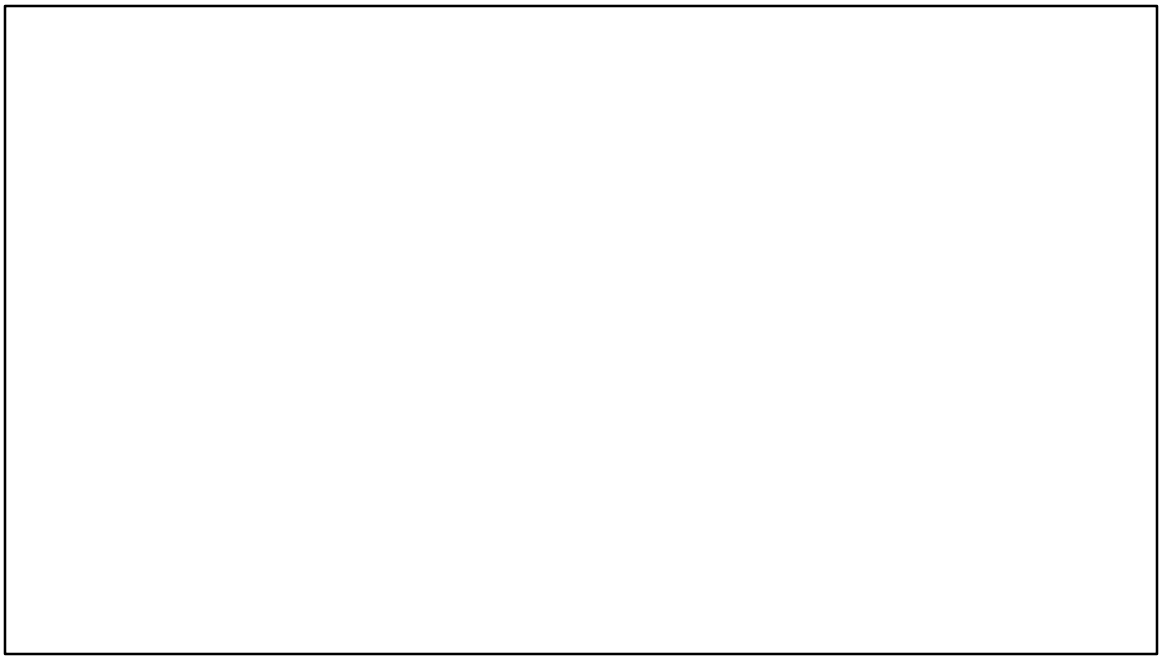


Deployment

Transact Extensibility for Java (English)

2020 Q1

©2020 Temenos Headquarters SA - all rights reserved.

