Appendix A

Progress Modernization Framework for OpenEdge

Common Operations

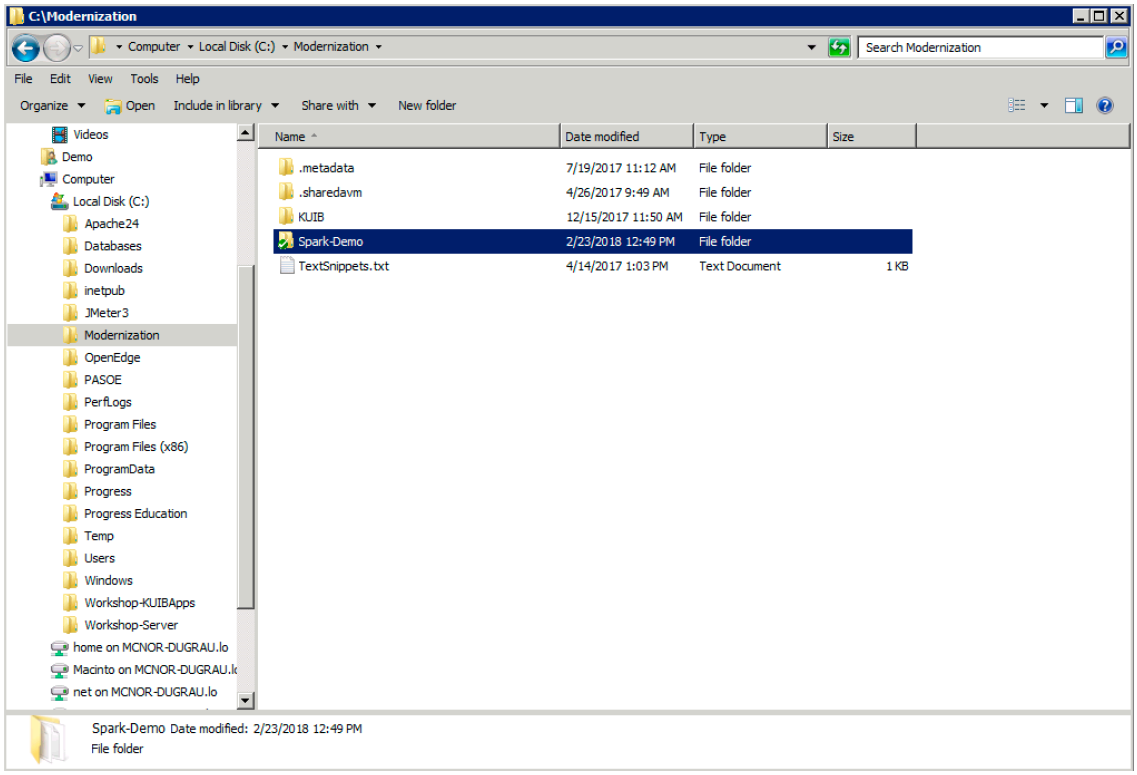
Purpose

This addendum serves to collect certain common operations in a single area, to avoid duplication throughout the material. It will be referenced from multiple sections of the accompanying chapters. All actions documented in this section cover the initial download of PMFO and configuration of the DynSports demo which is utilized for training purposes. If you wish to setup the demo application on your own computer, then this document will provide the full list of steps necessary to do so.

Obtaining Source Code

This operation assumes that you have installed an **Git** client, specifically **TortoiseGit**, to allow checkout of code from a GitHub repository. If using the VM provided for training, this should already be available.

1. Create a new directory to use as your PDSOE workspace, if not already available. The current practice has been to use **C:/Modernization** for consistency across various installations.
2. From the **C:/Modernization** directory right-click on an empty area within the directory and select the option "*Git Clone*". Enter the repository URL below:
   a. **https://github.com/progress/Spark-Demo**
   b. Be sure to set the destination directory to **C:/Modernization/Spark-Demo** and press **OK**.
3. You should now have a "*spark*" directory containing "*oe116*", "*oe117*", and "*support*".



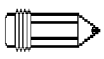
NOTE

If the code has already been checked out, it is a good idea to make sure you are up to date with any recent changes before you begin working. This is easy with **TortoiseGit** and a copy of the repository. Just **right-click** on the new "*spark*" directory from within the Windows Explorer. Select the "*Git Sync*" option from the context menu, click on the "Pull" button, and any obtained modifications will be reflected in your directory.

Creating a Database

For the initial training environment, there should already be databases created and configured for use. However, if you need to stand up a database for use, you may follow these procedures which assume that you have a structure (.st) file, a definitions (.df) file, and one or more data (.d) files. For our demo application, we should use the **WebState** and **Sports2000** databases.

1. Create a directory at **C:/Databases** if not already present.
    a. In this directory, create a suitable subdirectory for your database.
    b. Copy your relevant files (.st, .df, and .d) files to this location.
2. **Shift + Right-click** on the directory and select the "*Open command window here*" option.
    a. Run the command "`prostrct create <dbname>`", where <dbname> is the name of your structure (.st) file.
    b. Run the command "`procopy %DLC%/empty4 <dbname>`", where <dbname> is the name of your structure (.st) file.
    c. You may close the command prompt window.
3. Open the **Progress Data Administration** tool via the Start menu.
4. Use the option **Database → Connect** to select your <dbname>.db file, and click **OK**.
5. Go to **Admin → Load Data and Definitions → Data Definitions (.df file)…**
    a. Select your database's .df file, if not already set correctly by default.
    b. Click on **OK** to begin loading the definitions.
6. If you have a file "_sec-authentication-domain.d" then continue with the next steps.
    a. Go to **Admin → Load Data and Definitions → Security Domains**
    b. Select your database directory, and click OK.
7. Go to **Admin → Load Data and Definitions → Table Contents (.d file)…**
    a. Click on the **Select Some…** button and then click OK on the resulting screen.
    b. Click on **OK** to proceed, and select the directory where your .d files reside.
    c. Click on **OK** to continue loading the table contents.
8. If no issues were encountered, you may close the **Data Administrator**.
9. Log into the **OpenEdge Management** portal at http://localhost:9090
10. Create a new database via **Resources** tab at the top of the screen.
    a. Name your new databases as desired (or based on the .st file).
    b. Enter the full path to your database's .db file.
    c. Choose a port number that is not in use (eg. WebState: 8500, Sports2000: 8600).
    d. Set the database to **auto-start, enable AIW, and use 2 APW's**.
11. Save the new configuration, and start the database.
12. Repeat steps 2-11 above for each database (WebState and Sports2000).

