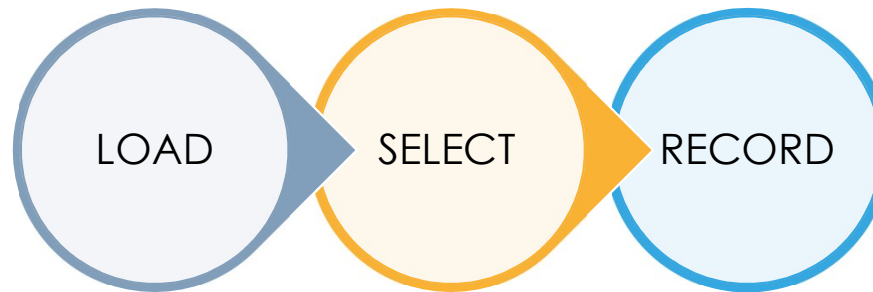


Lesson 5. SERVICE Hooks in Java

T24 Extensibility for Java

| Services in T24

- Multithreaded processes
- Can be run independently or as part of COB



- Must implement **com.temenos.t24.api.hook.system.ServiceLifecycle**
- Equivalent of
 - LOAD – initialize()
 - SELECT – getIds() or getTableName()
 - RECORD – inputRecord() or process()

Practice 5.1 - Service

- Create a simple service that updates the SHORT.NAME of the CUSTOMERS of SECTOR 1001 as TEST + Existing MNEMONIC
- Expected Output

CUSTOMER 130131	
Customer Mnemonic	KATRIN
Short Name	EN TEST KATRIN
Full Name	EN Abbey Adalwine
Street Name	EN FRANKFURT

Step 1 - Create the class

```
package com.newbank;

import java.util.List;

import com.temenos.api.TBoolean;
import com.temenos.api.TStructure;
import com.temenos.t24.api.complex.eb.servicehook.ServiceData;
import com.temenos.t24.api.hook.system.ServiceLifecycle;
import com.temenos.t24.api.records.customer.CustomerRecord;
import com.temenos.t24.api.system.DataAccess;

/* Service DEMO */

public class MyServiceDemo extends ServiceLifecycle {
    @Override
    public void inputRecord(String id, ServiceData serviceData, String controllItem, TBoolean setZeroAuth,
        List<String> versionNames, List<String> recordIds, List<TStructure> records) {

        DataAccess da= new DataAccess(this);
        versionNames.add("CUSTOMER");
        recordIds.add(id);
        CustomerRecord cusRec = new CustomerRecord(da.getRecord("CUSTOMER", id));
        cusRec.setShortName("TEST "+cusRec.getMnemonic().toString(), 0);
        records.add(cusRec.toStructure());
        setZeroAuth.set(true);
    }

    @Override
    public List<String> getIds(ServiceData serviceData, List<String> controllList) {

        List<String> recIds = null;
        DataAccess da = new DataAccess(this);
        recIds=da.selectRecords("BNK", "CUSTOMER", "", "WITH MNEMONIC EQ KATRIN KMART KOS ");
        return recIds;
    }
}
```

Export JAR and
restart jBoss

Step 2 - Create the EB.API records

```
public class MyServiceDemo extends ServiceLifecycle {  
    @Override  
    public void inputRecord(String id, ServiceData serviceData,  
        List<String> versionNames, List<String> recordIds,  
  
        DataAccess da= new DataAccess(this);  
        versionNames.add("CUSTOMER");  
        recordIds.add(id);  
        CustomerRecord cusRec = new CustomerRecord(da.getRecord  
        cusRec.setShortName("TEST "+cusRec.getMnemonic().toStri  
        records.add(cusRec.toStructure());  
        setZeroAuth.set(true);  
  
    }  
    @Override  
    public List<String> getIds(ServiceData serviceData, List<St  
  
        List<String> recIds = null;  
        DataAccess da = new DataAccess(this);  
        recIds=da.selectRecords("BNK", "CUSTOMER", "", "WITH MN  
        return recIds;  
  
    }  
}
```

EB.API SERVICE.DEMO	
Description	EN SERVICE DEMO
Protection Level	Full
Source Type	Method
Java Method	inputRecord
Java Class	MyServiceDemo
Java Package	com.newbank

EB.API SERVICE.DEMO.SELECT	
Description	EN SERVICE DEMO SELECT
Protection Level	Full
Source Type	Method
Java Method	getIds
Java Class	MyServiceDemo
Java Package	com.newbank

Step 3 - Create the PGM.FILE, BATCH & TSA.SERVICE records




PGM.FILE SERVICE.DEMO 

Type	B
Batch Job.1	@BATCH.JOB.CONTROL
Product	EB

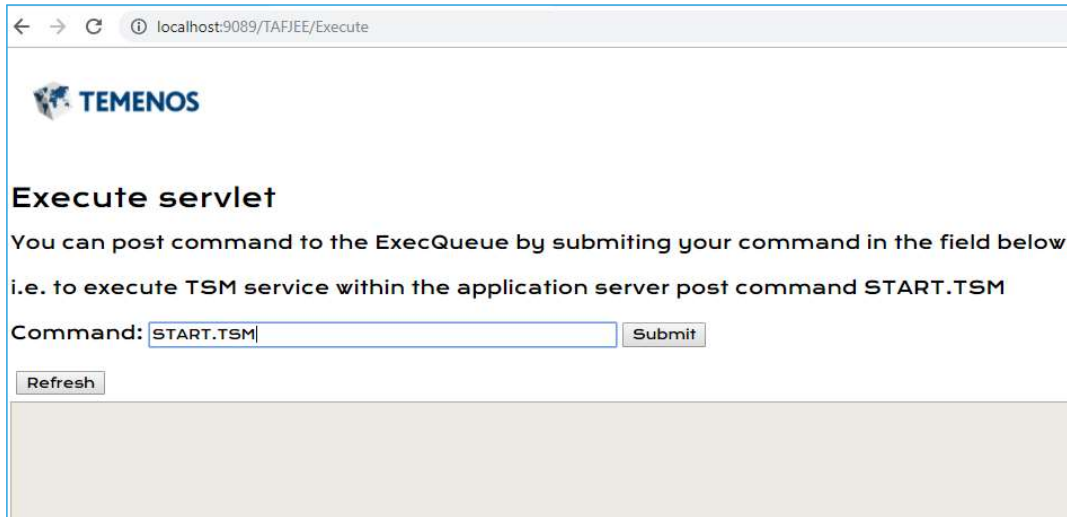
BATCH BNK/SERVICE.DEMO

Process Status	O
Batch Environment	F FOREGROUND
Job Name.1	SERVICE.DEMO
Frequency.1	D
Data.1.1	BULK.OFS


TSA.SERVICE BNK/SERVICE.DEMO

Description.1	SERVICE DEMO 
	
Server Name.1	<input type="text"/>
Work Profile.1	<input type="text"/>
Server Status.1	<input type="text"/>
User *	INPUTTER INPUTTER 
Service Control	<input type="radio"/> Stop <input checked="" type="radio"/> Start <input type="radio"/> Auto

Step 4 - Run the TSM and TSA



← → ↻ ⓘ localhost:9089/TAFJEE/Execute

 **TEMENOS**

Execute servlet

You can post command to the ExecQueue by submitting your command in the field below.
i.e. to execute TSM service within the application server post command START.TSM

Command:

- When the service is run, we can see the SHORT.NAME is updated



 TEMENOS CUSTOMER L L			
Customer No	Mnemonic	Name	Account Officer
<u>130131</u>	KATRIN	TEST KATRIN	Implementation
<u>100284</u>	KMART	TEST KMART	Branch Operations Manager
<u>100329</u>	KOS	TEST KOS	Derivatives Front Office Dealer

| Lesson 6. Local APPLICATION

T24 Extensibility for Java



TEMENOS
Learning Community

Practice 6.1 - Creating Local Application

- We can create a new application using EB.TABLE.DEFINITION. Raise OVERRIDE if the phone / email is used by existing customer

TABLE.NAME	WALKIN.CUSTOMER
PRODUCT	ST
FILE.TYPE	H
CLASSIFICATION	CUS

@ID, OVERRIDE, AUDIT
fields are added
automatically

Field Name	Data Type	Remarks
NAME	A	
PHONE		Raise override if this PHONE is available in CUSTOMER
EMAIL		Raise override if this EMAIL is available in CUSTOMER

Step 1 - Creating the TABLE – EB.TABLE.DEFINITION

- Create the table using EB.TABLE.DEFINITION
- ETD creates the required artefacts – PGM.FILE, FILE.CONTROL, STANDARD.SELECTION, DATA FILES

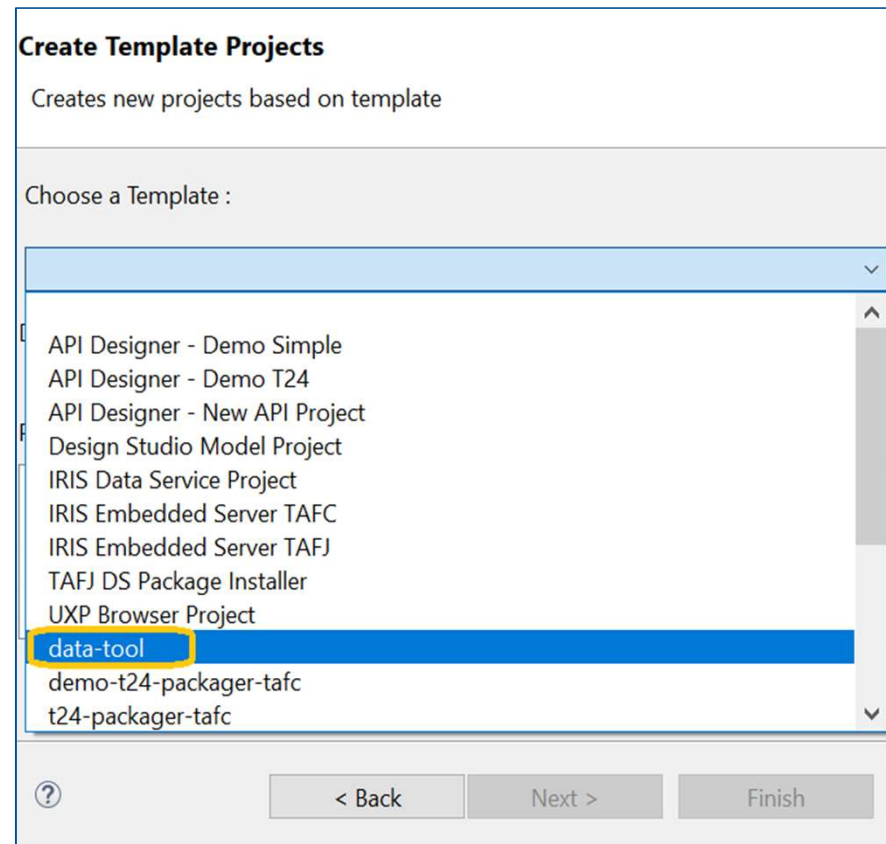
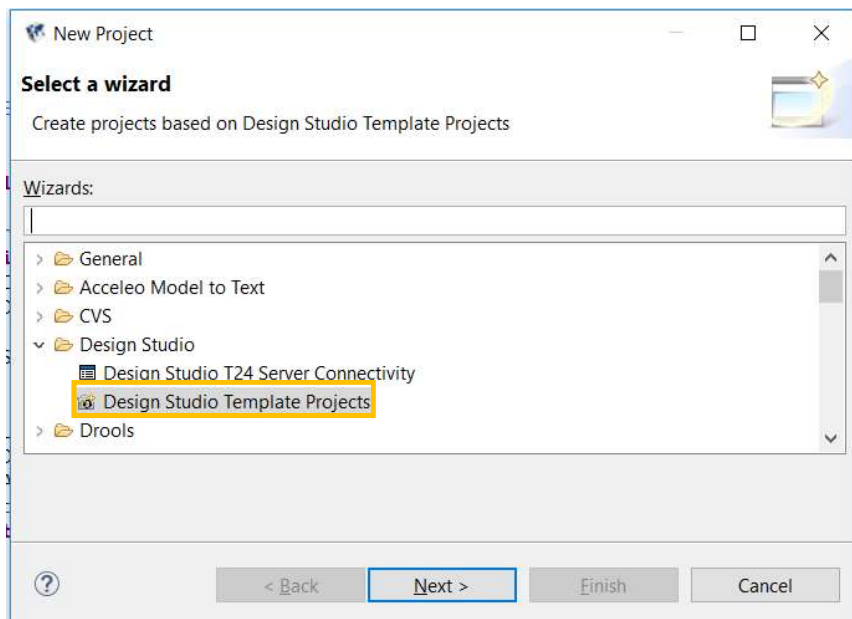
EB.TABLE.DEFINITION		WALKIN.CUSTOMER
Description	EN WALKIN CUSTOMER DEMO	
Product	ST	System Tables
Field Name.1	NAME	
Description.1	NAME.OF.CUST	
Max Char.1	35	
Data Type.1	ALPHANUMERIC	Alpha Numeric Charactes
Field Name.2	PHONE	
Description.2	PHONE	
Max Char.2	20	
Data Type.2	ALPHANUMERIC	Alpha Numeric Charactes
Field Name.3	EMAIL	
Description.3	EMAIL	
Max Char.3	35	
Data Type.3	ALPHANUMERIC	Alpha Numeric Charactes
Prefix	ST.WAL26	
File Type	Cus	
Link To Wfl	No	

| Step 2 - Generate the API for the new Table

- Using Design Studio, we can generate the API for the APPLICATION created using EB.TABLE.DEFINITION
- we need a server project – to establish connection with T24
- We need a models project – to import the application from T24

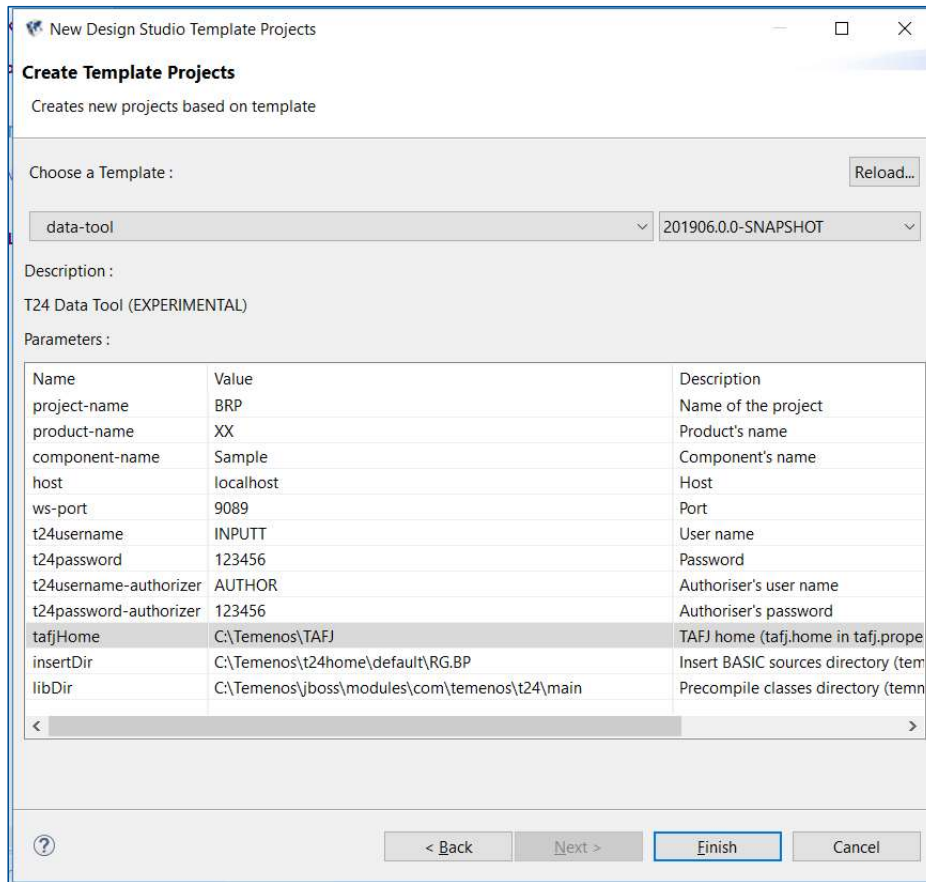
Step 2 - Create the –DataTool project

- Creating the server project
- File→New→Project



Step 3 - Create the DS-DataTool Project

- Create the DS-DataTool project



Create Template Projects
Creates new projects based on template

Choose a Template : Reload...

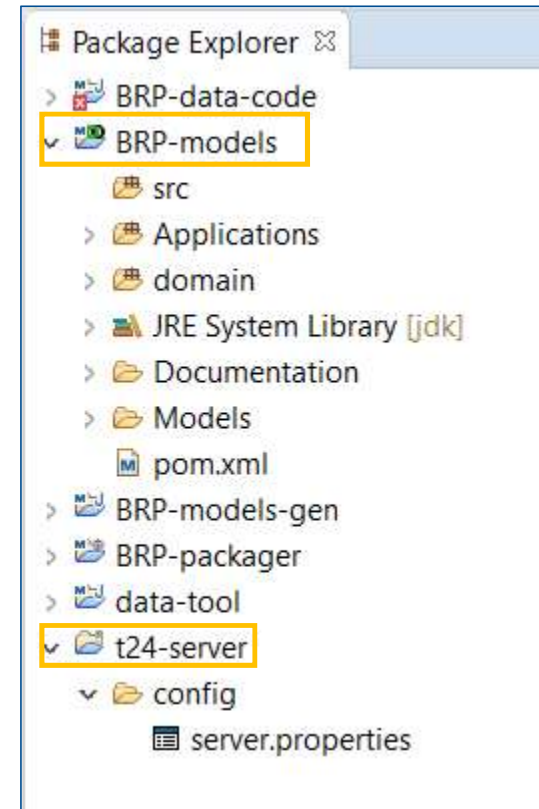
data-tool 201906.0.0-SNAPSHOT

Description :
T24 Data Tool (EXPERIMENTAL)

Parameters :

Name	Value	Description
project-name	BRP	Name of the project
product-name	XX	Product's name
component-name	Sample	Component's name
host	localhost	Host
ws-port	9089	Port
t24username	INPUTT	User name
t24password	123456	Password
t24username-authorizer	AUTHOR	Authoriser's user name
t24password-authorizer	123456	Authoriser's password
tafjHome	C:\Temenos\TAFJ	TAFJ home (tafj.home in tafj.prope
insertDir	C:\Temenos\t24home\default\RG.BP	Insert BASIC sources directory (tem
libDir	C:\Temenos\jboss\modules\com\temenos\t24\main	Precompile classes directory (temn

< Back Next > Finish Cancel



Step 4 - Import the Local Application

The screenshot illustrates the process of importing a local application into a T24 project. The main window is the 'T24 Applications Import Wizard', which is currently on the 'T24 Application Name Selection' step. The wizard prompts the user to 'Select the destination of imported Application' and 'Select the destination folder'. The destination folder is set to 'Applications' under the 'BRP-models' project. The application name is 'ST.WALKIN.CUSTOMER'. The wizard also shows the 'Projects visibility' section with 'Only model projects' selected and the 'Folder Structure' set to 'FLAT'.

The Package Explorer on the left shows the project structure. The 'EB_Foundation.domain' project is highlighted, and its contents are listed in the right pane:

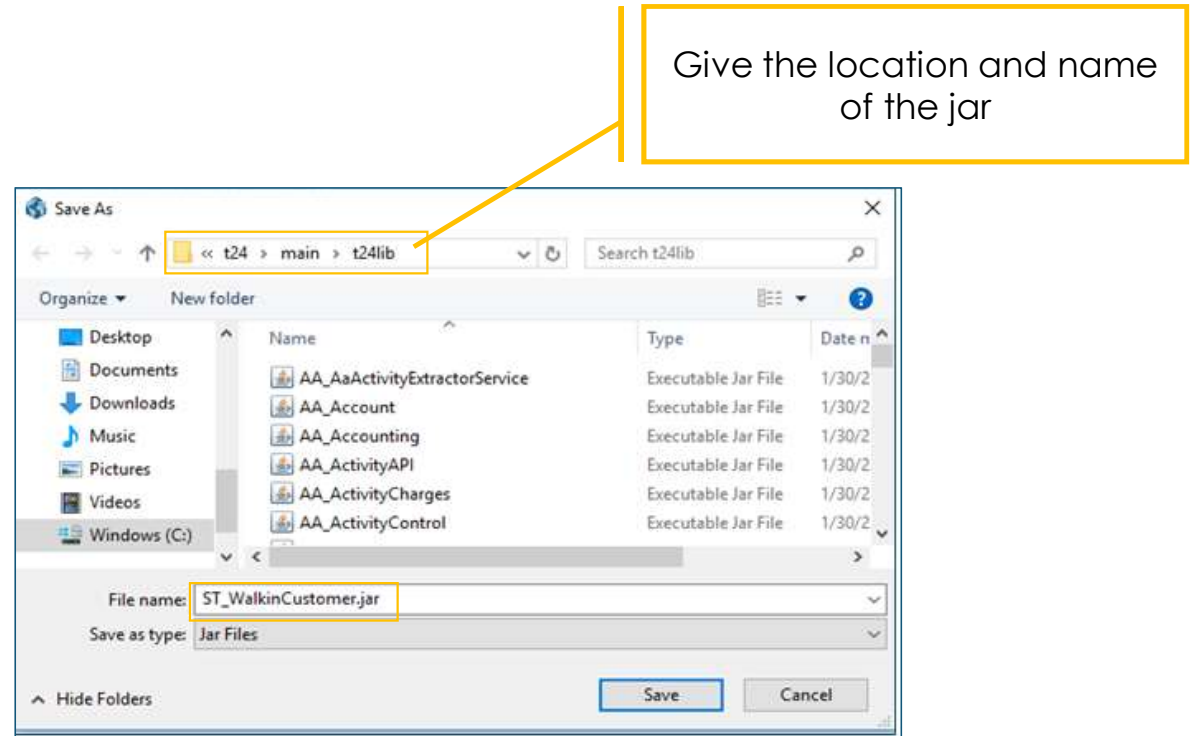
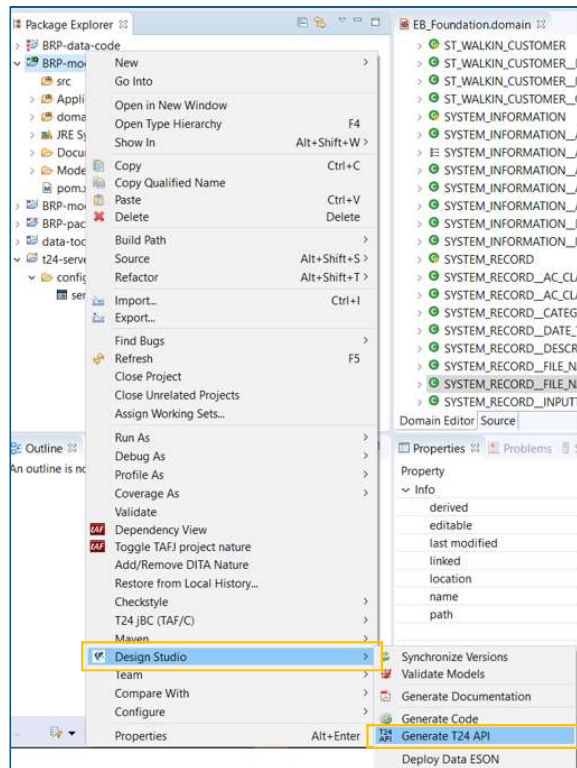
- EB_DatInterface.domain
- EB_Dealslip.domain
- EB_Delivery.domain
- EB_Desktop.domain
- EB_Dim.domain
- EB_ErrorProcessing.domain
- EB_FileUpload.domain
- EB_Foundation.domain
- EB_Interface.domain

The 'EB_Foundation.domain' project is expanded, showing the following sub-projects:

- ST_WALKIN_CUSTOMER
- ST_WALKIN_CUSTOMER_DATE_TIME
- ST_WALKIN_CUSTOMER_INPUTTER
- ST_WALKIN_CUSTOMER_OVERRIDE
- SYSTEM_INFORMATION
- SYSTEM_INFORMATION_ANALYS_ROUTINE
- SYSTEM_INFORMATION_ANALYS_ROUTINE_ANALYS_ROUTINE
- SYSTEM_INFORMATION_ANALYS_ROUTINE_ANALYS_VALUE
- SYSTEM_INFORMATION_APPLICATION

Step 5 - Generate the API for the new Table

- Toggle the DS project as a TAFJ project
- Right click on project → Design Studio → Generate T24 API



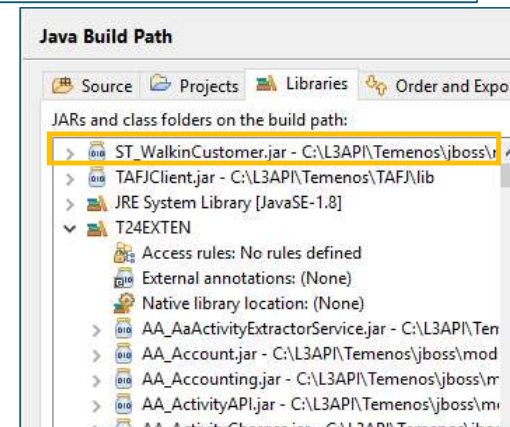
Update module.xml and Build path

C:\Temenos\jboss\modules\com\temenos\t24\main\t24lib\ST_WalkinCustomer.jar				
Name	Size	Packed Size	Modified	
WalkinCustomerRecord.class	16 803	6 058	2019-02-22 13:34	
WalkinCustomerRecord.java	6 397	955	2019-02-22 13:34	
WalkinCustomerTable.class	4 095	1 517	2019-02-22 13:34	
WalkinCustomerTable.java	2 082	517	2019-02-22 13:34	

- Update module.xml and restart jBoss

```
<module xmlns="urn:jboss:module:1.0" name="com.temenos.t24">
  <resources>
    <!-- Insert resources here -->
    <resource-root path="./T24Exten.jar" />
    <resource-root path="./t24lib/ST_WalkinCustomer.jar" />
    <resource-root path="./t24lib/AA_ARAccountsData.jar" />
    <resource-root path="./t24lib/AA_ARC.jar" />
  </resources>
</module>
```

- Add jar to the java Build path in DS



Step 6 - Create the class and implement ValidateRecord method

```
*WalkinCustomerOverride.java
package com.newbank;
import java.util.List;
import com.temenos.api.TBoolean;
import com.temenos.api.TString;
import com.temenos.api.TStructure;
import com.temenos.api.TValidationResponse;
import com.temenos.t24.api.hook.system.RecordLifecycle;
import com.temenos.t24.api.system.DataAccess;
import com.temenos.t24.api.tables.walkincustomer.WalkinCustomerRecord;

public TValidationResponse validateRecord(String application, String recordId, TStructure record,
    TStructure lastLiveRecord) {
    // validate phone and raise override
    DataAccess da = new DataAccess(this);
    String cusIds = "";
    List<String> recIds;
    int len;
    WalkinCustomerRecord WCR = new WalkinCustomerRecord(record);
    String phone = WCR.getPhone().toString();
    if (!phone.isEmpty()) {
        recIds = da.selectRecords("BNK", "CUSTOMER", "", "WITH PHONE.1 EQ '" + WCR.getPhone().toString() + "'");
        len = recIds.size();
        if (len > 0) {
            for (int i = 0; i < len; i++)
                cusIds = cusIds + " " + recIds.get(i);
            WCR.getPhone().setOverride("This phone number is used by CUSTOMER " + cusIds);
        }
    }
    // VALIDATE email and raise override
    String email = WCR.getEmail().toString();
    if (!email.isEmpty()) {
        recIds = da.selectRecords("BNK", "CUSTOMER", "", "WITH EMAIL.1 EQ '" + email + "'");
        len = recIds.size();
        cusIds = "";

        if (len > 0) {
            for (int i = 0; i < len; i++)
                cusIds = cusIds + " " + recIds.get(i);
            WCR.getEmail().setOverride("This email is used by CUSTOMER " + cusIds);
        }
    }
    return WCR.getValidationResponse();
}
```

Step 7 - Create the EB.API, VERSION

EB.API	WALKIN.CUSTOMER.OVERRIDE
Description	EN WALKIN CUST
Protection Level	Full
Source Type	Method
Java Method	validateRecord
Java Class	WalkinCustomerOverride
Java Package	com.newbank

Version	ST.WALKIN.CUSTOMER,DEMO
Records Per Page	1
Fields Per Line	Multi
Language Code.1	1
Field No.1	NAME
Text Char Max.1	35
Pre Syndication ID.1 .1	Name
Enrichm Char.1	35
Table Line.1	0
Field No.2	EMAIL
Text Char Max.2	35
Pre Syndication ID.2 .1	Email
Enrichm Char.2	35
Table Line.2	1
Field No.3	PHONE
Text Char Max.3	10
Pre Syndication ID.3 .1	Phone
Enrichm Char.3	10
Table Line.3	2
Input Routine.1	WALKIN.CUSTOMER.OVERRIDE

| Step 8 - Launch the VERSION

CUSTOMER 100100 ↑ i - Ple

Phone No.1	206 553-6947
Mobile Phone Numbers.1	206 553-6947
Email.1	harrycrisp@gmail.com

St Walkin Customer Demo 829

Name	HARRY THOMAS CRISP
Email	harrycrisp@gmail.com
Phone	206 553-6947
Override.1	This phone number is used by CUSTOMER 100100
Override.2	This email is used by CUSTOMER 100100

= This phone number is used by
CUSTOMER 100100
This email is used by CUSTOMER 100100

CANCEL ACCEPT

Using EB.TABLE.PROCEDURES

EB.API *	WALKI ▼				
	Key	Description	Protection Level	Source Type	Java Method
	WALKIN.CUSTOMER.DEFAULT	WALKIN CUST	FULL	METHOD	defaultFieldValues
	WALKIN.CUSTOMER.OVERRIDE	WALKIN CUST	FULL	METHOD	validateRecord

EB.TABLE.PROCEDURES	ST.WALKIN.CUSTOMER
Description *	EN WALKIN CUSTOMER
Initialise Proc.1	
Check Id Proc.1	
Check Rec Proc.1	WALKIN.CUSTOMER.DEFAULT WALKIN CUST ▼
Prs Fld Proc.1	
Check Fld Proc.1	
Crossval Proc.1	WALKIN.CUSTOMER.OVERRIDE WALKIN CUST ▼

PGM.FILE	WALKIN.CUSTOMER.DEFAULT
Type	S
Product	EB

PGM.FILE	WALKIN.CUSTOMER.OVERRIDES
Type	S
Product	EB

Using EB.TABLE.PROCEDURES

```
public class WalkInCustomerOverride extends RecordLifecycle {  
  
    public String checkId(String arg0) {  
  
    @Override  
    public void defaultFieldValues(String application, String recordId, TStruct  
        WalkinCustomerRecord WCR = new WalkinCustomerRecord(record);  
        if (WCR.getEmail().toString().isEmpty()) {  
            WCR.setEmail("abc@abc.com");  
            record.set(WCR.toStructure());  
        }  
  
    }  
  
}
```

St Walkin Customer Demo 456

Name

Email

abc@abc.com

Phone

Practice 6.2 – Service and Local Application

- Create a local application to hold the consolidated account balance of CUSTOMERS.
- Write a service to update the balance

Contract Screen

St Customer Balances
Demo *

Error message
No records matched the selection criteria

CUSTOMER BALANCES

Customer	Consolidated Balance
100105	589,905.00
100106	-60,085.00
100107	-80,090.00
100108	0.00
100352	0.00

| Lesson 7. Local Table

T24 Extensibility for Java



TEMENOS
Learning Community