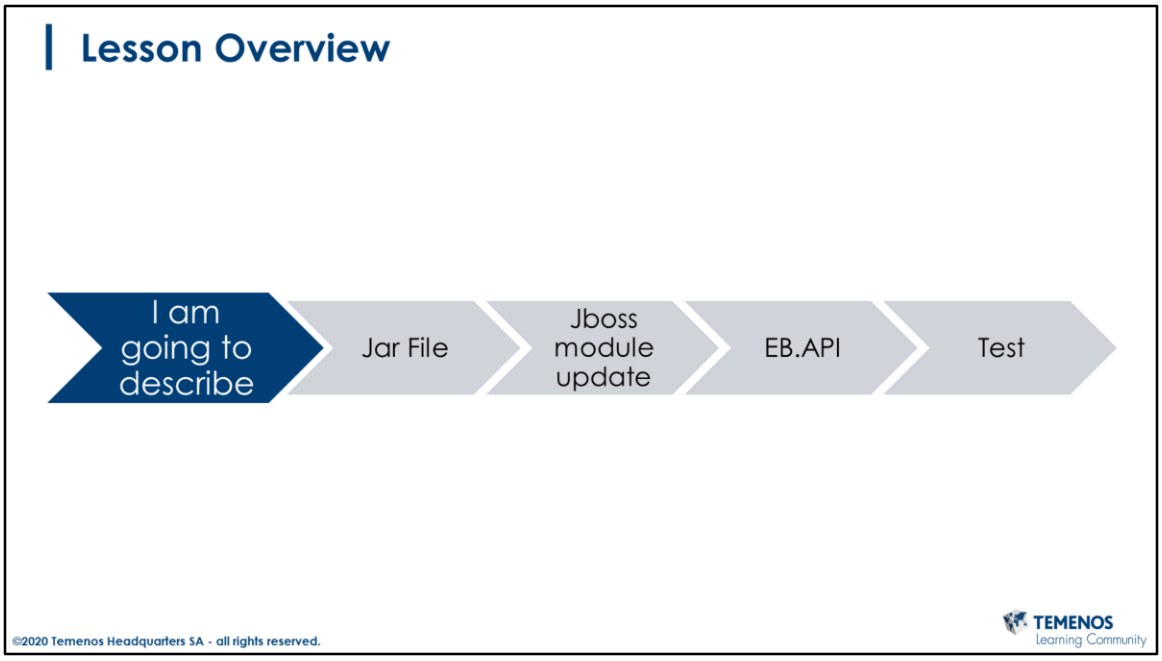
Copyright Notice

©2020 Temenos Headquarters SA - all rights reserved.

Warning: This document is protected by copyright law and international treaties. Unauthorised reproduction of this document, or any portion of it, may result in severe and criminal penalties, and will be prosecuted to the maximum extent possible under law.

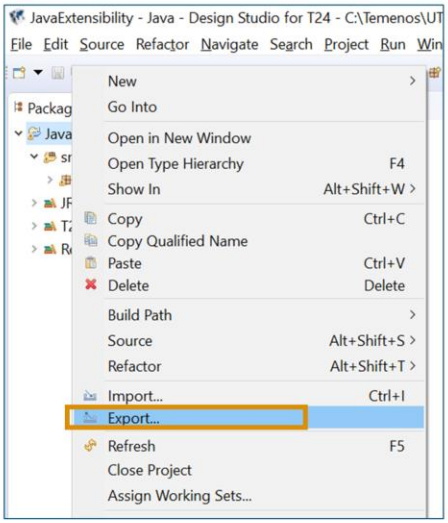
©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community



I am going to describeJar File
Jboss module update
EB.API
Test

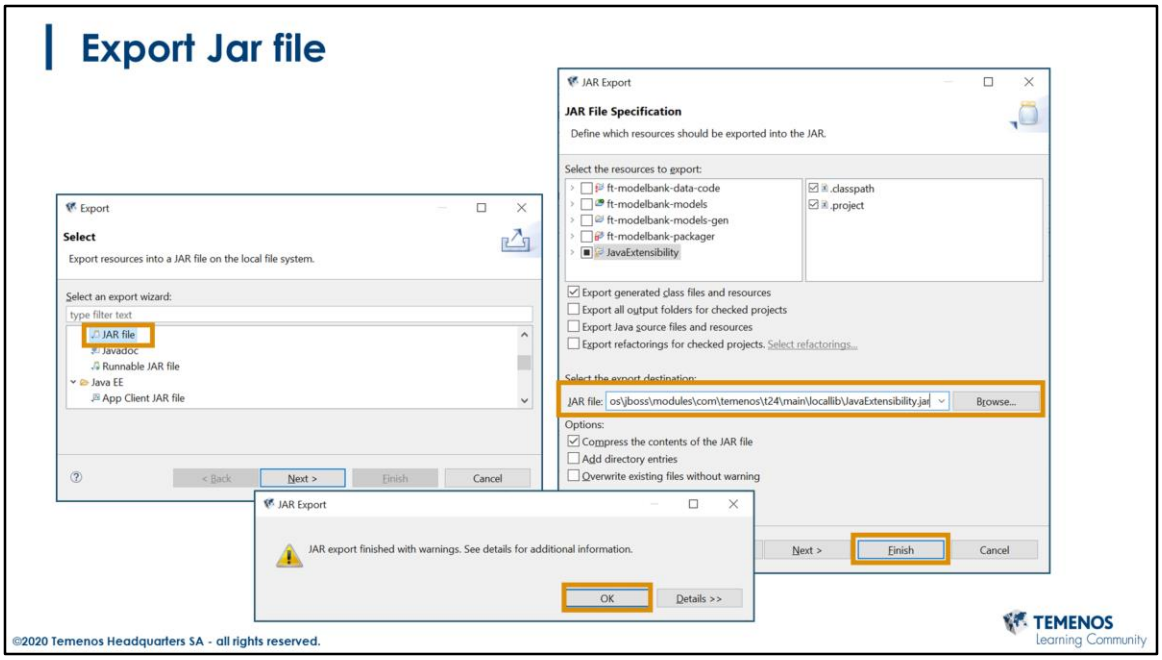
Export Jar File



©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

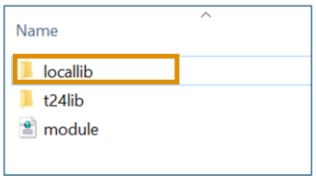
Right click on the project and select Export