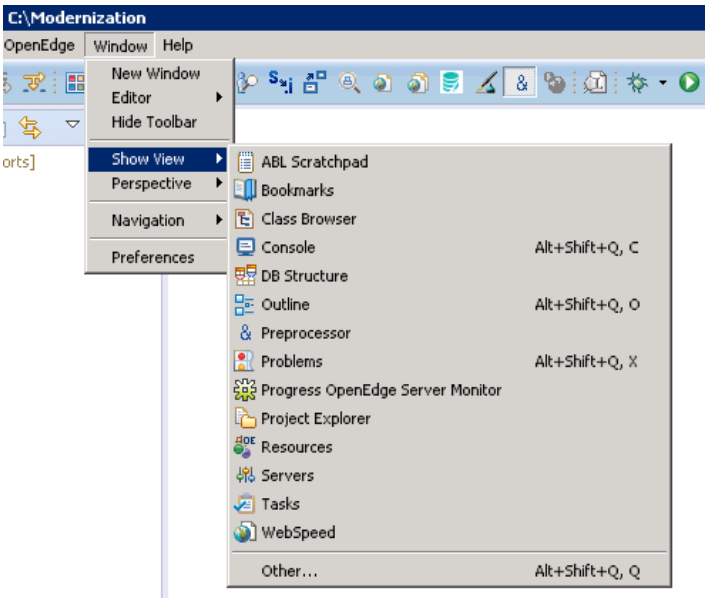


NOTE

You should add this database to your PDSOE workspace if you have not done so already. Since this was networked, use the server of "localhost" and the port as you chose in the steps above.

Preparing the PDSOE Workspace

1. Start the **Developer Studio** from the **Start Menu → All Programs → Progress → OpenEdge 11.7 (64-bit)** or directly from the Start Menu if available. (If prompted with a Windows Security Alert for Java, just click "Allow Access" to continue.)
2. If prompted for a workspace, select the **C:/Modernization** directory from earlier (be sure to select the option to use this as your default). If not prompted, you can still change your workspace via **File → Switch Workspace → Other** and select the appropriate path.
   a. If given a splash page on first use, select the "*Workbench*" option.
3. Select **Window → Show View** and look for an option called "Servers".



4. If the option is not available, then from this dialog select **"Other…**" menu filter results by typing "*Servers*" to view results as shown below.

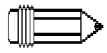
   a. Select the result under **Server → Servers** and click **OK**.

b. A new view should appear at the bottom of the PDSOE workbench, displaying all of the available servers as found on the Progress Admin Service.

c. At this time, you may remove all unnecessary servers from the view—we will add new servers later. Simply right-click on the other server options and select **Delete** to remove an item (this does not remove the server from the OpenEdge Management Console). Note: You do not need to stop the server before deleting.

Some options in PDSOE are considered helpful or provide consistency across projects, though they are not always enabled by default. To manage these settings, go to **Window → Preferences** and set the following options as indicated:

- General → Show heap status
- General → Editors → Text Editors → Insert spaces for tabs [enable]
- General → Editors → Text Editors → Displayed tab width: 4
- General → Editors → Text Editors → Show line numbers [enable]
- General → Startup and Shutdown → Refresh workspace on startup [enable]
- General → Workspace → Refresh using native hooks or polling
- General → Workspace → Refresh on access
- Progress OpenEdge → Advanced → Class Cache → Limit scope information to PROPATH
- Progress OpenEdge → Editor → Case: Lower [enable]
- Progress OpenEdge → Editor → Expand keywords [enable]
- Progress OpenEdge → Editor → Case keywords [enable]
- Progress OpenEdge → Server → Update properties from server before starting/launching [enable]



NOTE

If the option above for "Update properties from server" is not selected, PDS may attempt to overwrite your PAS instance's **openedge.properties** file with its last-known-good configuration, which may erase any changes you have made directly. This becomes apparent after later steps in this material which use the **oeprop** command to merge changes into your new **SportsPASOE** instance.

In the **Progress OpenEdge → Database Connections** section, add connections to your **WebState** and **Sports2000** databases. You may just use "localhost" as the server, and recall that these should have been set up with ports 8500 and 8600, respectively.

Preparing a PASOE Instance

Each instance of PASOE must have a place to live on your filesystem. The default shipping server is "oepas1" and lives under your %DLC%/WRK directory, which is fine for demonstrations but not so great for longevity. It is advised to create a more distinct home for your instances, especially to prevent any accidental erasure during upgrades or moving systems. For our purposes, this will be **C:/PASOE**

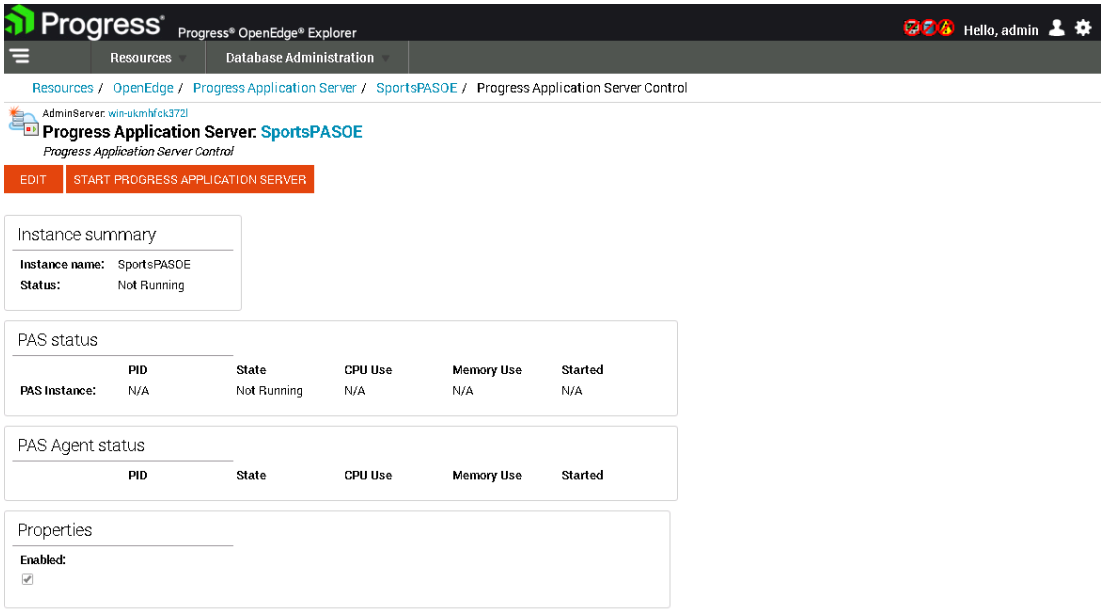
Please note, these instructions are only necessary if you are setting up a new environment and wish to use the ready-made example applications!

