The Backend of OE Mobile in OpenEdge 11.2

Mike Fechner, Consultingwerk Ltd.

PUG Challenge Americas, June 2013

Mike Fechner, Consultingwerk Ltd.

- Independent IT consulting organization
- Focusing on OpenEdge and .NET
- Located in Cologne, Germany
- Customers in Germany, EMEA, USA
- Vendor of tools and consulting programs (specialized on GUI for .NET, Modernization)
- 23 years of Progress experience (V5 … V11.2)
- Integrated OpenEdge Mobile into SmartComponent Library framework

OpenEdge® application modernization solutions

- WinKit
- **SmartComponent Library**
- Dynamics4.NET

- Tools can be used together or separately
- Shared common code base

- SmartComponents.Mobile
- SmartComponents.Web
- SmartBPMAdapter for OpenEdge BPM/Savvion

Agenda

- OpenEdge Mobile Overview
- Backend Architecture and Components
- JSDO Overview
- “Business Entity” class
- Working with ProDatasets
- Exposing ABL methods and procedures
- PDSOE support views

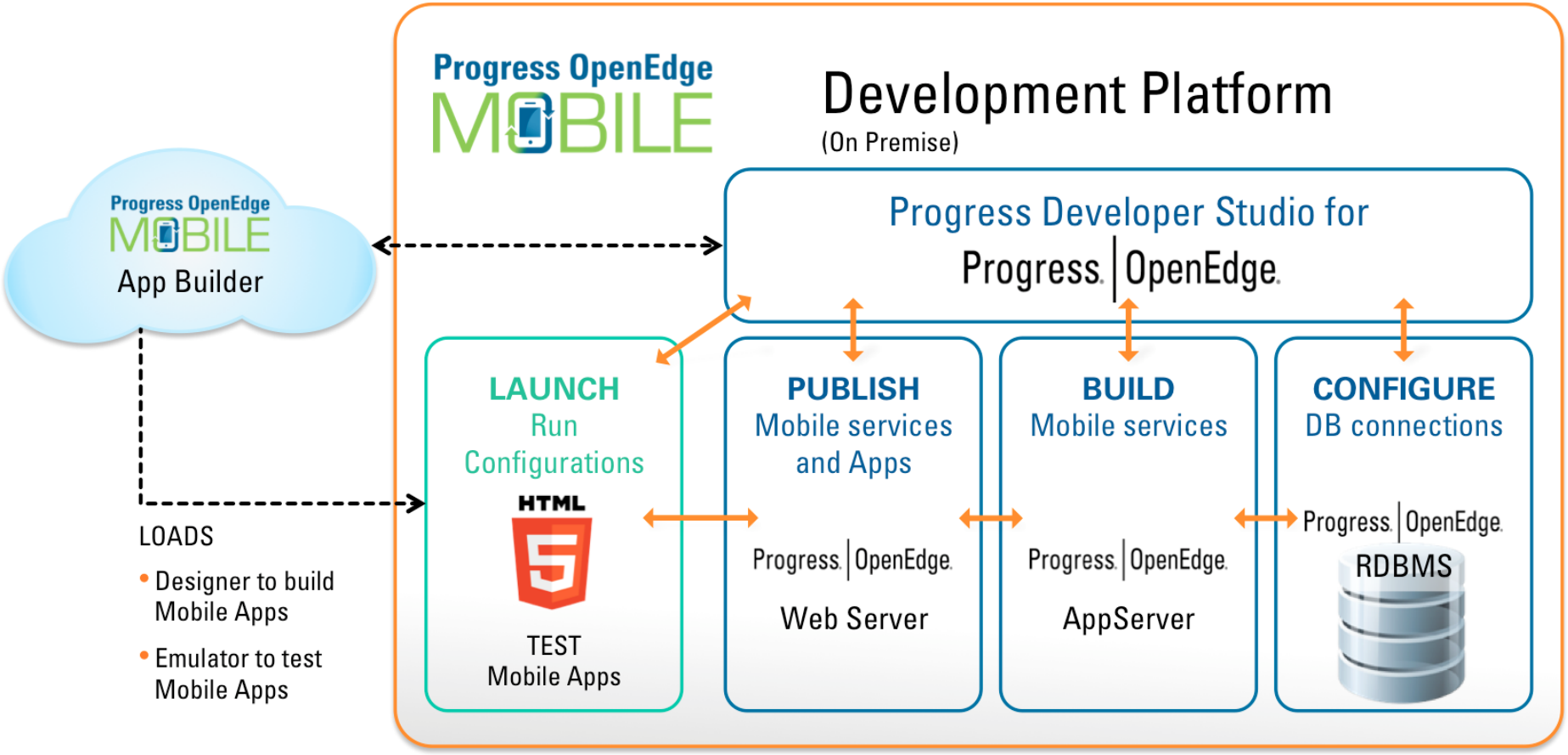
OpenEdge Mobile Overview

Progress OpenEdge MOBILE

- ***The*** new feature of OpenEdge 11.2
- Build hybrid apps for iOS and Android (WebApp or AppStore)
- Cloud based Mobile AppBuilder based on Tiggzi / Appery.io, HTML5 & JavaScript
- Integrated into PDSOE
- Progress AppServer Backend
- Get your 60 day Test-Drive at: http://www.progress.com/docs/gated/downloads/openedge/testdrive-pugs.htm

OpenEdge Mobile Overview

Integrated Development Environment



Mobility Sample App…

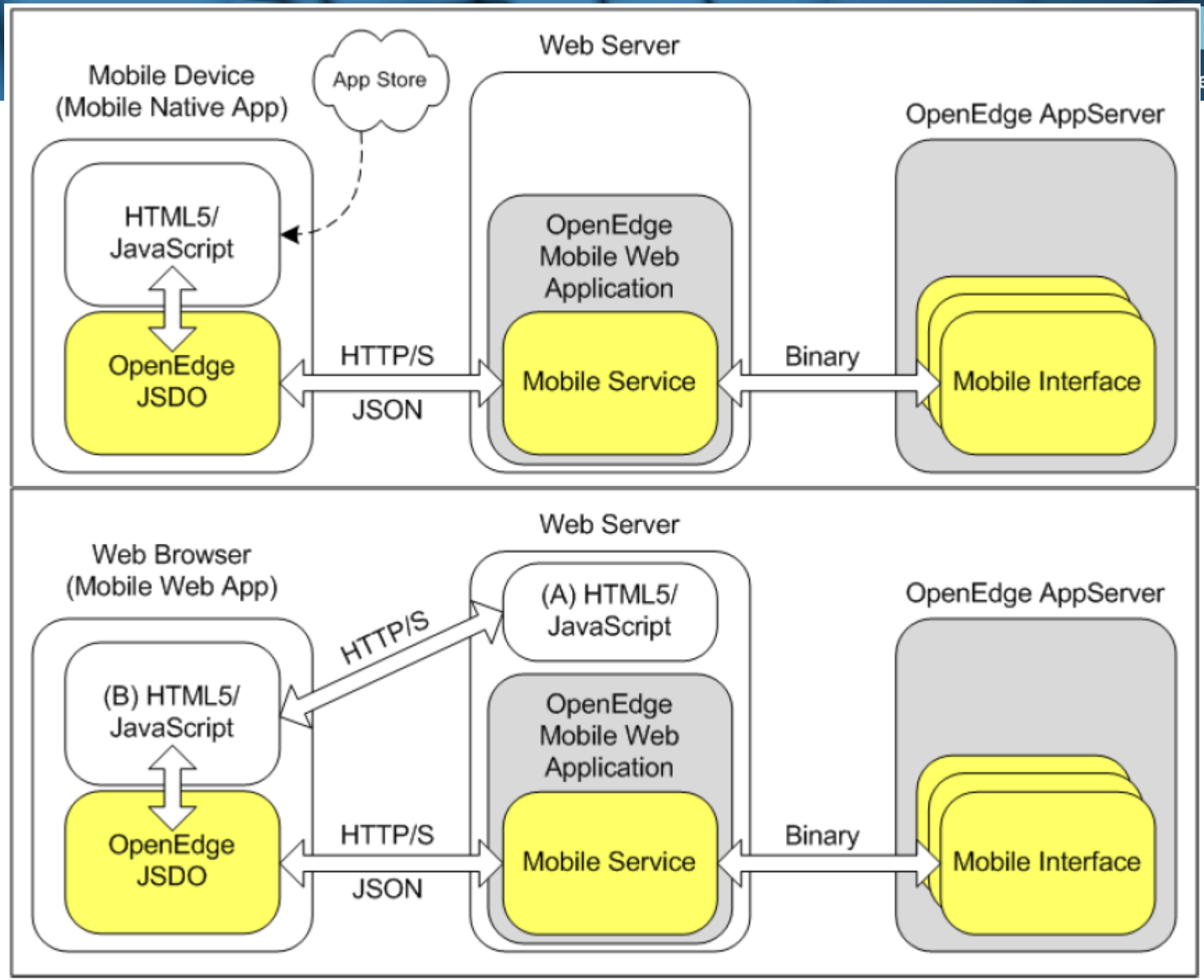
- PSDN post by Shelley Chase
- http://communities.progress.com/pcom/message/170354#170354
- Frontent app project
- Very small AppServer project
- Step-by-step guide

Agenda

- OpenEdge Mobile Overview
- Backend Architecture and Components
- JSDO Overview
- “Business Entity” class
- Parameter passing
- Working with ProDatasets
- Exposing ABL methods and procedures
- PDSOE support views

Backend Architecture

- Progress AppServer is the Backend for retrieving and updating data, execute ABL
- REST as the communication protocol between mobile device and AppServer
- **REST Adapter** executed in Tomcat 7 Java Servlet Container (similar to WSA and AIA)
- “ProxyGen” functionality integrated into PDSOE
- JSON as the data format between mobile device and AppServer
- Build on current standards and protocols


