OpenEdge Mobile and Bits Involved

Anil Kumar
Senior Software Engineer
7th Oct 2013

Divyatheja
Senior Software Engineer

PROGRESS EXCHANGE 2013
DISCOVER. DEVELOP. DELIVER.

Focus of the Session

Steps involved in developing a Mobile Application

Demo – Mobile App Development

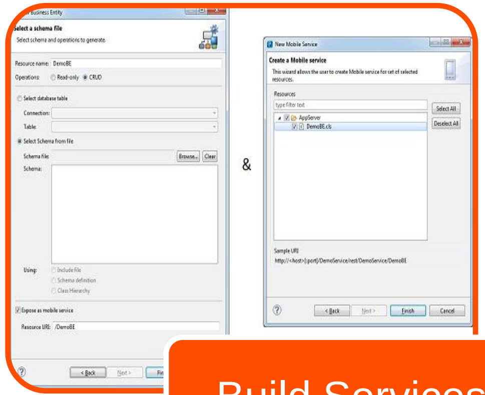
JSDO and Its Importance

Role of REST in OpenEdge Mobile

Architectural Elements

Mobile App and Steps Involved

Build Services → Create UI → Perform Mapping → Build and Deploy


Business Entity

Select a schema file

Select schema and operations to generate.

Resource name: DemoBE

Operations: ○ Read-only ● CRUD

○ Select database table

Connection:

Table:

● Select Schema from file

Schema file:

Schema:

Using: □ Include file ○ Schema definition ○ Class Hierarchy

☑ Expose as mobile service

Resource URI: /DemoBE

&

New Mobile Service

Create a Mobile service

This wizard allows the user to create Mobile service for set of selected resources.

Resources

type filter text

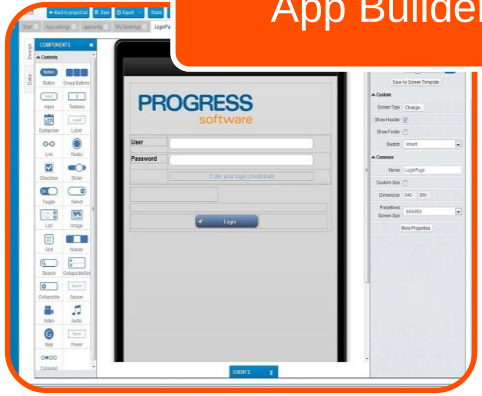
☑ AppServer

☑ DemoBE.cls

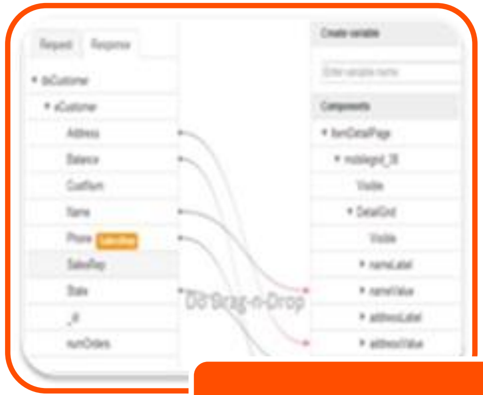
Sample URI

http://<host>[:port]/DemoService/rest/DemoService/DemoBE

Build Services in PDS OE

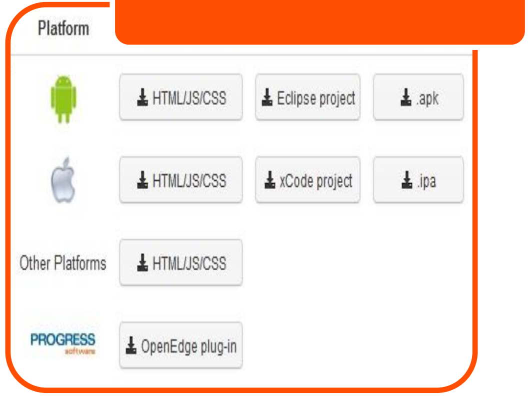


Build UI in Mobile App Builder



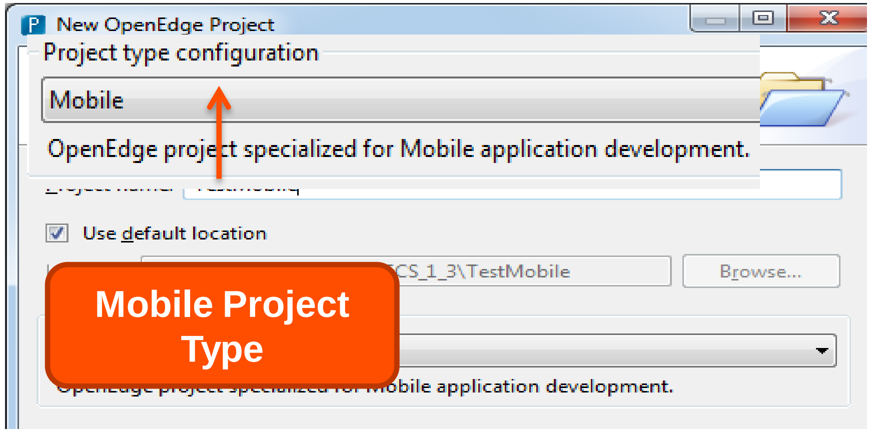
Bind data and UI

Build and Deploy



Mobile App Creation as Part of Mobile Project

- Creates project (or a mobile application) in cloud
- Requires user credentials (PSDN) for Mobile App Builder
- Choice of App types
  - Device
    - Phone or Tablet
  - Predefined Templates
    - Weather App
    - Session Enabled Phone and Tablet
  - From backup
    - Existing Mobile App



New OpenEdge Project

Project type configuration

Mobile

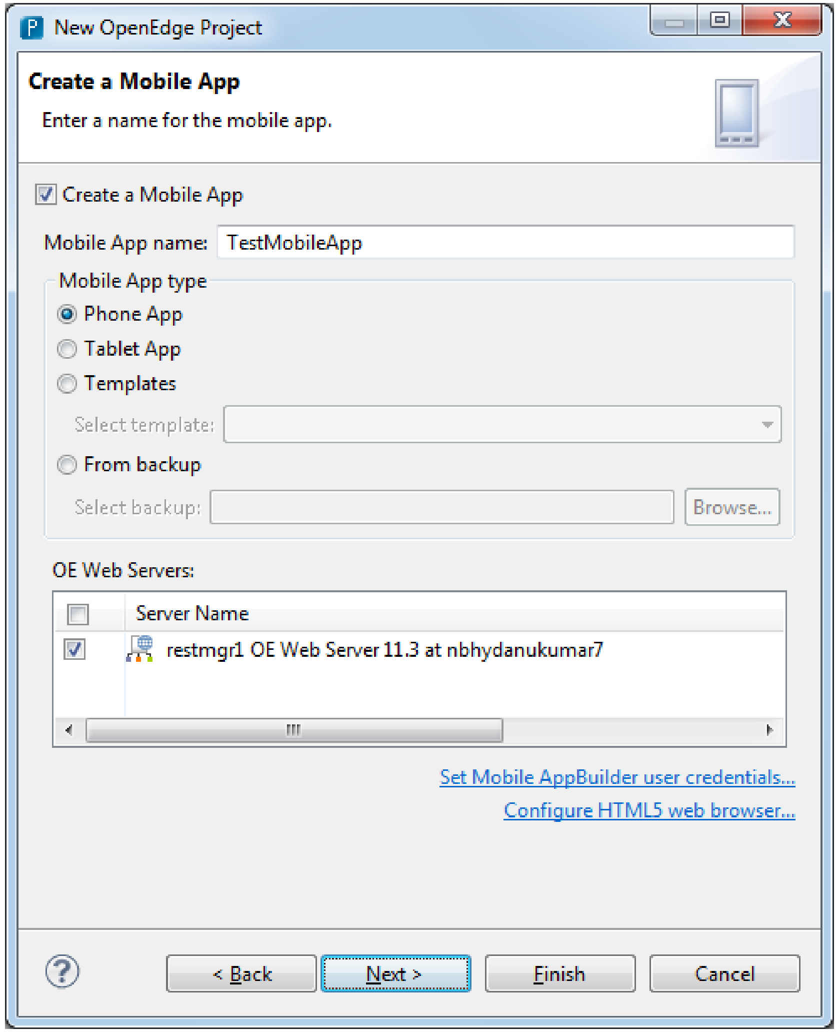
OpenEdge project specialized for Mobile application development.

☑ Use default location

CS_1_3\TestMobile

obile application development.

Mobile Project Type



New OpenEdge Project

Create a Mobile App

Enter a name for the mobile app.

☑ Create a Mobile App

Mobile App name: TestMobileApp

Mobile App type

● Phone App

○ Tablet App

○ Templates

Select template:

○ From backup

Select backup:

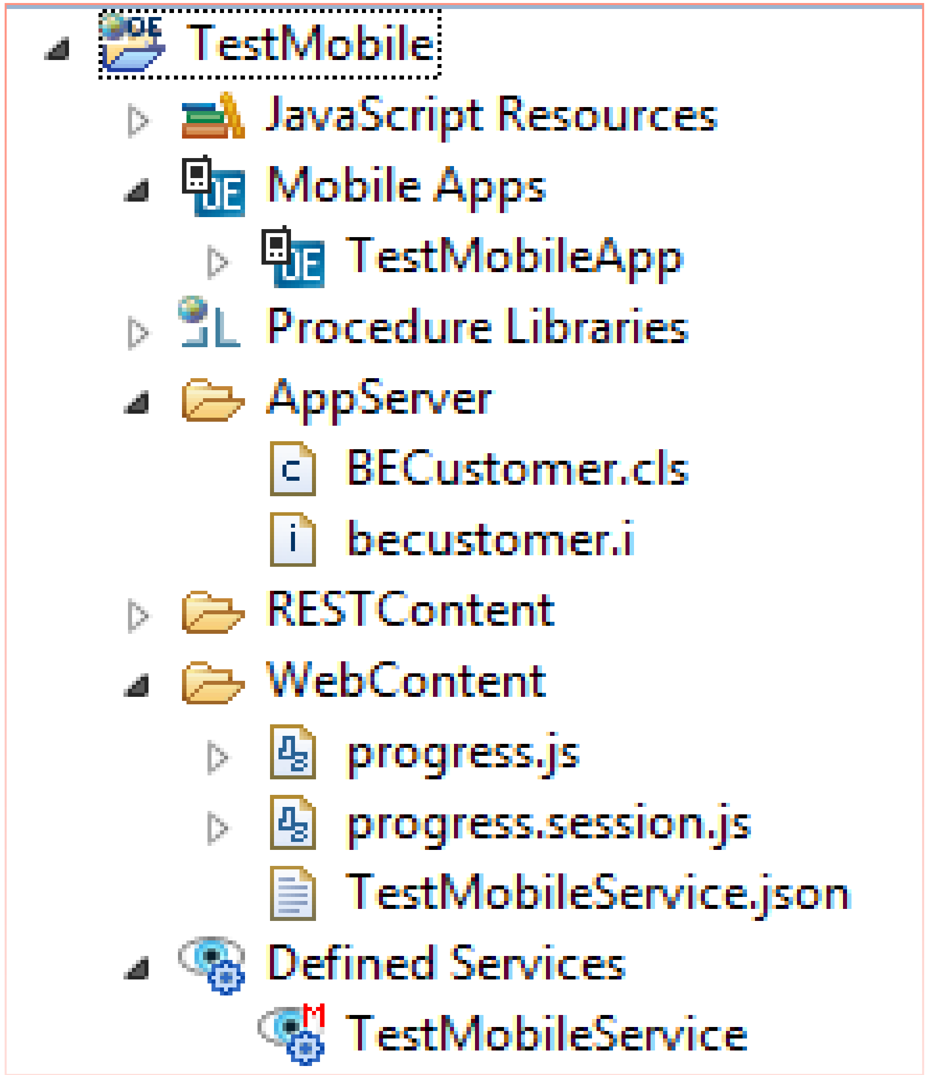
OE Web Servers:

□	Server Name
☑	restmgr1 OE Web Server 11.3 at nbhydanukumar7

Browse...
Set Mobile AppBuilder user credentials...
Configure HTML5 web browser...
< Back
Next >
Finish
Cancel

Mobile Project Folder Hierarchy

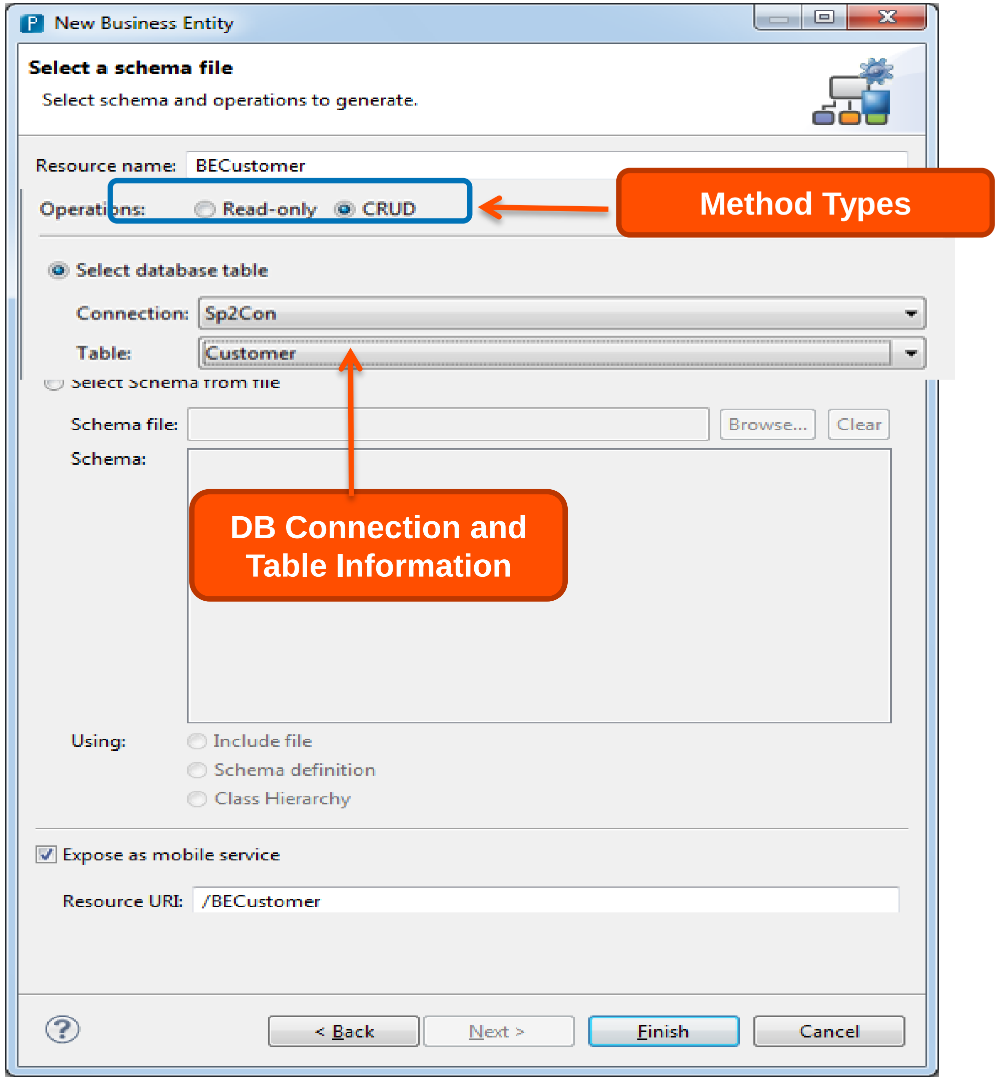
- **Mobile Apps**
  - Contains mobile apps that are created in the Mobile project
- **AppServer**
  - Contains files (business logic) to be published to server
- **RESTContent**
  - Contains files that will be used for generating a Mobile service (WAR) file
- **WebContent**
  - Progress JavaScript Data Object (JSDO) files: progress.js, progress.session.js, and the .json files
- **Defined Services**
  - Contains a list of defined/created Mobile and REST services.



```
◢ TestMobile
  ▷ JavaScript Resources
  ◢ Mobile Apps
    ▷ TestMobileApp
  ▷ Procedure Libraries
  ◢ AppServer
      BECustomer.cls
      becustomer.i
  ▷ RESTContent
  ◢ WebContent
    ▷ progress.js
    ▷ progress.session.js
      TestMobileService.json
  ◢ Defined Services
      TestMobileService
```

Business Entity

- Resembles same as ABL Class wizard
- Contains ABL Business Logic
- Default CRUD operations/methods are generated with stubs
- Schema file selection for CRUD operations
- Decorated with Mobile annotations
- Can also be generated for a given DB table



New Business Entity

Select a schema file

Select schema and operations to generate.