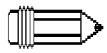
There are 2 easy ways to create a new instance: via the OpenEdge Management Console or via the command line. Using the OEM requires a bit more knowledge of pathing and internals of a PAS instance, so for beginners it is just as powerful and easy to use the command line. The training VM should have the correct paths configured so that you can do this from any shell in Windows. The following can be issued at a Windows command prompt to create the instance used by default:

```
pasman create -v -f -p 8820 -P 8821 -j 8822 -s 8823 C:/PASOE/SportsPASOE
```

The "**pasman**" command is similar to "asbman", "wtbman", and other utilities for managing Progress services. The –v option says to be verbose, and –f says to copy all files from the CATALINA_HOME directory (from DLC) into your instance. This includes the OpenEdge Manager used by PDSOE and the Tomcat Manager used to manage your webapps. The default PAS instance (oepas1) uses the port 8810, so we need to use another, unused port for HTTP connections. In our case we've selected 8820 for that, 8821 for HTTPS connections, 8822 for AJP connections, and 8823 for the "shutdown" port (used by Tomcat). The final entry is the path to your PAS instance. Note that we're just placing this directly under our PASOE directory, as we don't want to wade too deep in the filesystem to get to our instance. And notably the name "**SportsPASOE**" will be used from here on out to refer to our PAS instance by name. This will be both for our benefit and part of the normal behavior of OpenEdge.

With the instance created, you should now be able to view the server under the **OpenEdge Management Console** (aka. **The OpenEdge Explorer**). From here you can manage your instance, such as start/stop the server and modify any configuration options.



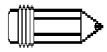


NOTE

To confirm from the command line that your environment was created, run the following:

```
pasman env -I SportsPASOE
```

```
C:\Windows\system32>pasman env -I SportsPASOE
catalina home:      C:\Progress\OpenEdge\servers\pasoe
catalina base:      C:\PASOE\SportsPASOE
catalina tmpdir     C:\PASOE\SportsPASOE\temp
catalina pid:       C:\PASOE\SportsPASOE\logs\catalina-SportsPASOE.pid
java home:          C:\Progress\OpenEdge\jdk
jre home:
manager http port: 8820
manager https port:8821
manager shut port: 8823
manager URL:        http://localhost:8820/manager
config type:        instance
config alias:       SportsPASOE
config parent:      C:/Progress/OpenEdge/servers/pasoe
server running:     0
instance tracking: True
instance file:      C:\Progress\OpenEdge\servers\pasoe\conf\instances.windows
server process-id: 0
window title:       n/a
security model:     developer
service:            false
```



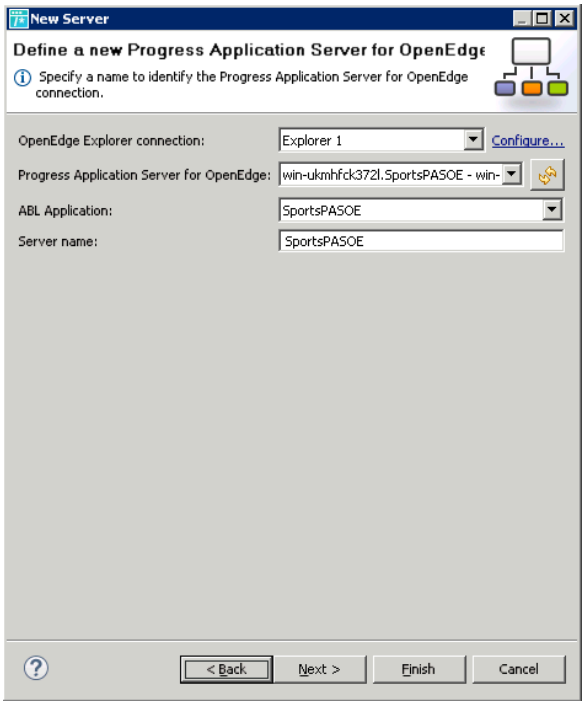
NOTE

To enable AJP proxying to Tomcat for your instance, run the following:

```
pasman feature AJP13=on -I SportsPASOE
```

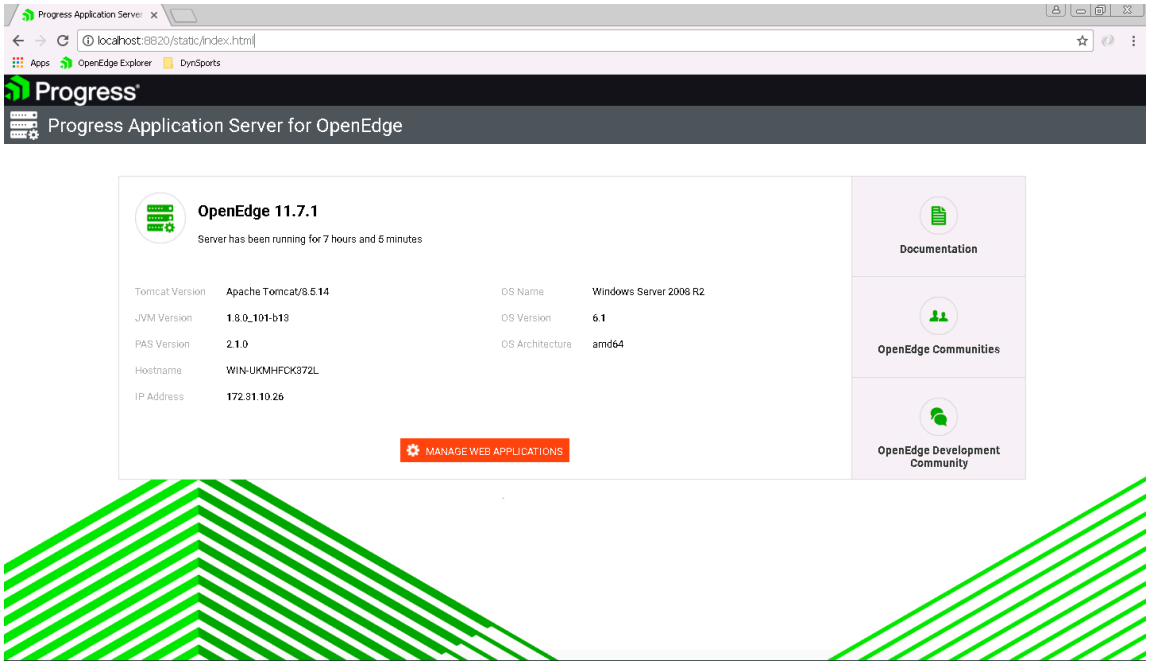
With an instance created, we should add the server to our PDSOE environment, and then we'll be able to add projects and services to it. This will be done via the Servers panel which we enabled previously.

