Kendo UI Builder 2.1

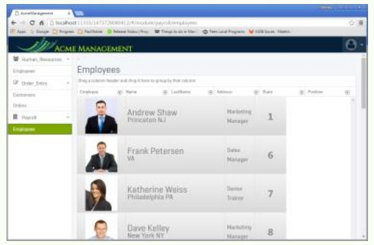
Peter Judge pjudge@progress.com

Jarmo Nieminen jniemine@progress.com

Kendo UI Builder by Progress
is a productivity tool for creating
a modern responsive Web App for
an OpenEdge application

Building Business Applications of the Future

Kendo UI® Builder
by Progress



JavaScript Data Object (JSDO)

OE Data Object Services (WEB or REST) | Catalog

Progress Application Server for OpenEdge | OpenEdge AppServer | OpenEdge BPM

OpenEdge RDBMS | OpenEdge DataServers

Progress Developer Studio for OpenEdge

OpenEdge Mgmt

Modernize Existing Applications to Web Applications

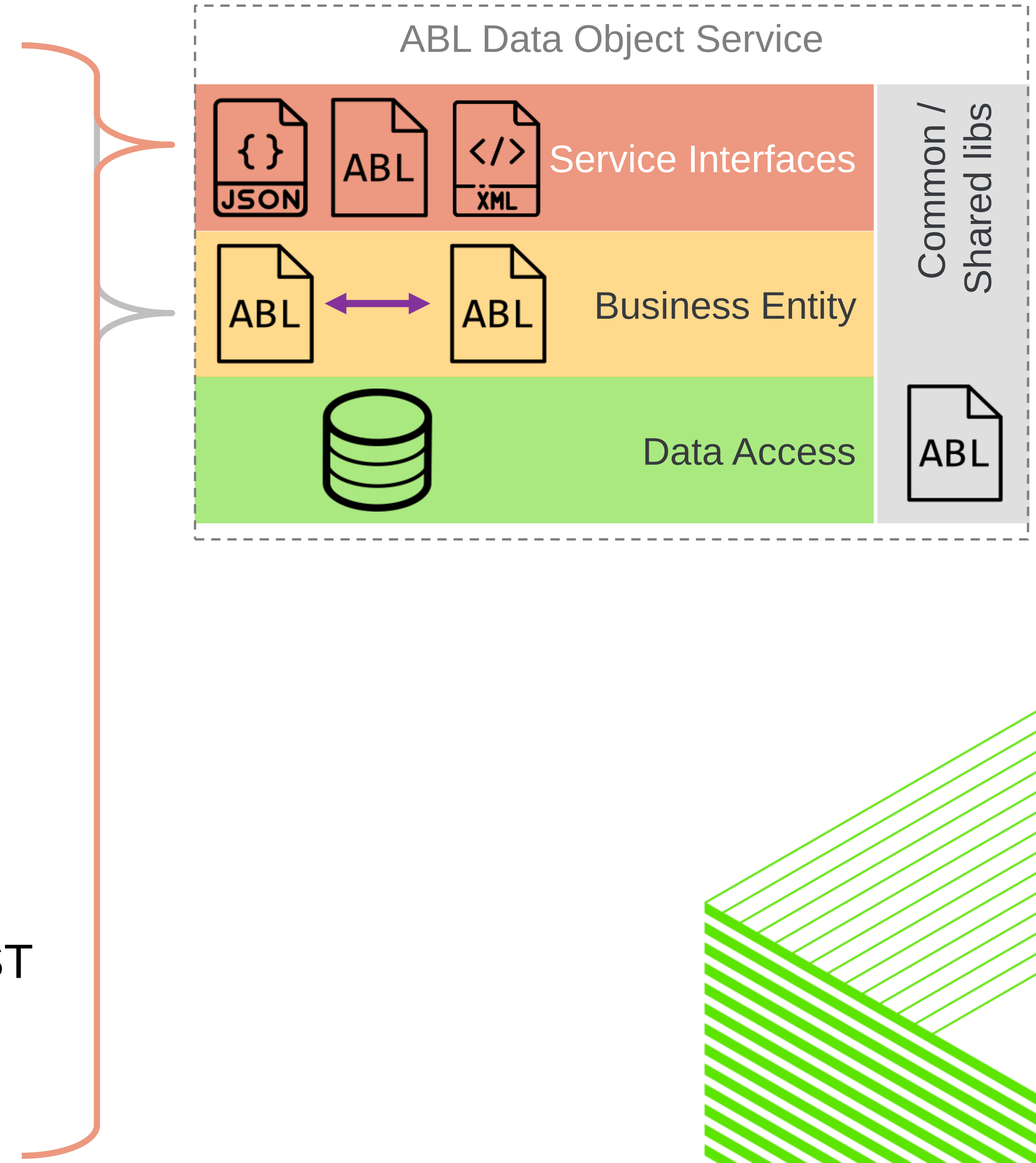
- Prebuilt Web Application Framework based on Angular
  - Application UI defined using Modules and Views
  - Supports Progress Data Objects (PDO), REST and OData Data Sources
  - Built-in data-driven Views (based on PDO Catalog)
  - Built-in authentication and session management
- Built-in Data Management
  - Data Providers manage authentication and session lifecycle
  - Access to CRUD operations, Submit, Count and other Business Logic
  - Client-side JSDO component offers dataset features like change tracking, data transactions and error handling
- Deployment to PAS for OE or other web server