Resource name: BECustomer

Operations: ○ Read-only ● CRUD

● Select database table

Connection: Sp2Con

Table: Customer

○ Select schema from file

Schema file:

Schema:

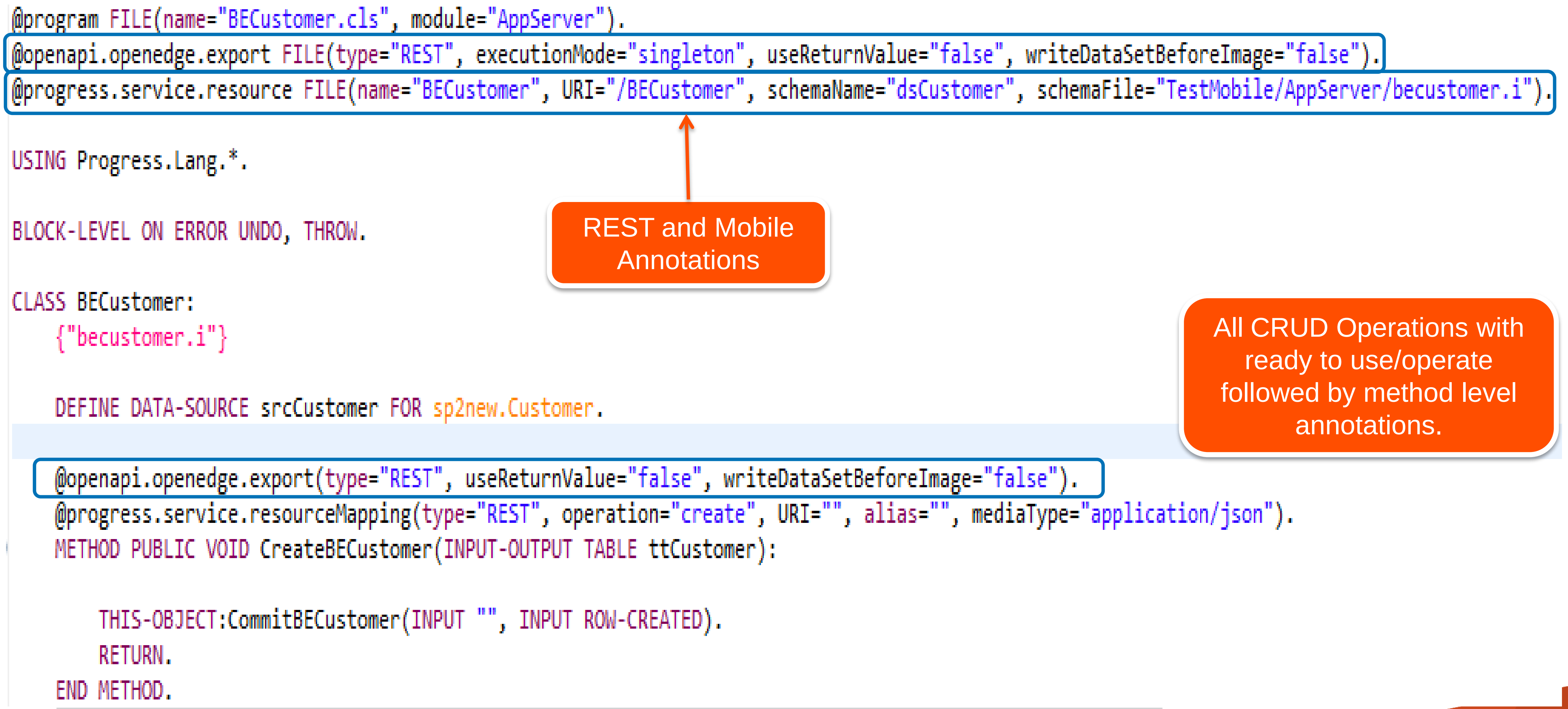
Using: ○ Include file ○ Schema definition ○ Class Hierarchy

☑ Expose as mobile service

Resource URI: /BECustomer

PROGRESS

Business Entity in Detail

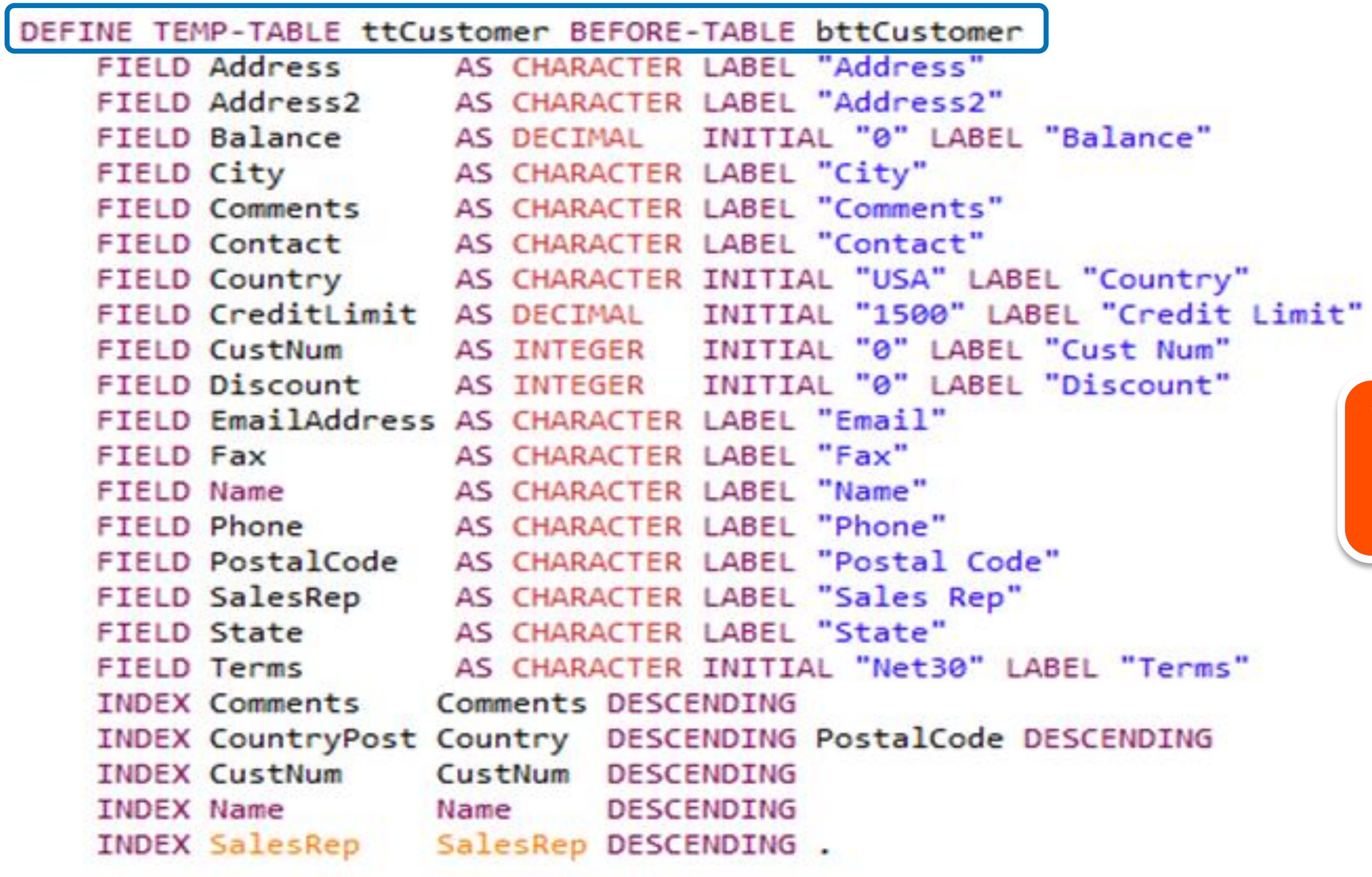


```
@program FILE(name="BECustomer.cls", module="AppServer").
@openapi.openedge.export FILE(type="REST", executionMode="singleton", useReturnValue="false", writeDataSetBeforeImage="false").
@progress.service.resource FILE(name="BECustomer", URI="/BECustomer", schemaName="dsCustomer", schemaFile="TestMobile/AppServer/becustomer.i").

USING Progress.Lang.*.

BLOCK-LEVEL ON ERROR UNDO, THROW.

CLASS BECustomer:
    {"becustomer.i"}

    DEFINE DATA-SOURCE srcCustomer FOR sp2new.Customer.

    @openapi.openedge.export(type="REST", useReturnValue="false", writeDataSetBeforeImage="false").
    @progress.service.resourceMapping(type="REST", operation="create", URI="", alias="", mediaType="application/json").
    METHOD PUBLIC VOID CreateBECustomer(INPUT-OUTPUT TABLE ttCustomer):

        THIS-OBJECT:CommitBECustomer(INPUT "", INPUT ROW-CREATED).
        RETURN.
    END METHOD.
```

REST and Mobile Annotations

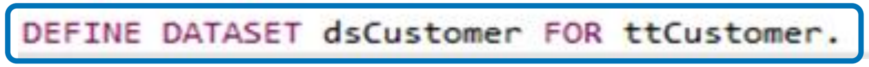
All CRUD Operations with ready to use/operate followed by method level annotations.

Business Entity - Include File Contents



```
DEFINE TEMP-TABLE ttCustomer BEFORE-TABLE bttCustomer
    FIELD Address      AS CHARACTER LABEL "Address"
    FIELD Address2     AS CHARACTER LABEL "Address2"
    FIELD Balance      AS DECIMAL   INITIAL "0" LABEL "Balance"
    FIELD City         AS CHARACTER LABEL "City"
    FIELD Comments     AS CHARACTER LABEL "Comments"
    FIELD Contact      AS CHARACTER LABEL "Contact"
    FIELD Country      AS CHARACTER INITIAL "USA" LABEL "Country"
    FIELD CreditLimit  AS DECIMAL   INITIAL "1500" LABEL "Credit Limit"
    FIELD CustNum      AS INTEGER   INITIAL "0" LABEL "Cust Num"
    FIELD Discount     AS INTEGER   INITIAL "0" LABEL "Discount"
    FIELD EmailAddress AS CHARACTER LABEL "Email"
    FIELD Fax          AS CHARACTER LABEL "Fax"
    FIELD Name         AS CHARACTER LABEL "Name"
    FIELD Phone        AS CHARACTER LABEL "Phone"
    FIELD PostalCode   AS CHARACTER LABEL "Postal Code"
    FIELD SalesRep     AS CHARACTER LABEL "Sales Rep"
    FIELD State        AS CHARACTER LABEL "State"
    FIELD Terms        AS CHARACTER INITIAL "Net30" LABEL "Terms"
    INDEX Comments    Comments DESCENDING
    INDEX CountryPost Country  DESCENDING PostalCode DESCENDING
    INDEX CustNum     CustNum  DESCENDING
    INDEX Name        Name     DESCENDING
    INDEX SalesRep    SalesRep DESCENDING .
```

Temp -Table definition with complete field information as that of DB Table



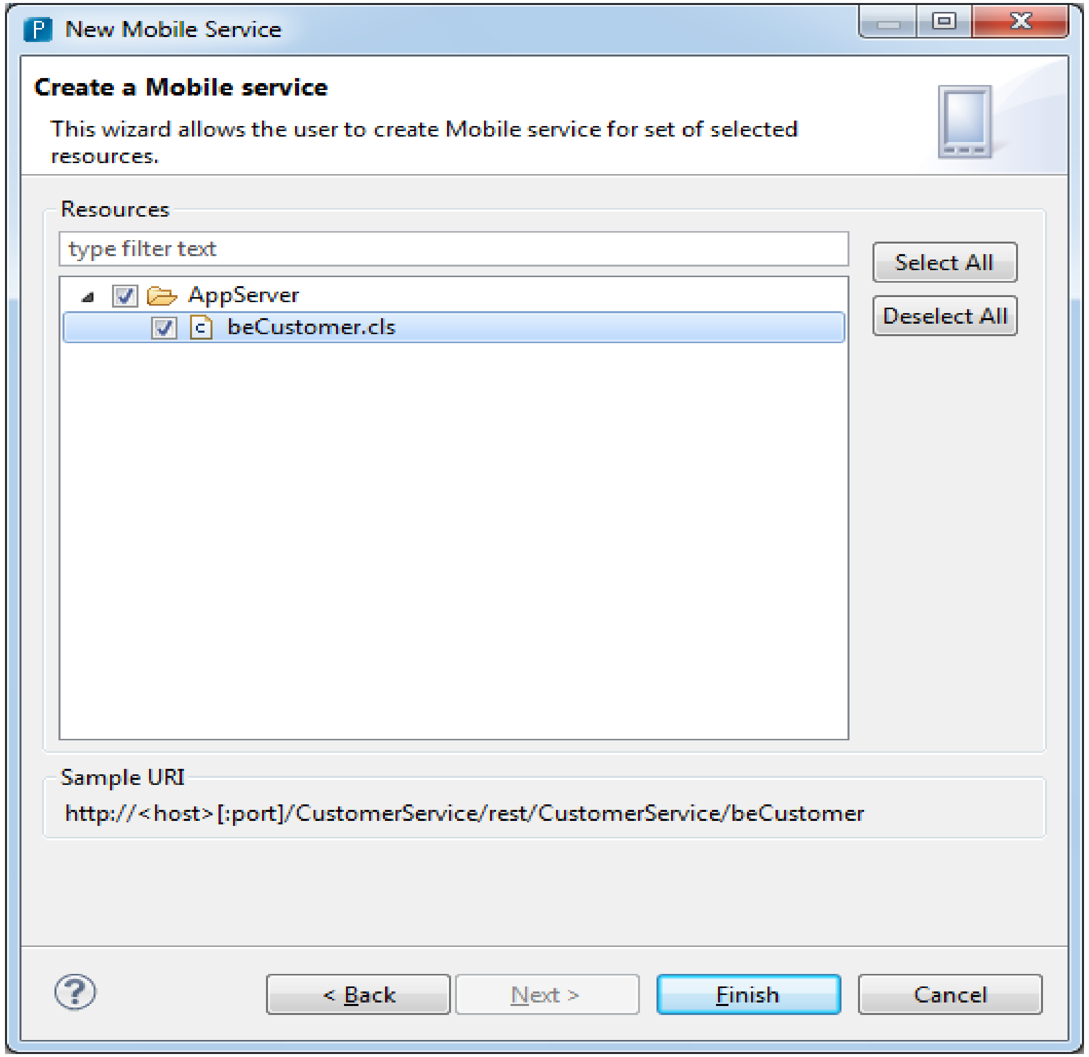
```
DEFINE DATASET dsCustomer FOR ttCustomer.
```

Dataset Name

Mobile Service

- Created for a Business Entity or any Mobile annotated file
- A mobile service results in REST and Mobile artifacts (.json) in PDS OE
- JSON file contains complete information (schema, operations) of the mobile resources
- JSON file termed as Catalog file
- JSON file location:

  <Project Name>/WebContent/<Service Name>.json



Create a Mobile service

This wizard allows the user to create Mobile service for set of selected resources.