1. Right-click within the Servers panel, and select **New → Server**
2. On the new modal, select the option **Progress Application Server for OpenEdge**
3. Click the **Next** button and select your OpenEdge Explorer instance.
4. Find your PAS instance "**SportsPASOE**" in the next dropdown field and select it.
5. To make life easier later, shorten your server name to just "*SportsPASOE*"
6. At this point just click **Finish** to add the server to our view.



You should now see your server in the panel, and we can begin configuring some options which will be utilized for the AppServer portion of operation.

1. First double-click on the new "*SportsPASOE*" server.
2. Expand the "**Publishing**" tab at the top right, and select "**Never publish automatically**".
3. Press Ctrl+S (or use **File → Save**) to save these changes.
4. Start the server via **Right-click** and selecting "*Start*".
5. Wait for the progress bar at the bottom left to complete, and for the server to state the status of "**Started, Synchronized**".
6. At this point you should be able to visit the server at http://localhost:8820/ and view the default server landing page. If this does not work as expected, then please refer to the section labelled "**Troubleshooting Checklist**" found later in this chapter.



Project Setup

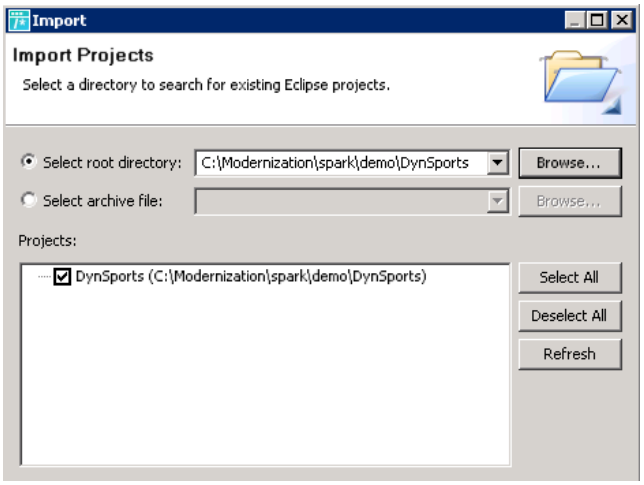
These are the setup instructions for the built-in applications that come with the current release of the PMFO. **Therefore, these instructions are only necessary if you are setting up a new environment and wish to use the ready-made example application.**

NOTE

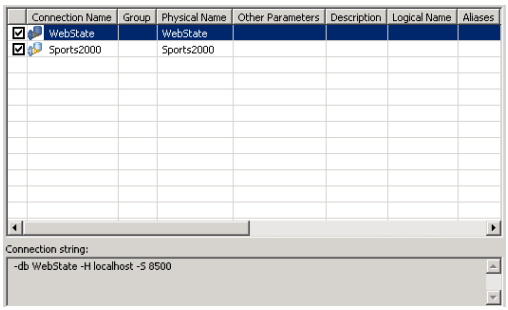
Before proceeding, please see the sections **"Obtaining Source Code"** and **"Preparing a PDSOE Workspace"** earlier in this appendix. These steps will ensure a consistent working directory for your projects and ready your development space in the Progress Developer Studio for OpenEdge.

Importing the DynSports Project

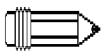
1. From the **File → Import** menu, select **General → Existing Projects into Workspace**, **Next**.
   a. For the selected root directory, browse to **C:/Modernization/Spark-Demo**
   b. You should be given the option to select multiple projects. For our immediate purposes, we only need to check the box for the **DynSports** project. There is no need to select any other checkboxes on this screen, so just select **Finish**.



2. Before continuing, it's a good idea to make sure your database connections are available. For training purposes, we will use the 2 databases configured previously: **WebState** at port **8500** and **Sports2000** database at port **8600**.
   a. From within PDSOE, **right-click** on your project folder and select **Properties**.
   b. Under the **Progress OpenEdge** option select **Database Connections**.
   c. Check both databases if they are available as shown below.



3. If no such entry exists for WebState or Sports2000, click the link at the top-right for "*Configure database connections*", then click New to create a new connection. Using the Sports2000 database as an example, perform the following steps:
    a. The connection name and physical name should be "*Sports2000*"
    b. For the Hostname use "*localhost*" and for Service/Port use 8600
    c. Click on **Test Connection**, and if successful click the **Next** button.
    d. Check the box to define a new SQL connection, then click **Next**.
    e. Use the default options and test the connection. If successful click **Next**.
    f. Make sure the auto-start option is **unchecked**, as we will start the database automatically via the Progress Admin Service. Click **Finish** to create the entry.
    g. You should now see the new connection available in the list. Click **OK** to continue.
    h. Repeat the above for "*WebState*" if necessary, using port 8500.
4. From the list of available connections, you should see Sports2000. If no connections are shown, select the "*Show All*" radio button to view all available connections. Once the connections are visible, check the boxes and click **OK**.