Mobile Device
(Mobile Native App)
App Store
Web Server
OpenEdge AppServer
HTML5/
JavaScript
OpenEdge
Mobile Web
Application
OpenEdge
JSDO
HTTP/S
JSON
Mobile Service
Binary
Mobile Interface
Web Server
Web Browser
(Mobile Web App)
(A) HTML5/
JavaScript
OpenEdge AppServer
HTTP/S
(B) HTML5/
JavaScript
OpenEdge
Mobile Web
Application
OpenEdge
JSDO
HTTP/S
JSON
Mobile Service
Binary
Mobile Interface

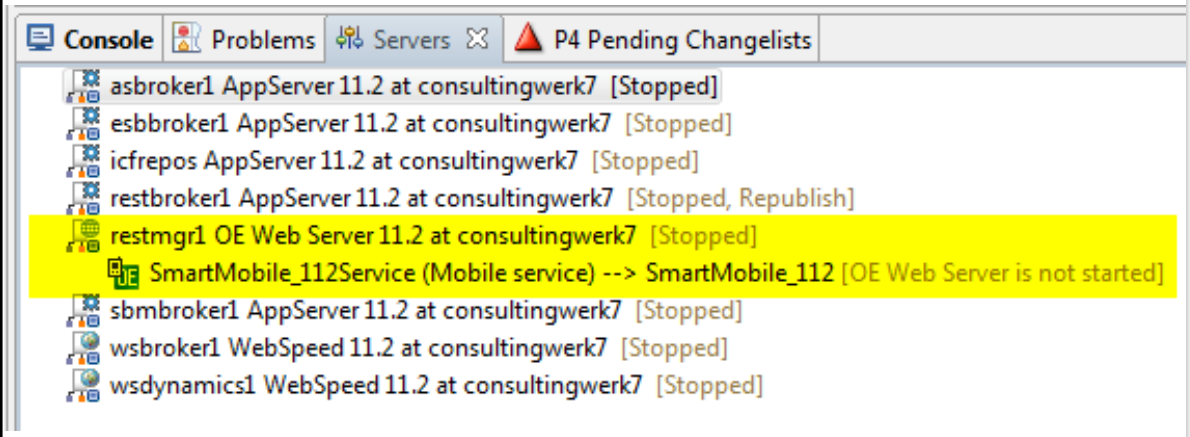
OpenEdge Web Server

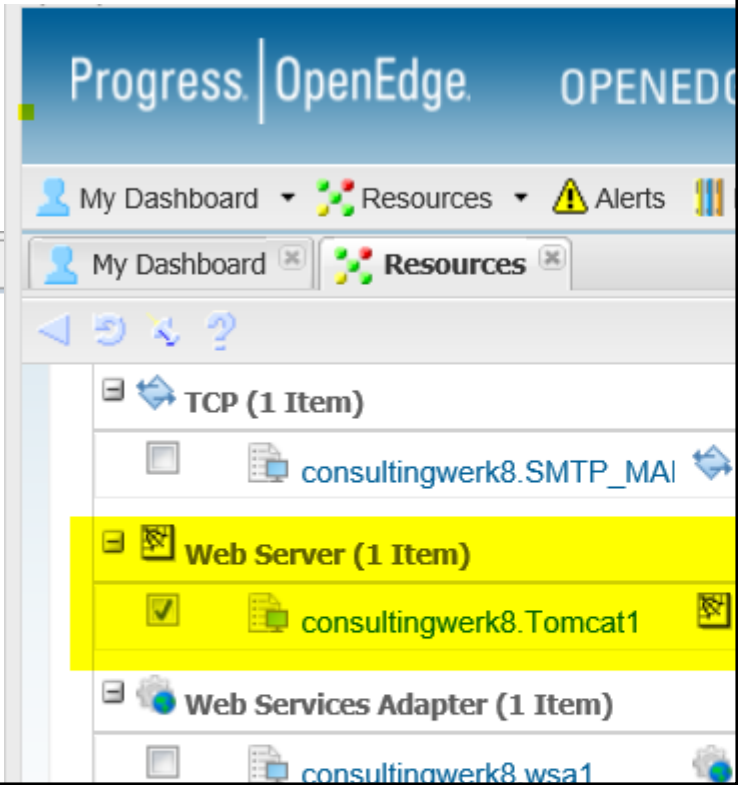
- Tomcat 7, installed with Progress for development
- %DLC%\servers\tomcat
- No need to fiddle around with Tomcat installation anymore

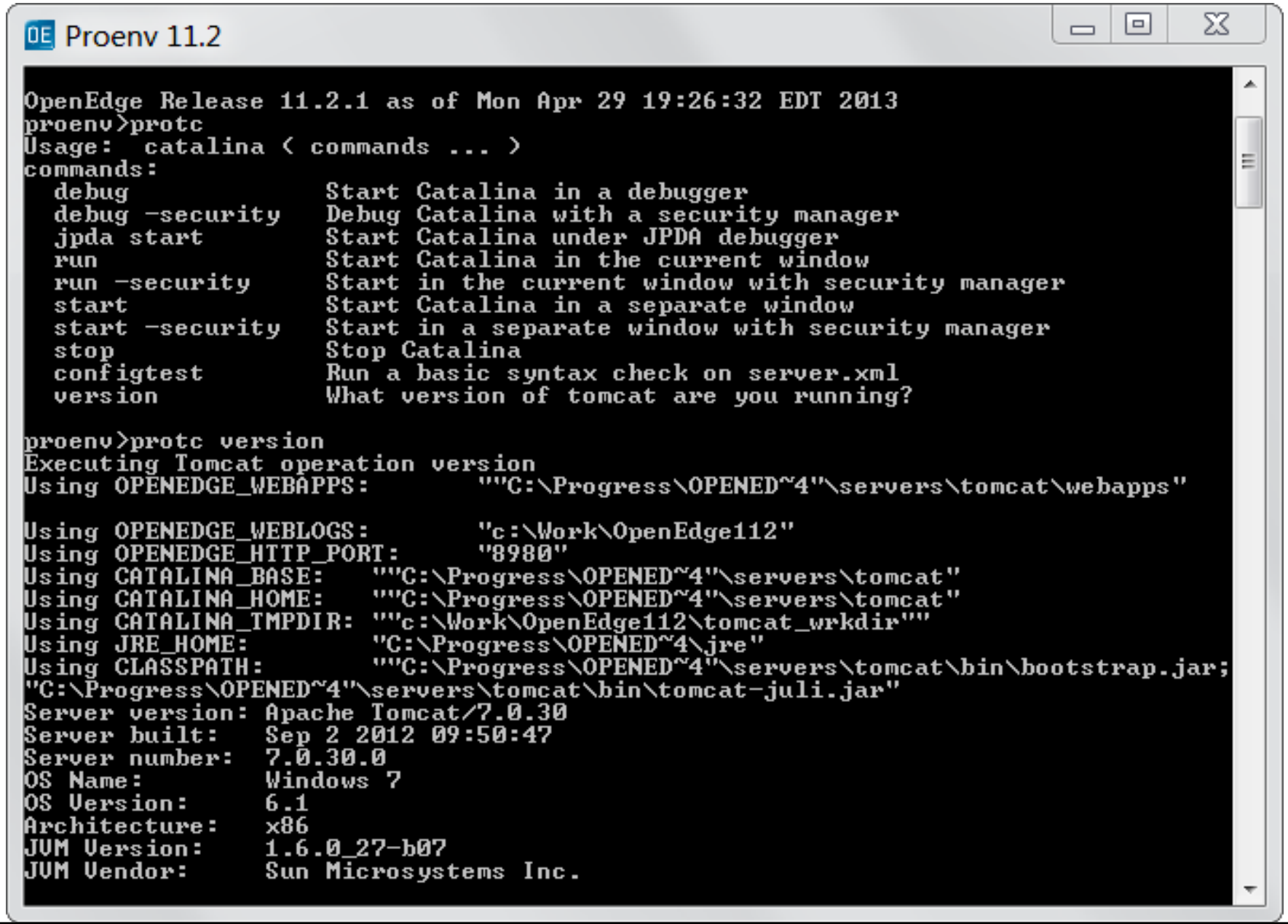
- Default port: 8980
- Hosting of
  - OERM (REST Manager)
  - Mobile App (Web Server for a mobile app)
  - Mobile Service (REST access for an application)

OpenEdge Web Server

- Started from OpenEdge Explorer / OpenEdge Management
- Servers view in Progress Developer Studio
- protc command






Proenv 11.2
OpenEdge Release 11.2.1 as of Mon Apr 29 19:26:32 EDT 2013
proenv>protc
Usage: catalina (commands ...)
commands:
debug Start Catalina in a debugger
debug -security Debug Catalina with a security manager
jpda start Start Catalina under JPDA debugger
run Start Catalina in the current window
run -security Start in the current window with security manager
start Start Catalina in a separate window
start -security Start in a separate window with security manager
stop Stop Catalina
configtest Run a basic syntax check on server.xml
version What version of tomcat are you running?
proenv>protc version
Executing Tomcat operation version
Using OPENEDGE_WEBAPPS: ""C:\Progress\OPENED~4"\servers\tomcat\webapps"
Using OPENEDGE_WEBLOGS: "c:\Work\OpenEdge112"
Using OPENEDGE_HTTP_PORT: "8980"
Using CATALINA_BASE: ""C:\Progress\OPENED~4"\servers\tomcat"
Using CATALINA_HOME: ""C:\Progress\OPENED~4"\servers\tomcat"
Using CATALINA_TMPDIR: ""c:\Work\OpenEdge112\tomcat_wrkdir""
Using JRE_HOME: "C:\Progress\OPENED~4\jre"
Using CLASSPATH: ""C:\Progress\OPENED~4"\servers\tomcat\bin\bootstrap.jar;
"C:\Progress\OPENED~4"\servers\tomcat\bin\tomcat-juli.jar"
Server version: Apache Tomcat/7.0.30
Server built: Sep 2 2012 09:50:47
Server number: 7.0.30.0
OS Name: Windows 7
OS Version: 6.1
Architecture: x86
JVM Version: 1.6.0_27-b07
JVM Vendor: Sun Microsystems Inc.