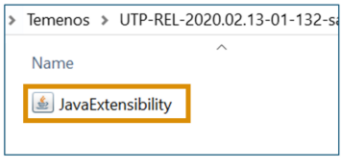


Save the jar file under Jboss modules

Update module.xml

- Add the jar file in module.xml






```
<resources>
  <!-- Insert resources here -->
  <resource-root path="./locallib/JavaExtensibility.jar" />
  <resource-root path="./t24lib/AA_ARAccountsData.jar" />
  <resource-root path="./t24lib/AA_ARC.jar" />
  <resource-root path="./t24lib/AA_AaActivityExtractorService.jar" />
```

©2020 Temenos Headquarters SA - all rights reserved.

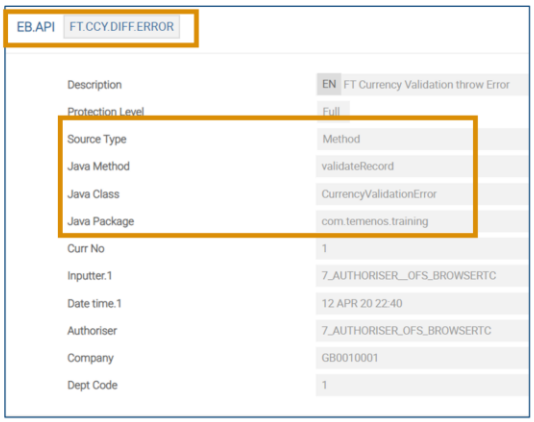
TEMENOS Learning Community

Make an Entry in module.xml with the jar file name (file inside JBOSS\modules\com\Temenos\t24\main)

Always add the new jar file name at the top

EB.API

- Hooks are invoked while transactions are validated or commited
- All the Hooks must have an entry in EB.API with the SourceType = method



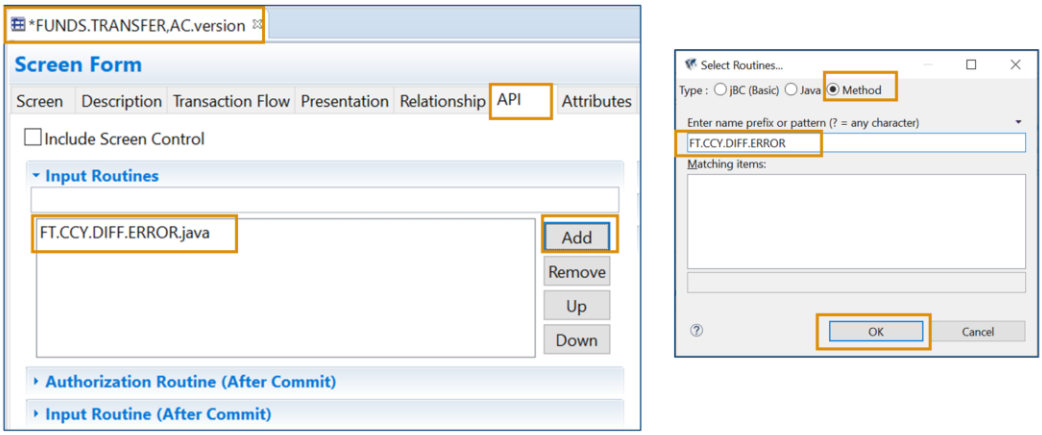
©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Hooks are invoked when a transaction is validated/committed
Input Hooks are attached to the field INPUT.ROUTINE in the VERSION application
All hooks must have an entry in EB.API

Source Type must be given as **method**
Java Method - name of the java method in the class
Java Class – name of the class file
Java Package - name of the package

Attach the EB.API to Exit Point

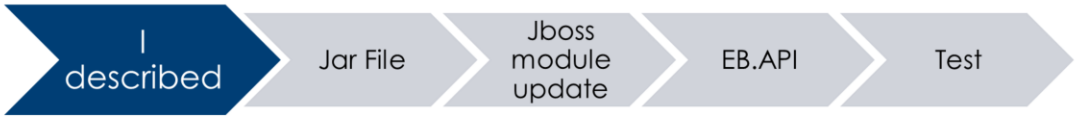


©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

In the API tab, Attach the EB.API Record created in the previous slide as java Method

Lesson Summary



©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Practice 6.1

- Create EB.API entry for the method implemented in Practice 5.1
- Add the above created EB.API ID to the Version Exit point

©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Practice 6.2 ValidateRecord

- Set an EB.ERROR or throw **Error** Message when the Credit Currency and Debit Currency are not same in FUNDS.TRANSFER transaction

©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Set an EB.ERROR or throw Error Message when the Credit Currency and Debit Currency are not same in FUNDS.TRANSFER transaction
Remember to create an EB.API record and attach the same to version as like the last practice.