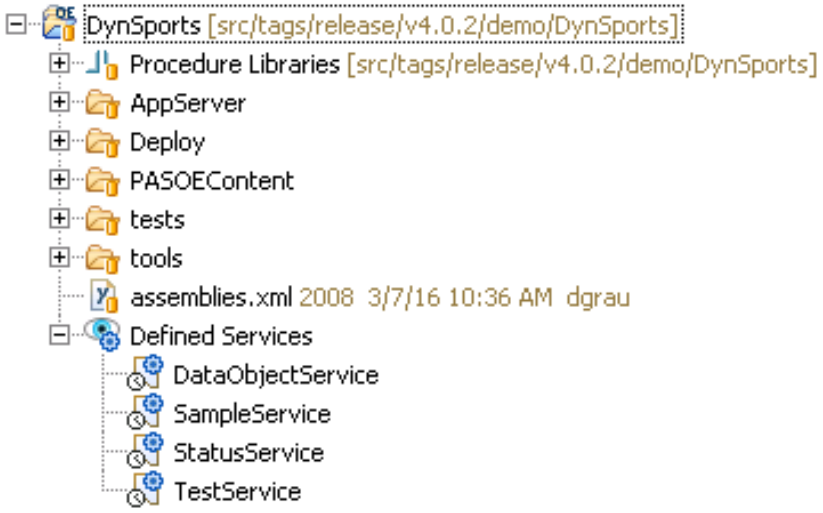


NOTE

After making changes to any database connections, it may be necessary to restart the AVM.

1. Right-click on the project and select **Progress OpenEdge → Restart OpenEdge AVM**.
2. If any error icons appear over any folders, again right-click and this time select **Progress OpenEdge → Clear OpenEdge Compile Errors**.
3. To compile your code, either **right-click** on the AppServer directory and select **Progress OpenEdge → Compile** or from the **Project → Clean**, then select "*Clean all projects*" and press **OK**. The latter may take more time, as it will evaluate the entire project to find anything that needs to be cleaned/compiled.

After import, the project should contain the following directories and services.



Publishing to PASOE

While it is useful to deploy a WAR file to a PASOE instance for production, such a process is more cumbersome for development purposes. Therefore, the easiest way is to publish to a PAS instance via PDSOE. This is due to the need to generate some metadata and artifacts for the server which can only be done from the IDE. In our case, we already imported an existing project (DynSports) into PDSOE which will contain everything we need for our running application, but we just need to publish it to an instance.

From the filesystem, it is easy to determine if PDSOE has published to the PASOE instance:

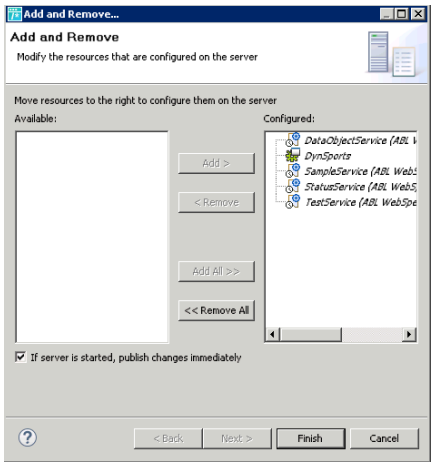
1. First navigate to **C:/PASOE/SportsPASOE/webapps** and note if "**DynSports**" exists.
2. If the folder **does not** exist, then no code has been published yet from PDSOE.

From PDSOE you can check on, or add, a project to a PAS instance. For our PMFO demo application we will perform the following actions to illustrate a simplified means of configuring a project manually.

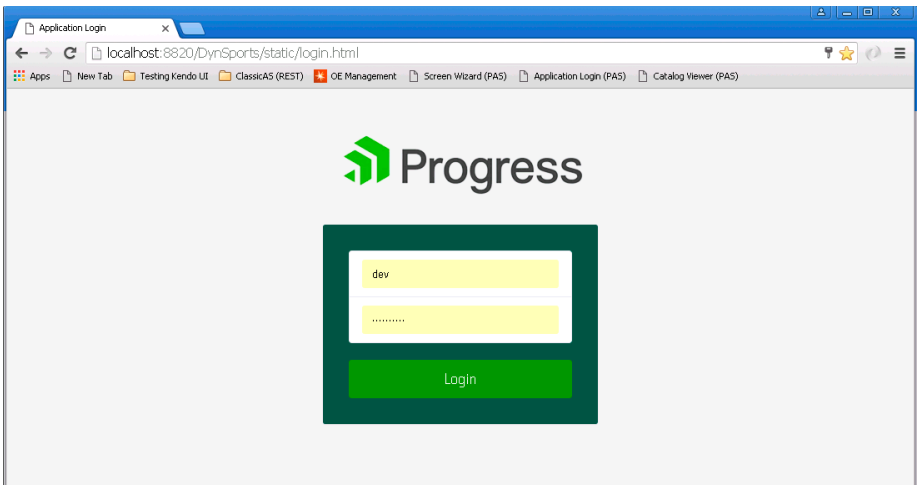
1. From the DynSports project, open the "**AppServer**" folder and copy the file "**Ccs.pl**" from "**AppServer**" to **C:/PASOE/SportsPASOE/openedge**
2. Open the file "**startup.pf**" (Right-click, **Open With → Text Editor**) to confirm the connection parameters for the WebState and Sports2000 databases are correct. Copy this file to **C:/PASOE/SportsPASOE/openedge**
3. Note the file "**AppServer/merge.openedge.properties**" which contains options to set the PROPATH and startup options for our PAS instance. We will merge these options next.
    a. Open a Windows command prompt, and navigate to "**C:/PASOE/SportsPASOE/bin**"
    b. Run the following command:

```
oeprop -f C:/Modernization/Spark-Demo/oe117/DynSports/AppServer/merge.openedge.properties
```

    c. You can confirm the changes made by the **oeprop** command by opening "**C:/PASOE/SportsPASOE/conf/openedge.properties**" and searching for "*startup.pf*", which should now be present in the file.
4. Navigate to the "**Deploy/Conf/**" directory of the **DynSports** project and copy all **.json** and **.cp** files within the directory to a new folder **C:/PASOE/SportsPASOE/conf/DynSports**.
5. Copy "**Deploy/Conf/Realm/SparkRealm.cp**" to **C:/PASOE/SportsPASOE/common/lib**
6. Copy "**Deploy/Conf/Realm/SparkRealm.json**" to **C:/PASOE/SportsPASOE/conf/spark**
7. From the Servers view, Right-click on **SportsPASOE** and click the option "*Add and Remove*".
    a. Select the project and all services on the left and click the **Add** button to add it to the PASOE instance. Click **Finish**.
    b. Publishing of the project files should begin immediately, which will copy files and restart the PAS instance automatically.