REST protocol

- **Re**presentional **s**tate **t**ransfer
- W3C standard
- Typically http/1.1 transport
- Simpler then SOAP web services
- Client and Server communicate about the state of an object
- State transitions as the message
- Client may request (GET) using URI
- Client may post using request content
- Twitter API, JIRA API, Jenkins API

REST „verbs“

- Additional http REQUEST_METHOD’s
- Multiple interactions on the same URI
- **GET** – client requests resource (record), should not modify the resource
- **POST** – client posts a new instance of the resource (create a record)
- **PUT** – client posts a modification of a resource (update record)
- **DELETE** – client requests deletion of a resource
- …

Sample

- ABL Client invoking Twitter API

- http://search.twitter.com/search.json?q=progress sw&rpp=50&include_entities=false&result_type=mixed

JSON – JavaScript Object Notation

- The “little brother of XML“
- Originates from JavaScript development
- JavaScript objects can be written to and read from JSON, also supported in other languages
- More lightweight, typically smaller than XML (no need for end tag)
- OE10.2B, support for READ/WRITE-JSON of ProDataset and Temp-Table
- OE11.0, support for JSON ObjectModel Parser
- OE11.2, document format of the REST Adapter
- Mime-Type: application/json

Sample ProDataset JSON output

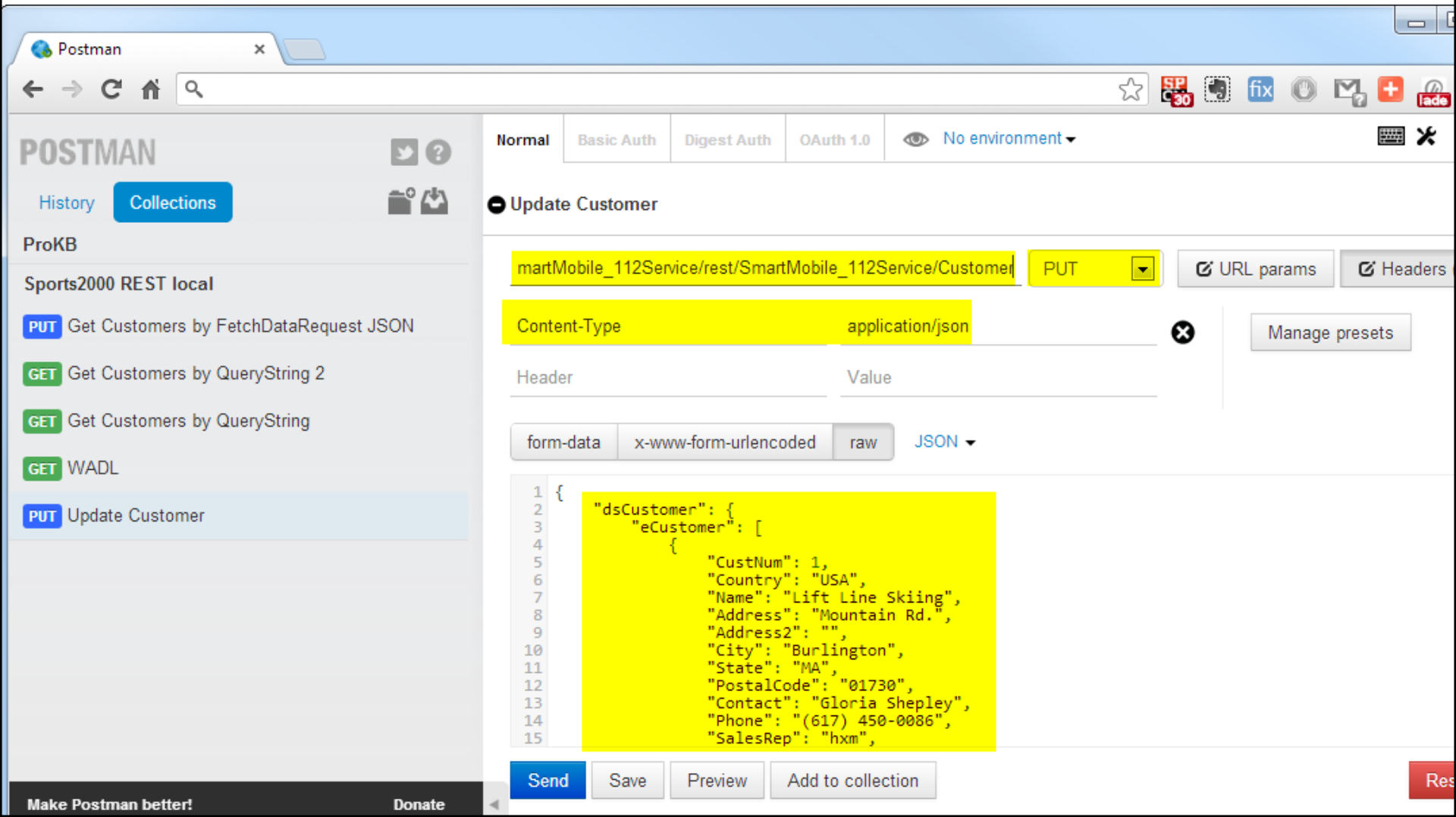
- { } wraps a single object
- [ ] wraps an array of objects
- All strings are quoted
- Data types: Number, String, Boolean, Array, Object, Null
- Everything else must be passed as a String (e.g. Date)


```
{"dsOrder": {
  "eOrder": [
    {
      "Ordernum": 1,
      "CustNum": 53,
      "OrderDate": "2009-01-23",
      "ShipDate": "2009-01-28",
      "PromiseDate": "2009-01-28",
      "Carrier": "FlyByNight Courier",
      "Instructions": "Handle with care",
      "SalesRep": "RDR",
      "OrderStatus": "Shipped",
      "Creditcard": "Master Card",
      "eOrderLine": [
        {
          "Ordernum": 1,
          "Linenum": 1,
          "Itemnum": 54,
          "Price": 4.86,
          "Qty": 30,
          "Discount": 10,
          "ExtendedPrice": 131.22,
          "OrderLineStatus": "Shipped"
        }
      ]
    }
  ]
}}
```

Testing REST Operations

- Several Clients for testing REST applications are available
- Example: **Postman** extension to the Google Chrome browser
  - Build all request parameters
  - Run all REST request methods
  - History of previous requests
  - Store Request for reuse in Collections
  - Export Collections to local files

Demo – Postman for OE AppServer



Agenda

- OpenEdge Mobile Overview
- Backend Architecture and Components
- JSDO Overview
- “Business Entity” class
- Working with ProDatasets
- Exposing ABL methods and procedures
- PDSOE support views

JSDO – JavaScript Data Object

- Provides client side access to the backend resources (“Business Entity”)
- A JavaScript class provided by Progress
- progress.data.JSDO class
- A single instance provides access to a single OpenEdge resource (class, procedure)

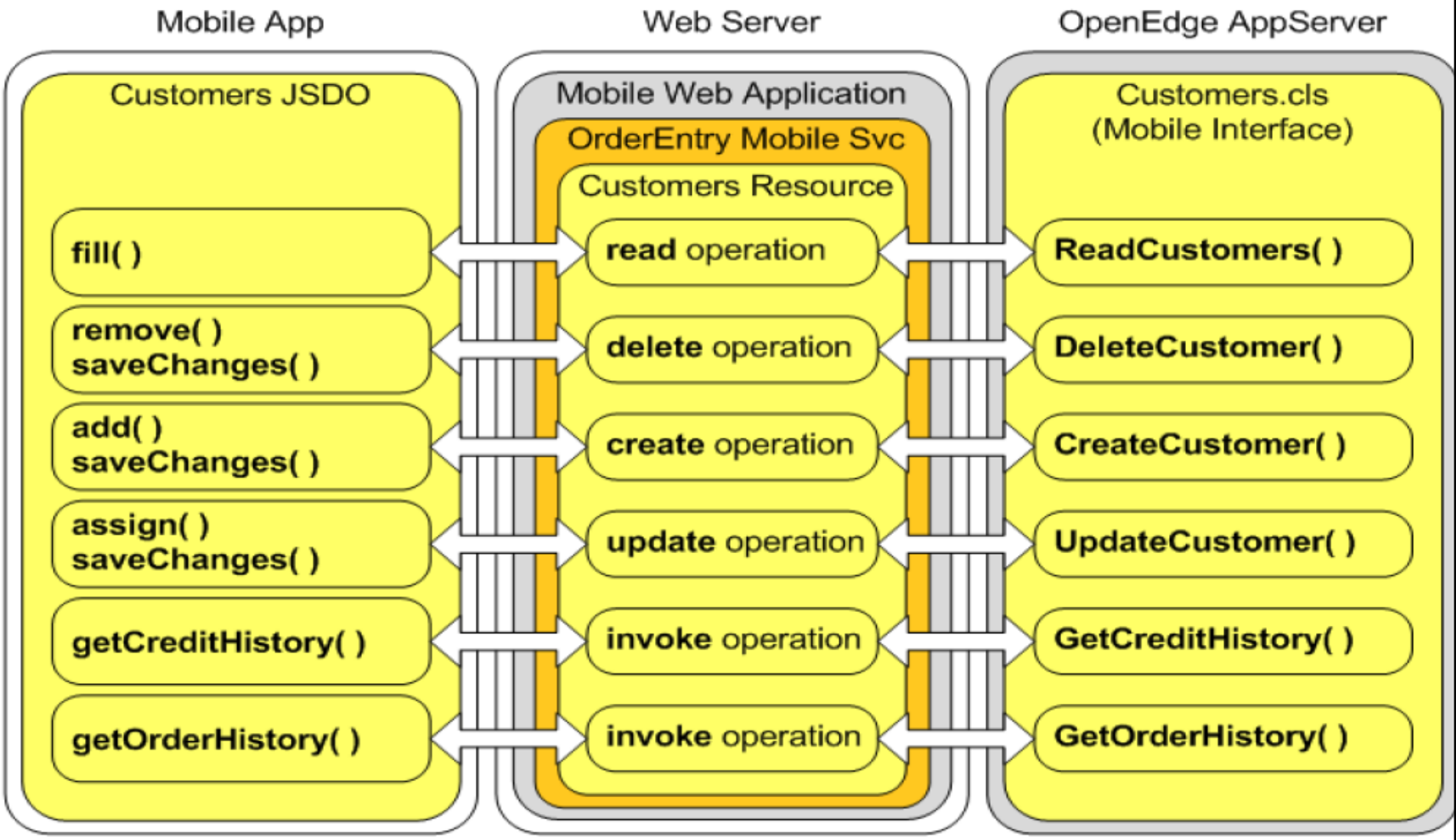
Catalog file

- The JSDO needs the catalog during initialization
- The catalog provides information
  - about the schema of the Temp-Tables and ProDatasets exposed by the mobile service
  - about the methods exposed by the mobile service resources and their parameters
- Automatically generated by PDSOE
- Required when defining JSDO service in the mobile AppBuilder