Resources

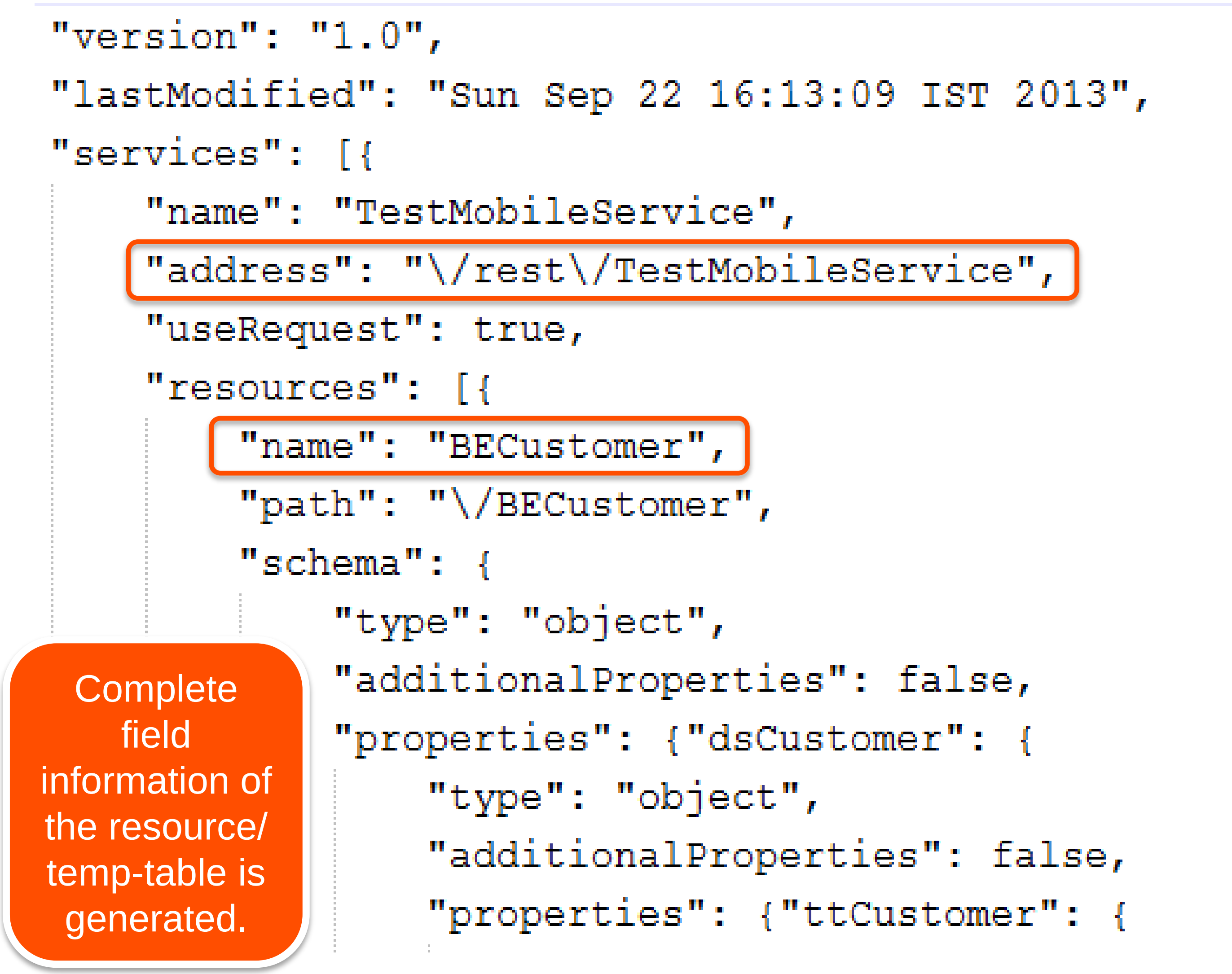
type filter text

- ☑ AppServer
  - ☑ beCustomer.cls

Sample URI

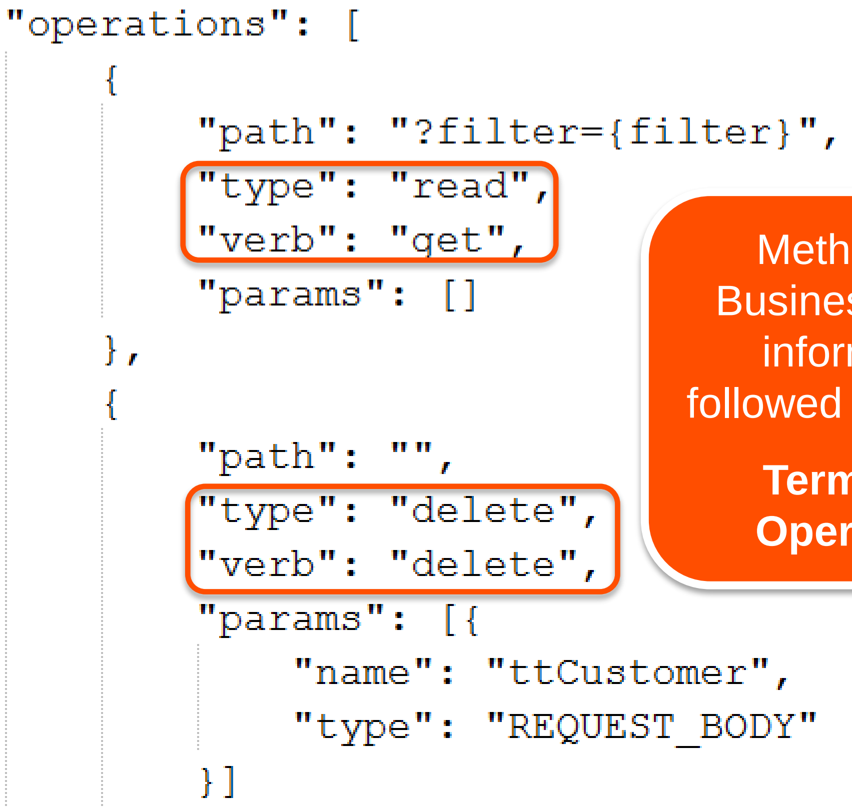
http://<host>[:port]/CustomerService/rest/CustomerService/beCustomer

Catalog File – JSON Representation

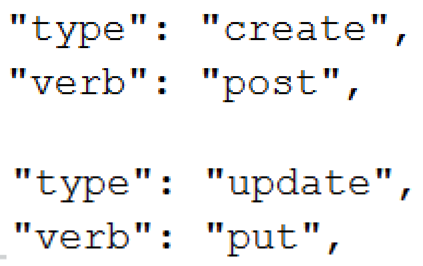


```
"version": "1.0",
"lastModified": "Sun Sep 22 16:13:09 IST 2013",
"services": [{
    "name": "TestMobileService",
    "address": "\/rest\/TestMobileService",
    "useRequest": true,
    "resources": [{
        "name": "BECustomer",
        "path": "\/BECustomer",
        "schema": {
            "type": "object",
            "additionalProperties": false,
            "properties": {"dsCustomer": {
                "type": "object",
                "additionalProperties": false,
                "properties": {"ttCustomer": {
```

Complete field information of the resource/ temp-table is generated.



```
"operations": [
    {
        "path": "?filter={filter}",
        "type": "read",
        "verb": "get",
        "params": []
    },
    {
        "path": "",
        "type": "delete",
        "verb": "delete",
        "params": [{
            "name": "ttCustomer",
            "type": "REQUEST_BODY"
        }]
```



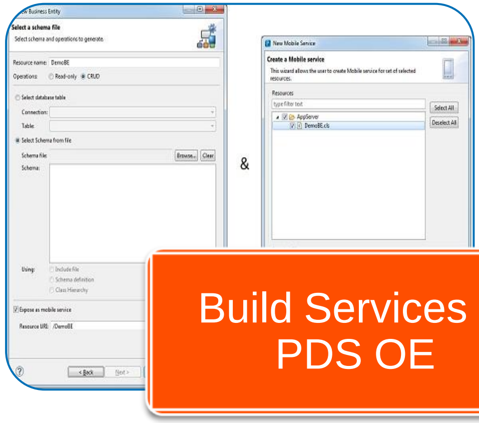
```
"type": "create",
"verb": "post",

"type": "update",
"verb": "put",
```

Methods (in Business Entity) information followed by its verb

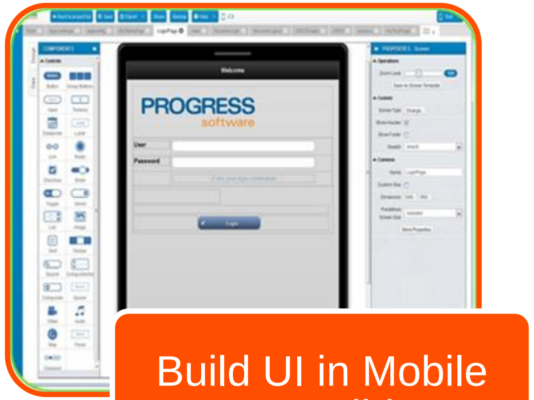
Termed as Operations

Mobile App and Steps Involved

Build Services → Create UI → Perform Mapping → Build and Deploy

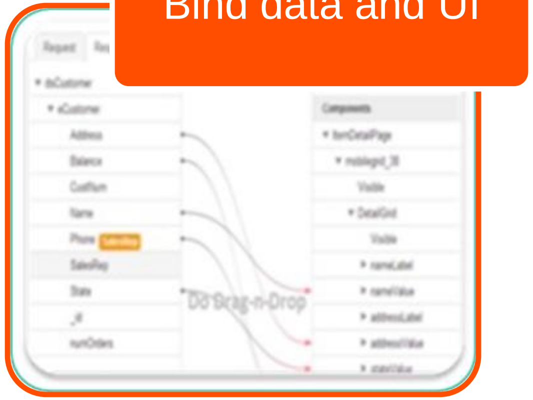


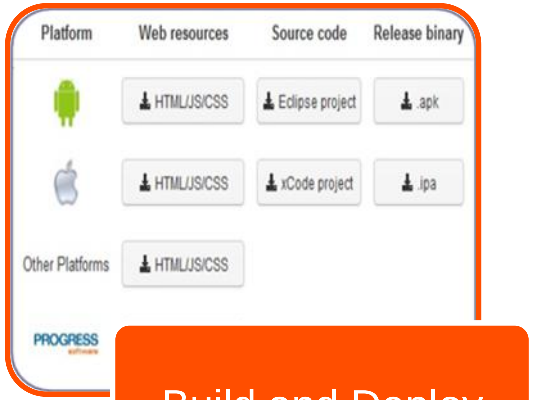
Build Services in PDS OE



Build UI in Mobile App Builder

Bind data and UI

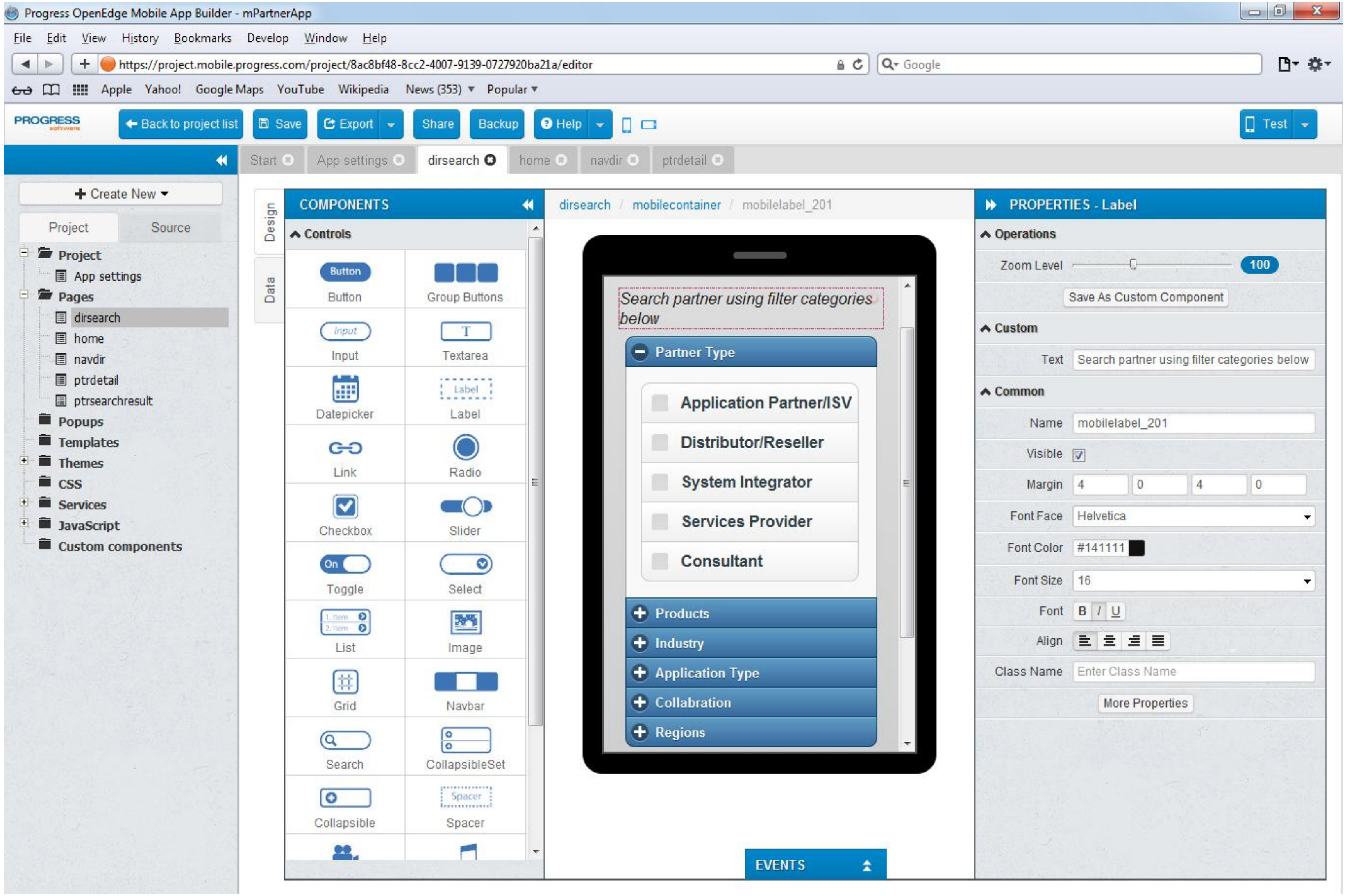




Platform	Web resources	Source code	Release binary
	HTML/JS/CSS	Eclipse project	.apk
	HTML/JS/CSS	xCode project	.ipa
Other Platforms	HTML/JS/CSS		