8. Once restarted, if we return to the **C:/PASOE/SportsPASOE/webapps** directory, we should now find a "*DynSports*" folder that contains our current project. This is where our static content and AppServer code lives within the instance.
9. WebApps are deployed in the format of **http://<server>:<port>/<webapp>** and you should be able to visit this application at http://localhost:8820/DynSports to view the page below.
    a. Login with the username "**dev**" and password "**bravepoint**".



NOTE

After a successful deployment of a new project to a PAS instance, it is a recommended practice to make a backup of the "*openedge.properties*" file in the **CATALINA_BASE/conf** directory.

NOTE

The log files for this particular PASOE instance will be in the following location:
C:/PASOE/SportsPASOE/logs

NOTE

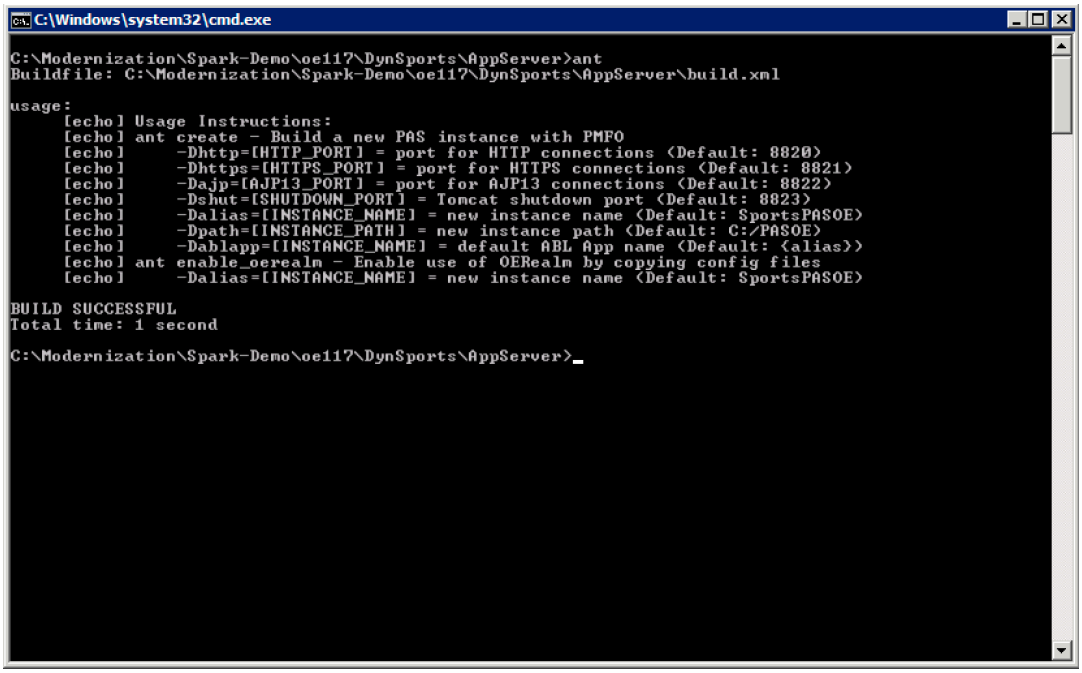
As an alternative to many of the steps above, an ANT script has been created which can create and configure the initial PAS instance via a single command. To utilize this tooling, open a Windows Command Prompt (as Administrator) to the following where a build.xml file should reside:

C:/Modernization/Spark-Demo/oe117/DynSports/AppServer

Running just the command "*ant*" should provide usage instructions to using this script. By default, all ports, paths, and names should match what was just covered in the sections previous. Running the command "*ant create*" without further changes will use all defaults as listed to create the PAS instance as follows:

- Execute "*pasman*" with the noted properties to create the instance.
- Copy .pl and .pf files from *AppServer* to CATALINA_BASE/openedge
- Copy all *OERealm* configuration files to their respective paths.
- Merge configuration options to openedge.properties via "*oeprop*".

This should prepare the new instance with the correct databases and startup options, and avoid the need to alter any properties via the launch configuration screen in PDSOE.



Troubleshooting Checklist

Questions to ask when attempting to troubleshoot your REST adapter:

- Can I ping my project server (assuming use of a hosts file entry)?
- Are all of the services started (PAS instance, Databases, etc.)?
- Can you access your database via the Data Dictionary using server/port parameters?
- Is your instance even running? **C:/PASOE/<instance>/logs/catalina-<project>.pid**
- Is the server responding with any valid page when you visit **http://localhost:8820/** ?
- Any response from the manager page at **http://localhost:8820/manager/html** ?
- Can you reach your webapp at **http://localhost:8820/<project_name>/** ?
- Can you reach the webapp's **/static/auth/login.html** page?
- Can you authenticate via this login.html page (using dev/bravepoint)?
- Did you attempt to access the Catalog Viewer (/static/catview.html)?
- Can you run any of the API's listed by the Catalog Viewer?
- What does the agent log file say: **C:/PASOE/<instance>/logs/<project>.agent.log** ?
- What does the broker log file say: **C:/PASOE/<instance>/logs/<project>.<date>.log** ?
- When did you last reboot? ☺

Questions to ask when attempting to troubleshoot your web application:

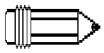
- What responses are present when viewing the AJAX requests from the web browser's network panel (Developer Tools for Chrome, Firebug for Firefox, etc.)?
- Were there any HTTP/404 or HTTP/500 errors encountered?
- For the SPA templates, did you see both an HTML and JS file be delivered?

Advanced Topics

In order to make full use of the PAS environment we need to make sure that users are properly authenticated. The projects imported so far will already have a security scheme in place and so these steps have already been performed. But for sake of understanding and documenting the process we will outline how to change your security model and to provide the proper security against your database.

Security Configuration

In order to illustrate the differences between public and private REST resources, our projects alter the default security model of "**Anonymous**" to at least "**Form-Local**". This model requires the use of the "**users.properties**" file within our REST project, to provide a set of username/password pairs along with valid roles for each configured user account. The "**Form**" portion indicates that credentials will be sent via POST request to a specific URL for authentication. Additionally, upgrading from "**Local**" to "**OERealm**" allows for use of an ABL file to programmatically confirm user credentials. So, for the imported project this will already be pre-configured via the **WEB-INF/oeablSecurity.[csv|properties]** files, specifying the options necessary to illustrate usage of an ABL class to perform authentication of user credentials. This is covered in more depth as part of Chapters 2 and 4 of the related material.