Catalog location

- catalogURI: http://<server>:8980/MyMobileService/static/mobile/MyMobileService.json
- Resource in PDSOE project: <projectName>/WebContent/MyMobileService.json
- JSON File itself
- **Hint:** verify contents of catalog file after changing exposed methods and data structures to make sure the catalog is up to date!

```
{
    "version": "1.0",
    "lastModified": "Sat Jun 08 08:09:03 CEST 2013",
    "services": [{
        "name": "SmartMobile_112Service",
        "address": "\/rest\/SmartMobile_112Service",
        "useRequest": true,
        "resources": [
            {
                "name": "Customer",
                "path": "\/Customer",
                "schema": {
                    "type": "object",
                    "additionalProperties": false,
                    "properties": {"dsCustomer": {
                        "type": "object",
                        "additionalProperties": false,
                        "properties": {"eCustomer": {
                            "type": "array",
                            "items": {
                                "additionalProperties": false,
                                "properties": {
                                    "_id": {"type": "string"},
                                    "CustNum": {
                                        "type": "integer",
                                        "default": 0,
                                        "title": "Cust Num"
                                    },
                                    "Country": {
                                        "type": "string",
                                        "default": "USA",
                                        "title": "Country"
                                    },
                                    "Name": {
                                        "type": "string",
                                        "default": "",
                                        "title": "Name"
                                    },
                                    "Address": {
                                        "type": "string",
                                        "default": "",
                                        "title": "Address"
```


Mobile App
Web Server
OpenEdge AppServer
Customers JSDO
Mobile Web Application
OrderEntry Mobile Svc
Customers Resource
Customers.cls
(Mobile Interface)
fill( )
remove( )
saveChanges( )
add( )
saveChanges( )
assign( )
saveChanges( )
getCreditHistory( )
getOrderHistory( )
read operation
delete operation
create operation
update operation
invoke operation
invoke operation
ReadCustomers( )
DeleteCustomer( )
CreateCustomer( )
UpdateCustomer( )
GetCreditHistory( )
GetOrderHistory( )

JSDO – „Business Entity“ mapping

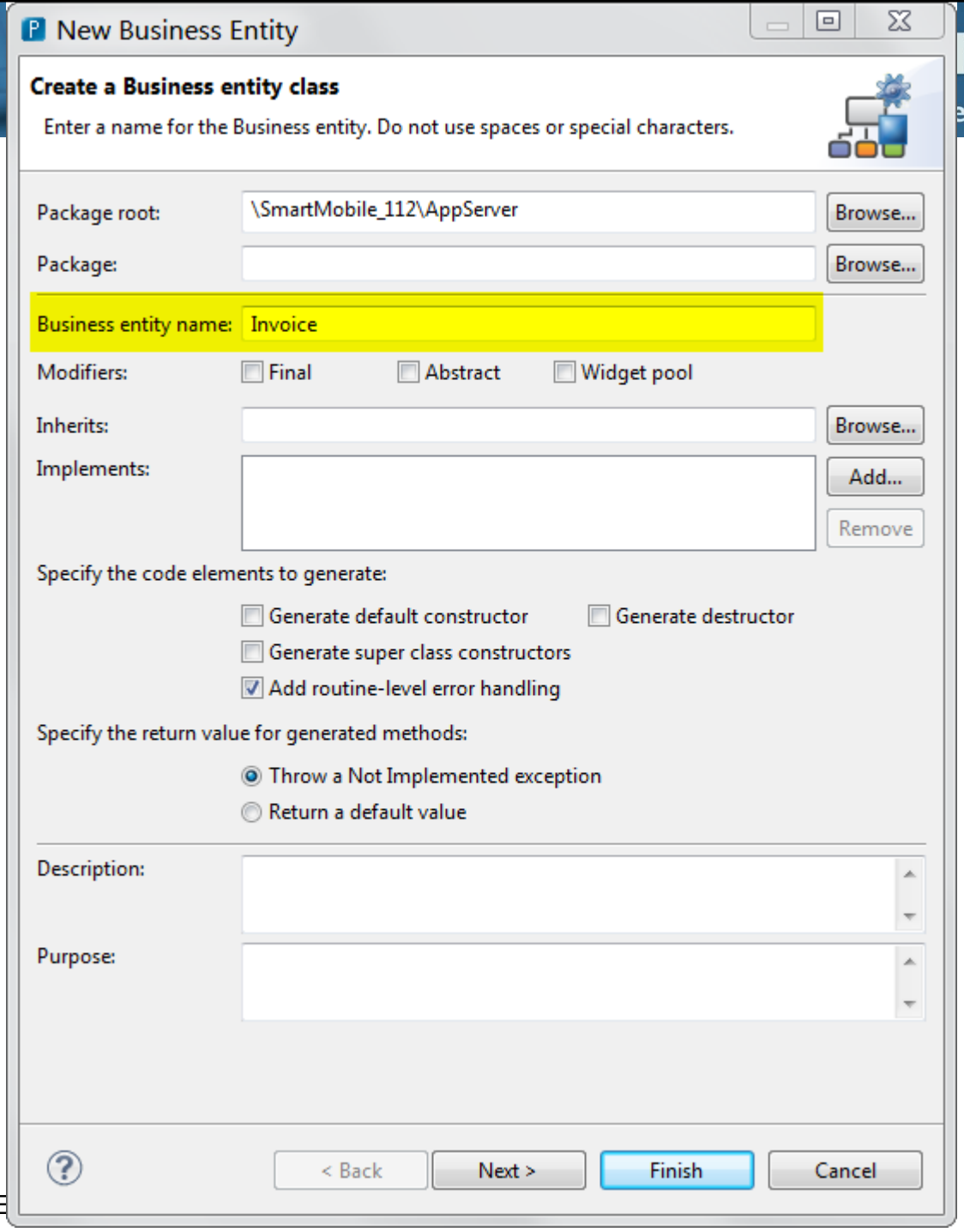
- The ABL Resource may support standard CRUD operations
  - fill (filter AS CHARTER, OUTPUT dataset)
  - create (INPUT-OUTPUT dataset)
  - update (INPUT-OUTPUT dataset)
  - delete (INPUT-OUTPUT dataset)
- Signature of standard CRUD methods is fixed
- Other methods may be exposed with custom signature

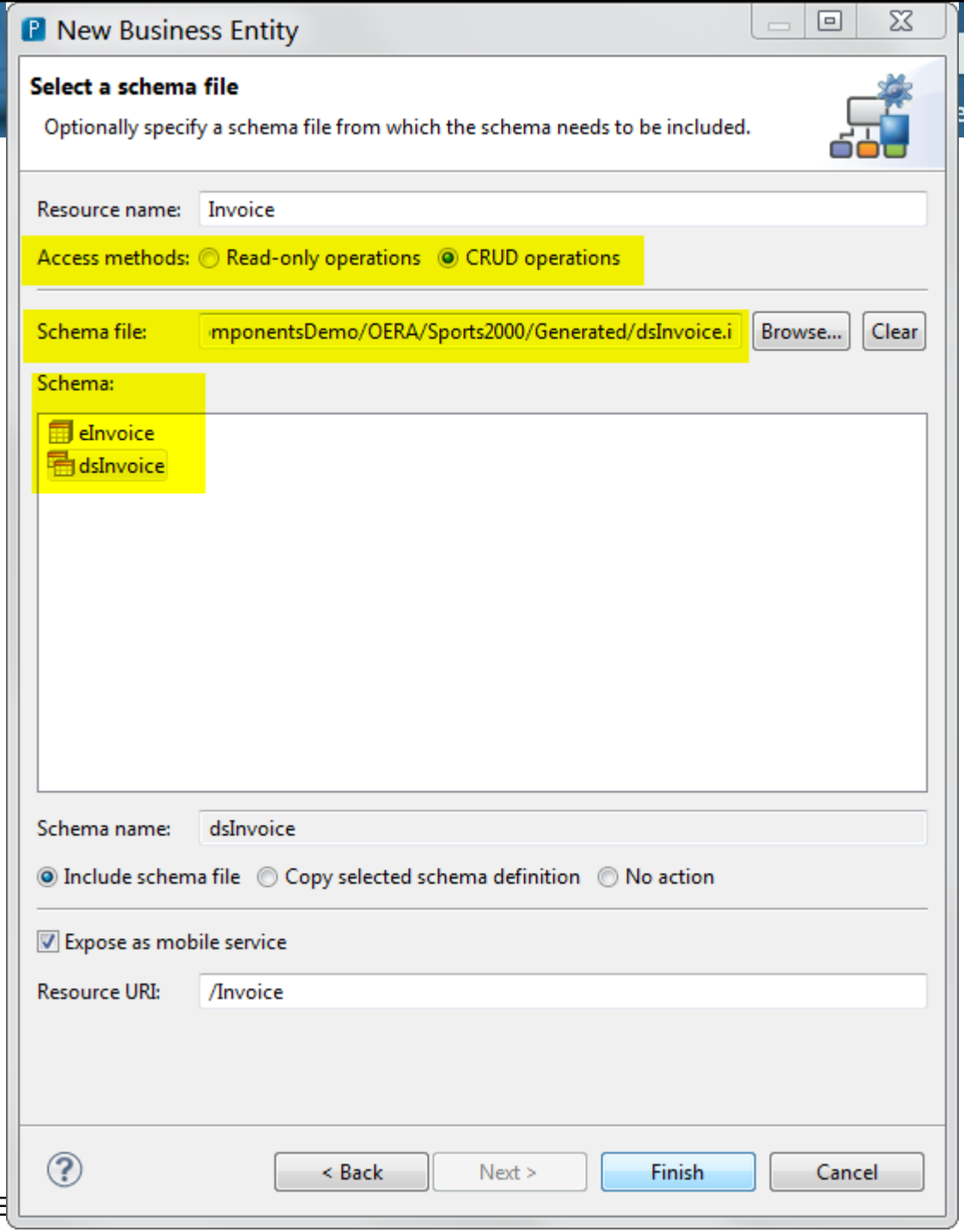
Agenda

- OpenEdge Mobile Overview
- Backend Architecture and Components
- JSDO Overview
- “Business Entity” class
- Working with ProDatasets
- Exposing ABL methods and procedures
- PDSOE support views