Solution 6.2

- In this exercise, an **error** is raised when the debit and credit currency are not the same

```
public class CurrencyValidationError extends RecordLifecycle {

    @Override
    public TValidationResponse validateRecord(String application, String currentRecordId, TStructure currentRecord,
            TStructure unauthorisedRecord, TStructure liveRecord, TransactionContext transactionContext) {

        FundsTransferRecord ftRecord = new FundsTransferRecord(currentRecord); //Cast TStructure to Valid Application Record

        TField debitCurrency = ftRecord.getDebitCurrency(); // Read Debit Currency in a TField Object
        TField creditCurrency = ftRecord.getCreditCurrency(); // Read Credit Currency in TField Object

        if (!debitCurrency.getValue().equals(creditCurrency.getValue())) { // Compare the both the currencies. getValue() is used to
                                                                           // read value from the object

            creditCurrency.setError("FT-CCY.DIFF"); // Set the EB.ERROR ID
        }

        return ftRecord.getValidationResponse(); //Return the validated response
    }

}
```

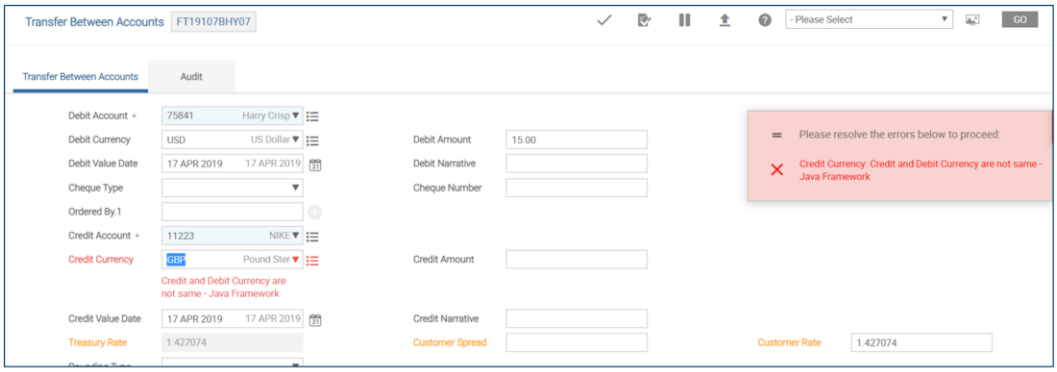
©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

In this exercise, an **error** is raised when the debit and credit currency are not the same. Creditcurrency.setError is passed the ID of the EB.ERROR record

Solution 6.2

- Error is raised when the debit and credit currency are not the same



©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Error is raised when the debit and credit currency are not the same

Practice 6.3 updateCoreRecord

- Update SHORT.TITLE of all accounts that belong to the customer with the customers SHORT.NAME when a CUSTOMER record is created/updated using the version CUSTOMER,JAVA.

©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Update SHORT.TITLE of all accounts that belong to the customer with the customers SHORT.NAME when a CUSTOMER record is created/updated using the version CUSTOMER,JAVA.