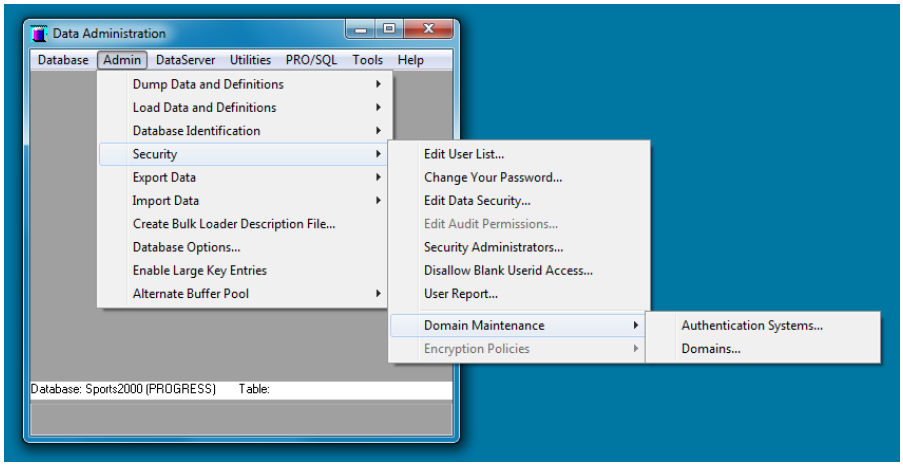
NOTE

Traditionally the REST service was designed to handle either all-public request or all-secured requests—not both in the same instance. However, in real-world situations it is more likely to have security at a per-URI level in a single service, securing all URI endpoints with some basic role requirement and opening only the necessary public URI's where needed. For our purposes, we have enabled the use of anonymous access along with providing a security model to achieve this goal. In Chapters 2 and 4 it will be explained how this granular control can be implemented.

Domain Configuration

Recently in OpenEdge releases, the idea of creating a domain primarily implied the use of a multi-tenant database. However, for the REST adapter and the Spring Security layer in Tomcat the use of a domain allows securing the connection between an AppServer and Tomcat (for OERealm security) or in our case verifying the identity of a user after authentication. The process is simple, and can even be performed on a running database.

1. Open the Data Administration tool from **Start Menu → All Programs → Progress → OpenEdge 11.7 (64-bit)** or directly from the Start Menu if available.
2. From **Database → Connect** browse to **C:/Databases/Sports2000/Sports2000.db**
3. Click on **Options** and check the **Multiple Users** checkbox then **OK**.
4. Go to **Admin → Security → Domain Maintenance → Domains…**



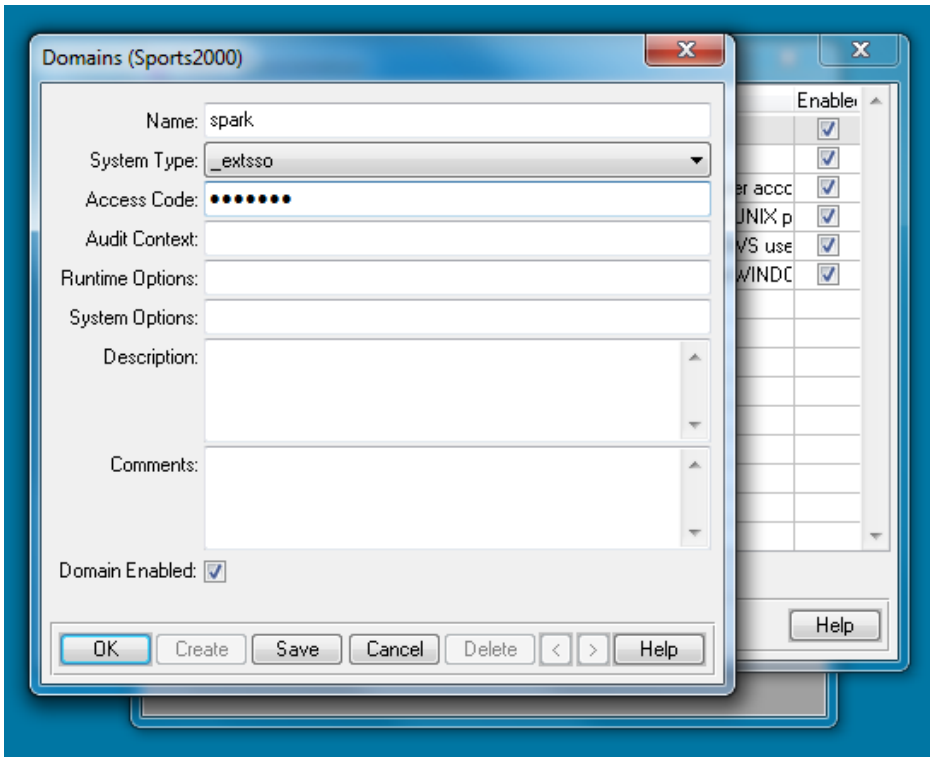
5. Click on **Create**, and provide the following options:

   a. Name: spark

   b. System Type: _extsso

   c. Access Code: spark01

   d. NOTE: For real-world cases you should use a stronger access code, and your own choice of domain! Because these domains have nothing to do with multitenancy, you are free to have as few or as many as you wish. If you intend to have a Single Sign-On for multiple applications then you may wish to use only 1, or you could configure 1 domain per REST service—this is up to you as the designer.