Build and Deploy

Mobile App Builder - “WYSIWYG” Designer



https://project.mobile.progress.com/project/8ac8bf48-8cc2-4007-9139-0727920ba21a/editor

Search partner using filter categories below

Partner Type

- □ Application Partner/ISV
- □ Distributor/Reseller
- □ System Integrator
- □ Services Provider
- □ Consultant

Products

Industry

Application Type

Collaboration

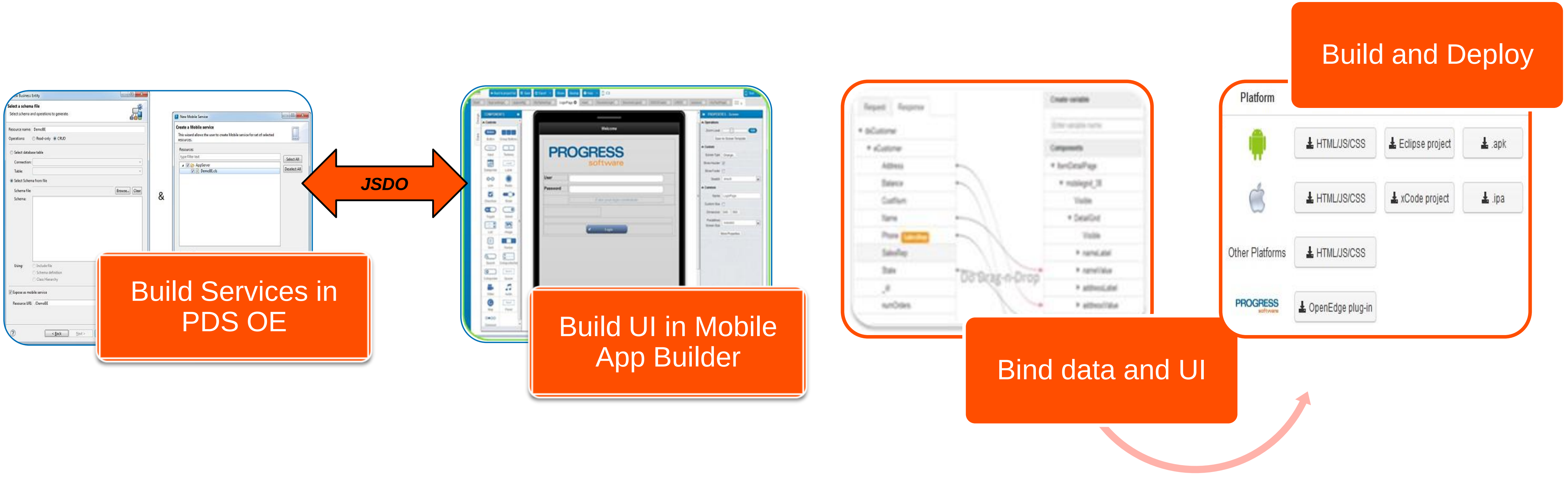
Regions

Zoom Level	100
Text	Search partner using filter categories below
Name	mobilelabel_201
Visible	☑
Margin	4 0 4 0
Font Face	Helvetica
Font Color	#141111
Font Size	16
Font	B I U
Align	
Class Name	Enter Class Name

PROGRESS

Mobile App and Steps Involved

Build Services → Create UI → Perform Mapping → Build and Deploy



Build Services in PDS OE

&

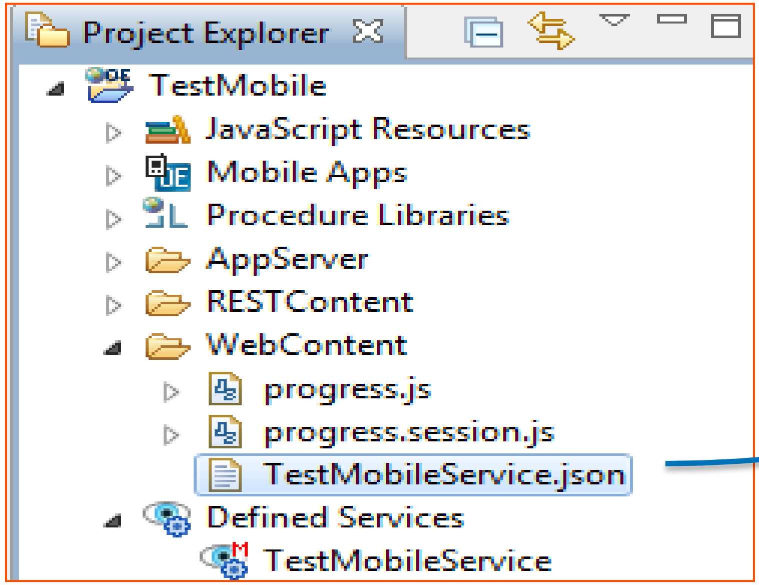
JSDO

Build UI in Mobile App Builder

Bind data and UI

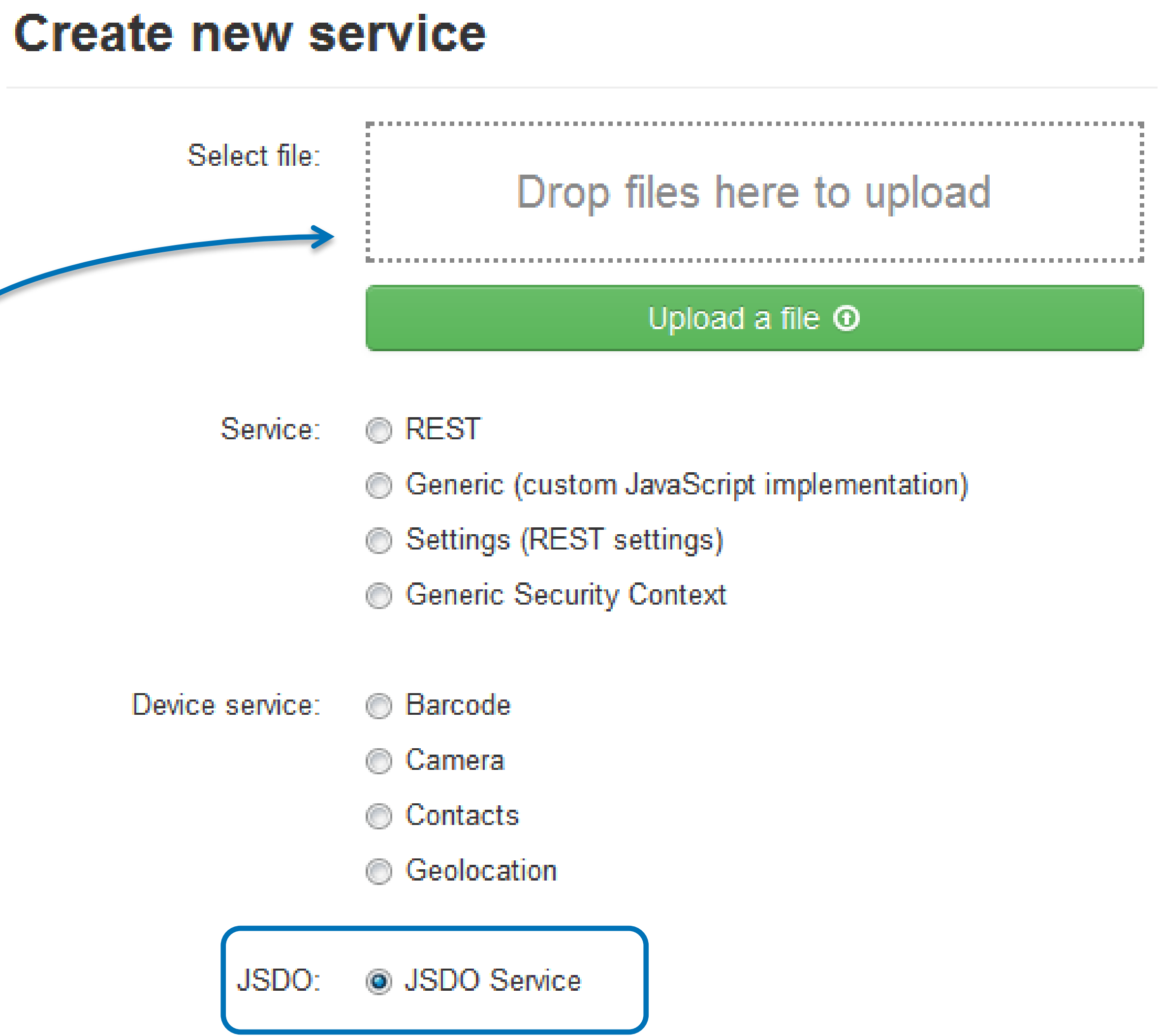
Build and Deploy

JSDO Service Creation



Project Explorer

- TestMobile
  - JavaScript Resources
  - Mobile Apps
  - Procedure Libraries
  - AppServer
  - RESTContent
  - WebContent
    - progress.js
    - progress.session.js
    - TestMobileService.json
  - Defined Services
    - TestMobileService



Create new service

Select file: Drop files here to upload

Upload a file

Service:
- ○ REST
- ○ Generic (custom JavaScript implementation)
- ○ Settings (REST settings)
- ○ Generic Security Context

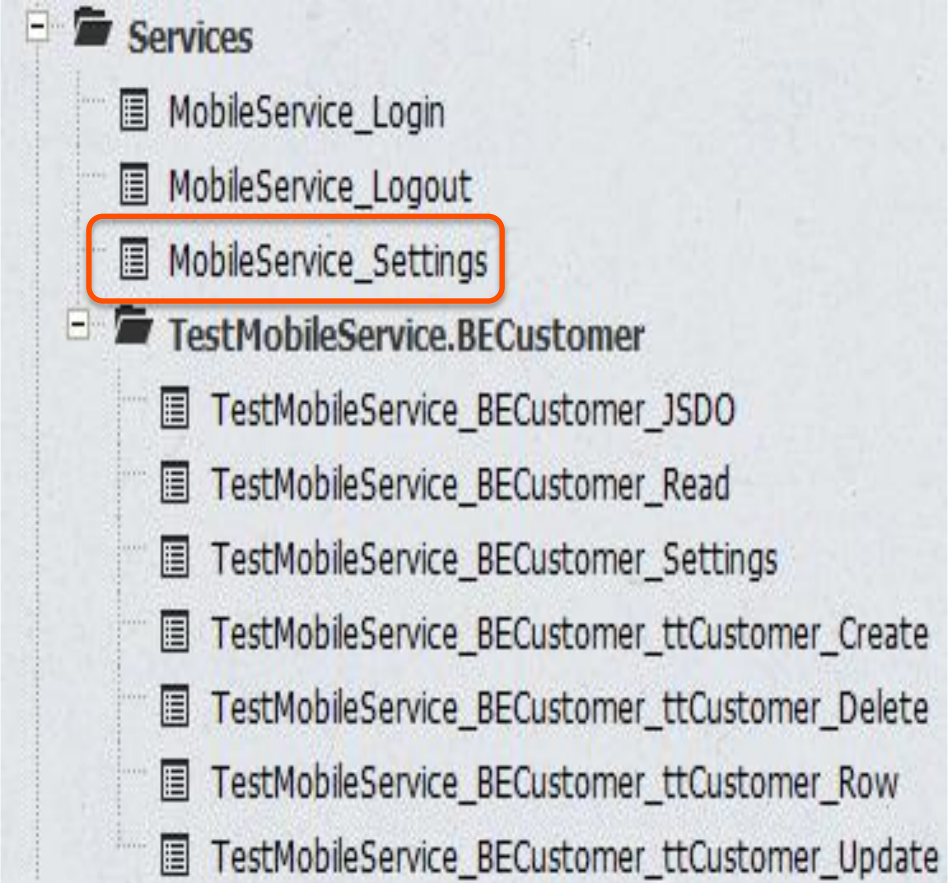
Device service:
- ○ Barcode
- ○ Camera
- ○ Contacts
- ○ Geolocation

JSDO: ● JSDO Service

- Option to create JSDO Services in Mobile App Builder
- Drag and Drop Catalog file

JSDO Services

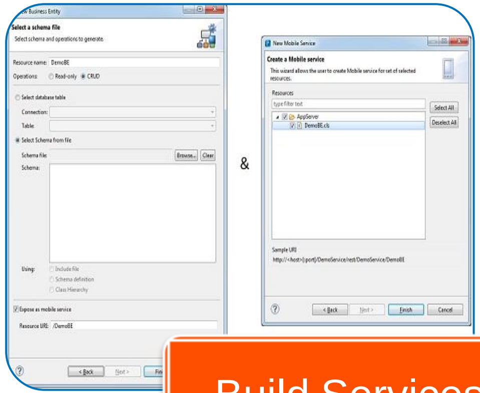
- Contains default Login Service for JSDO Load
- One for each method gets created
- Catalog and Service URI are mandatory
- Further Mappings are performed for respective services against UI elements
- JavaScript Implementation for all methods are available by default



Services
- MobileService_Login
- MobileService_Logout
- MobileService_Settings
- TestMobileService.BECustomer
  - TestMobileService_BECustomer_JSDO
  - TestMobileService_BECustomer_Read
  - TestMobileService_BECustomer_Settings
  - TestMobileService_BECustomer_ttCustomer_Create
  - TestMobileService_BECustomer_ttCustomer_Delete
  - TestMobileService_BECustomer_ttCustomer_Row
  - TestMobileService_BECustomer_ttCustomer_Update