„Business Entity“ class

- PDSOE Wizard for a class with full CRUD support
- New Singleton run mode
- Term “Business Entity” is more than confusing as in proper OERA terms this is supposed to be part of the **Service Interface**
- I prefer to talk of an **Business Entity (Service) Interface** that provides access to an Business Entity that must not be specific to OE Mobile / The REST Adapter (used also in GUI app)


New Business Entity
Create a Business entity class
Enter a name for the Business entity. Do not use spaces or special characters.
Package root:
\SmartMobile_112\AppServer
Browse...
Package:
Browse...
Business entity name:
Invoice
Modifiers:
Final
Abstract
Widget pool
Inherits:
Browse...
Implements:
Add...
Remove
Specify the code elements to generate:
Generate default constructor
Generate destructor
Generate super class constructors
Add routine-level error handling
Specify the return value for generated methods:
Throw a Not Implemented exception
Return a default value
Description:
Purpose:
< Back
Next >
Finish
Cancel


New Business Entity
Select a schema file
Optionally specify a schema file from which the schema needs to be included.
Resource name:
Invoice
Access methods:
Read-only operations
CRUD operations
Schema file:
mponentsDemo/OERA/Sports2000/Generated/dsInvoice.i
Browse...
Clear
Schema:
eInvoice
dsInvoice
Schema name:
dsInvoice
Include schema file
Copy selected schema definition
No action
Expose as mobile service
Resource URI:
/Invoice
< Back
Next >
Finish
Cancel

Demo

- Create „Business Entity“ class
- Review generated code

„Business Entity“ methods

- Standard methods need to be VOID
- All standard CRUD methods need to use the same data structure (same Temp-Table or same ProDataset)
- Filter parameter of read method MUST be called “filter”
- Usage/purpose of the filter parameter is up to us
- Custom methods may use other Datasets or Temp-Tables

„Business Entity“ methods

- Dataset Relations may not be NESTED
- Methods may throw error to the client

Critical issue in OpenEdge 11.2 and 11.2.1

- When ProDataset Temp-Table fields are defined with XML attributes, the Temp-Table will not be parsed properly
- Typically leads to Temp-Tables with only a single field (parser seems to stop after first field)
- May require generating a “clean” duplicate of the Temp-Table definition, if the same schema is also required for XML output ☹
- **Can be verified in the catalog file**!

This Temp-Table definition will fail

```
DEFINE {&ACCESS} TEMP-TABLE eCustomer NO-UNDO {&REFERENCE-ONLY} &IF DEFINED (NO-BEFORE) EQ 0 &THEN BEFORE-TABLE eCustom
    FIELD CustNum AS INTEGER FORMAT ">>>>9":U INIT "0":U LABEL "Cust Num":T SERIALIZE-NAME "CustNum":U XML-DATA-TYPE "
    FIELD Country AS CHARACTER FORMAT "x(20)":U INIT "USA":U LABEL "Country":T SERIALIZE-NAME "Country":U XML-DATA-TYPE
    FIELD Name AS CHARACTER FORMAT "x(30)":U LABEL "Name":T SERIALIZE-NAME "Name":U XML-DATA-TYPE "string":U XML-NODE-T
    FIELD Address AS CHARACTER FORMAT "x(35)":U LABEL "Address":T SERIALIZE-NAME "Address":U XML-DATA-TYPE "string":U X
    FIELD Address2 AS CHARACTER FORMAT "x(35)":U LABEL "Address2":T SERIALIZE-NAME "Address2":U XML-DATA-TYPE "string"
    FIELD City AS CHARACTER FORMAT "x(25)":U LABEL "City":T SERIALIZE-NAME "City":U XML-DATA-TYPE "string":U XML-NODE-T
    FIELD State AS CHARACTER FORMAT "x(20)":U LABEL "State":T SERIALIZE-NAME "State":U XML-DATA-TYPE "string":U XML-NOD
    FIELD PostalCode AS CHARACTER FORMAT "x(10)":U LABEL "Postal Code":T SERIALIZE-NAME "PostalCode":U XML-DATA-TYPE "s
    FIELD Contact AS CHARACTER FORMAT "x(30)":U LABEL "Contact":T SERIALIZE-NAME "Contact":U XML-DATA-TYPE "string":U X
    FIELD Phone AS CHARACTER FORMAT "x(20)":U LABEL "Phone":T SERIALIZE-NAME "Phone":U XML-DATA-TYPE "string":U XML-NOD
    FIELD SalesRep AS CHARACTER FORMAT "x(4)":U LABEL "Sales Rep":T SERIALIZE-NAME "SalesRep":U XML-DATA-TYPE "string"
    FIELD CreditLimit AS DECIMAL FORMAT "->,>>>,>>9":U INIT "1500":U LABEL "Credit Limit":T SERIALIZE-NAME "CreditLimit
    FIELD Balance AS DECIMAL FORMAT "->,>>>,>>9.99":U INIT "0":U LABEL "Balance":T SERIALIZE-NAME "Balance":U XML-DATA-
    FIELD Terms AS CHARACTER FORMAT "x(20)":U INIT "Net30":U LABEL "Terms":T SERIALIZE-NAME "Terms":U XML-DATA-TYPE "st
    FIELD Discount AS INTEGER FORMAT ">>9%":U INIT "0":U LABEL "Discount":T SERIALIZE-NAME "Discount":U XML-DATA-TYPE "
    FIELD Comments AS CHARACTER FORMAT "x(80)":U LABEL "Comments":T SERIALIZE-NAME "Comments":U XML-DATA-TYPE "string"
    FIELD Fax AS CHARACTER FORMAT "x(20)":U LABEL "Fax":T SERIALIZE-NAME "Fax":U XML-DATA-TYPE "string":U XML-NODE-TYPE
    FIELD EmailAddress AS CHARACTER FORMAT "x(50)":U LABEL "Email":T SERIALIZE-NAME "EmailAddress":U XML-DATA-TYPE "str
    FIELD Flags AS CHARACTER FORMAT "X(8)":U LABEL "Notes":T
    FIELD SmartRecordKey AS CHARACTER FORMAT "X(80)":U
    FIELD SmartAttachments AS LOGICAL FORMAT "yes/no":U INIT "no":U LABEL "Attachments":T
    FIELD SmartComments AS LOGICAL FORMAT "yes/no":U INIT "no":U LABEL "Comments":T

    INDEX Comments AS WORD-INDEX Comments ASCENDING
    INDEX CountryPost Country ASCENDING PostalCode ASCENDING
    INDEX CustNum AS UNIQUE PRIMARY CustNum ASCENDING
    INDEX Name Name ASCENDING
    INDEX SalesRep SalesRep ASCENDING
```

This Temp-Table definition will succeed

```
DEFINE {&ACCESS} TEMP-TABLE eCustomer NO-UNDO {&REFERENCE-ONLY} &IF DEFINED (NO-BEFORE) EQ 0 &THEN BEFORE-TABLE eCustom
    FIELD CustNum AS INTEGER FORMAT ">>>>9":U INIT "0":U LABEL "Cust Num":T
    FIELD Country AS CHARACTER FORMAT "x(20)":U INIT "USA":U LABEL "Country":T
    FIELD Name AS CHARACTER FORMAT "x(30)":U LABEL "Name":T
    FIELD Address AS CHARACTER FORMAT "x(35)":U LABEL "Address":T
    FIELD Address2 AS CHARACTER FORMAT "x(35)":U LABEL "Address2":T
    FIELD City AS CHARACTER FORMAT "x(25)":U LABEL "City":T
    FIELD State AS CHARACTER FORMAT "x(20)":U LABEL "State":T
    FIELD PostalCode AS CHARACTER FORMAT "x(10)":U LABEL "Postal Code":T
    FIELD Contact AS CHARACTER FORMAT "x(30)":U LABEL "Contact":T
    FIELD Phone AS CHARACTER FORMAT "x(20)":U LABEL "Phone":T
    FIELD SalesRep AS CHARACTER FORMAT "x(4)":U LABEL "Sales Rep":T
    FIELD CreditLimit AS DECIMAL FORMAT "->,>>>,>>9":U INIT "1500":U LABEL "Credit Limit":T
    FIELD Balance AS DECIMAL FORMAT "->,>>>,>>9.99":U INIT "0":U LABEL "Balance":T
    FIELD Terms AS CHARACTER FORMAT "x(20)":U INIT "Net30":U LABEL "Terms":T
    FIELD Discount AS INTEGER FORMAT ">>9%":U INIT "0":U LABEL "Discount":T
    FIELD Comments AS CHARACTER FORMAT "x(80)":U LABEL "Comments":T
    FIELD Fax AS CHARACTER FORMAT "x(20)":U LABEL "Fax":T
    FIELD EmailAddress AS CHARACTER FORMAT "x(50)":U LABEL "Email":T

    INDEX Comments AS WORD-INDEX Comments ASCENDING
    INDEX CountryPost Country ASCENDING PostalCode ASCENDING
    INDEX CustNum AS UNIQUE PRIMARY CustNum ASCENDING
    INDEX Name Name ASCENDING
    INDEX SalesRep SalesRep ASCENDING

    .
```

Agenda

- OpenEdge Mobile Overview
- Backend Architecture and Components
- JSDO Overview
- “Business Entity” class
- Working with ProDatasets
- Exposing ABL methods and procedures
- PDSOE support views

Working with ProDatasets and Temp-Tables

- The JSDO supports passing a ProDataset (Temp-Table) between client and AppServer
  - fill: OUTPUT
  - create: INPUT-OUTPUT
  - update: INPUT-OUTPUT
  - delete: INPUT-OUTPUT
- Currently no support for ProDataset Before-Table and ROW-STATE
- This may be a serious limitation if you plan to reuse existing Business Entities!