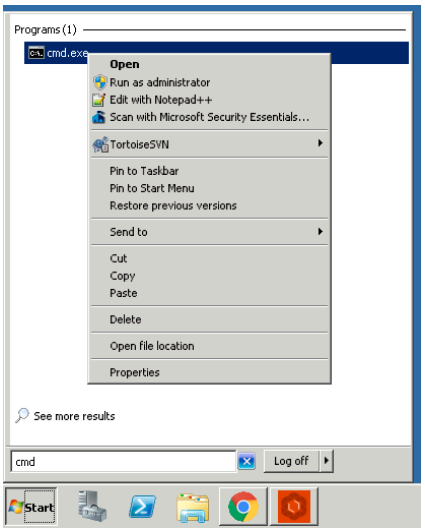


6. Click on **OK** to complete creation and confirm on the Domains screen, then click **Done**.

Launching the Wizard

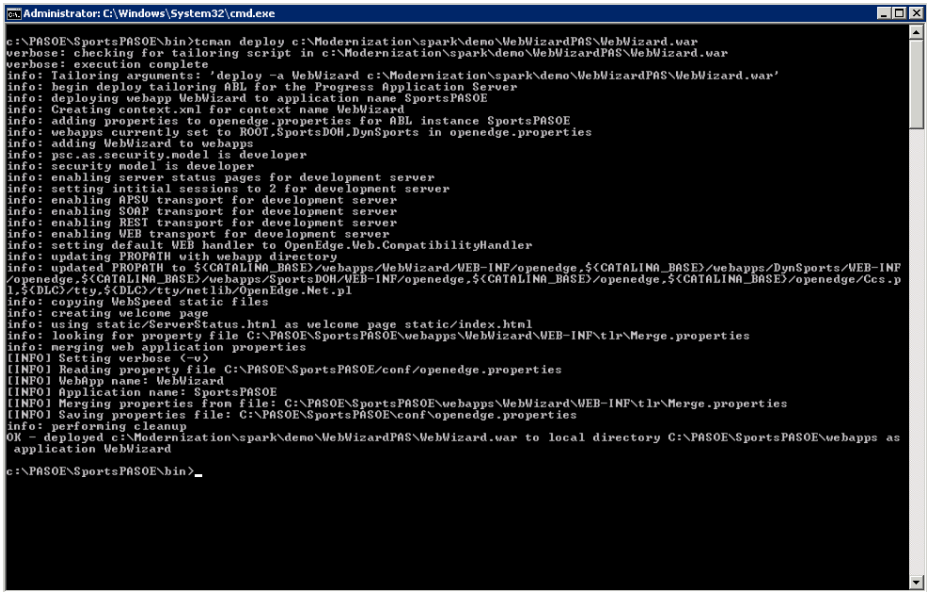
As of the 3.1.0 release, you can officially deploy the **WebWizard** via a simple WAR file, by utilizing the "*tcman*" command in your PAS instance. This will allow you to deploy the WebApp to any available PAS instance.

1. From the **Windows Start Menu**, search for the text "cmd" to list the cmd.exe program.
2. **Right-click** on the item to bring up the context menu, and select "*Run as administrator*", entering the password "**Pr0gress2016**" when prompted.



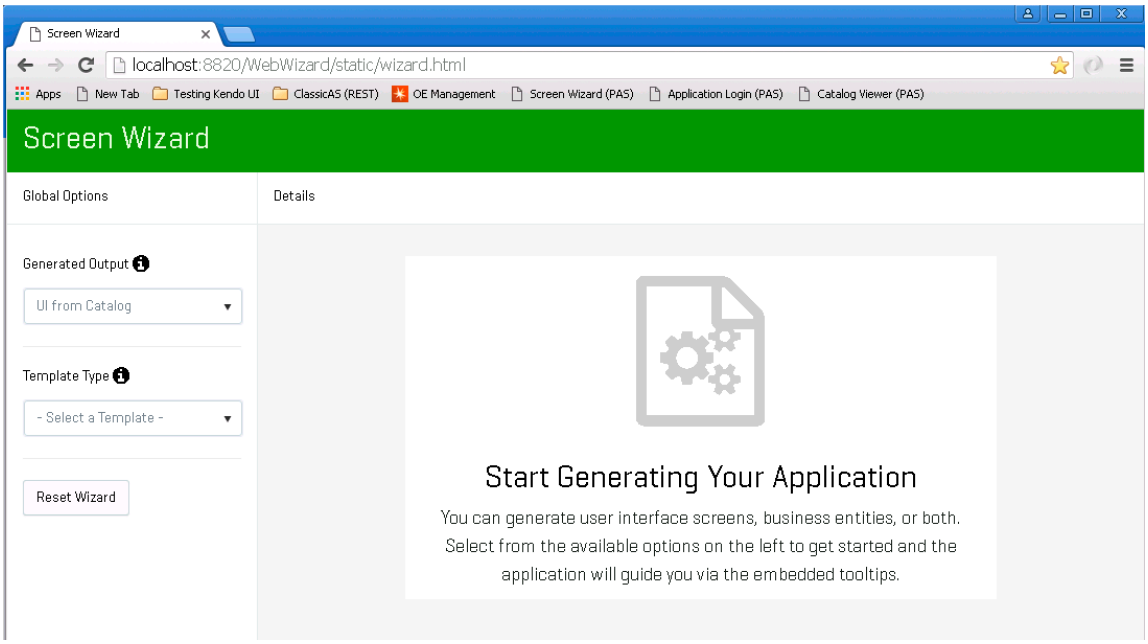
3. Change directory (via "**cd**" command) to **C:/PASOE/SportsPASOE/bin**
4. Run the command "`tcman stop`" to ensure the PAS instance is not currently running.
5. Run the following command to deploy the WAR file to this PAS instance:

```
tcman deploy C:/Modernization/Spark-Demo/oe117/WebWizardPAS/WebWizard.war
```



6. With the WAR file successfully deployed and tailored to this instance, start the PAS instance by any valid means (command line, PDSOE servers view, etc.).

It should now be possible to attempt to reach the application. Your PAS instance should be started, which means the Tomcat web server and built-in AppServer are responding. That still leaves a lot of room for error, so next we'll see what kind of issues may be encountered. To start, you can attempt to reach the site at **http://localhost:8820/WebWizard/static/wizard.html**



If you reached a page with a header that reads "Screen Wizard" then you're on the right track. If you could not reach that page (ie. you got a 404 error), then several problems may be present:

- The project and related services did not get deployed as expected. Make sure that you have both a **DynSports** and **WebWizard** folder in your PAS instance's webapps folder.
- If you get the screen above but see a loading bar at 0% at the top, that means the initial API's that deliver data are not responding. Open your browser's debug tools to check the requests being sent. It could be anything from a 404 (can't find the code), a 401/403 (can't authenticate), or just a 500 that indicates some other error.
- For any other issues, you may want to consult the "**Troubleshooting Checklist**" found earlier for some helpful tips on what to be on the lookout for when things go sideways.

If any additional corrections have been made, then at this point we can return to our browser and refresh the page (wizard.html)—we should now see the remainder of the page, along with requests for additional data from the REST service that was deployed.

NOTES: