Sample - AutoEdgeTheFactory Mobile App

Contents

Intro

This document describes the installation and setup steps for the mobile sample app based on **AutoEdge|TheFactory**. This document contains the goals of developing this sample, as well as some takeaways of the development process.

The attached ZIP file contains all the artifacts needed for this sample mobile app, including backend services and the mobile UI.

Goals

- Reasonable “real-life” app
- Device interaction (true mobile, not ‘just’ web)
- Reduce amount of manual JavaScript written
    - Some JS unavoidable
- Test process, feedback into engineering
- Provide engineering with sample

App detail

- Fits into AutoEdge story
    - Alternative to BPM BizSolo flow
- Vehicle ordering app
    - Brand data used to select car, options
    - Dealer data for physical car pickup etc
    - Shopping cart contains multiple (potential) orders
- Design Wireframe UI
    - App actually reasonably close to design

Wireframe UI design

Click on the thumbnail for a more detailed view

Installation

> **Installation via Progress Developer Studio Samples**
> This sample should soon be available from within Progress Developer Studio via the Samples page. For details see AETF Mobile Sample - Providing samples in PDSOE.

Start Progress Developer Studio
File -> Import -> General -> Existing Projects into workspace
Select the AutoEdgeTheFactoryMobileSample.zip file and import it into a workspace.
Alternatively, unzip the file into a location, and import it using a folder location. Make sure in this case to select 'Copy into workspace'.

The project's workspace folder is now effectively `$INSTALL` (so something like `c:/workspaces/mobile/autoedge`).

Setup

`$INSTALL` is the path where the sample has been installed/unzipped. You can find this by right-clicking on the project's name. The full path is in the Resource tab, in the Location field.

In all cases below, manually substitute `$DLC` and `$INSTALL` for the appropriate paths. `$DLC` being where your OE 11.2 install is located: `C:\Progress\OpenEdge` is the default.

Database

Create the database from the backup

The backup is located in `$INSTALL/`db directory, named aetf.bup
In ProEnv, run the following

proenv

```
prorest aetf aetf.bup
```

Start a server for the DB.

You can start the db with -S/-H if you insist, but there's no need for it

proenv

```
proserve aetf
```

Add DB to PDSOE

- Once the DB server has started, add the database to your workspace in PDSOE, via File > Window > Preferences > Progress OpenEdge > Database Connections. Name it **AETF-Mobile** and point the DB location to `$INSTALL\db\aetf.db`

If you call it **AETF-Mobile** the project will automatically use it; if your name is something different, then you will have to associate the DB with the project manually.

AppServer

Create the AppServer

There are 2 alternatives to creating the necessary AppServer

- Manually create an AppServer called **`mobile_asAutoEdgeTheFactory`**. The parameters listed below are the only important ones.

ubroker.properties

```
Operating Mode:  State-free
Server startup parameters:  -pf ./cfg/server.pf
PROPATH:   ./src/Appserver;./src/Common;$DLC
Startup procedure: as_startup.p
Working directory: $INSTALL
```

- Use the provided `$INSTALL/cfg/aetf.ubroker.properties` file.

- Open aetf.ubroker.properties and replace $INSTALL and $DLC with your values.
- In ProEnv, run the following

proenv

```
mergeprop -action update -type ubroker -delta $INSTALL/cfg/aetf.ubroker.properties
```

making sure to replace `$INSTALL` with the appropriate value. Note that sometimes mergeprop has trouble with spaces in paths, so you may have to enclose the delta argument in quotes , eg `-delta "$INSTALL/cfg/aetf.ubroker.properties"`

Add the AppServer to PDSOE

In the server view, right-click and select New > Server. Select OpenEdge AppServer; after pressing Next to get to the 2nd page, the newly-added **`mobile_asAutoEdgeTheFactory`** appserver should appear. Select this. There is no need to add the current project to this AppServer because the AppServer's `$PROPATH` points directly at the project location, which means there's no deployment necessary.

Add the OE Web Server to PDSOE

In the server view, right-click and select New > Server. Select OE Web Server; after pressing Next to get to the 2nd page, select the default **`restmgr1`**. Click Next and add the **VehicleOrderService** and **SecurityTokenService** to the OE Web Server.