JSDO fill / Read method

- INPUT cFilter AS CHARACTER
- A single CHARACTER parameter, may be passed for filtering, batching, …, we are free to use it as needed
- No additional parameters (context) may be returned back by the method
- All context information (like last row num or ROWID for batching) needs to be part of the ProDataset

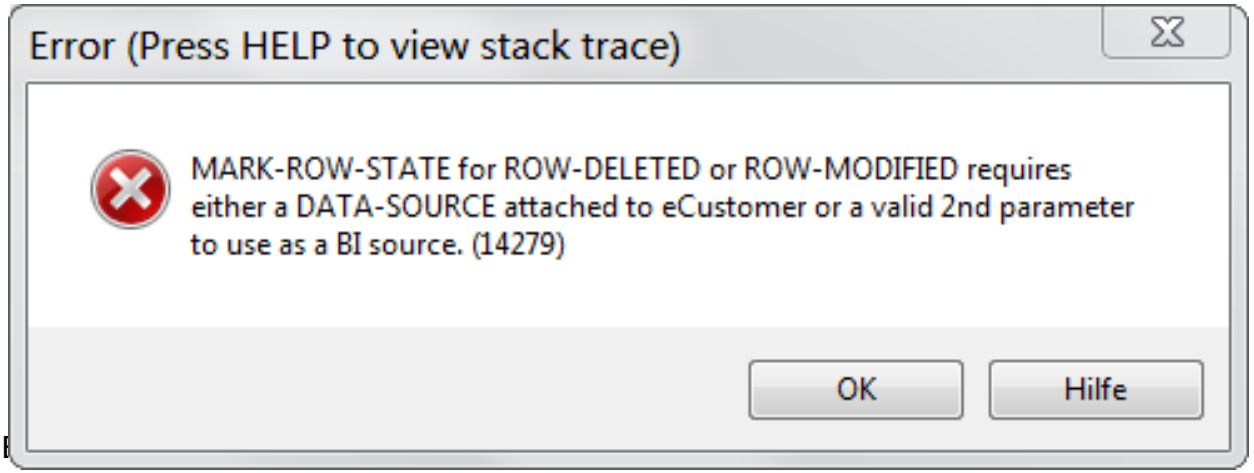
JSDO create / Create Method

- INPUT-OUTPUT Parameter for the ProDataset of Temp-Table
- No BEFORE-TABLE information is provided
- The MARK-ROW-STATE method of the ProDataset Buffer may be used to create BEFORE-TABLE with ROW-CREATED state

```
FOR EACH eCustomer:
    BUFFER eCustomer:MARK-ROW-STATE (ROW-CREATED) .
END.
```

JSDO assign / Update Method

- INPUT-OUTPUT Parameter for the ProDataset of Temp-Table
- No BEFORE-TABLE information is provided
- The MARK-ROW-STATE method of the ProDataset Buffer may be used to create BEFORE-TABLE with ROW-MODIFIED state



ProDataset Update without before records

- We need to either attach a DATA-SOURCE to the Temp-Table (but in my OERA understanding the BE Interface should not know the DATA-SOURCE)
- Or provide another Buffer to be used as the source for the before record
- It's similar to classic ABL: You need to read a record before you can modify it!
- In OERA: Use the same Business Entity to fetch the record from the Data-Source (into a dynamic ProDataset)

```
FOR EACH eCustomer:
    /* Build a request object to fetch the record from the Business Entity
       using the unique key information */
    oFetchDataRequest =
        NEW FetchDataRequest
            ("eCustomer":U,
             SUBSTITUTE ("FOR EACH eCustomer WHERE eCustomer.CustNum = &1",
                         eCustomer.CustNum),
             1) .

    /* Perform the read Request for the original record
       into a dynamic dataset */
    ServiceInterface:FetchData ({&BusinessEntityName},
                                oFetchDataRequest,
                                OUTPUT DATASET-HANDLE hReferenceDataset) .

    hReferenceDataset::eCustomer:FIND-FIRST () .

    /* Set the eCustomer record (from OE Mobile) to
       the modified ROW-STATE) */
    BUFFER eCustomer:MARK-ROW-STATE (ROW-MODIFIED, hReferenceDataset::eCustomer:HANDLE) .

    FINALLY:
        DELETE OBJECT hReferenceDataset .
    END FINALLY.
END.

ServiceInterface:SaveChanges ({&BusinessEntityName},
                              INPUT-OUTPUT DATASET dsCustomer) .
```

Demo

- Use Postman to update customer record in sports2000 database
- Review output in AppServer logfile
- Compare with ABL GUI for .NET application

JSDO remove / Delete Method

- INPUT-OUTPUT Parameter for the ProDataset of Temp-Table
- No BEFORE-TABLE information is provided
- The MARK-ROW-STATE method of the ProDataset Buffer may be used to create BEFORE-TABLE with ROW-DELETED state
- Same as with ROW-MODIFIED/Update method

Alternative solutions…

- Don‘t rely on the BEFORE-TABLE in your Data-Access (update) logic
  - Update database without DATA-SOURCE
  - Don‘t use the SAVE-ROW-CHANGES
  - Loss of a lot of flexibility provided by ProDataset
  - May loose code-reuse with other clients
- Store “real” before image when records are read in a session store on the AppServer
  - OpenEdge Database
  - File System

Agenda

- OpenEdge Mobile Overview
- Backend Architecture and Components
- JSDO Overview
- “Business Entity” class
- Working with ProDatasets
- Exposing ABL methods and procedures
- PDSOE support views

Exposing ABL methods and procedures

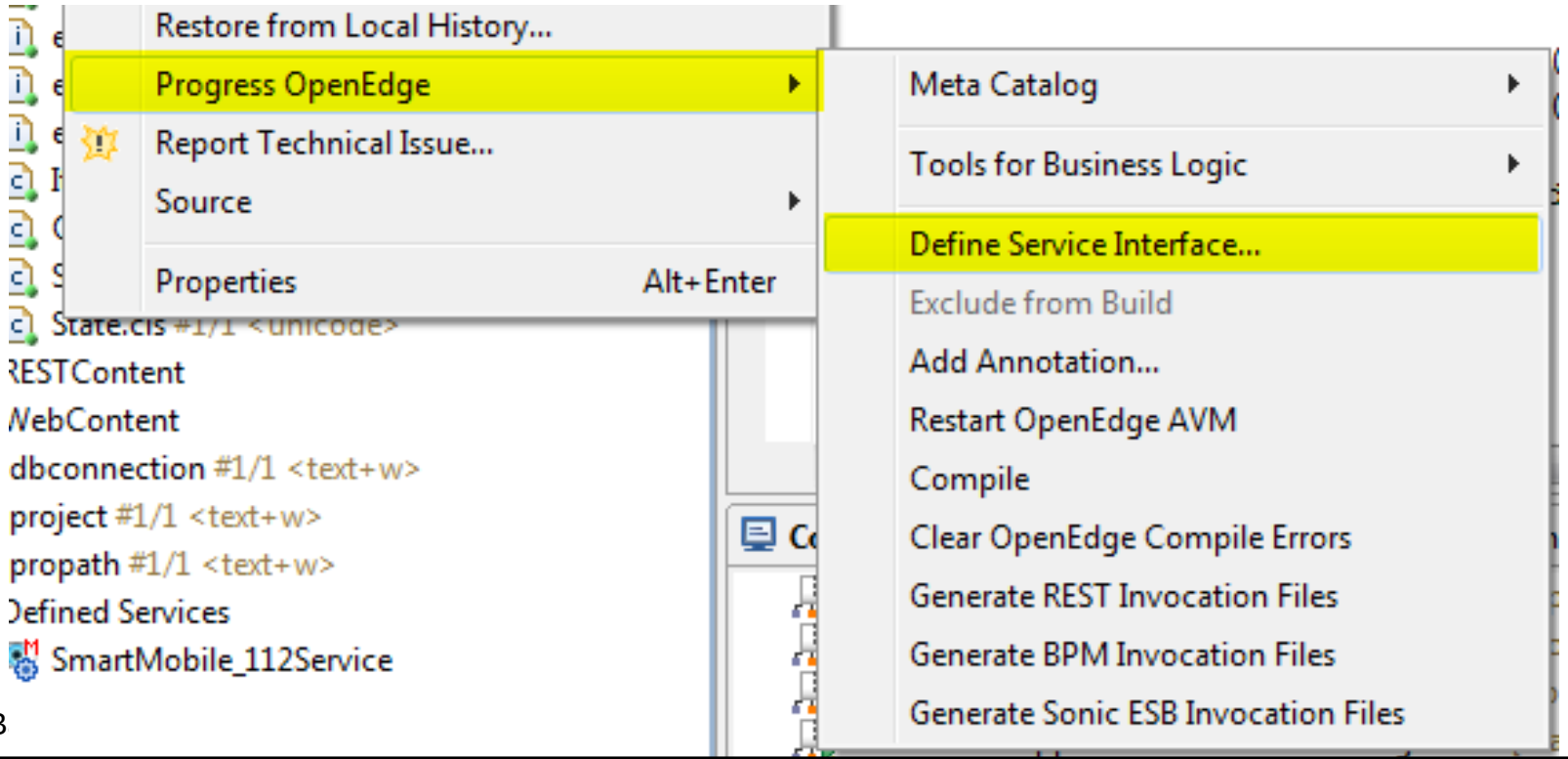
- The REST Adapter can be used to expose “any” ABL class or procedure to REST clients and OE Mobile
- The REST Adapter is currently the only client to the AppServer that can call into classes directly! (need more of that)
- Definition is based on @Annotations ().
- Annotations are like technical comments in the source code
- Annotations make the ProxyGen obsolete!

