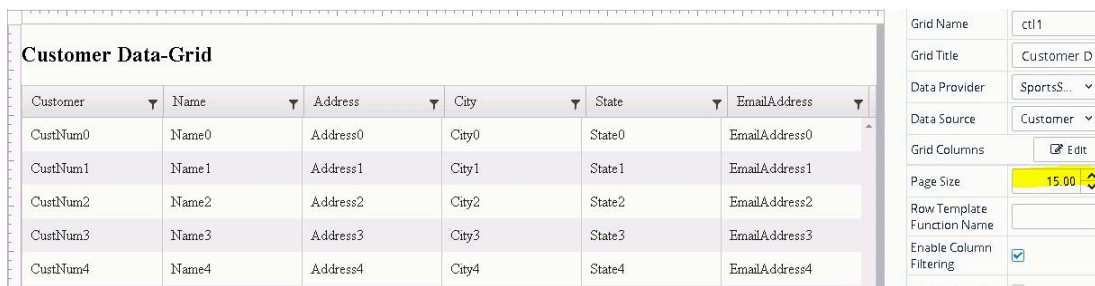
## Modifying a View

In this section you make a change to the Customer Data-Grid view created earlier.

28. In the Chrome browser window, click on the tab labelled **Star Nova** to return to the KUIB designer page:

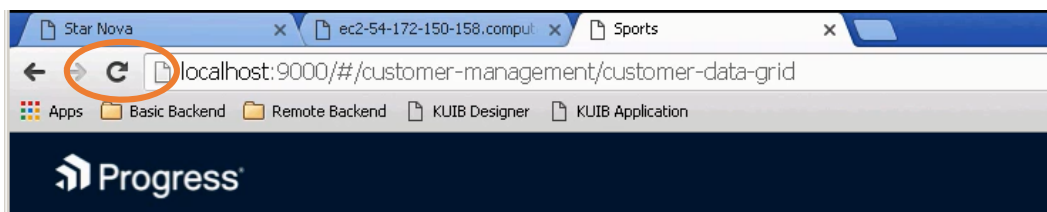


29. Change the setting of the **Page Size** attribute to **15**:




30. Press **Save** and then press **Build** to re-generate your application. Wait for the completion message to appear and dismiss it.

31. Because your web application is already running, you don't need to start it again. Simply click on the **Sports** tab in the Chrome browser window and **refresh** the page in the browser:



### 32. The modified Customer Grid-View displays:

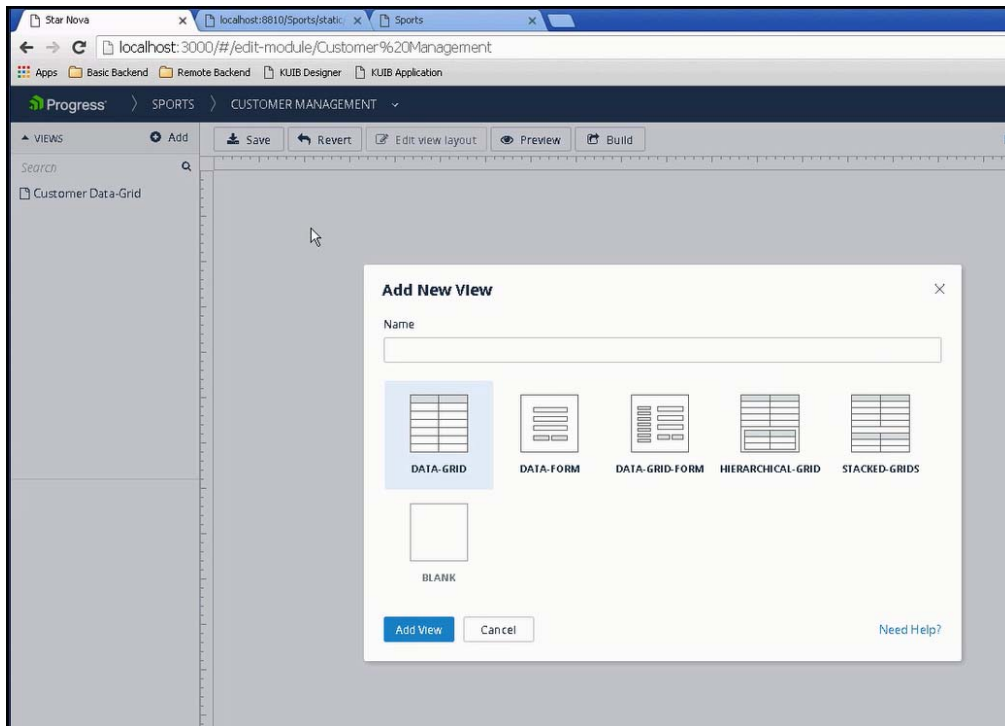
| Customer | Name                 | Address              | City       | State           | EmailAddress |
|----------|----------------------|----------------------|------------|-----------------|--------------|
| 1        | Lift Tours           | 276 North Dr.        | Burlington | MA              |              |
| 2        | Urpon Frisbee        | Rattipolku 3         | Oslo       | Uusima          |              |
| 3        | Hoops                | Suite 415            | Atlanta    | GA              |              |
| 4        | Go Fishing Ltd       | Unit 2               | Harrow     | Middlesex       |              |
| 5        | Match Point Tennis   | 66 Homer Pl          | Boston     | MA              |              |
| 6        | Fanatical Athletes   | 20 Bicep Bridge Road | Montgomery | AL              |              |
| 7        | Aerobics valine Ky   | Peltolantie 3        | Tikkurila  | Uusimaa         |              |
| 8        | Game Set Match       | Box 60               | Deatsville | AL              |              |
| 9        | Pihtiputaan Pyora    | Putikontie 2         | Pihtipudas | Etelä-Savo      |              |
| 10       | Just Joggers Limited | Fairwind Trading Est | Ramsbottom | Sussex          |              |
| 11       | Keilailu ja Biljardi | Vattuniemenkuja 8 A  | Helsinki   | Uusimaa         |              |
| 12       | Surf Lautaveikkoset  | Venemestarinkatu 35  | Salo       | Varsinais-Suomi |              |
| 13       | Biljardi ja tennis   | Urheilutie 1         | Mantsala   | Uusimaa         |              |
| 14       | Paris St Germain     | 113, avenue du Stade | Paris      | Seine           |              |
| 15       | Hoopla Basketball    | 87 Calumnet St       | Egg Harbor | NJ              |              |

33. If you haven't done so already, take a few moments to try out the built-in features of the Customer Data-Grid: Page through the data using the page navigation at the bottom; Sort on any column by clicking the column heading; Filter the data records displayed by clicking on the Filter icon  in any column.

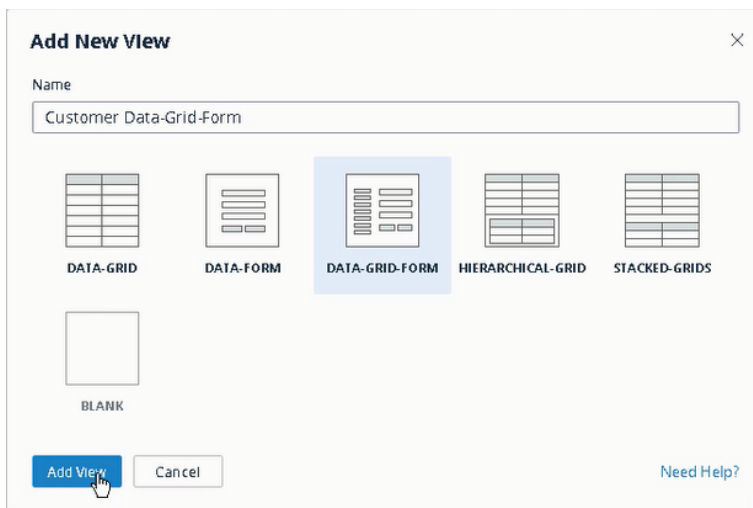
## Creating a DataGridForm View

In this section, you will add another Customer View using the DataGridForm template. The DataGridForm view combines a data grid and a data modification form on a single page.