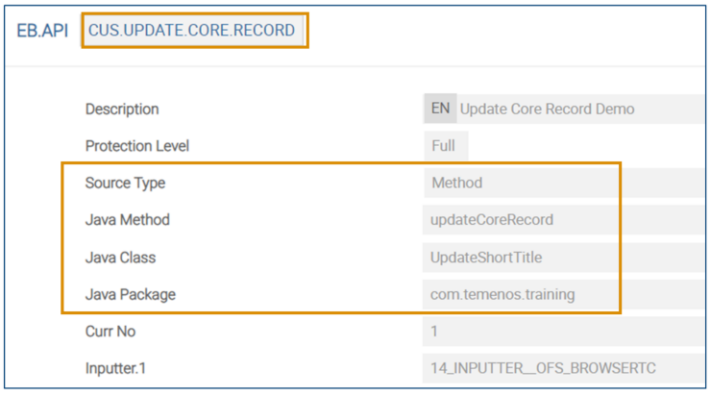
Solution 6.3

```
61  public void updateCoreRecord(String application, String currentRecordId, TStructure currentRecord,
62          TStructure unauthorisedRecord, TStructure liveRecord, List<String> versionNames, TBoolean isZeroAuth,
63          List<String> currentRecordIds, List<TStructure> currentRecords, TransactionContext transactionContext) {
64
65      Customer customer = new Customer(this);
66
67      CustomerRecord cusRec = new CustomerRecord(currentRecord); // Cast TStructure to CustomerRecord
68
69      try {
70          List<String> accountNumbers = customer.getAccountNumbers(); // Using the core API capture the list of
71                                                                       // account numbers belongs to the customer
72          for (String accNum : accountNumbers) {
73              AccountRecord accRec = new AccountRecord(); // Create empty AccountRecord
74
75              accRec.setShortTitle(cusRec.getShortName(0), 0); // Set Short title to the empty Account Record
76              versionNames.add("ACCOUNT,JAVA"); // Set the name of the version
77              currentRecordIds.add(accNum); // Set the Record Id
78              currentRecords.add(accRec.toStructure()); // Set the Record in TStructure format
79          }
80          isZeroAuth.set(true); // set the zero auth flag to true
81
82      } catch (T24CoreException e) {
83          throw new T24CoreException("Unable to update the Accounts");
84      }
```

©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

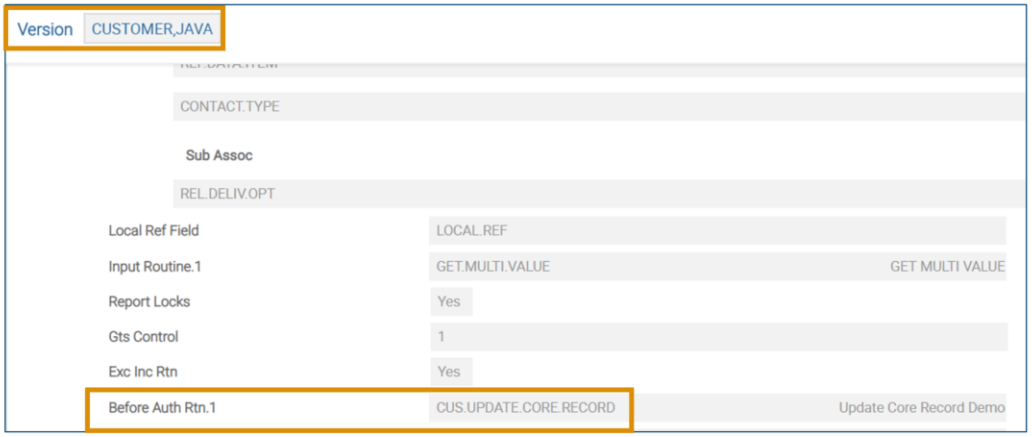
Solution 6.3 – EB.API



©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Solution 6.3 - Version



©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Practice 6.4 - postUpdateRequest

- Create an Account with below Information as result of Creating a Customer
  - Category is 6001
  - Currency is USD

©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Create an Account with below Information as result of Creating a Customer

Category is 6001

Currency is USD

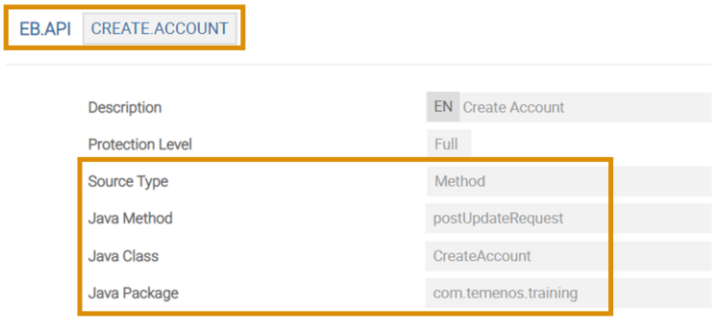
Solution 6.4

```
@Override
public void postUpdateRequest(String application, String currentRecordId, TStructure currentRecord,
        List<TransactionData> transactionData, List<TStructure> currentRecords,
        TransactionContext transactionContext) {

    AccountRecord accRecNew = new AccountRecord(); // Create a new Account Record

    accRecNew.setCustomer(currentRecordId);//Set Customer ID
    accRecNew.setCategory("6001"); // Set Category
    accRecNew.setCurrency("USD"); // Set Currency

    TransactionData txnData = new TransactionData();
    txnData.setFunction("INPUT"); // OFS Function
    txnData.setNumberOfAuthoriser("0"); // NO OF AUTH
    txnData.setSourceId("BULK.OFS"); // OFS Source ID
    txnData.setVersionId("ACCOUNT,JAVA"); // Version to use in OFS Request

    transactionData.add(txnData);
    currentRecords.add(accRecNew.toStructure());

}
```