Singleton run

- New run mode for classes and procedures
- Currently only supported by REST Adapter
- Allows exposing of class methods and internal procedures without the overhead of state-less bound
- No need to instantiate, run procedure, destroy
- Class or procedure instantiated automatically at first use
- Cannot be unloaded… without AppServer agent restart

Defining the Service Interface

- Add the necessary annotations to a procedure or class use the **context menu** of the **PDSOE project**:



Consultingwerk

Define Service Interface

Edit Annotation

Edit annotations for selected files.

Select a file

SmartMobile_112/AppServer/Test/mat

Annotation details for the selected file:

Main annotation | CRUD annotations | Invoke annotations

Enter annotation details for the selected file

Invoke	Routine name	Alias	Return value
☑	divide (integer, inte...		☐
☑	plus (integer, intege...		☐

Mobile detail annotation:

```
@openapi.openedge.export(type="REST", useReturnValue="false", writeDataSetBeforeImage="false").
@progress.service.resourceMapping(type="REST", operation="", URI="", alias="", mediaType="application/json").
```

< Back | Next > | Finish | Cancel

Demo

- Create a custom .p file and add @Annotations. using the “Define Service Interface” dialog
- Add new procedure to “Service”
- Review math.p internal procedures
- Test in Postman

- http://localhost:8980/SmartMobile_112Service/rest/SmartMobile_112Service/?_wadl

Defining the Service Interface

- Be careful when using the dialog with existing REST resources
- The tool may overwrite @Annotations. without warning
- May have severe impact on “Business Entity”
- May be used only for new methods

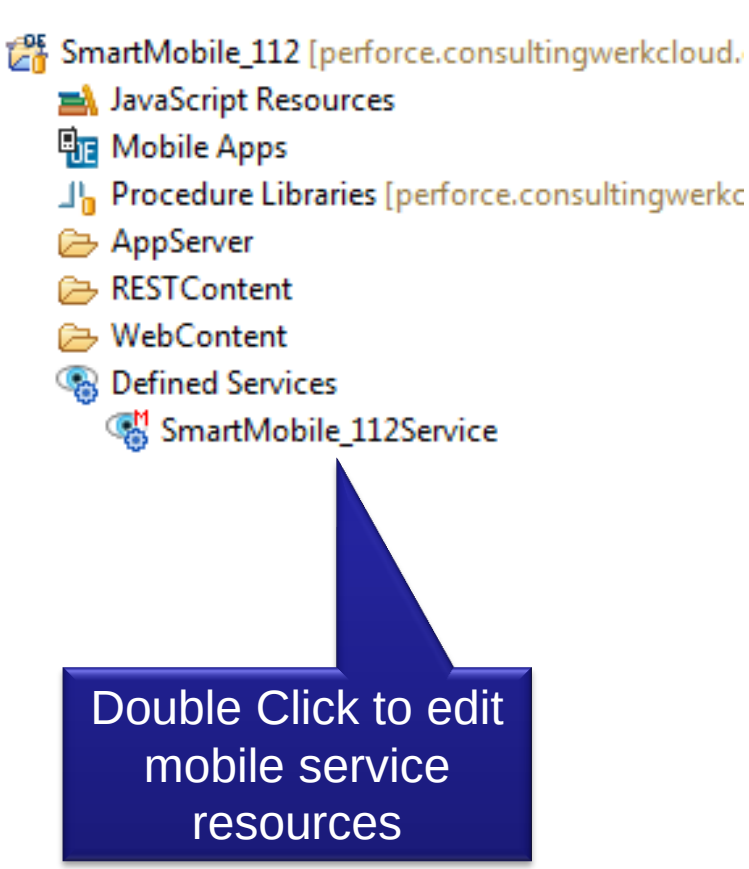
Agenda

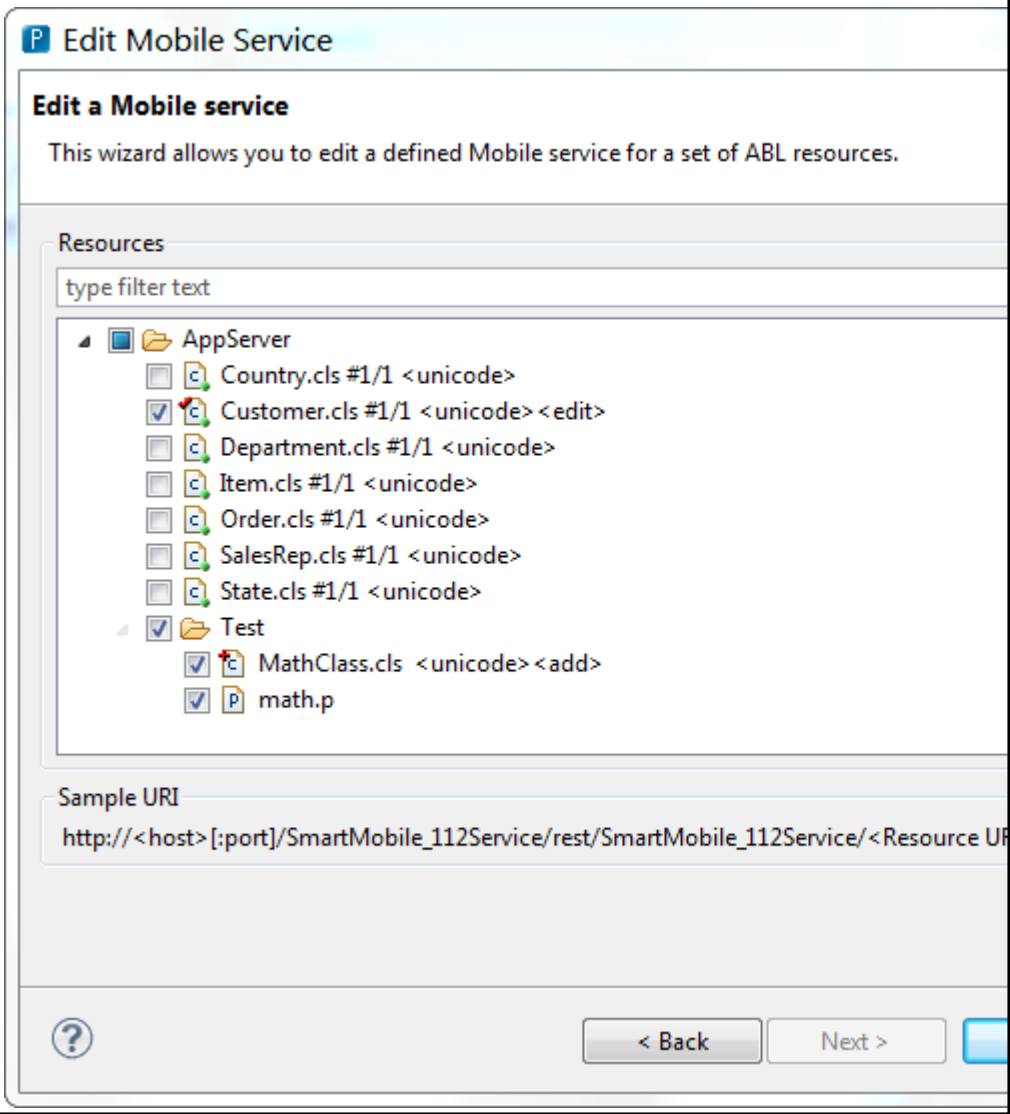
- OpenEdge Mobile Overview
- Backend Architecture and Components
- JSDO Overview
- “Business Entity” class
- Working with ProDatasets
- Exposing ABL methods and procedures
- PDSOE support views

PDSOE Support Views

- Key functionality to expose ABL logic and interact with the REST Adapter is available from PDSOE views
  - Resource View, access to context menu Tools
    - Defined Services Node
  - Servers View
    - Server Editor, AppServer Run Configuration