34. Click on the **Star Nova** tab in your Chrome browser to return to the KUIB design page, then click on the **+Add** button to add a new view to your Sports application:

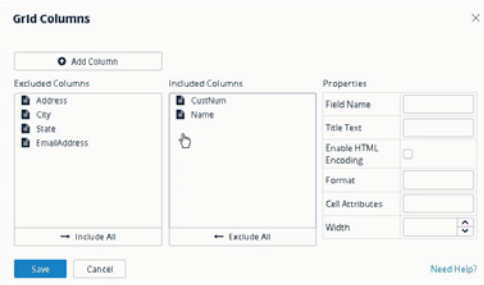


35. Select the **DATA-GRID-FORM** template, give the View a name such as **Customer Data-Grid-Form**, and click **Add View**:

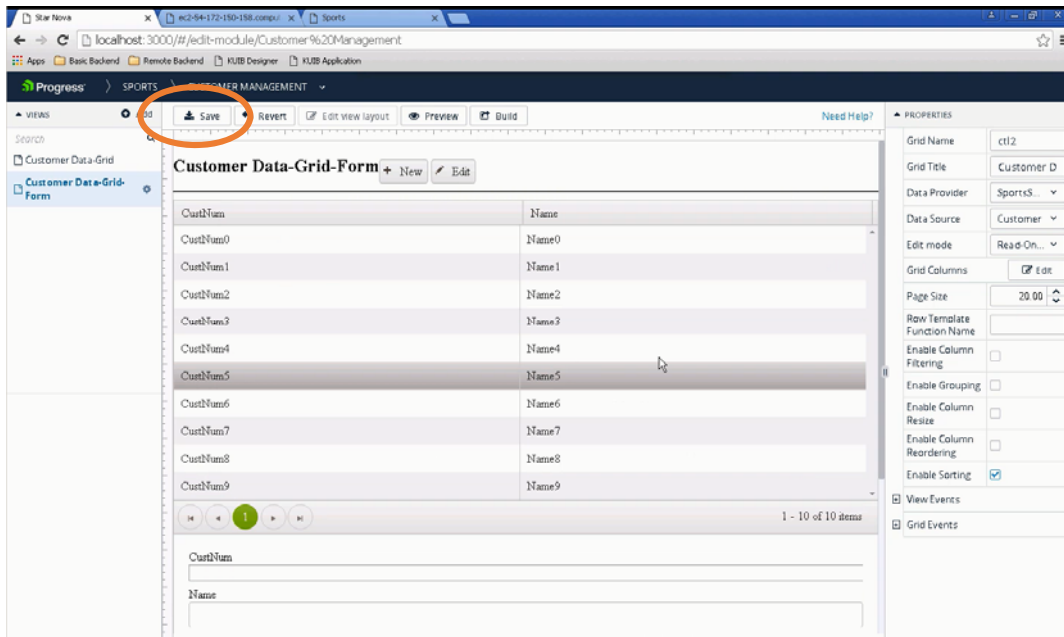


36. In the Properties pane, specify:
- Grid Title **Customer Data-Grid-Form**
  - Data Provider **SportsService**
  - Data Source **Customer**
  - Page Size **15.00**
  - Toggle on **Enable Column Filtering**

Click **Edit** Grid Columns and modify the column list to include only **CustNum** and **Name** and **Click Save** in the Grid Columns editor:

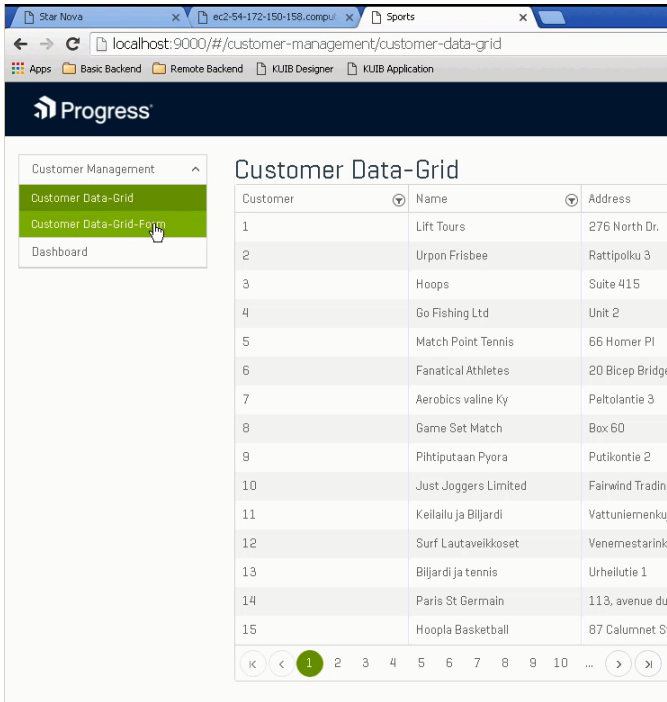


Accept the default values for other properties of the Customer Data-Grid-Form view and click **Save**:

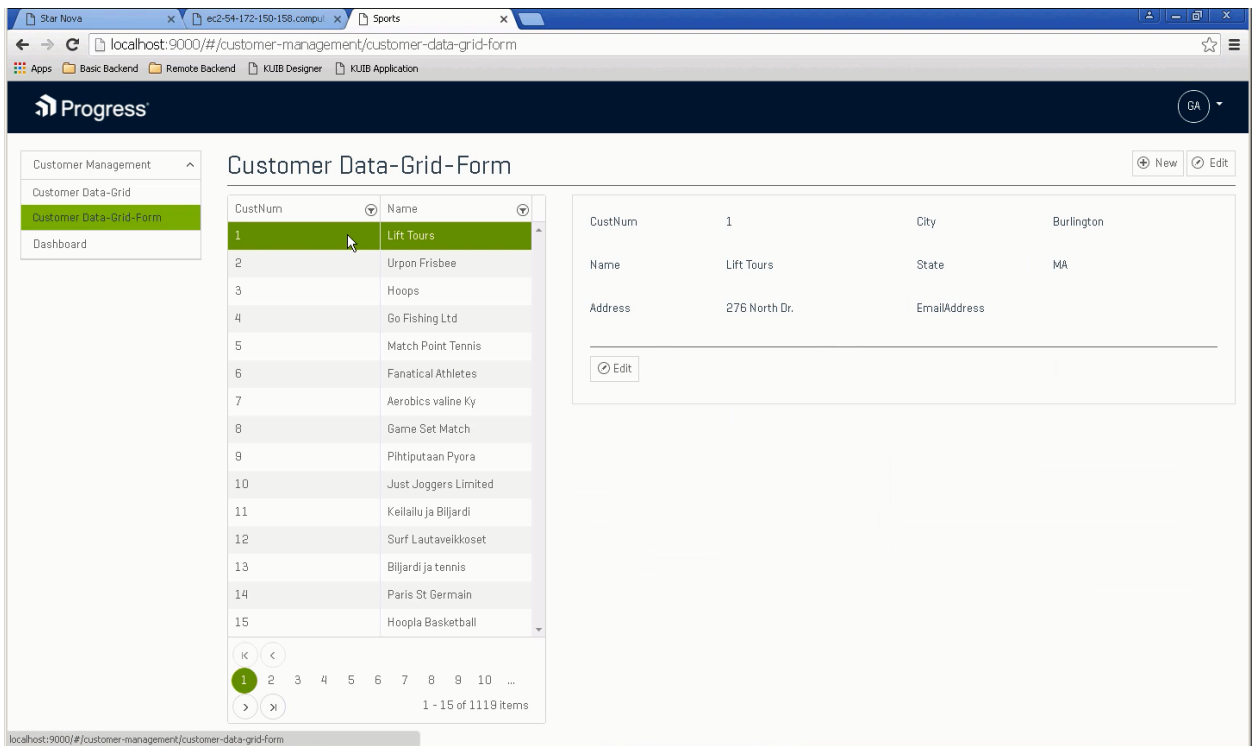



37. Press **Build** to re-generate your application. Dismiss the successful completion message box when it appears.

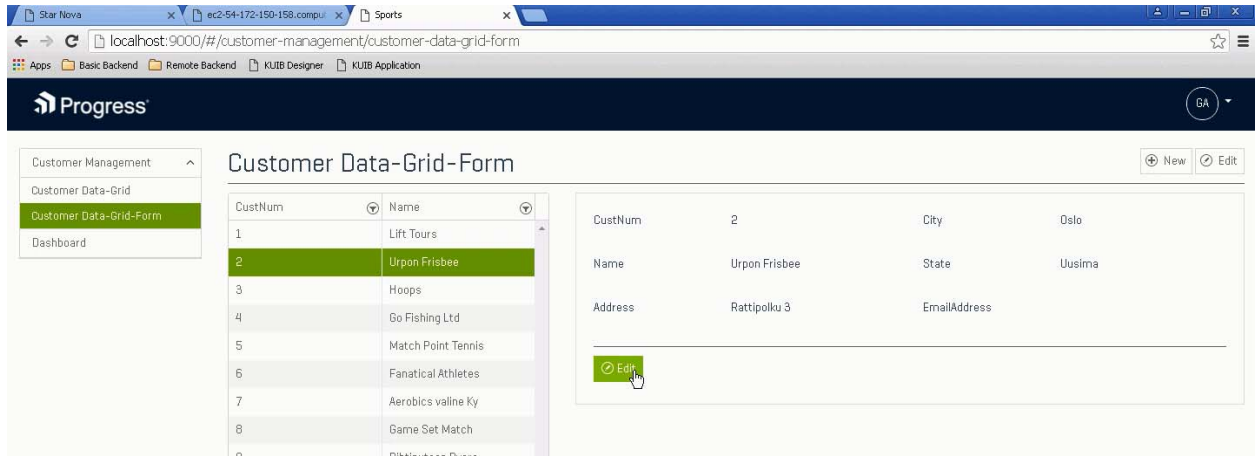
38. Select the **Sports** tab in your Chrome browser and **refresh** the application page. The new view is shown in the menu at the top left:



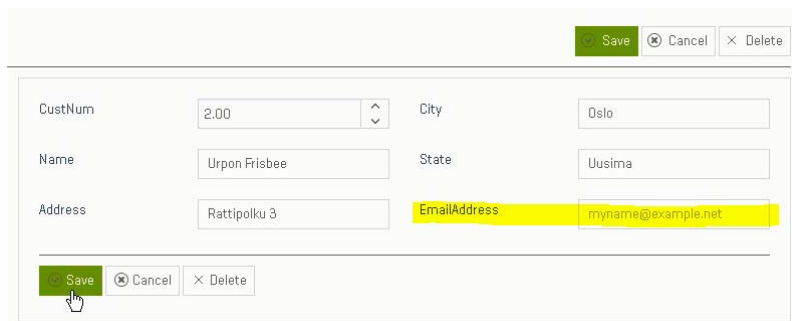
39. Select the Customer Data-Grid-Form from the menu to display the new view:



40. With a Customer record highlighted, click the **Edit**  button in the Form:



41. Notice that the EmailAddress entry field in the form prompts with the format required for the Customer.EmailAddress field based on the semantic data type you defined earlier for the Customer Data Source:



42. Enter an email address for the Customer and press **Save**.

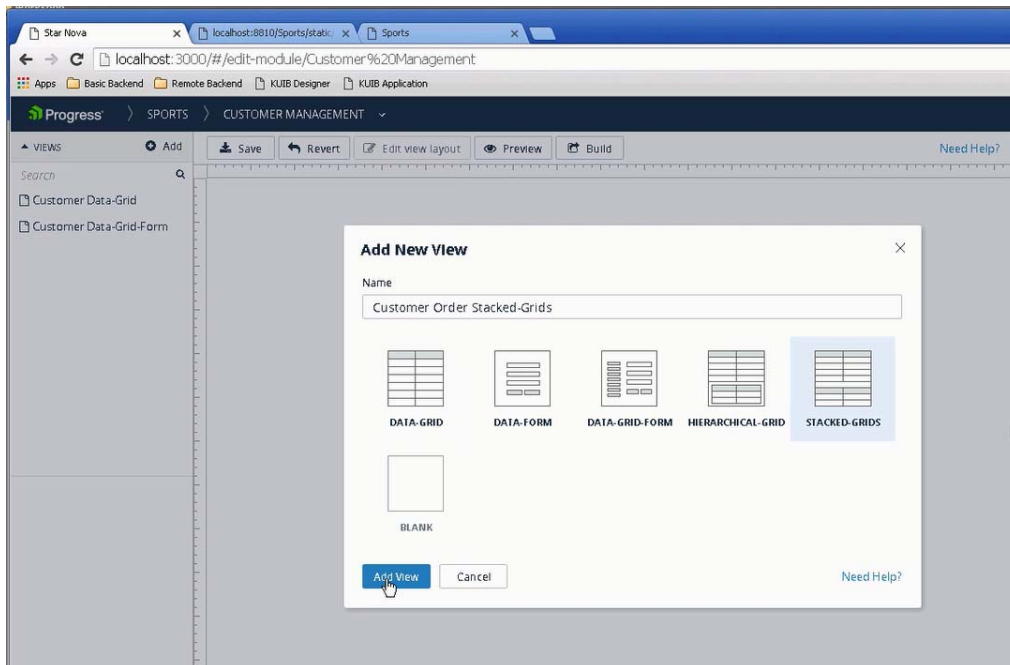
43. To test the responsive sizing of this View, **double-click** in the browse window **title bar** to **resize** the Chrome browser window, forcing the View to adjust to the change.