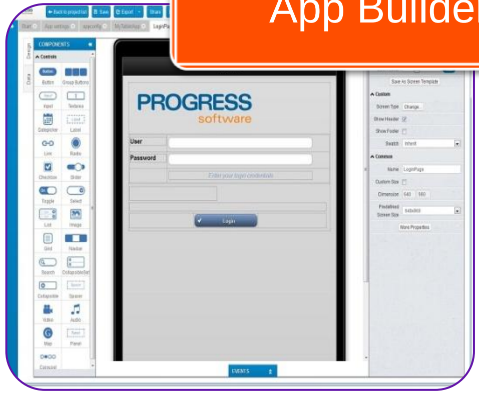
Binding Data Elements

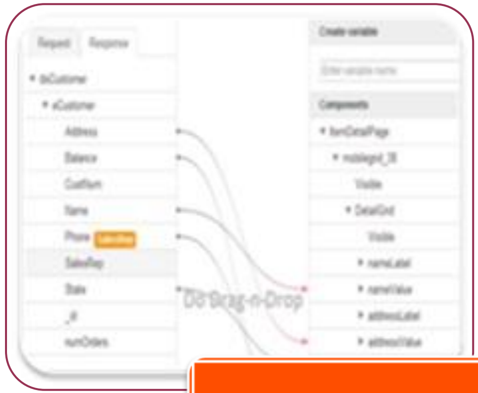
Build Services → Create UI → Perform Mapping → Build, Deploy



Build Services in PDS OE

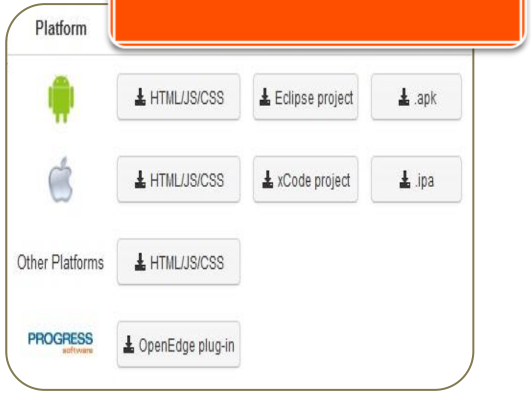
Build UI in Mobile App Builder





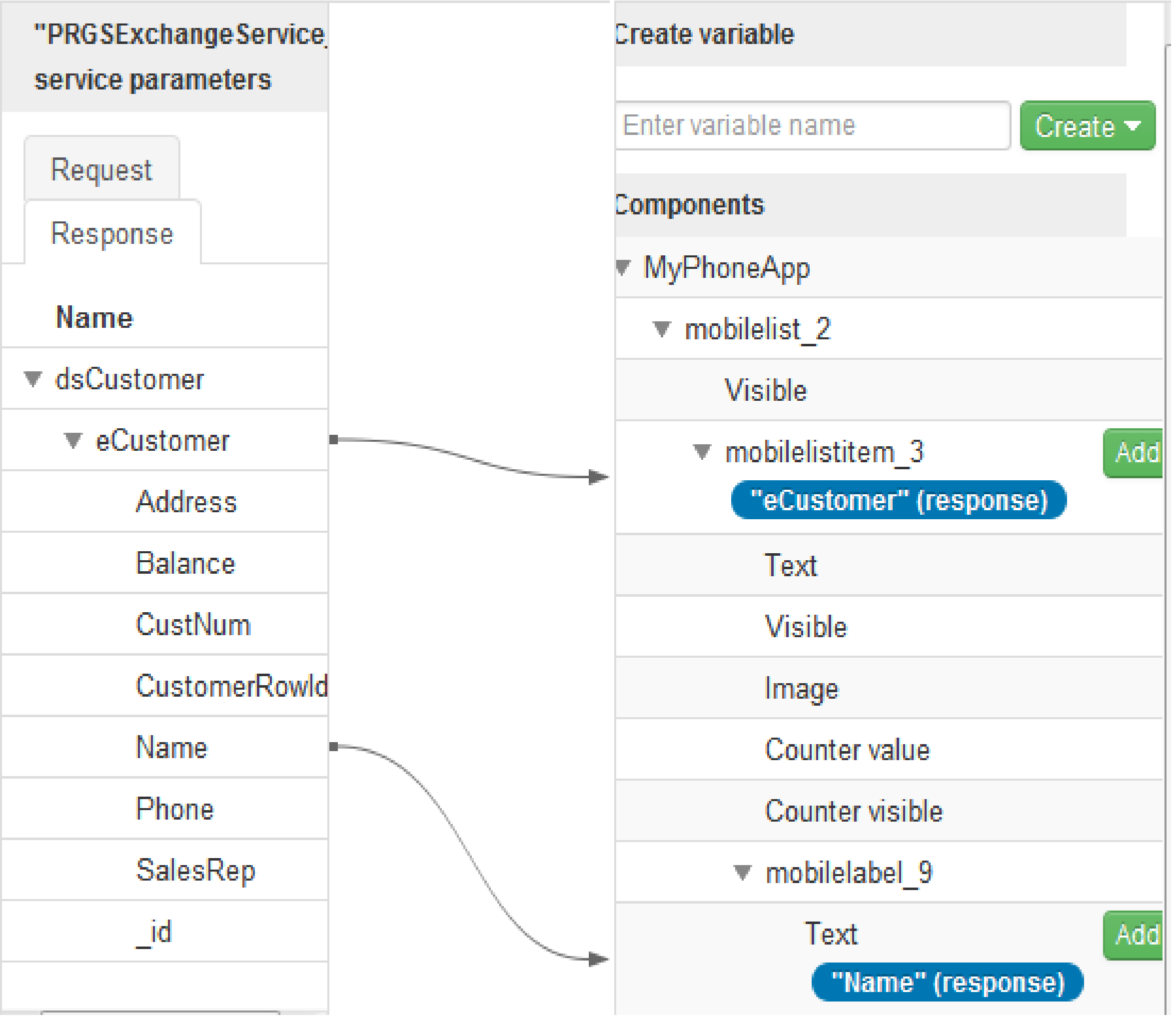
Bind data and UI

Build , Deploy



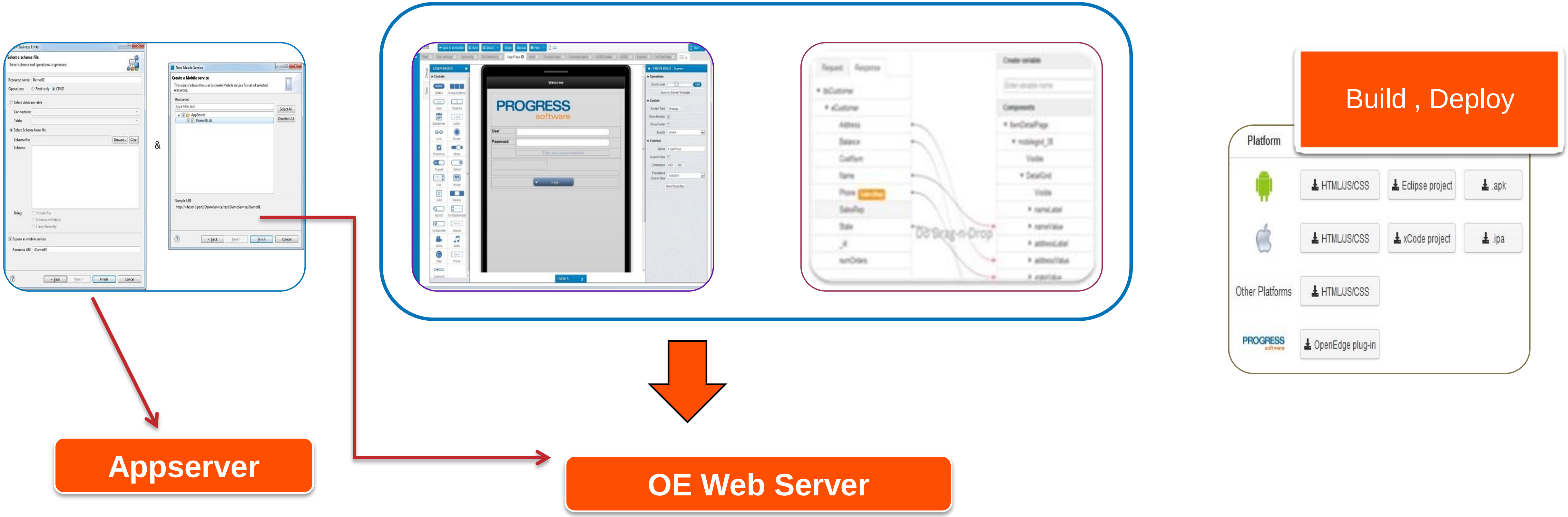
Bind Data and UI Elements

- Data elements from catalog file
- UI Elements in Mobile App Builder
- Perform mappings with respect to data and UI elements
- Custom JS implementation can also be added along with mapping



Publish Business Logic and UI

Build Services → Create UI → Perform Mapping → Build, Deploy

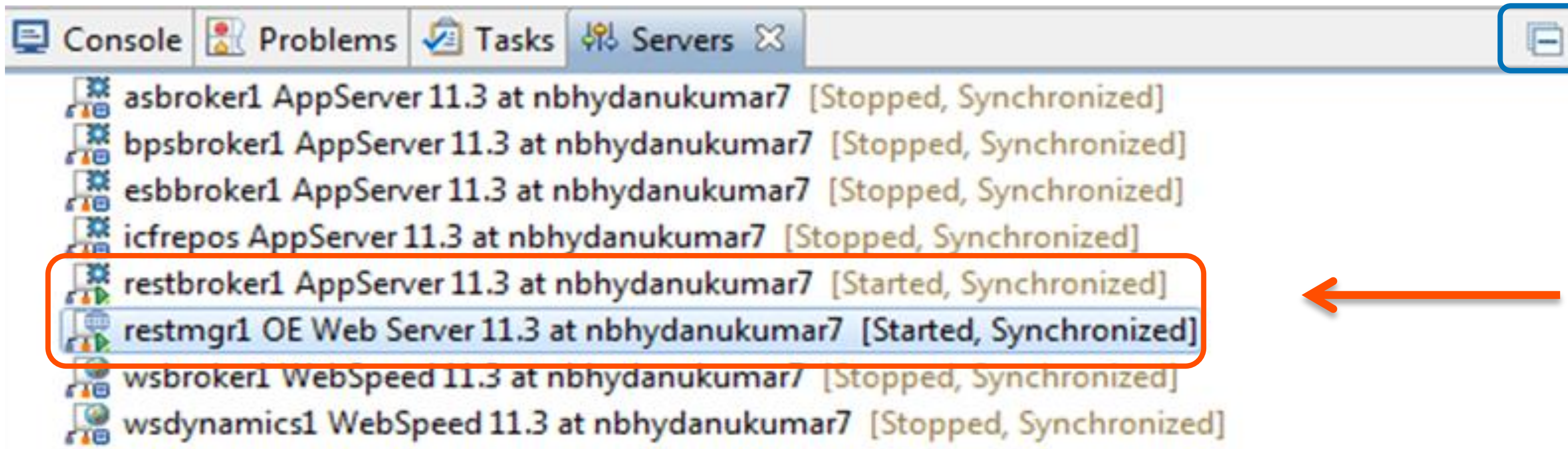


Appserver

OE Web Server

Servers in PDSOE

- OpenEdge REST/Mobile applications need two types of servers
  - OpenEdge AppServer (restbroker1)
  - OE Web Server (restmgr1)
- Servers can be stopped and started from Servers view in PDSOE



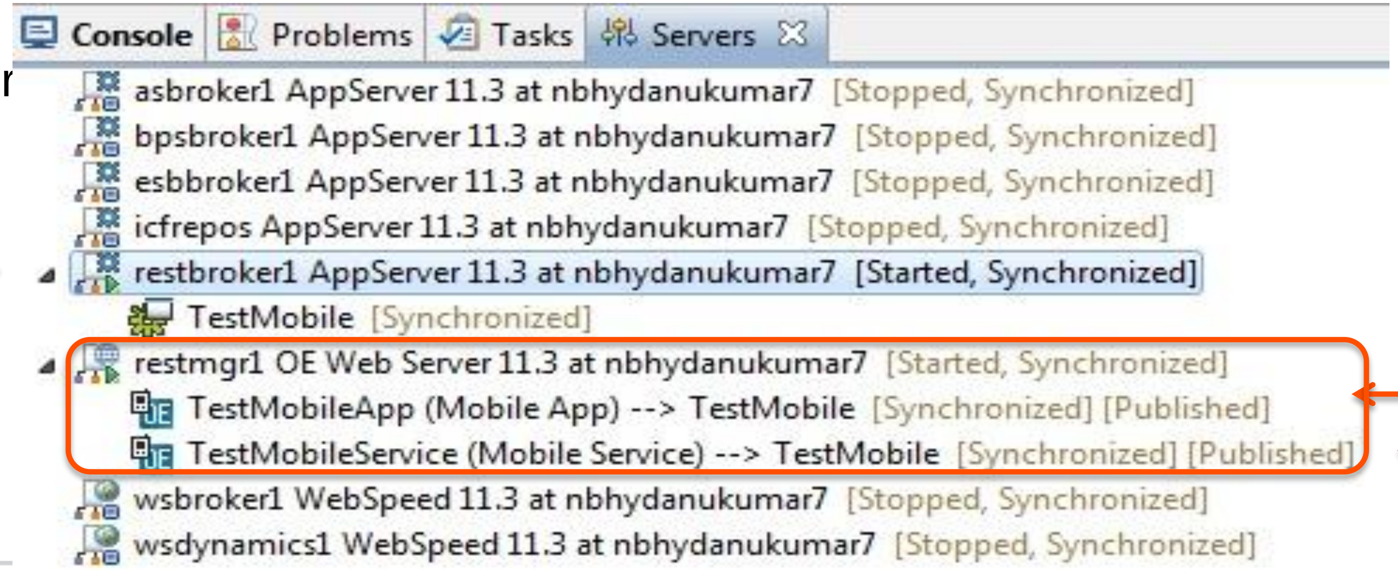
asbroker1 AppServer 11.3 at nbhydanukumar7 [Stopped, Synchronized]
bpsbroker1 AppServer 11.3 at nbhydanukumar7 [Stopped, Synchronized]
esbbroker1 AppServer 11.3 at nbhydanukumar7 [Stopped, Synchronized]
icfrepos AppServer 11.3 at nbhydanukumar7 [Stopped, Synchronized]
restbroker1 AppServer 11.3 at nbhydanukumar7 [Started, Synchronized]
restmgr1 OE Web Server 11.3 at nbhydanukumar7 [Started, Synchronized]
wsbroker1 WebSpeed 11.3 at nbhydanukumar7 [Stopped, Synchronized]
wsdynamics1 WebSpeed 11.3 at nbhydanukumar7 [Stopped, Synchronized]