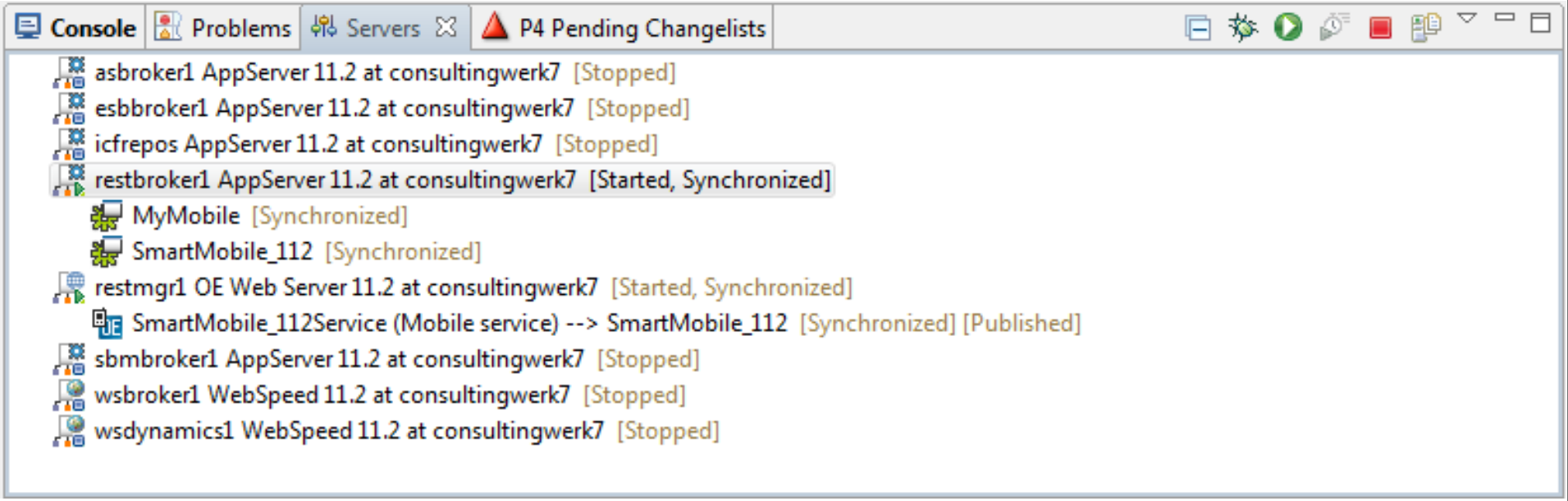
Defined Services in the Resource View





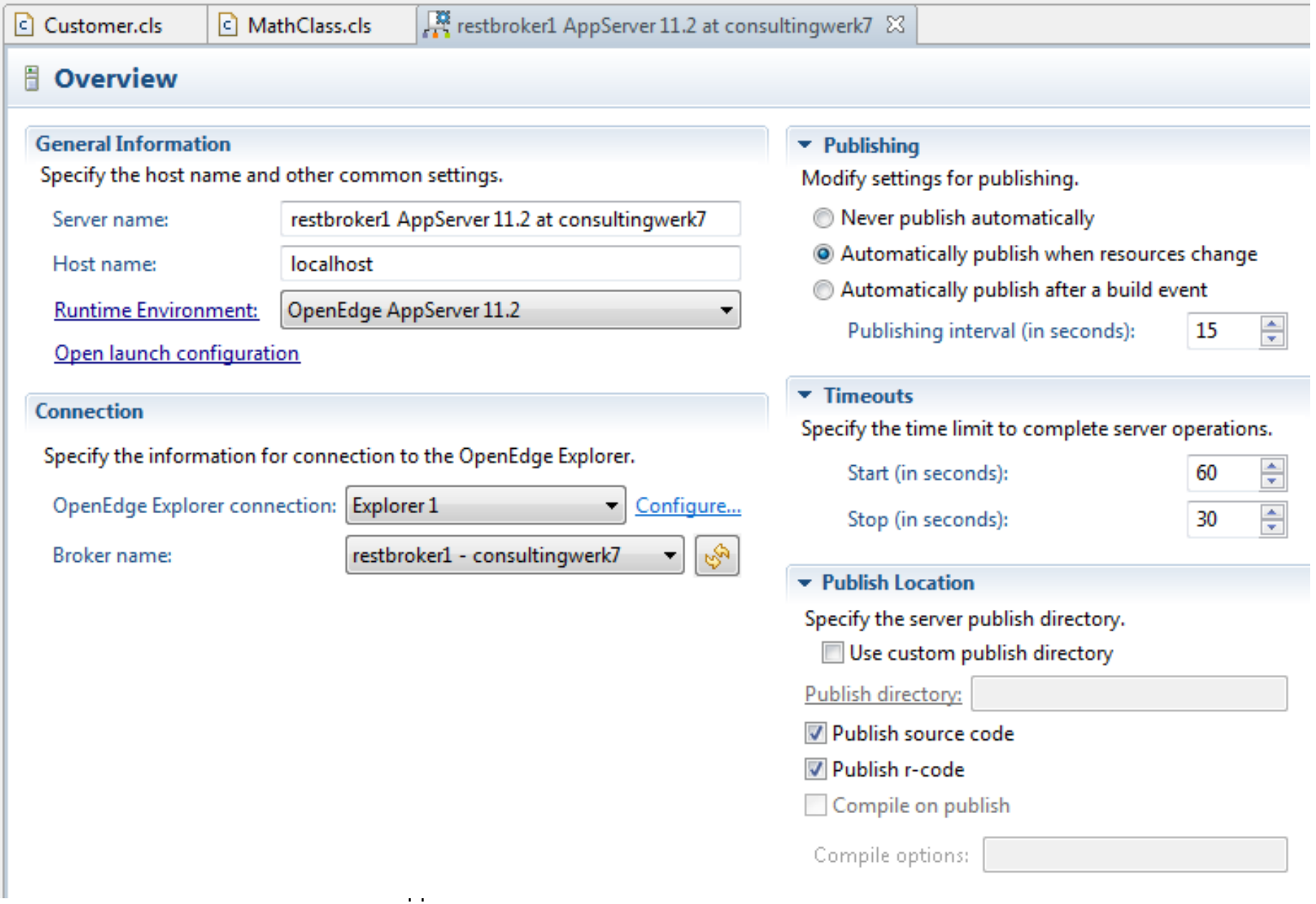
Servers View

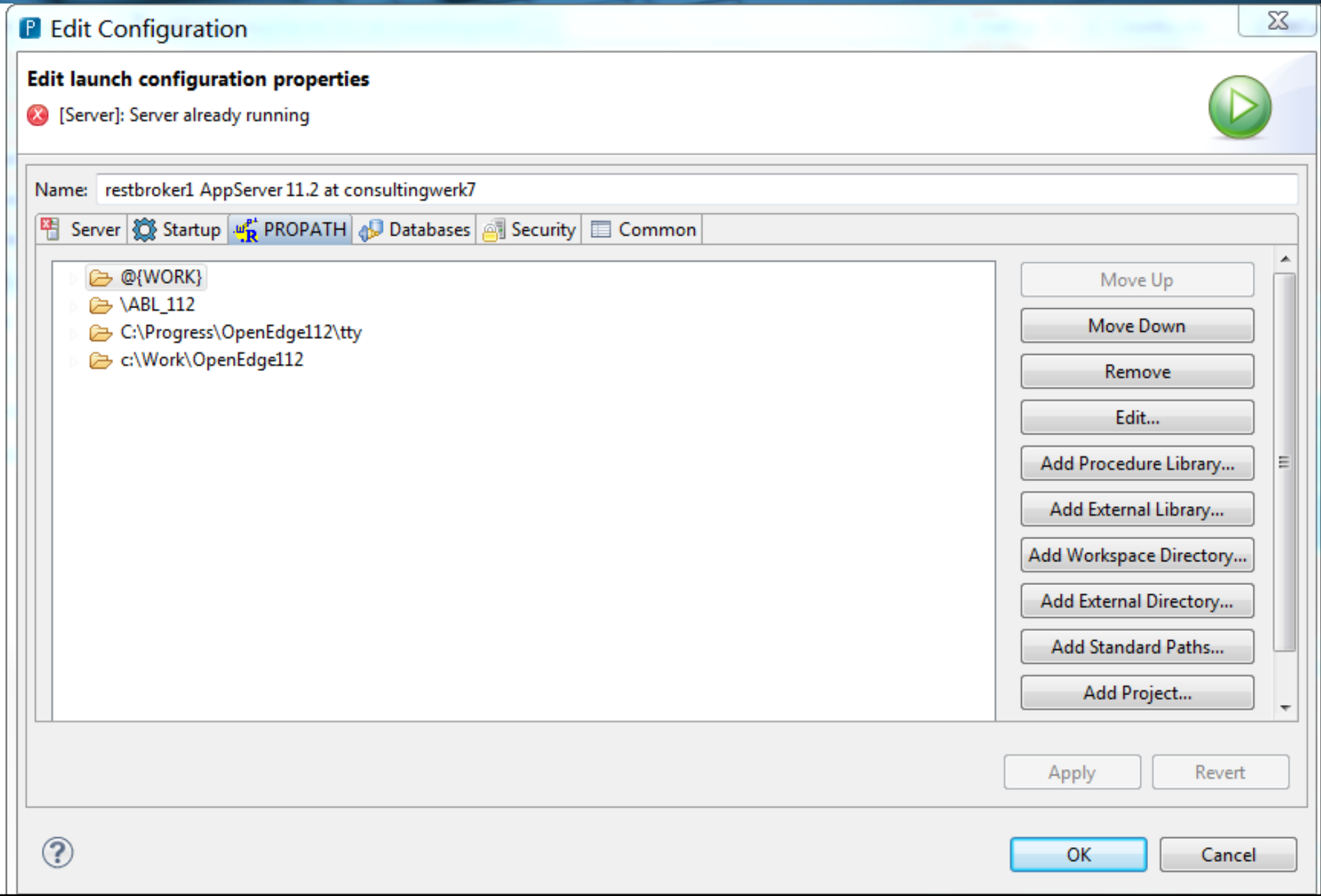
- Allows you to publish changes to the AppServer and the OpenEdge Web Server (Tomcat)



AppServer functionality

- Start / Stop / Publish
- Add/Remove: Associates projects with the AppServer
- Double Click AppServer for Server Editor
- Edit Publish properties, edit AppServer Run Configuration (DB's, PROPATH)


Customer.cls
MathClass.cls
restbroker1 AppServer 11.2 at consultingwerk7
Overview
General Information
Specify the host name and other common settings.
Server name:
restbroker1 AppServer 11.2 at consultingwerk7
Host name:
localhost
Runtime Environment:
OpenEdge AppServer 11.2
Open launch configuration
Connection
Specify the information for connection to the OpenEdge Explorer.
OpenEdge Explorer connection:
Explorer 1
Configure...
Broker name:
restbroker1 - consultingwerk7
Publishing
Modify settings for publishing.
Never publish automatically
Automatically publish when resources change
Automatically publish after a build event
Publishing interval (in seconds):
15
Timeouts
Specify the time limit to complete server operations.
Start (in seconds):
60
Stop (in seconds):
30
Publish Location
Specify the server publish directory.
Use custom publish directory
Publish directory:
Publish source code
Publish r-code
Compile on publish
Compile options:


Edit Configuration
Edit launch configuration properties
[Server]: Server already running
Name: restbroker1 AppServer 11.2 at consultingwerk7
Server
Startup
PROPATH
Databases
Security
Common
@{WORK}
\ABL_112
C:\Progress\OpenEdge112\tty
c:\Work\OpenEdge112
Move Up
Move Down
Remove
Edit...
Add Procedure Library...
Add External Library...
Add Workspace Directory...
Add External Directory...
Add Standard Paths...
Add Project...
Apply
Revert
OK
Cancel

Web Server functionality

- Start / Stop / Publish
- Add/Remove: Associates Mobile Apps or REST Services with the Web Server
- Double Click Web Server for Server Editor
- Edit Publish properties, Admin Server

Questions ?

Thank you!