©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Solution 6.4 – EB.API



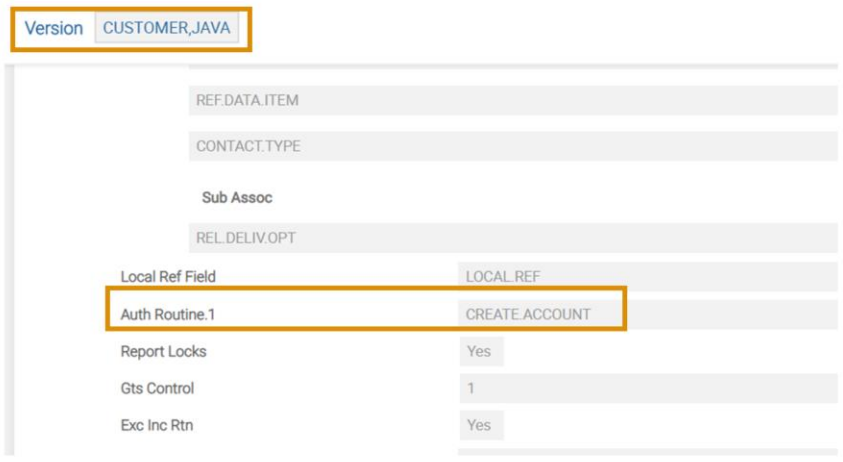
©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Create EB.API Record with below information

Source Type = method
Java Method = postUpdateRequest
Java Class = CurrentAccount
Java Package = com.Temenos.training

Solution 6.4 – AUTH.ROUTINE

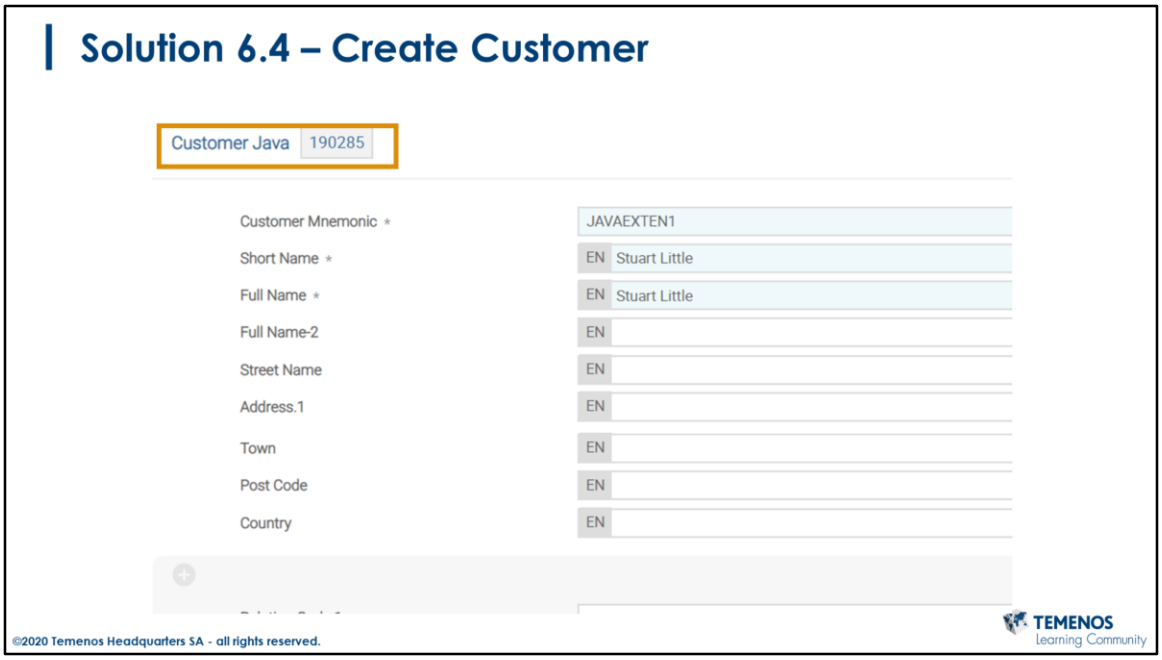


©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Attach the EB.API created as AUTH.ROUTINE
AUTH.ROUTINE = CREATE.ACCOUNT