- Other Approaches
  - Proenv command line
  - OpenEdge Explorer / OpenEdge Management

Publishing Mobile App and Mobile Service

- Complete Mobile Project constitutes both Mobile Application and Mobile Service
- Same OE Web Server is used for publishing both Mobile Application and Mobile Service
- Two different .WAR files are published to OE Web Server
- Publishing a mobile application does:
  - Fetch sources from Mobile App Builder
  - Package it as WAR file
  - Publish to Tomcat director

Business Logic to Appserver

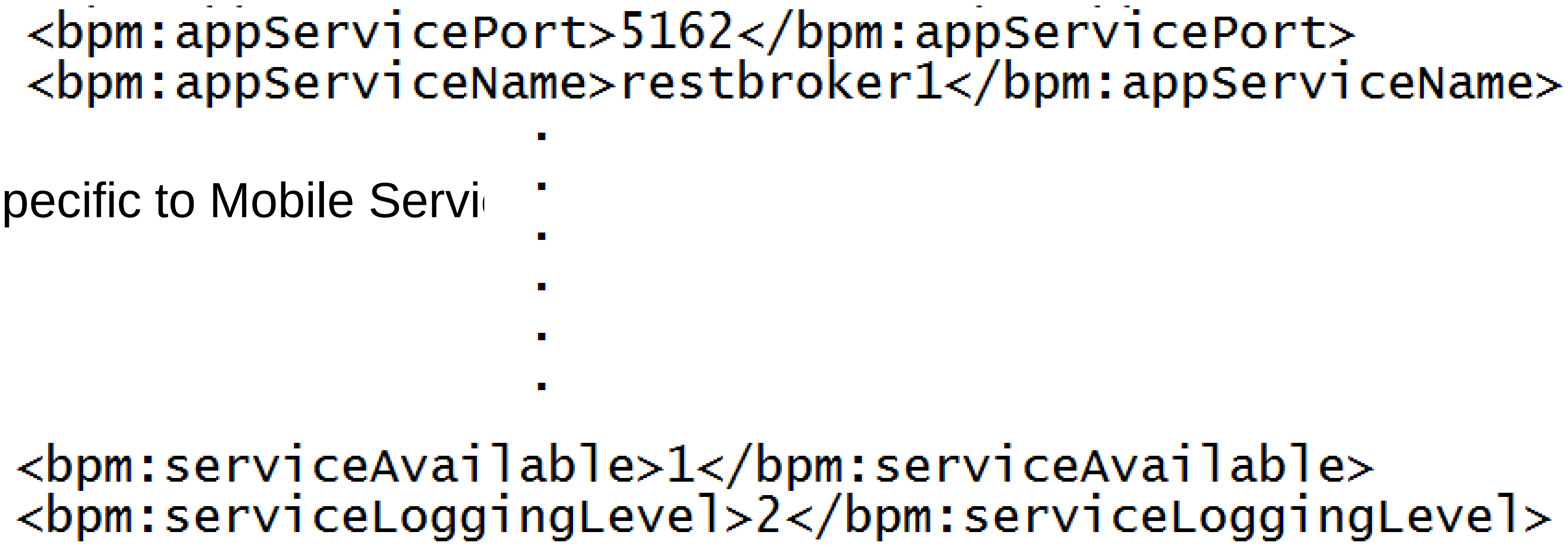


asbroker1 AppServer 11.3 at nbhydanukumar7 [Stopped, Synchronized]
bpsbroker1 AppServer 11.3 at nbhydanukumar7 [Stopped, Synchronized]
esbbroker1 AppServer 11.3 at nbhydanukumar7 [Stopped, Synchronized]
icfrepos AppServer 11.3 at nbhydanukumar7 [Stopped, Synchronized]
restbroker1 AppServer 11.3 at nbhydanukumar7 [Started, Synchronized]
TestMobile [Synchronized]
restmgr1 OE Web Server 11.3 at nbhydanukumar7 [Started, Synchronized]
TestMobileApp (Mobile App) --> TestMobile [Synchronized] [Published]
TestMobileService (Mobile Service) --> TestMobile [Synchronized] [Published]
wsbroker1 WebSpeed 11.3 at nbhydanukumar7 [Stopped, Synchronized]
wsdynamics1 WebSpeed 11.3 at nbhydanukumar7 [Stopped, Synchronized]

Deployed Application to OE Webserver

- A WAR file comprises of many artifacts
  - PAAR File(s)
  - Library Files
  - Log Files
  - Security Files
  - <Service Name>.json specific to Mobile Servi
  - Runtime properties

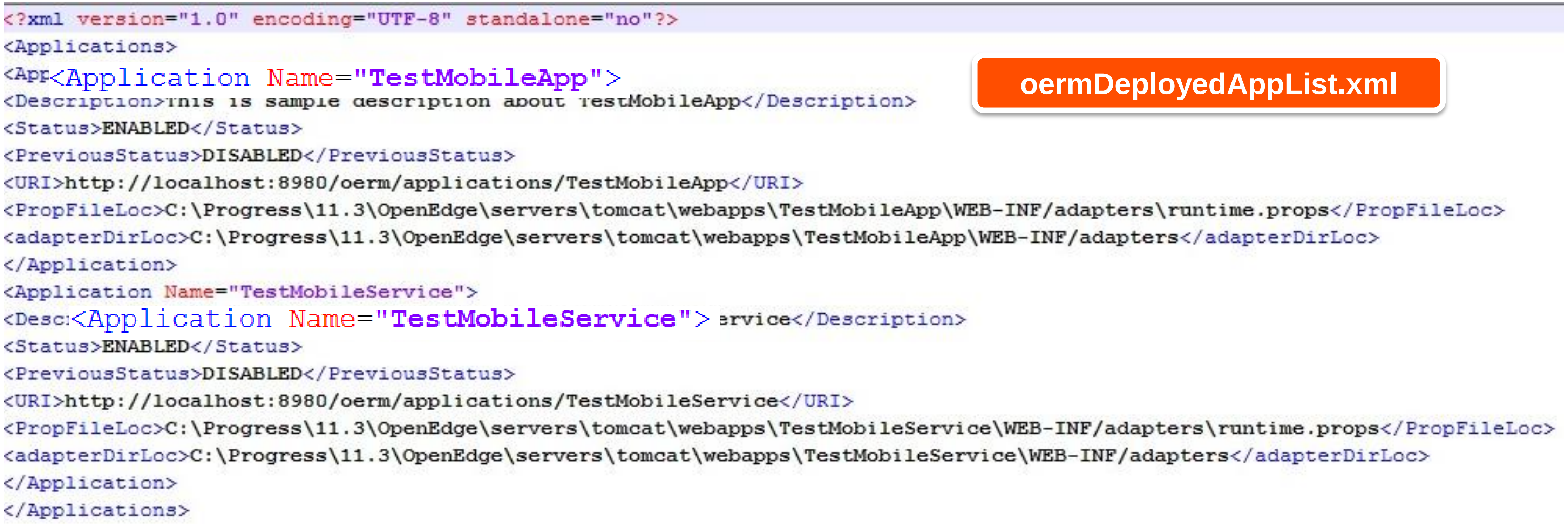
runtime.props



```
<bpm:appServicePort>5162</bpm:appServicePort>
<bpm:appServiceName>restbroker1</bpm:appServiceName>
                  .
                  .
                  .
                  .
                  .
                  .

<bpm:serviceAvailable>1</bpm:serviceAvailable>
<bpm:serviceLoggingLevel>2</bpm:serviceLoggingLevel>
```

What Does oerm Contain?

- Library Files, Log Information
- List of deployed applications
- Security related files

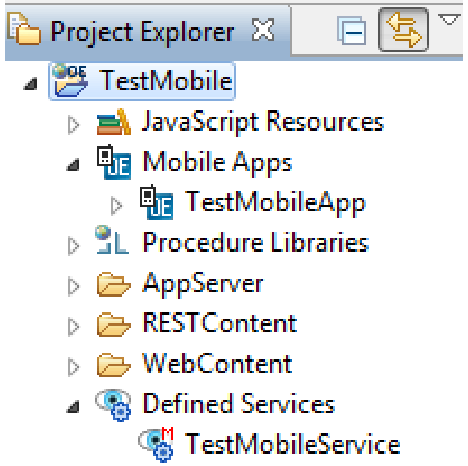


oermDeployedAppList.xml

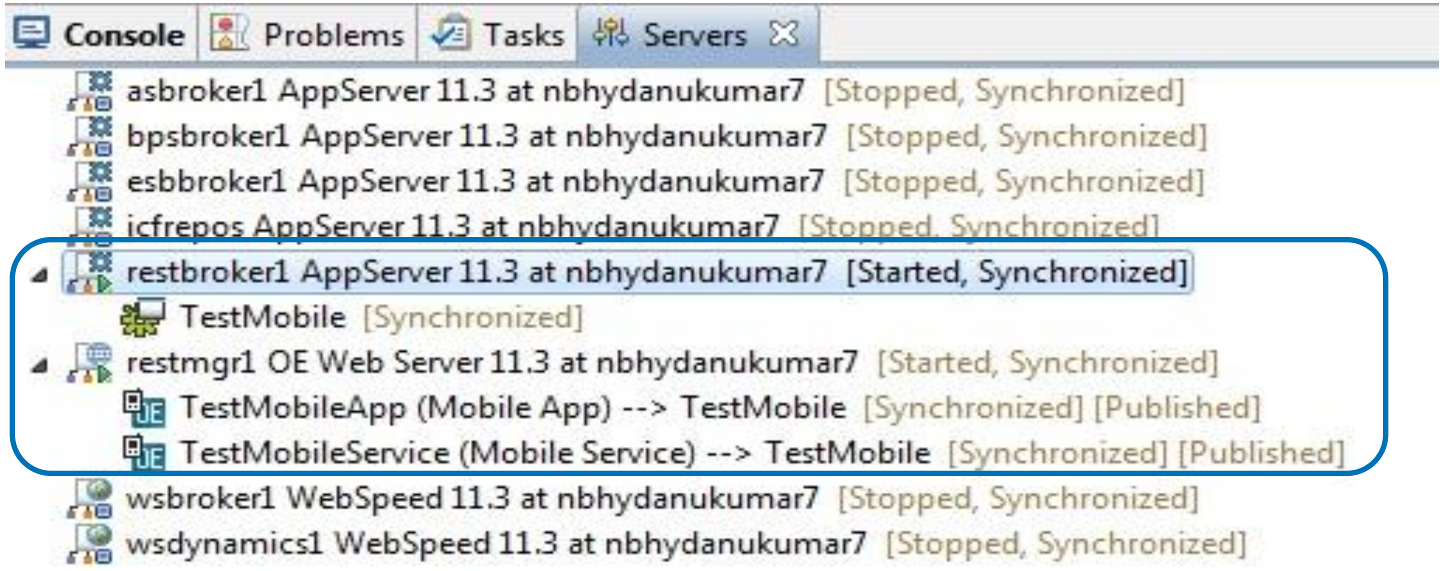
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Applications>
<App<Application Name="TestMobileApp">
<Description>This is sample description about TestMobileApp</Description>
<Status>ENABLED</Status>
<PreviousStatus>DISABLED</PreviousStatus>
<URI>http://localhost:8980/oerm/applications/TestMobileApp</URI>
<PropFileLoc>C:\Progress\11.3\OpenEdge\servers\tomcat\webapps\TestMobileApp\WEB-INF/adapters\runtime.props</PropFileLoc>
<adapterDirLoc>C:\Progress\11.3\OpenEdge\servers\tomcat\webapps\TestMobileApp\WEB-INF/adapters</adapterDirLoc>
</Application>
<Application Name="TestMobileService">
<Desc<Application Name="TestMobileService"> ervice</Description>
<Status>ENABLED</Status>
<PreviousStatus>DISABLED</PreviousStatus>
<URI>http://localhost:8980/oerm/applications/TestMobileService</URI>
<PropFileLoc>C:\Progress\11.3\OpenEdge\servers\tomcat\webapps\TestMobileService\WEB-INF/adapters\runtime.props</PropFileLoc>
<adapterDirLoc>C:\Progress\11.3\OpenEdge\servers\tomcat\webapps\TestMobileService\WEB-INF/adapters</adapterDirLoc>
</Application>
</Applications>
```

Publishing: Sum-up

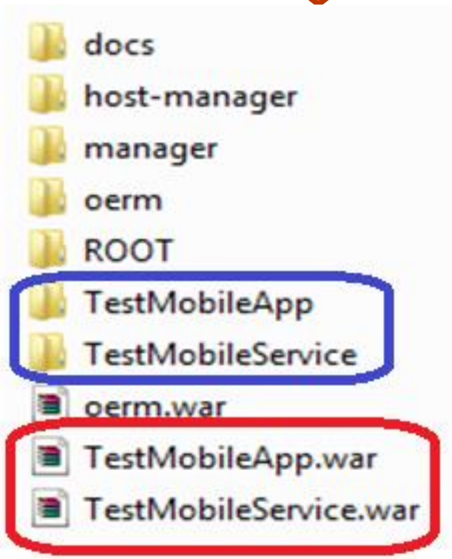
- Defined Service(s) created
- Configured to publish on server
- Published to application directory



