openlegacy

COMPUWARE DATA SHEET

Plugin to Compuware Topaz

Overview

Compuware is the only software company solely focused on mainframe innovation. Compuware's Topaz allows mainframe developers to develop faster and leverage DevOps. OpenLegacy's API integration platform now comes as a plugin to Topaz. OpenLegacy's platform is the fastest way for mainframe applications to expose APIs externally by connecting directly to the system and automating code generation. With a couple of clicks, users generate consumable APIs. There is no hand-coding or additional configuration needed. The combination of Topaz along with the OpenLegacy plugin converts any Topaz visible legacy asset into a modern RESTful or SOAP microservice-based API in minutes.

Key Benefits of OpenLegacy

- Both tools support for the Eclipse platform makes a simpler API generation process from your existing environments
- Direct connection to any mainframe system including, CICS, IMS, and many more.
- Automatic code generation of java APIs that can be deployed as microservices.
- Parses metadata and generates SDK that includes runtime connection to the mainframe
- The ability to templatize the code generation during the SDK and/or API stage to support content like security, code standards, and even deployment methods
- Deployment into any infrastructure (Docker, PCF, Tomcat, etc.).

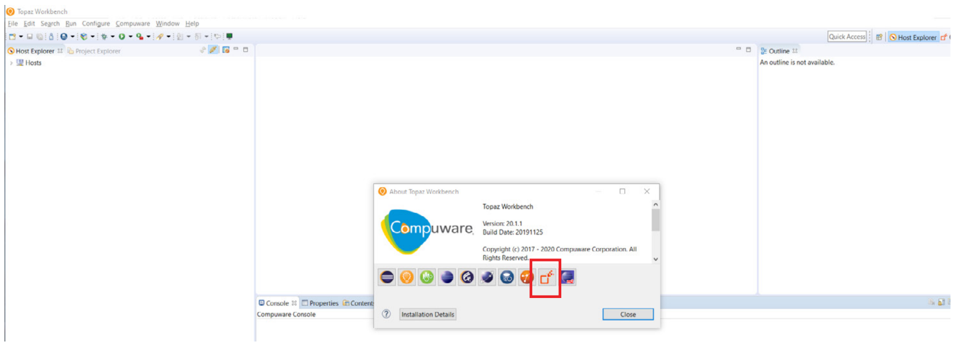
Key Benefits of Compuware

- Ability to design, build and view mainframe applications
- Integrated complete mainframe DevOps toolchain
- Powerful COBOL and other language debugging
- Instant visual summary of your program

How It Works

Topaz is Eclipse-based, and OpenLegacy's platform operates as an Eclipse plugin. This allows the OpenLegacy plugin to be installed into the Compuware Topaz IDE. The following are the steps to leverage both tools together to build and execute an integration.

1. **Verify the OpenLegacy plugin is installed inside Topaz and if not install it**



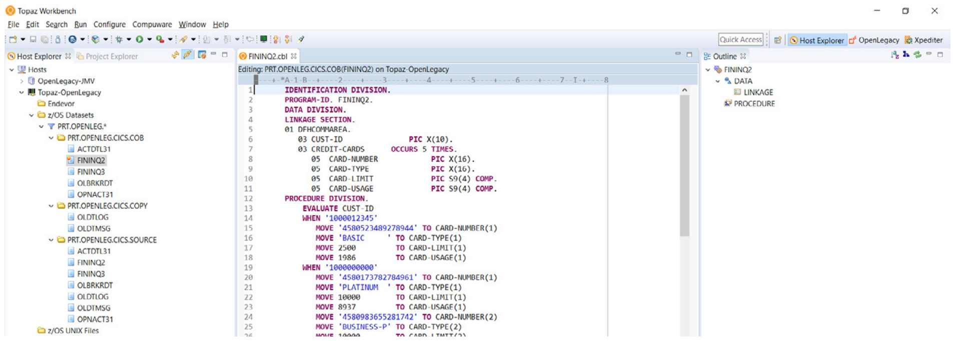
About Topaz Workbench

Topaz Workbench

Version: 20.1.1
Build Date: 20191125

Copyright (c) 2017 - 2020 Compuware Corporation. All Rights Reserved.

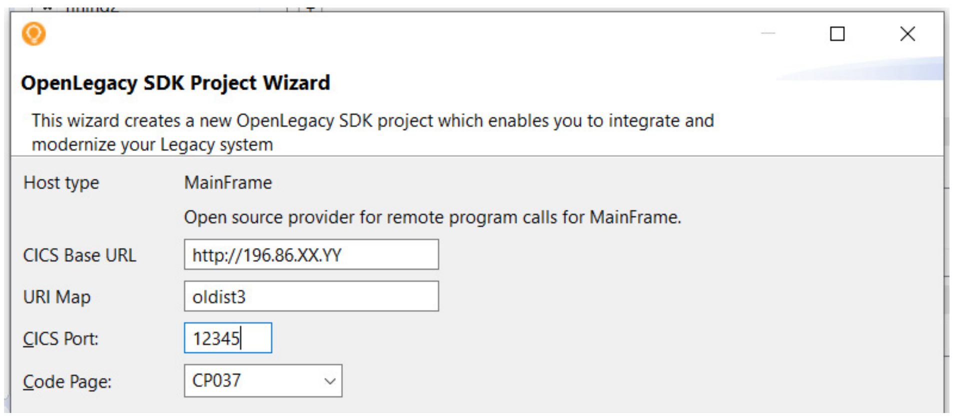
2. **Topaz allows for easy navigation to find the appropriate COBOL copybooks.** Find the copybook with the code you are looking to interface too.



Editing: PRT.OPENLEG.CICS.COB(FININQ2) on Topaz-OpenLegacy

```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. FININQ2.
       DATA DIVISION.
       LINKAGE SECTION.
       01 DFHCOMMAREA.
          03 CUST-ID                 PIC X(10).
          03 CREDIT-CARDS      OCCURS 5 TIMES.
             05  CARD-NUMBER             PIC X(16).
             05  CARD-TYPE               PIC X(16).
             05  CARD-LIMIT              PIC S9(4) COMP.
             05  CARD-USAGE              PIC S9(4) COMP.
       PROCEDURE DIVISION.
           EVALUATE CUST-ID
           WHEN '1000012345'
              MOVE '4580523489278944' TO CARD-NUMBER(1)
              MOVE 'BASIC      ' TO CARD-TYPE(1)
              MOVE 2500         TO CARD-LIMIT(1)
              MOVE 1986         TO CARD-USAGE(1)
           WHEN '1000000000'
              MOVE '4580173782784961' TO CARD-NUMBER(1)
              MOVE 'PLATINUM  ' TO CARD-TYPE(1)
              MOVE 10000        TO CARD-LIMIT(1)
              MOVE 8937         TO CARD-USAGE(1)
              MOVE '4580983655281742' TO CARD-NUMBER(2)
              MOVE 'BUSINESS-P' TO CARD-TYPE(2)
```

3. **Create an SDK project (container for the connection and data translation logic for mainframe data and business logic) in the OpenLegacy IDE and populate the connection configuration (host, port, username, pass, etc.).**

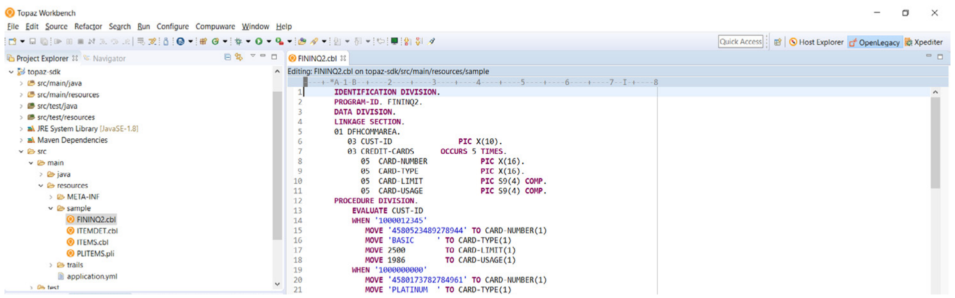


OpenLegacy SDK Project Wizard

This wizard creates a new OpenLegacy SDK project which enables you to integrate and modernize your Legacy system

Host type	MainFrame
	Open source provider for remote program calls for MainFrame.
CICS Base URL	http://196.86.XX.YY
URI Map	oldist3
CICS Port:	12345
Code Page:	CP037

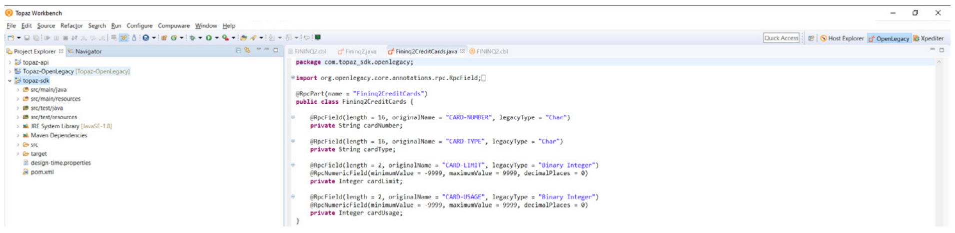
4. **Copy the selected copybook into this SDK project.** This code acts as a source for generating the Java-based SDK.



Editing: FININQ2.cbl on topaz-sdk/src/main/resources/sample

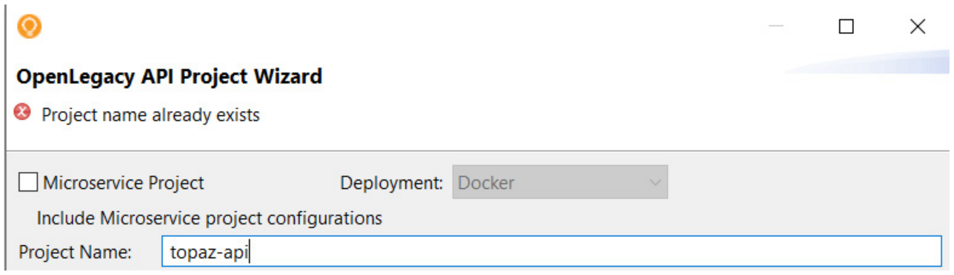
```
       IDENTIFICATION DIVISION.
       PROGRAM-ID. FININQ2.
       DATA DIVISION.
       LINKAGE SECTION.
       01 DFHCOMMAREA.
          03 CUST-ID                 PIC X(10).
          03 CREDIT-CARDS      OCCURS 5 TIMES.
             05  CARD-NUMBER             PIC X(16).
             05  CARD-TYPE               PIC X(16).
             05  CARD-LIMIT              PIC S9(4) COMP.
             05  CARD-USAGE              PIC S9(4) COMP.
       PROCEDURE DIVISION.
           EVALUATE CUST-ID
           WHEN '1000012345'
              MOVE '4580523489278944' TO CARD-NUMBER(1)
              MOVE 'BASIC      ' TO CARD-TYPE(1)
              MOVE 2500         TO CARD-LIMIT(1)
              MOVE 1986         TO CARD-USAGE(1)
           WHEN '1000000000'
              MOVE '4580173782784961' TO CARD-NUMBER(1)
              MOVE 'PLATINUM  ' TO CARD-TYPE(1)
```

5. **Generate SDK from the copybook. OpenLegacy automates this whole process, including the creation of the connection to the mainframe, translation of data, and any orchestration.**



```
package com.topaz_sdk.openlegacy;

import org.openlegacy.core.annotations.rpc.RpcField;

@RpcPart(name = "Fininq2CreditCards")
public class Fininq2CreditCards {

    @RpcField(length = 16, originalName = "CARD-NUMBER", legacyType = "Char")
    private String cardNumber;

    @RpcField(length = 16, originalName = "CARD-TYPE", legacyType = "Char")
    private String cardType;

    @RpcField(length = 2, originalName = "CARD-LIMIT", legacyType = "Binary Integer")
    @RpcNumericField(minimumValue = -9999, maximumValue = 9999, decimalPlaces = 0)
    private Integer cardLimit;

    @RpcField(length = 2, originalName = "CARD-USAGE", legacyType = "Binary Integer")
    @RpcNumericField(minimumValue = -9999, maximumValue = 9999, decimalPlaces = 0)
    private Integer cardUsage;
}
```

6. **Create an API project in the OpenLegacy IDE.** This is the project where the API will be generated into.



OpenLegacy API Project Wizard

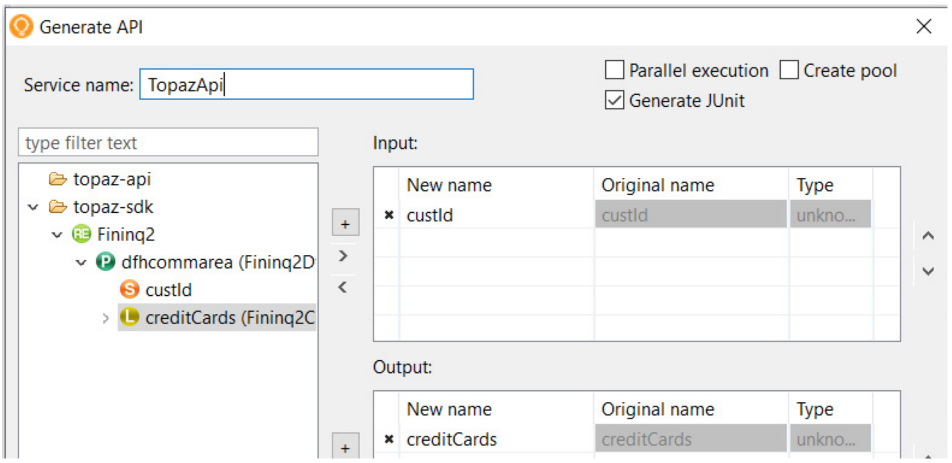
Project name already exists

☐ Microservice Project   Deployment: Docker

Include Microservice project configurations

Project Name: topaz-api

7. **Generate Services (APIs) based on the SDK you generated previously.** The API includes the actual interface and the code to call any SDKs.



Generate API

Service name: TopazApi

☐ Parallel execution ☐ Create pool

☑ Generate JUnit

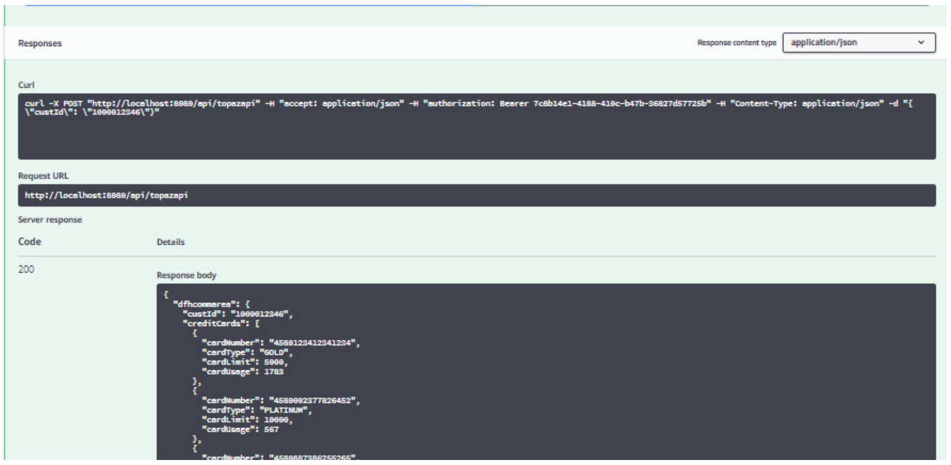
Input:

New name	Original name	Type
custId	custId	unkno...

Output:

New name	Original name	Type
creditCards	creditCards	unkno...

8. Test as Swagger Spec, deploy as .jar file, along with various other deployment methods.



Responses

Response content type: application/json

Curl

```
curl -X POST "http://localhost:8080/api/topazapi" -H "accept: application/json" -H "authorization: Bearer 7c8b14e1-4188-410c-b47b-36827d57725b" -H "Content-Type: application/json" -d "{
\"custId\": \"1000012346\"}"
```

Request URL

```
http://localhost:8080/api/topazapi
```

Server response

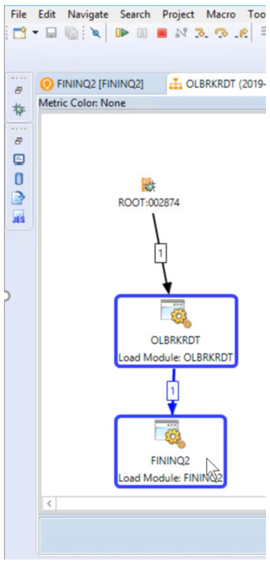
Code: 200

Details

Response body

```
{
  "dfhcommarea": {
    "custId": "1000012346",
    "creditCards": [
      {
        "cardNumber": "4580123412341234",
        "cardType": "GOLD",
        "cardLimit": 5000,
        "cardUsage": 1783
      },
      {
        "cardNumber": "4580002377826452",
        "cardType": "PLATINUM",
        "cardLimit": 10000,
        "cardUsage": 567
      },
      {
        "cardNumber": "4580087386255265",
```

9. Topaz supports debugging of both Java and Cobol code and run-time flowcharting of the Cobol execution on the mainframe



About OpenLegacy

OpenLegacy accelerates delivery of innovative digital services from legacy systems in days or weeks versus months. Our microservices-based API integration and management software reduces manual effort by automating API creation, simplifies the process by avoiding layers of complexity, and improves staff efficiency and API performance. Our software directly accesses and extends business logic to web, mobile or cloud innovations in the form of Java objects, REST APIs, ODATA APIs, or SOAP. Most importantly, this process is not only fast, easy and secure, but also does not require special staff skills or changes to existing systems or architecture. Together, business and IT teams can quickly, easily and securely meet consumer, partner or employee demands for digital services without altering or replacing core systems. Learn why leading companies choose OpenLegacy at www.openlegacy.com.

openlegacy
OPEN INNOVATION

www.openlegacy.com
sales@openlegacy.com