Solution 6.4 – Create Customer

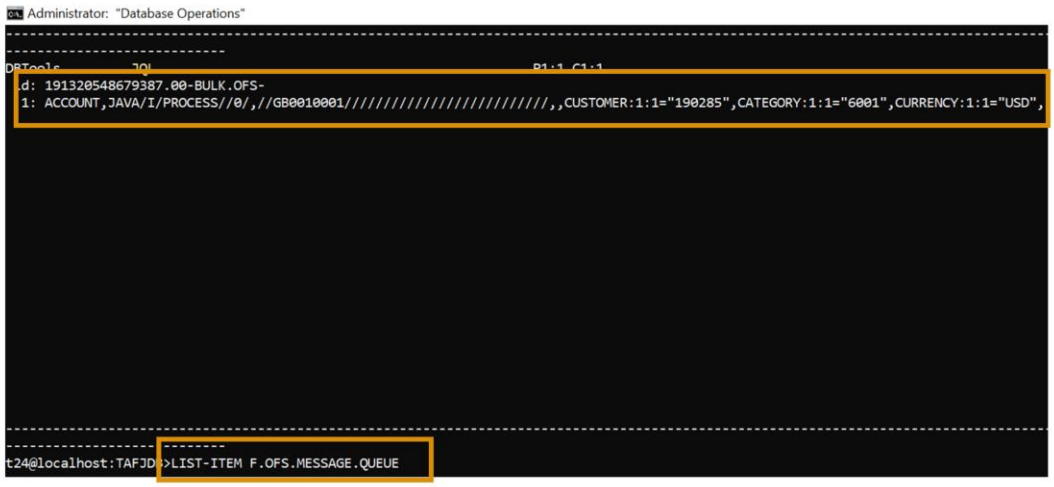


©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Create a customer using the version CUSTOMER,JAVA with mandatory inputs

Solution 6.4 – List OFS.MESSAGE.QUEUE

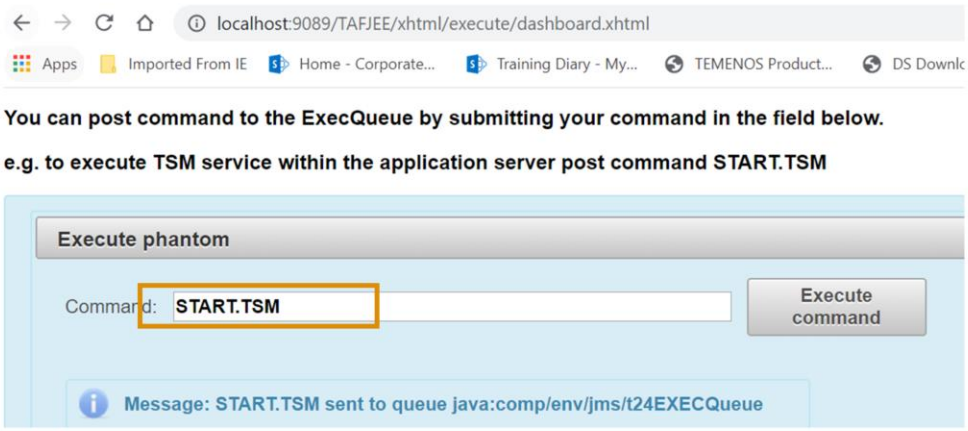


©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Once the CUSTOMER record is authorized, check the F.OFS.MESSAGE.REQUEST file to see the list of OFS Message Created.
Use DBTools console to view the above list

Solution 5.4 – Execute TSM

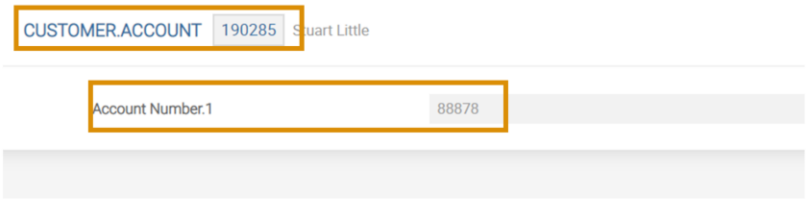


©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Start the TSM from TAFJEE Execute Servlet page