Mobile web app

- Enter your credentials for the Mobile App Builder. Window > Preferences > Progress OpenEdge > Mobile App Builder.
- Add a new Mobile App. File > New > Other (or CTRL-N). Filter on 'mobile' or select Progress OpenEdge > Mobile > Mobile App

- Select the project, and give the app a name. We use **VehicleOrderApp**.
- Select the "From backup" option, and select the included ZIP file, aetf_vehicleorderapp.zip (in `$INSTALL`).

- This takes you to a web browser and you can now edit your application.
- You need to edit the host name for the various services that provide data to the **VehicleOrderApp**.

- In the left-hand tree, expand the Services node. You should see 5 services: 3 **VehicleOrderService.***, 1 **SecurityTokenService.SecurityTokenService** and a **GeolocationService**. We ignore the latter service for the host name changes.

- For each of the 3 VehicleOrderService services (BrandDataService, DealerService and ShoppingCartService), expand the service node and select the VehicleOrderService_*_Settings node (eg VehicleOrderService_BrandDataService_Settings).

- Change the value of the **catalogURL** property to `http://$HOSTNAME:8980/VehicleOrderService/static/mobile/VehicleOrderService.json`, where `$HOSTNAME` is your machine name, in a form that's accessible from the network in which you wish to use the app.

- Change the value of the **RESTapplicationURL** property to `http://$HOSTNAME:8980/VehicleOrderService`

- For the SecurityTokenService, the **catalogURL** and **RESTapplicationURL** also need to change to `http://$HOSTNAME:8980/SecurityTokenService/static/mobile/SecurityTokenService.json` and `http://$HOSTNAME:8980/SecurityTokenService` resp.

- Make sure you Save your changes.

Misc

- Once the AppServer and DB are set up for the project, compile and build the project, via Project > Clean. Select the AETF Mobile Sample project and make sure that it gets built too.

- If you're on an Arcade instance - or otherwise strongly-firewalled machine - make sure the part 8980 is open.

Runtime

Deploy mobile services

- In Project Explorer, select & edit (by double-clicking) the SecurityTokenService from under the project's "Defined Services" node. Select Finish to cause the service to deploy to the OE WebServer (**`restmgr1`**).
- Repeat for the VehicleOrderService.

Important
There is a bug which causes the deployed services to be pointing at the wrong AppServer (**`restbroker1`** instead of **`mobile_asAutoEdgeTheFactory`**). You can verify this in the Progress OpenEdge Server Monitor view. Expand the nodes under OE Web Servers
until you see the services deployed above. Select each one, and check in the Properties tab for the AppService Name property. This
should be **`mobile_asAutoEdgeTheFactory`**. If it is not, there are 2 methods of fixing this.

1) Using ProEnv, type in the following commands.
For the VehicleOrderService:

proenv

```
restman -i restmgr1 -appname VehicleOrderService -disable
restman -i restmgr1 -appname VehicleOrderService -setprops -prop appServiceName -value
mobile_asAutoEdgeTheFactory
restman -i restmgr1 -appname VehicleOrderService -enable
```

For the SecurityTokenService:

proenv

```
restman -i restmgr1 -appname SecurityTokenService -disable
restman -i restmgr1 -appname SecurityTokenService -setprops -prop appServiceName -value
mobile_asAutoEdgeTheFactory
restman -i restmgr1 -appname SecurityTokenService -enable
```

2) Using OE Management/Explorer.
In the Resources view, select **<your-machine-name>.restmgr1** in the OE Web Servers group by clicking on the blue (link) text. In the
next screen, select the REST Applications like. Now, for each service, disable the service, change the property value and re-enable
the service.

Running the app

Security Model

AutoEdge|TheFactory uses an OpenEdge application table - called **`ApplicationUser`** - for authentication. This table contains user login names, domains and passwords, as well as some extra information. Login and Logout services are provided via the **`SecurityTokenService`**, which is accessed via a REST interface (as installed above).

The OE Web Server has multiple authentication models available. This mobile sample uses the **anonymous** method.

More detail is available at the AETF Mobile Sample Security page.

Application user credentials

The doc at `/AETF Mobile Sample/doc/applicationusers.txt` contains user names and credentials for the application. The AutoEdge|TheFactory security model requires a brand/tenant in all cases.