```
TestMobile
  JavaScript Resources
  Mobile Apps
    TestMobileApp
  Procedure Libraries
  AppServer
  RESTContent
  WebContent
  Defined Services
    TestMobileService
```



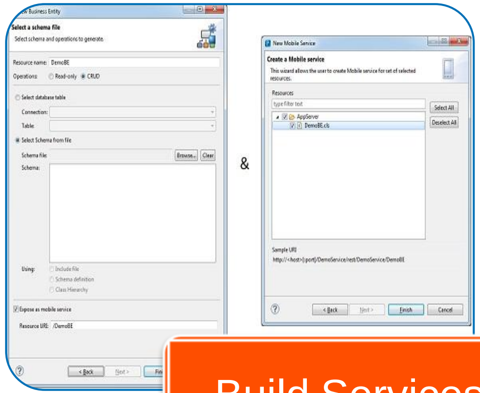
```
asbroker1 AppServer 11.3 at nbhydanukumar7  [Stopped, Synchronized]
bpsbroker1 AppServer 11.3 at nbhydanukumar7  [Stopped, Synchronized]
esbbroker1 AppServer 11.3 at nbhydanukumar7  [Stopped, Synchronized]
icfrepos AppServer 11.3 at nbhydanukumar7  [Stopped, Synchronized]
restbroker1 AppServer 11.3 at nbhydanukumar7  [Started, Synchronized]
    TestMobile  [Synchronized]
restmgr1 OE Web Server 11.3 at nbhydanukumar7  [Started, Synchronized]
    TestMobileApp (Mobile App) --> TestMobile  [Synchronized] [Published]
    TestMobileService (Mobile Service) --> TestMobile  [Synchronized] [Published]
wsbroker1 WebSpeed 11.3 at nbhydanukumar7  [Stopped, Synchronized]
wsdynamics1 WebSpeed 11.3 at nbhydanukumar7  [Stopped, Synchronized]
```



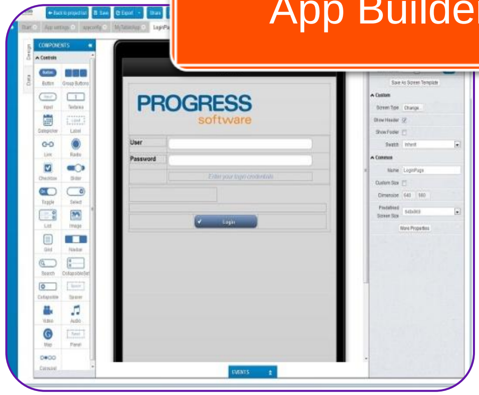
```
docs
host-manager
manager
oerm
ROOT
TestMobileApp
TestMobileService
oerm.war
TestMobileApp.war
TestMobileService.war
```

Mobile App and Steps Involved

Build Services → Create UI → Perform Mapping → Build and Deploy

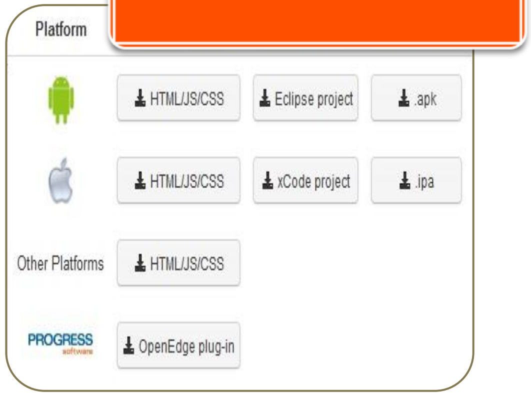


Build Services in PDS OE



Build UI in Mobile App Builder

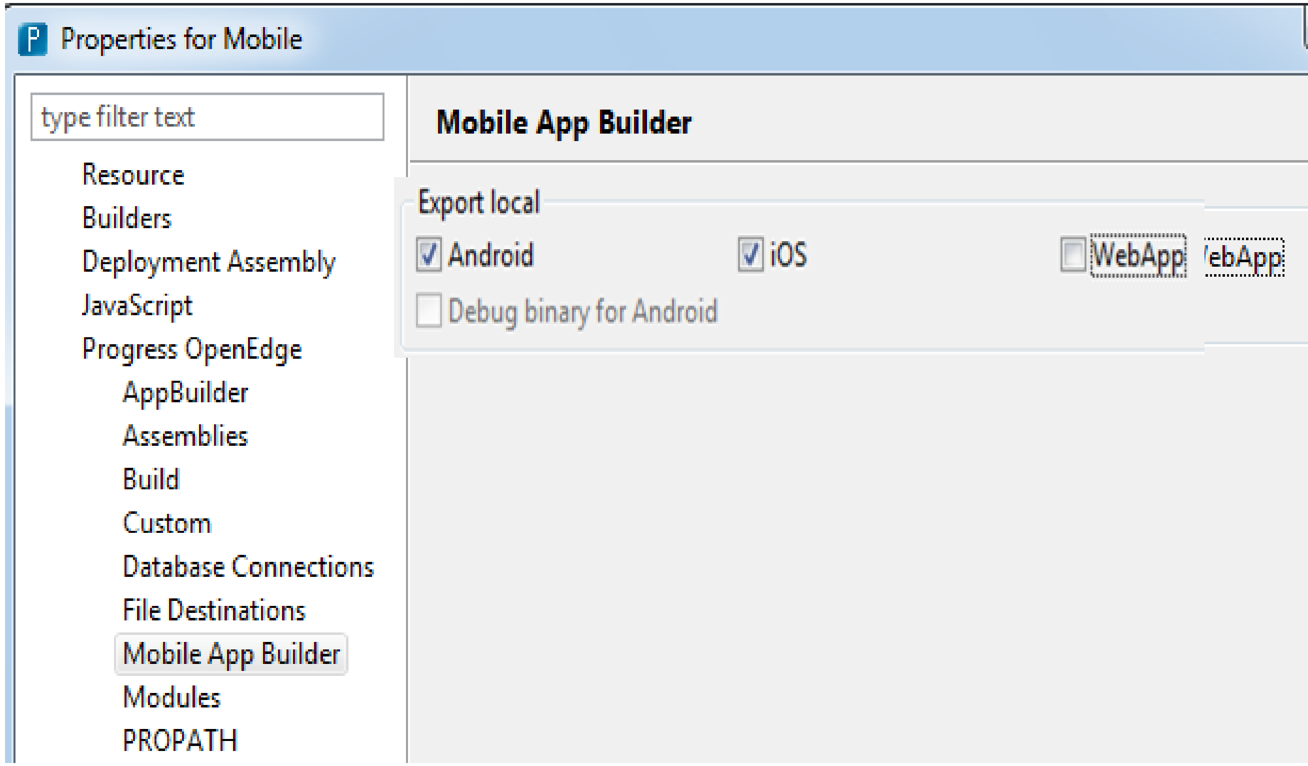
Bind data and UI



Build and Deploy

Mobile App Builder Options in PDS OE

- PDS OE allows user to perform following options on a mobile application
  - Open
  - Run
  - Backup Local
  - Copy Source Local
  - Export Local
  - Delete
  - Rename
- Android (.apk) and iOS (.ipa) files are generated based on the Project Properties settings
- Deleting an application in PDSOE removes from Cloud also.



Mobile App Builder

Export local

☑ Android ☑ iOS ☐ WebApp 'ebApp

☐ Debug binary for Android

Demo

Focus of the Session

Steps involved in developing a Mobile Application

Demo – Mobile App Development

JSDO and Its Importance

Role of REST in OpenEdge Mobile

Architectural Elements

Role of REST in OpenEdge Mobile

REST Protocol

OpenEdge REST Adapter Deployment

REST and Mobile Comparison

Role of REST in Mobile Solution

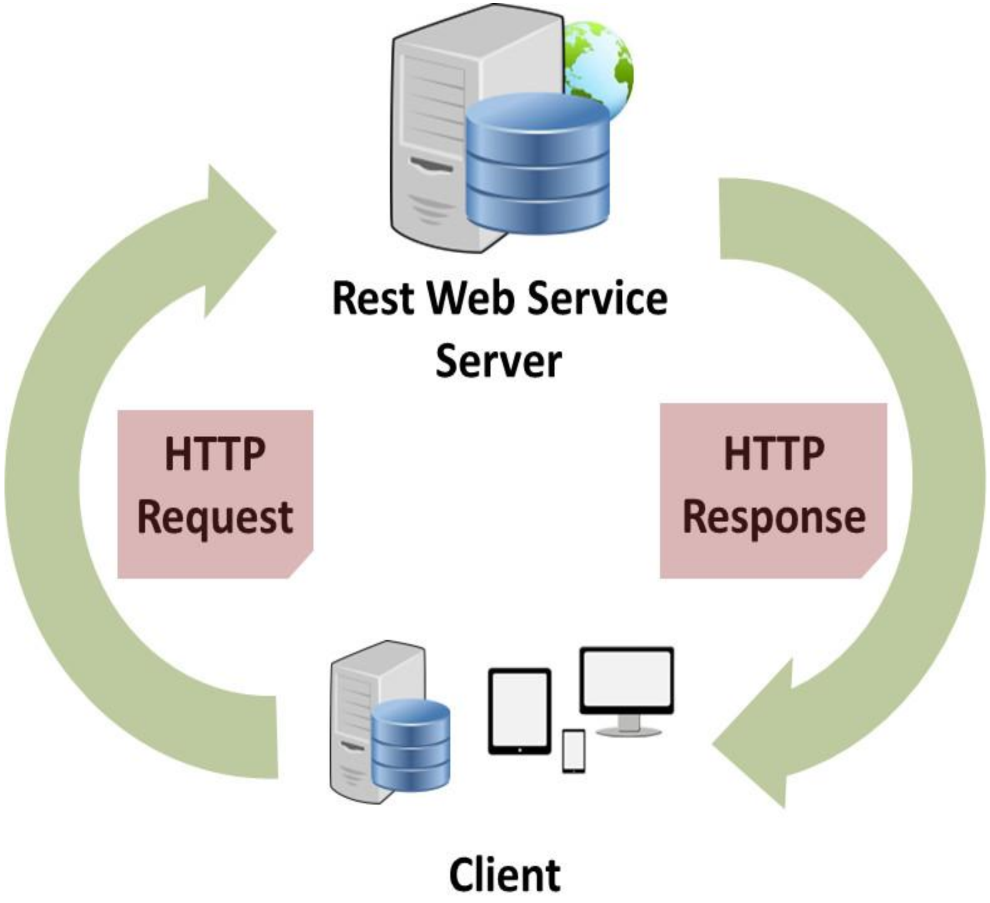
REST Protocol

OpenEdge REST Adapter Deployment

REST and Mobile Comparison

- REST as communication channel for Mobile Solution
- **RE**presentational **S**tate **T**ransfer
- Simpler than SOAP web services
- Client and Server communicate about the state of an object

- REST Verbs:
  - GET : Requests resource
  - PUT : Client posts modification of record
  - POST : Client posts new creation of the record
  - DELETE : Requests deletion of record



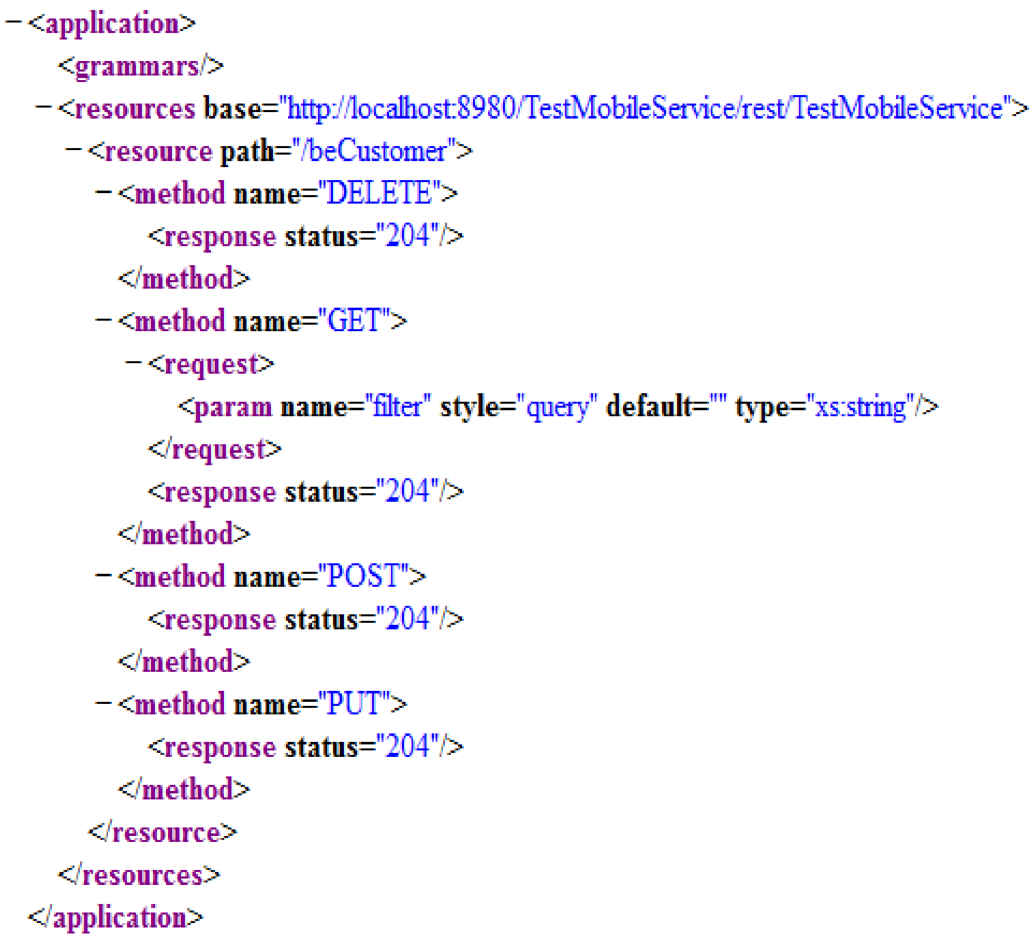
WADL Structure

REST Protocol

OpenEdge REST Adapter Deployment

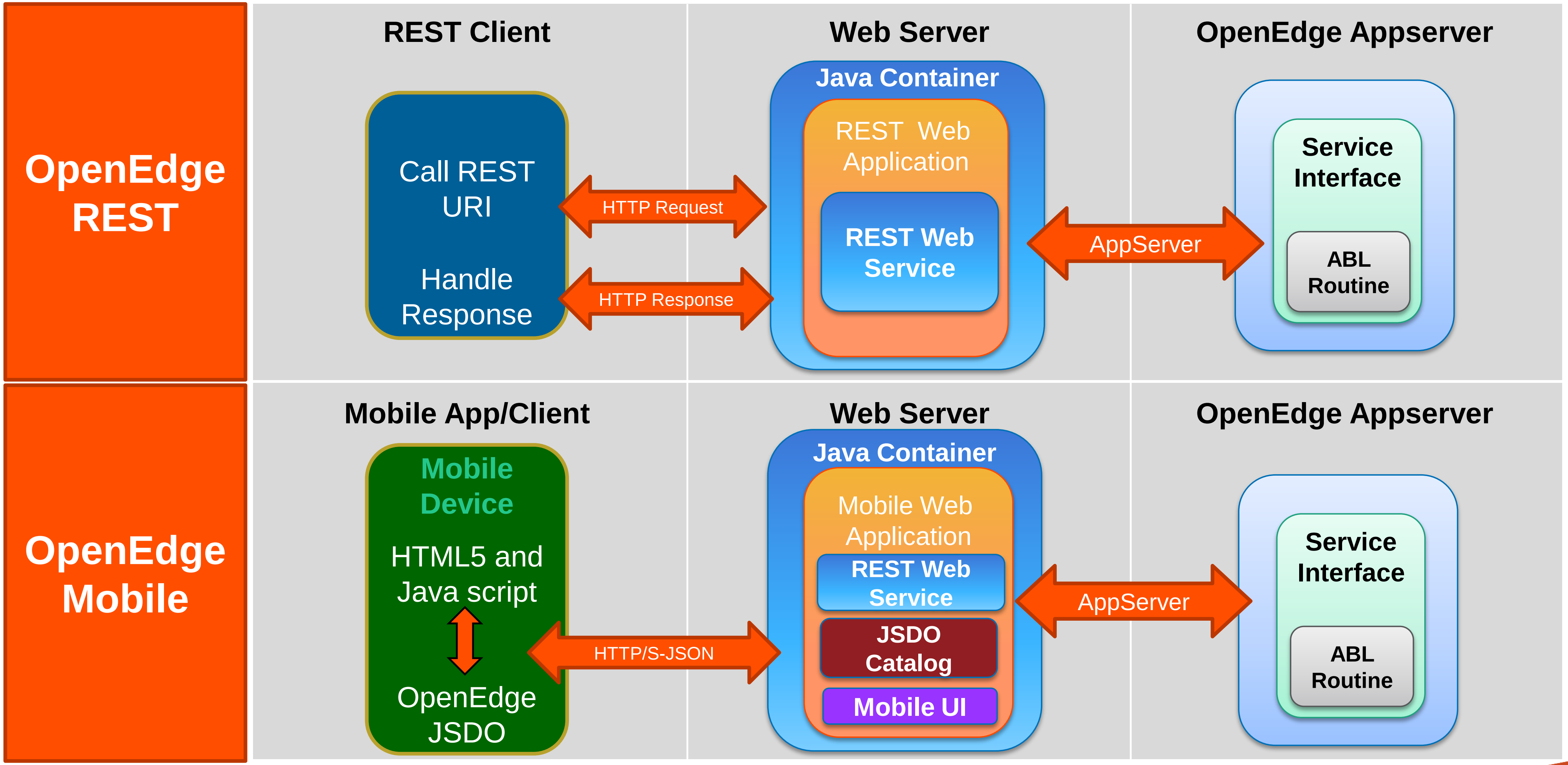
REST and Mobile Comparison

- WADL resembles same as that of WSDL (Web Service Description language) for Web Services
- Provides complete information about the services that are deployed
- Access to list of all resources and the parameter information
- Different tags in the WADL File
  - application
  - resources (base)
  - resource path
  - Method
  - Request
  - response