Solution 6.4 – Account Created



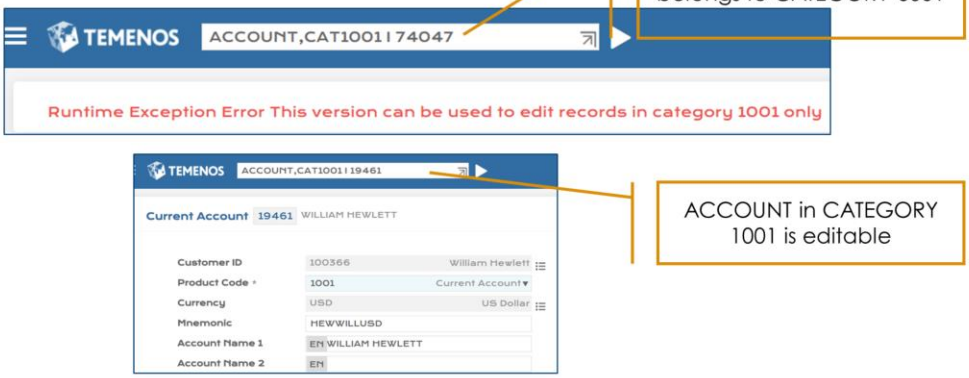
©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Account Created with the specifications give in the program

Practice 6.5 – defaultFieldValues

- Create a VERSION for ACCOUNT application that allows user to edit records belonging to CATEGORY 1001 only
- Expected Output



©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Create a VERSION for ACCOUNT application that allows user to edit records belonging to CATEGORY 1001 only

Practice 6.6 - checkId

- Create a VERSION for the CUSTOMER that prefixes "99" to the ID given by the user
- Expected Output

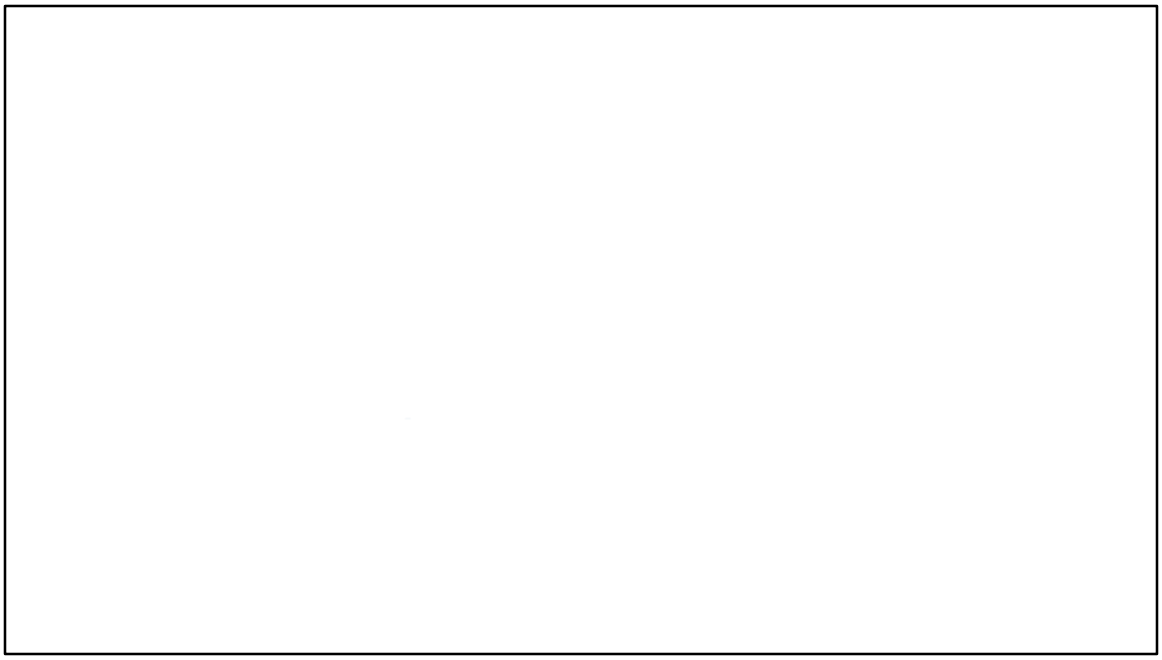
©2020 Temenos Headquarters SA - all rights reserved.

TEMENOS Learning Community

Create a VERSION for the CUSTOMER that prefixes "99" to the ID given by the user

TEMENOS

Learning Community

thank.you

tlc.temenos.com