Server-side

Server-side Business Entities

- Define Service Interface
  - Source code Annotations
    - Define the Resource and data model
    - Specify the ABL methods to expose

- Encapsulate data and data transactions
  - Prescriptive method signatures (REST)
    - Read-only
    - CRUD (Create, Read, Update, Delete)
    - CRUD + Submit with Before-imaging
  - Business logic with free-form signatures (REST RPC)
    - Invoke

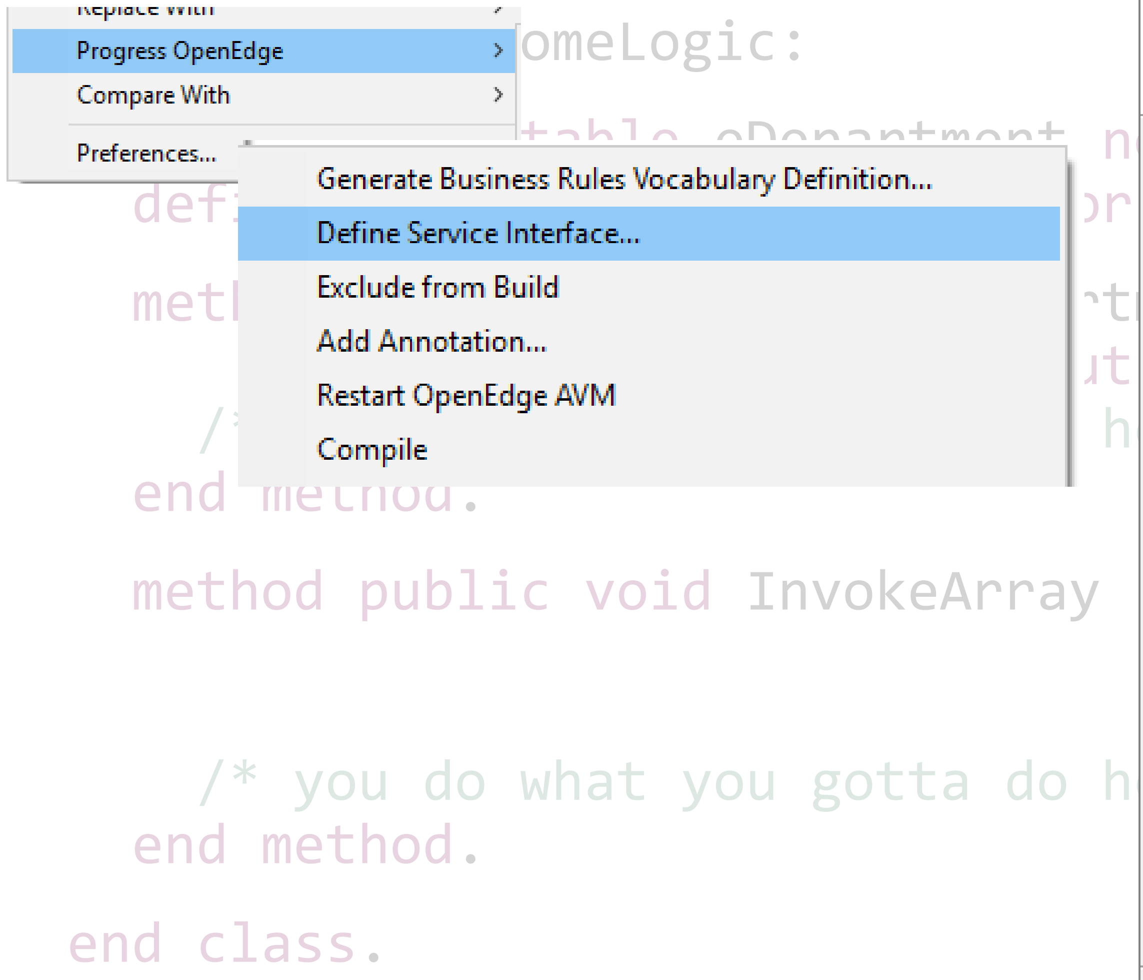


Annotated Data Object Service

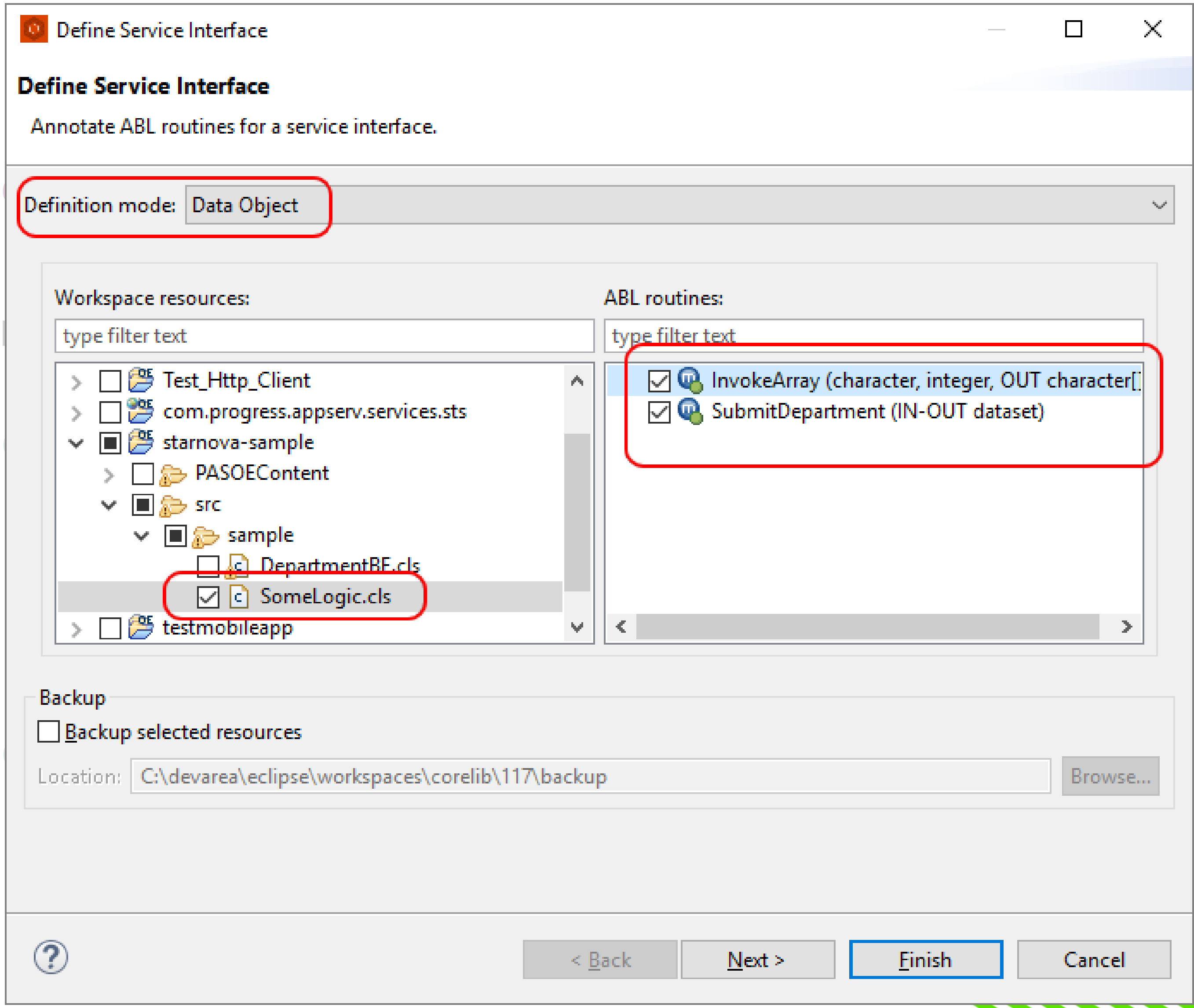
```
class sample.DepartmentUpdateObject:

  define temp-table eDepartment no-undo like Department.
  define dataset dsDepartment for eDepartment.

  method public void SubmitDepartment(
                          input-output dataset dsDepartment):
    /* you do what you gotta do here */
  end method.

  method public void InvokeArray (input pcName as char,
                                  input piSize as integer,
                                  output pcChars as character extent):
    /* you do what you gotta do here */
  end method.

end class.
```

Annotated Data Object Service

- Add annotations



```
omeLogic:
table eDepartment n
def
meth
end method.
method public void InvokeArray
   /* you do what you gotta do h
end method.
end class.
```



Define Service Interface

Annotate ABL routines for a service interface.

Definition mode: Data Object

Workspace resources:

- Test_Http_Client
- com.progress.appserv.services.sts
- starnova-sample
  - PASOEContent
  - src
    - sample
      - DepartmentBE.cls
      - ☑ SomeLogic.cls
- testmobileapp

ABL routines:

- ☑ InvokeArray (character, integer, OUT character[]
- ☑ SubmitDepartment (IN-OUT dataset)

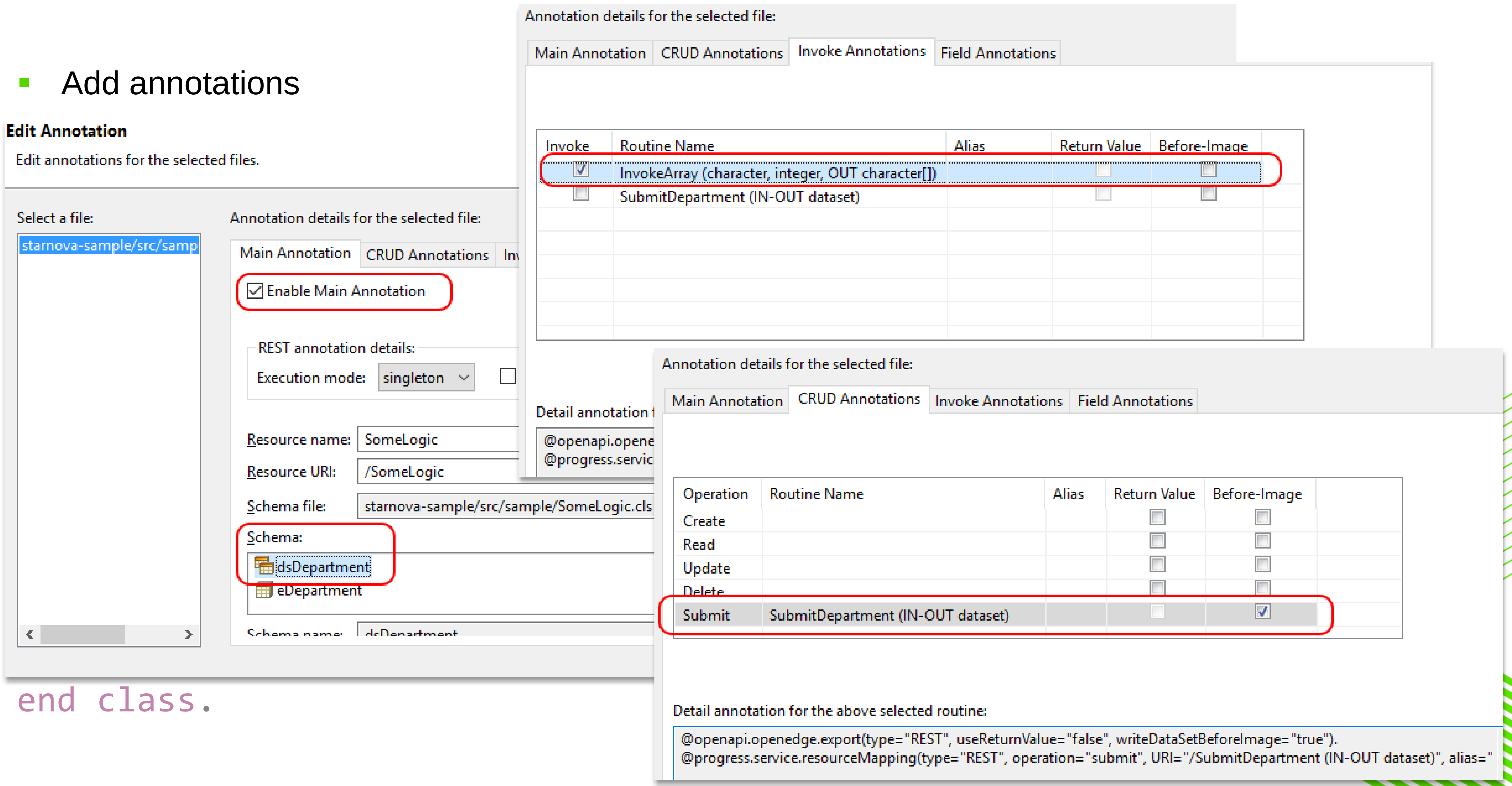
Backup

☐ Backup selected resources

Location: C:\devarea\eclipse\workspaces\corelib\117\backup

Annotated Data Object Service

- Add annotations



Edit Annotation

Edit annotations for the selected files.

Select a file:

starnova-sample/src/samp

Annotation details for the selected file:

Main Annotation | CRUD Annotations | In

☑ Enable Main Annotation

REST annotation details:

Execution mode: singleton

Resource name: SomeLogic

Resource URI: /SomeLogic

Schema file: starnova-sample/src/sample/SomeLogic.cls

Schema:

dsDepartment

eDepartment

Schema name: dsDepartment

Annotation details for the selected file:

Main Annotation | CRUD Annotations | Invoke Annotations | Field Annotations

Invoke	Routine Name	Alias	Return Value	Before-Image
☑	InvokeArray (character, integer, OUT character[])		☐	☐
☐	SubmitDepartment (IN-OUT dataset)		☐	☐

Detail annotation f

@openapi.opene

@progress.servic

Annotation details for the selected file:

Main Annotation | CRUD Annotations | Invoke Annotations | Field Annotations

Operation	Routine Name	Alias	Return Value	Before-Image
Create			☐	☐
Read			☐	☐
Update			☐	☐
Delete			☐	☐
Submit	SubmitDepartment (IN-OUT dataset)		☐	☑

Detail annotation for the above selected routine:

```
@openapi.openedge.export(type="REST", useReturnValue="false", writeDataSetBeforeImage="true").
@progress.service.resourceMapping(type="REST", operation="submit", URI="/SubmitDepartment (IN-OUT dataset)", alias="
```

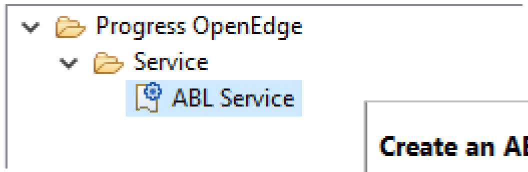
```
end class.
```

Annotated Data Object Service – final result

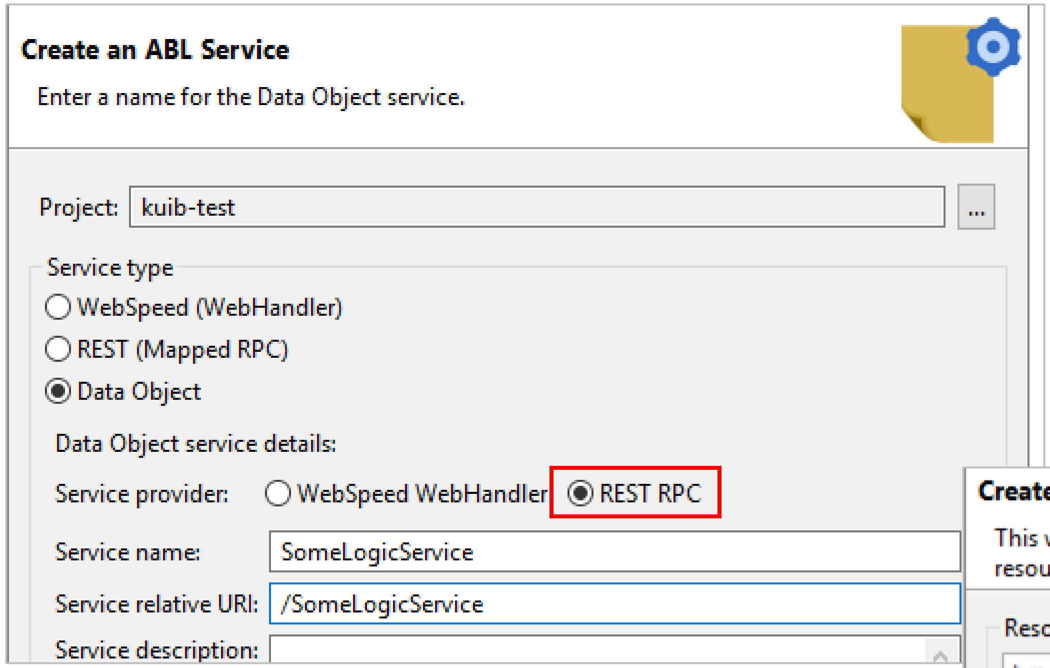
```
@openapi.openedge.export FILE(type="REST", executionMode="singleton", useReturnValue="false",
                              writeDataSetBeforeImage="false").
@progress.service.resource FILE(name="SomeLogic", URI="/DepartmentEntity", schemaName="dsDepartment", schemaFile="").

class sample.DepartmentEntity:
  define temp-table eDepartment no-undo like Department.
  define dataset dsDepartment for eDepartment.
  @openapi.openedge.export(type="REST", useReturnValue="false", writeDataSetBeforeImage="true").
  @progress.service.resourceMapping(type="REST", operation="submit", URI="/SubmitDepartment", alias="",
                                    mediaType="application/json" ).
  method public void SubmitDepartment(input-output dataset dsDepartment):
    /* you do what you gotta do here */
  end method.

  @openapi.openedge.export(type="REST", useReturnValue="false", writeDataSetBeforeImage="false").
  @progress.service.resourceMapping(type="REST", operation="invoke", URI="/InvokeArray", alias="",
                                    mediaType="application/json").
  method public void InvokeArray (input pcName as char,
                                  input piSize as int,
                                  output pcChars as character extent):
    /* you do what you gotta do here */
   end method.
end class.
```

Annotated Data Object Service Deployment Target

- Create an ABL Service (PASOE) or Data Object (Classic AppServer)



- Progress OpenEdge
  - Service
    - ABL Service



Create an ABL Service

Enter a name for the Data Object service.

Project: kuib-test

Service type

- WebSpeed (WebHandler)
- REST (Mapped RPC)
- [✓ Data Object]

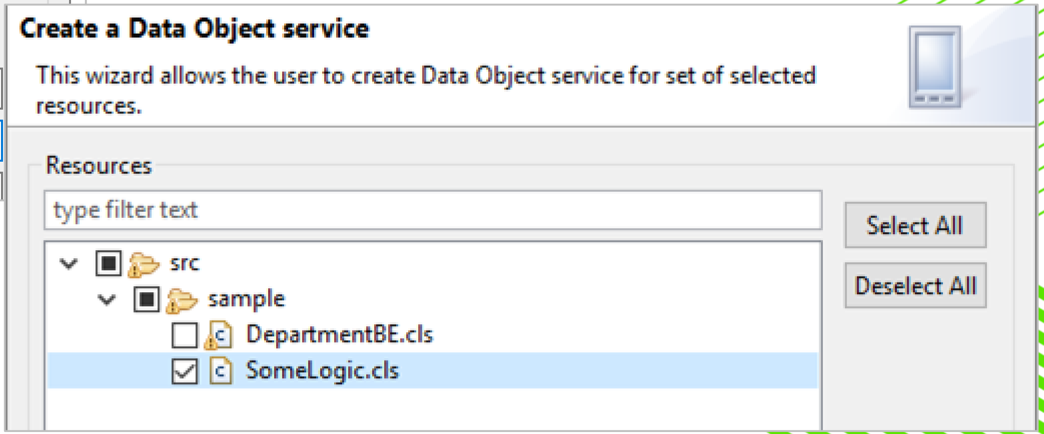
Data Object service details:

Service provider: WebSpeed WebHandler [✓ REST RPC]

Service name: SomeLogicService

Service relative URI: /SomeLogicService

Service description:



Create a Data Object service

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

Resources

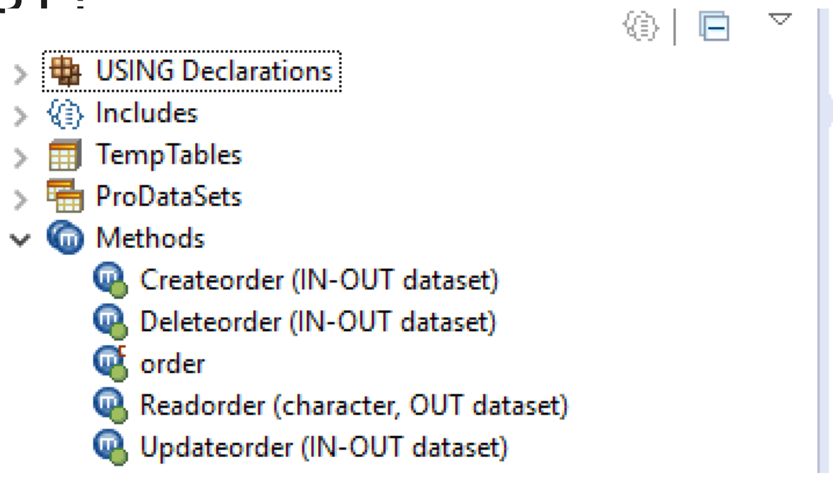
type filter text

- ■ src
  - ■ sample
    - ☐ DepartmentBE.cls
    - ☑ SomeLogic.cls

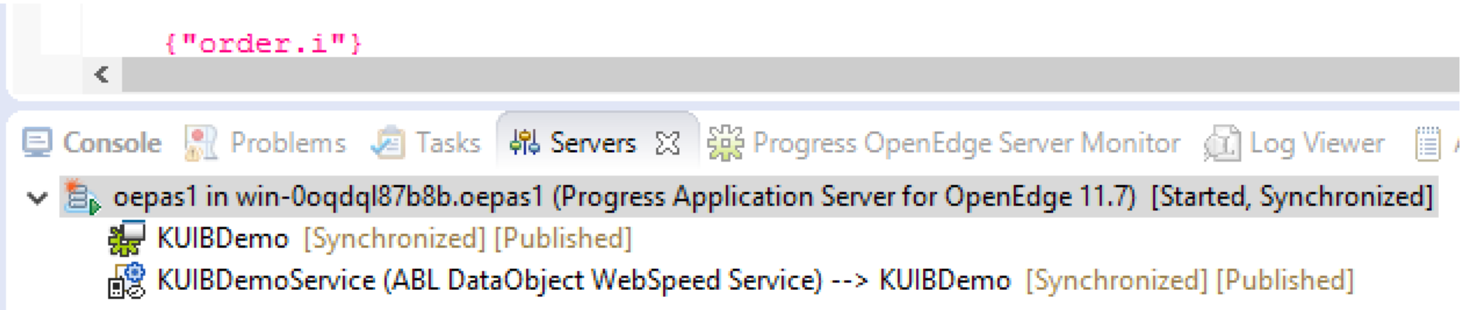
Annotated Data Object Service – Catalog

```
{"services": [{
        "name": "DepartmentEntityService",
        "address": "/web/pdo/DepartmentEntityService",
        "useRequest": true,
        "resources": [{
            "name": "DepartmentEntity",
            "path": "/ DepartmentEntity ",
            "autoSave": false,
            "schema": { "properties": {"dsDepartment": {"properties": {"eDepartment": { ... } } } }} }} },
            "operations": [ {
                    "name": "InvokeArray",
                    "path": "\/InvokeArray",
                    "useBeforeImage": false,
                    "type": "invoke",
                    "verb": "put",
                    "params": [
                        { "name": "pcName",  "type": "REQUEST_BODY" },
                        { "name": "piSize",  "type": "REQUEST_BODY" },
                        { "name": "pcChars", "type": "RESPONSE_BODY" }
                ]}, {
                    "name": "SubmitDepartment",
                    "path": "\/SubmitDepartment",
                    "useBeforeImage": true,
                    "type": "submit",
                    "verb": "put",
                    "params": [
                        { "name": "dsDepartment", "type": "REQUEST_BODY"  },
                        { "name": "dsDepartment", "type": "RESPONSE_BODY" }
] } ] }] }] }
```

How to expose my existing or new business logic to web/mobile using standard REST?



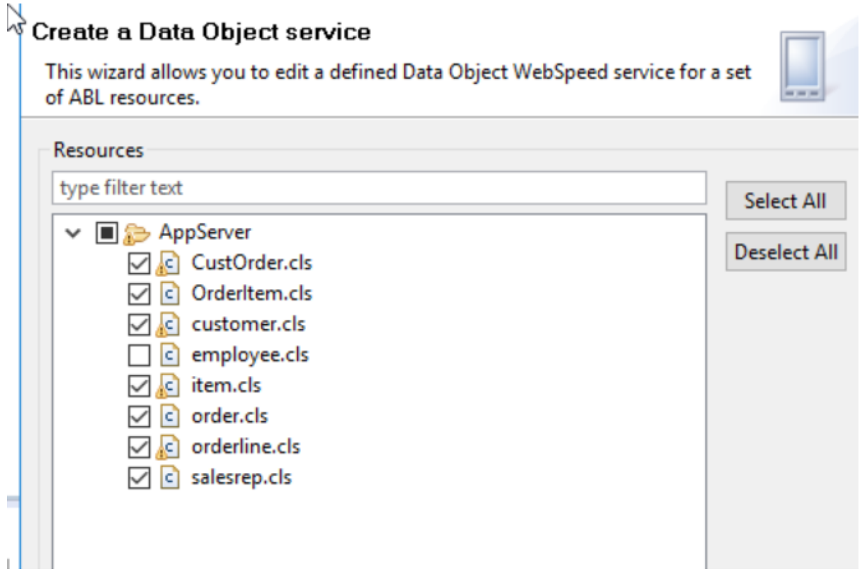
- USING Declarations
- Includes
- TempTables
- ProDataSets
- Methods
  - Createorder (IN-OUT dataset)
  - Deleteorder (IN-OUT dataset)
  - order
  - Readorder (character, OUT dataset)
  - Updateorder (IN-OUT dataset)



```
{"order.i"}
```

Console | Problems | Tasks | Servers | Progress OpenEdge Server Monitor | Log Viewer

```
oepas1 in win-0oqdql87b8b.oepas1 (Progress Application Server for OpenEdge 11.7)  [Started, Synchronized]
    KUIBDemo  [Synchronized] [Published]
    KUIBDemoService (ABL DataObject WebSpeed Service) --> KUIBDemo  [Synchronized] [Published]
```



Create a Data Object service

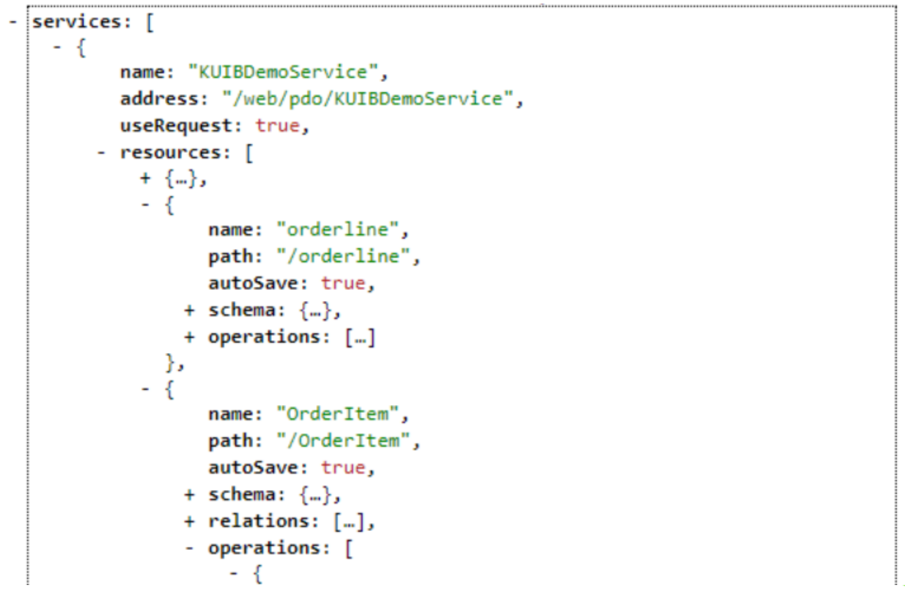
This wizard allows you to edit a defined Data Object WebSpeed service for a set of ABL resources.

Resources

type filter text

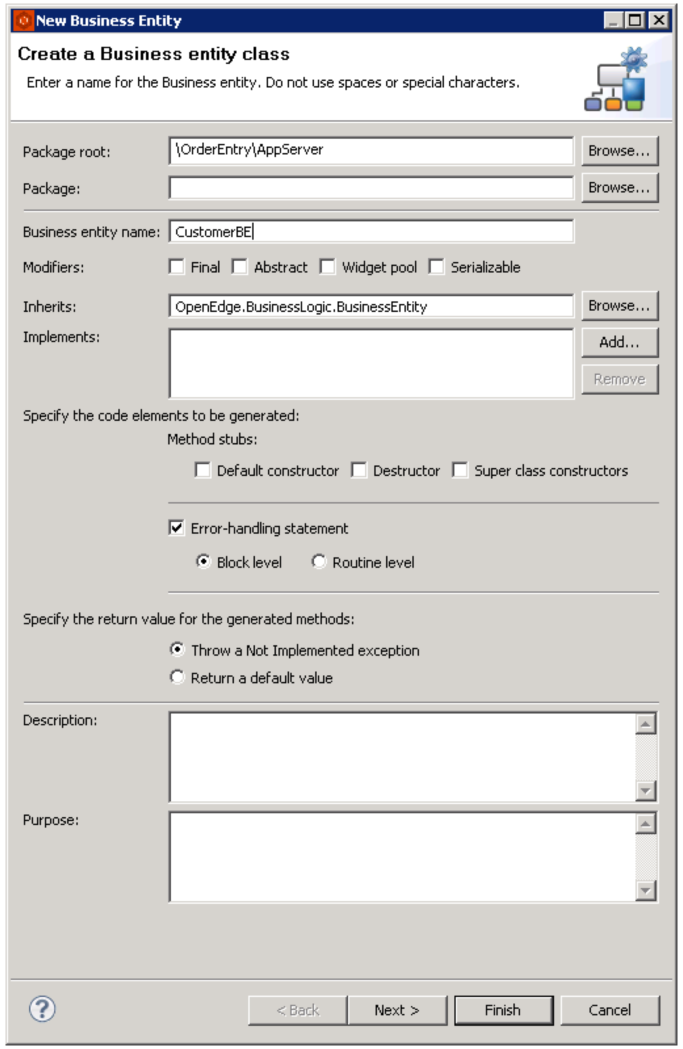
Select All | Deselect All

- [■] AppServer
  - [✓] CustOrder.cls
  - [✓] OrderItem.cls
  - [✓] customer.cls
  - [ ] employee.cls
  - [✓] item.cls
  - [✓] order.cls
  - [✓] orderline.cls
  - [✓] salesrep.cls



```
- services: [
   - {
          name: "KUIBDemoService",
          address: "/web/pdo/KUIBDemoService",
          useRequest: true,
        - resources: [
            + {…},
            - {
                  name: "orderline",
                  path: "/orderline",
                  autoSave: true,
                + schema: {…},
                + operations: […]
              },
            - {
                  name: "OrderItem",
                  path: "/OrderItem",
                  autoSave: true,
                + schema: {…},
                + relations: […],
                - operations: [
                    - {
```

Demo – Business Entities


New Business Entity

Create a Business entity class

Enter a name for the Business entity. Do not use spaces or special characters.

Package root: \OrderEntry\AppServer

Package:

Business entity name: CustomerBE

Modifiers: ☐ Final ☐ Abstract ☐ Widget pool ☐ Serializable

Inherits: OpenEdge.BusinessLogic.BusinessEntity

Implements:

Specify the code elements to be generated:

Method stubs:

☐ Default constructor ☐ Destructor ☐ Super class constructors

☑ Error-handling statement

[✓ Block level] Routine level

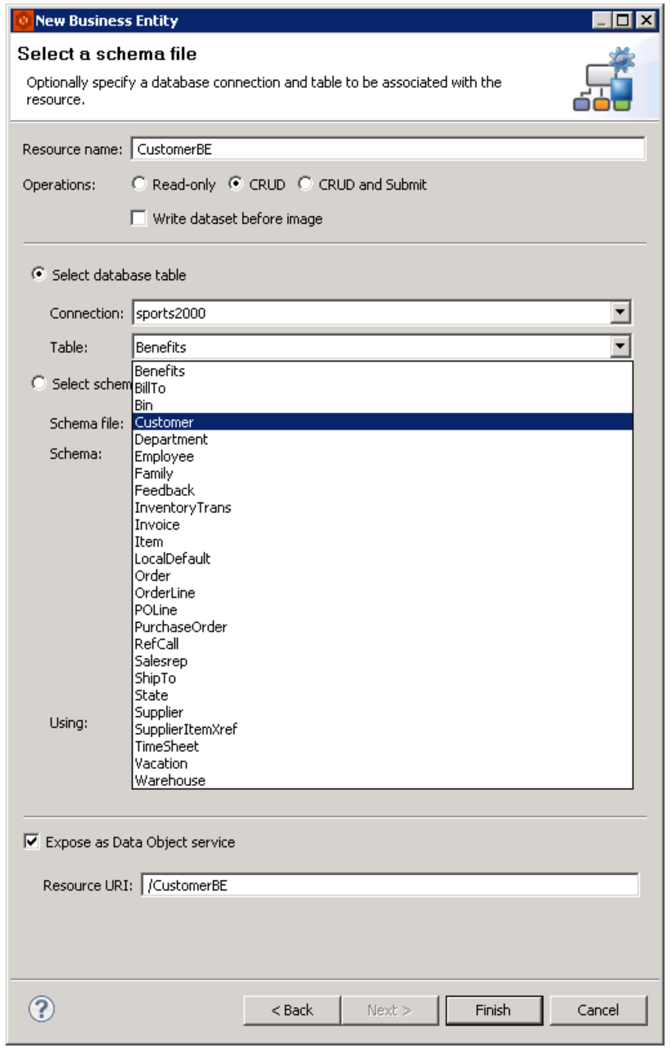
Specify the return value for the generated methods:

[✓ Throw a Not Implemented exception]

Return a default value

Description:

Purpose:


New Business Entity

Select a schema file

Optionally specify a database connection and table to be associated with the resource.

Resource name: CustomerBE

Operations: Read-only [✓ CRUD] CRUD and Submit

☐ Write dataset before image

[✓ Select database table]

Connection: sports2000

Table: Benefits

Benefits
BillTo
Bin
Customer
Department
Employee
Family
Feedback
InventoryTrans
Invoice
Item
LocalDefault
Order
OrderLine
POLine
PurchaseOrder
RefCall
Salesrep
ShipTo
State
Supplier
SupplierItemXref
TimeSheet
Vacation
Warehouse

Select schem

Schema file:

Schema:

Using:

☑ Expose as Data Object service

Resource URI: /CustomerBE