```
- <application>
    <grammars/>
  - <resources base="http://localhost:8980/TestMobileService/rest/TestMobileService">
    - <resource path="/beCustomer">
      - <method name="DELETE">
          <response status="204"/>
        </method>
      - <method name="GET">
        - <request>
            <param name="filter" style="query" default="" type="xs:string"/>
          </request>
          <response status="204"/>
        </method>
      - <method name="POST">
          <response status="204"/>
        </method>
      - <method name="PUT">
          <response status="204"/>
        </method>
      </resource>
    </resources>
  </application>
```

REST and Mobile Comparison



OpenEdge REST

REST Client

Call REST URI

Handle Response

HTTP Request

HTTP Response

Web Server

Java Container

REST Web Application

REST Web Service

AppServer

OpenEdge Appserver

Service Interface

ABL Routine

OpenEdge Mobile

Mobile App/Client

Mobile Device

HTML5 and Java script

OpenEdge JSDO

HTTP/S-JSON

Web Server

Java Container

Mobile Web Application

REST Web Service

JSDO Catalog

Mobile UI

AppServer

OpenEdge Appserver

Service Interface

ABL Routine

Why JSDO Service but not generic REST Service?

REST Hurdles:

- It is possible to access data on a server prior to OpenEdge 11.2
  - For example using JavaScript XMLHttpRequest( )
  - Very labor intensive to write and maintain low level code

Functionality offered by JSDO:

- Support for tracking changes
  - Transparent access to OpenEdge REST web services
    - Implementation details are hidden
  - Developers do not need to know the URIs
    - Local Storage and Offline transport support
    - Change tracking
- Using generic REST Service for development in Mobile App builder will leave out JSDO features.

Focus of the Session

Steps involved in developing a Mobile Application

Demo – Mobile App Development

Role of REST in OpenEdge Mobile

JSDO and its Importance

Architectural Elements

Architectural Elements

Runtime Architecture of Mobile Web App

Runtime Architecture of Mobile Web App including JSDO

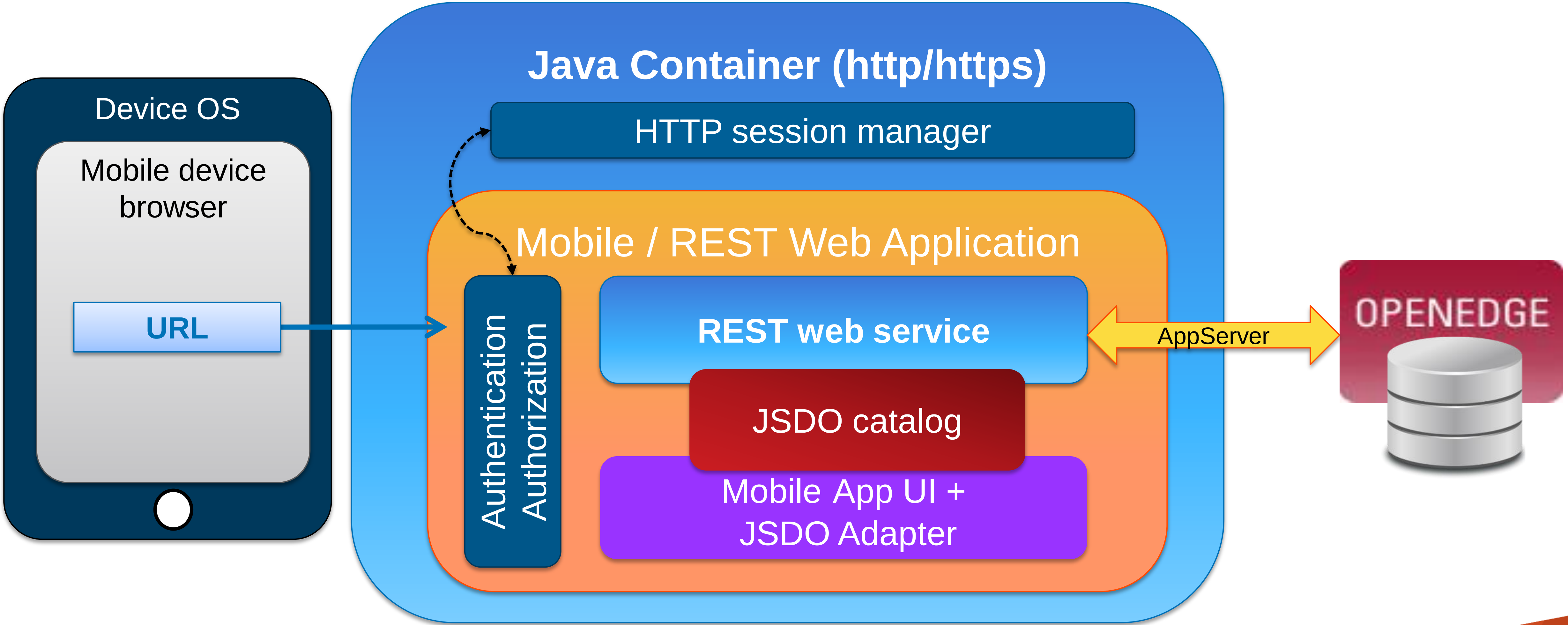
Runtime Architecture of Native Application

Runtime Architectural Elements

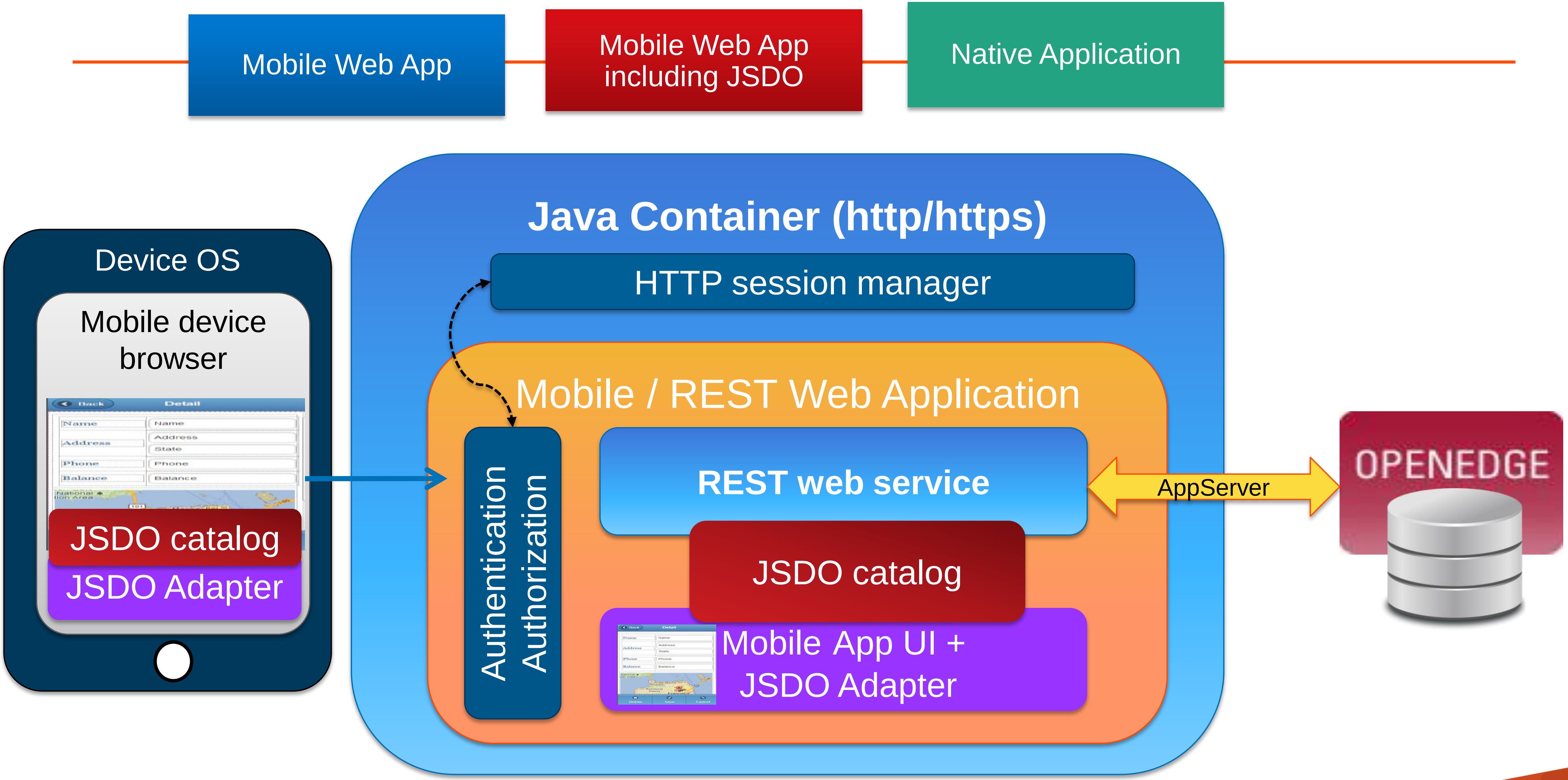
Mobile Web App

Mobile Web App including JSDO

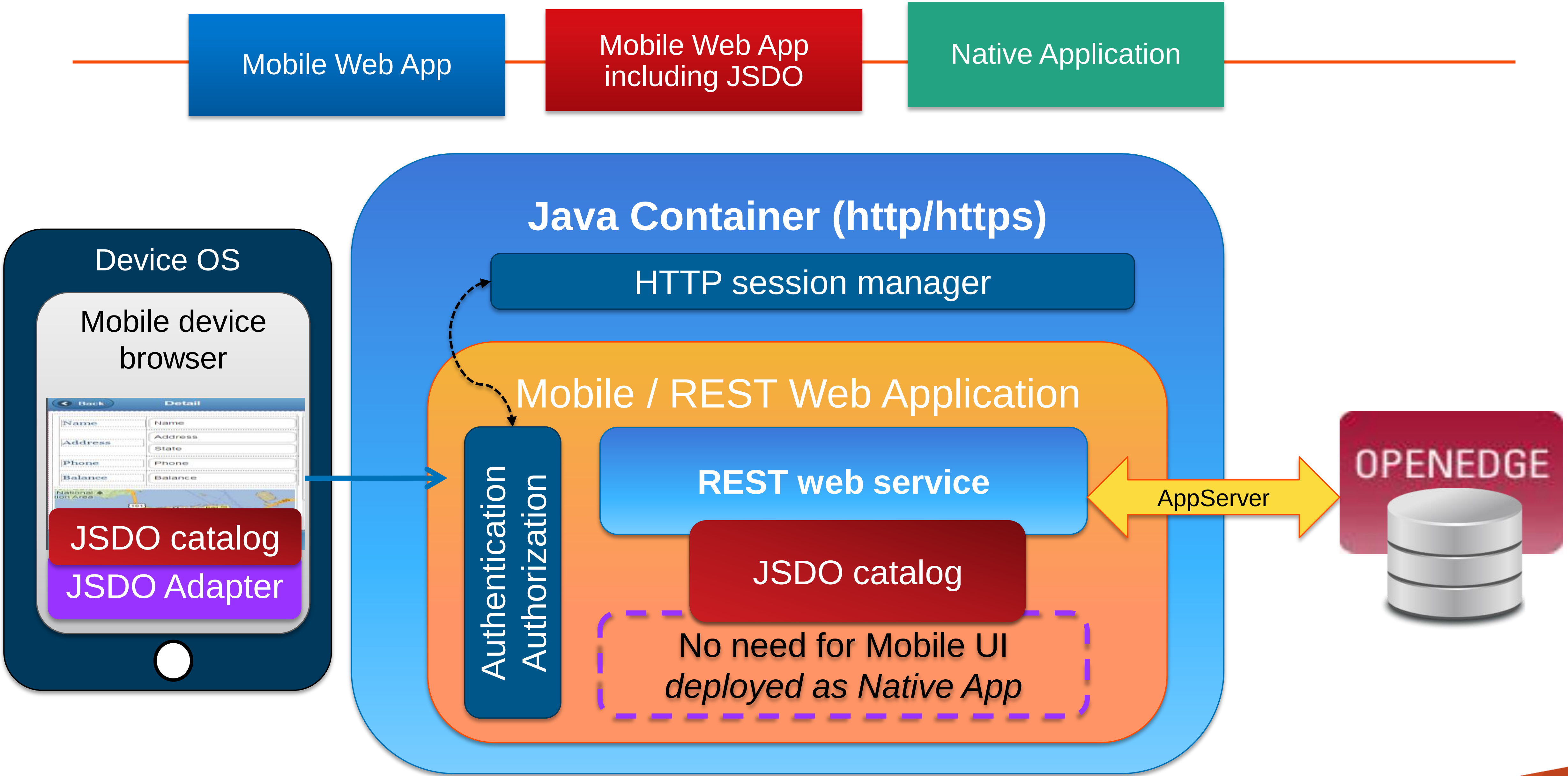
Native Application



Runtime Architectural Elements



Runtime Architectural Elements



Benefits and Why Progress OpenEdge Mobile

- A Complete End to End Solution for the Mobile Development for OpenEdge Applications
- Quick way of providing mobile support for existing ABL Applications
- Rapid Development
- Drag and Drop support over Cloud Platform
- Mapping with respect to Data and UI Elements
- Ease of Publishing/Deployment
- Seamless communication with Appserver data through JSDO

PROGRESS OpenEdge

Mobile

PROGRESS