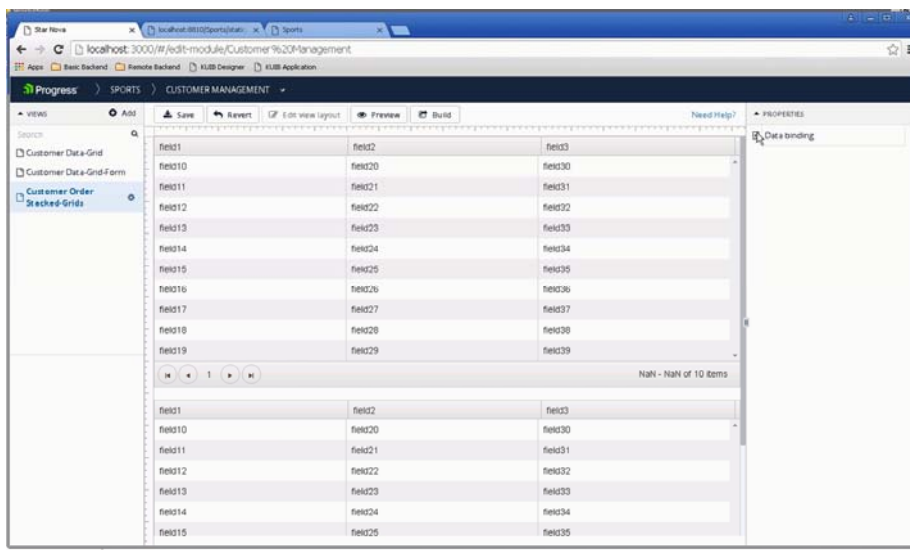
## Creating a StackedGrid View

In this section you will create a Stacked Grid View of Master/Detail data using the Customer table as the Master record and the Order table as the related Detail records.

44. Return to the KUIB Designer by clicking on the Star Nova tab, and **Add** a new view. Provide a name such as **Customer Order Stacked-Grids**, select the **STACKED-GRIDS** template and click **Add View**:



The KUIB displays the new View for editing:



45. Enter the Grid Title **Customer Order Stacked-Grids**.

Expand the **DataBinding** node and select the following values from the drop-down lists:

Master Data Source: **Customer**

Details Data Source: **Order**

Master Primary Key: **CustNum**

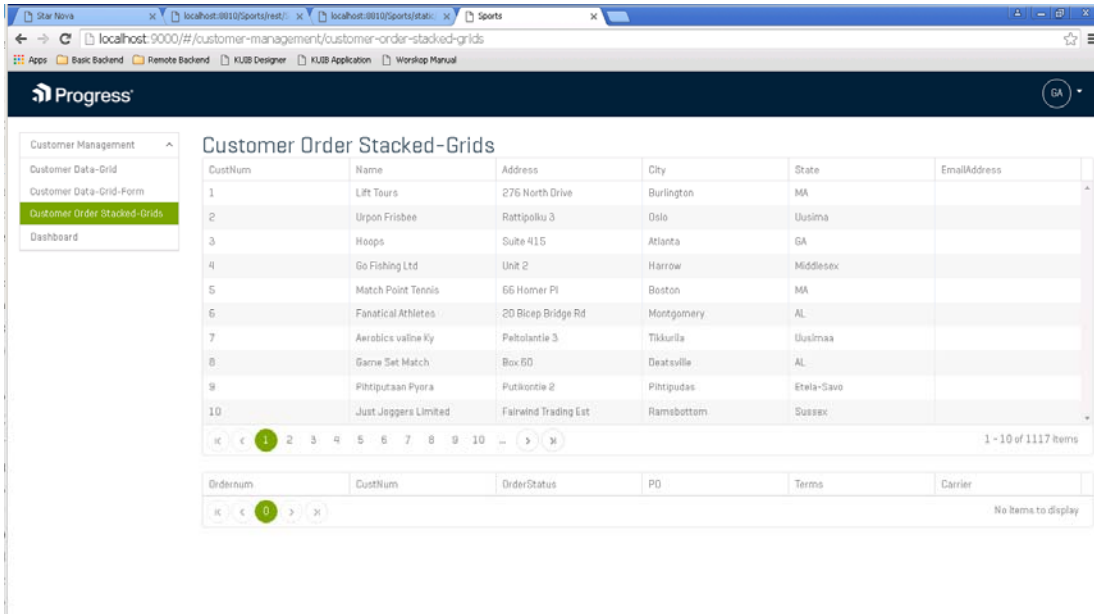
Details Foreign Key: **CustNum**

The screenshot shows the Kendo UI Builder interface for a 'Customer Order Stacked-Grids'. The main area displays two stacked data grids. The top grid has columns: CustNum, Name, Address, City, State, and EmailAddress. The bottom grid has columns: OrderNum, CustNum, OrderStatus, PO, Terms, and Carrier. The Properties pane on the right shows the following configuration:

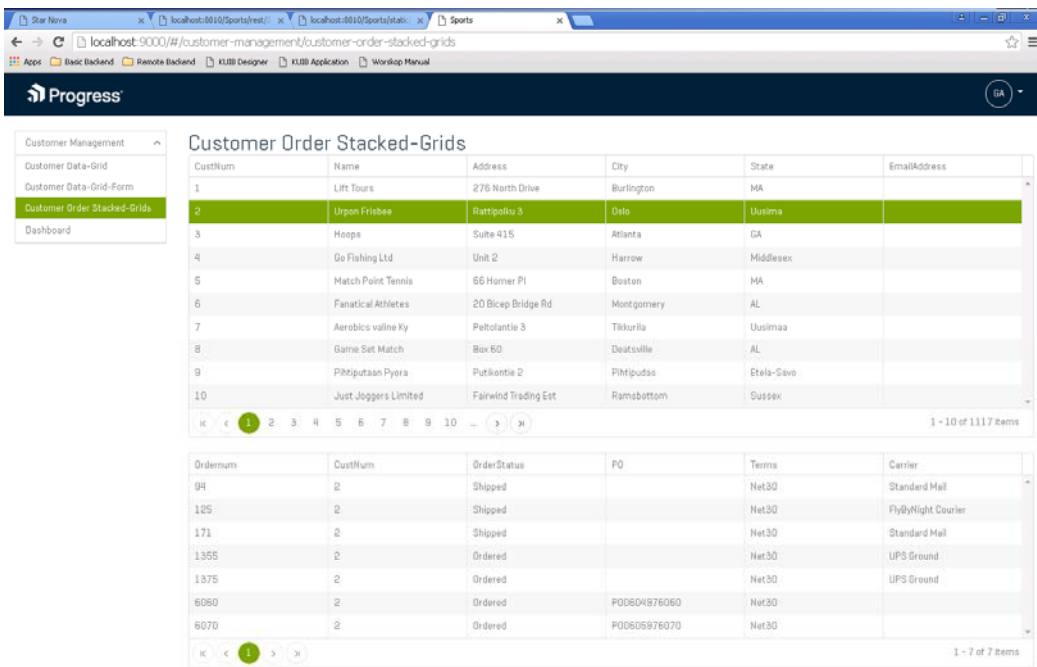
- Grid Name: ct18
- Grid Title: Customer O
- Data binding:
  - Data Provider: Sports..
  - Master Data Source: Custom...
  - Details Data Source: Order
  - Master Primary Key: CustNu..
  - Details Foreign Key: CustNu..

46. Press **Save** and **Build**.

47. Switch to the **Sports** application tab and **refresh** the page. Select the **Customer Order Stacked-Grids** view from the application menu:



Notice that there are no detail records (Orders) displayed initially. Select a Customer row to see the associated Order records in the Details grid:



This completes the workshop. Thank you for attending!