

Integration with Compuware Topaz In 10 Easy Steps



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

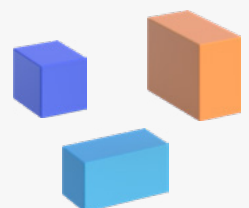
- Both tools support for the Eclipse platform makes a simpler API generation process from your existing environments
- Direct connection to any mainframe system including, CICS, IMS, and many more.
- Automatic code generation of java APIs that can be deployed as microservices.
- Parses metadata and generates 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

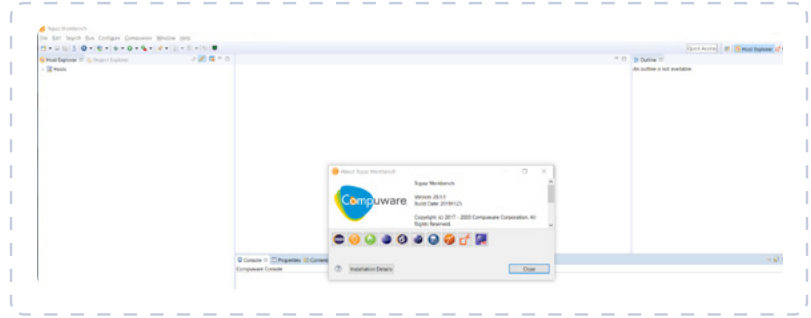
How It Works

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.



01

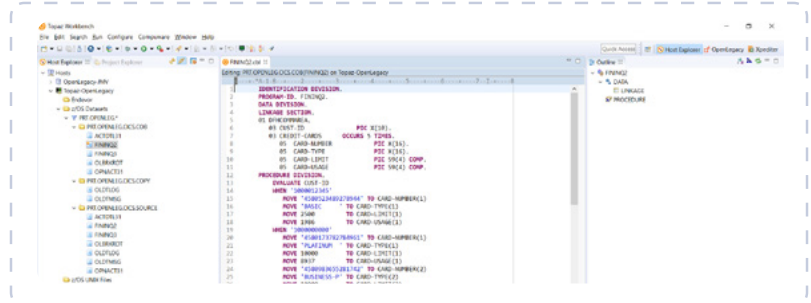
Verify the OpenLegacy plugin is installed inside Topaz and if not install it.



02

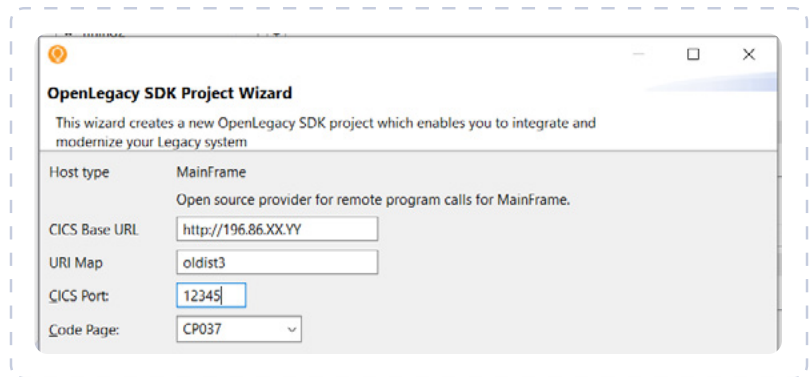
Topaz allows for easy navigation to find the appropriate COBOL copybooks.

Find the copybook with the code you are looking to interface too.



03

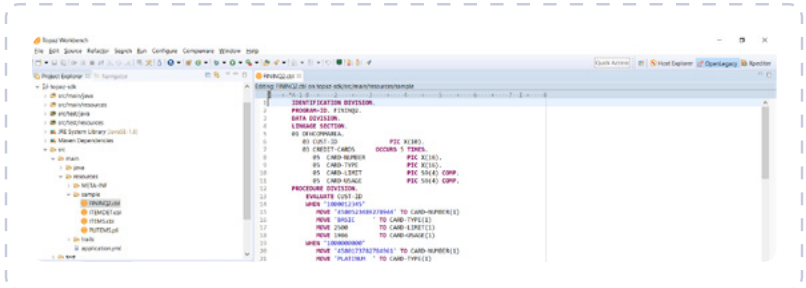
Create an SDK project (container for the connection and data translation logic for mainframe data and business process) in the OpenLegacy IDE and populate the connection configuration (host, port, username, pass, etc.).



04

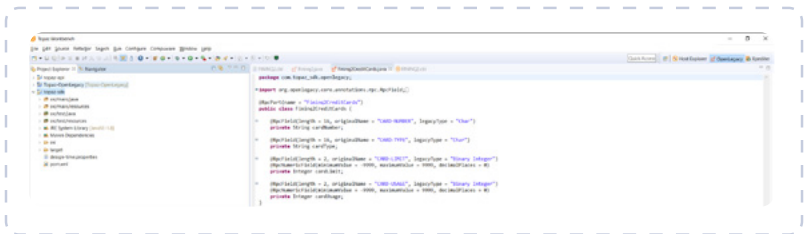
Copy the selected copybook into this SDK project.

This code acts as a source for generating the Java-based SDK.



05

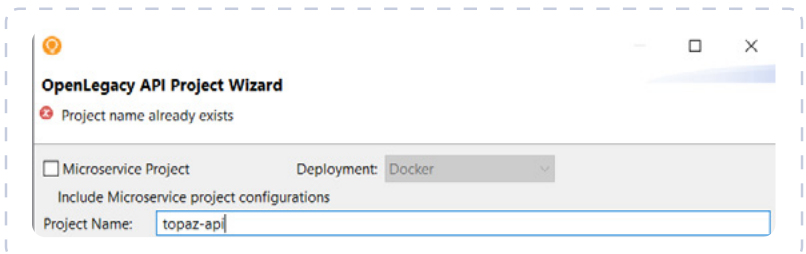
Generate SDK from the copybook. OpenLegacy automates this whole process, including the creation of the connection to the mainframe, translation of data, and any orchestration.



06

Create an API project in the OpenLegacy IDE.

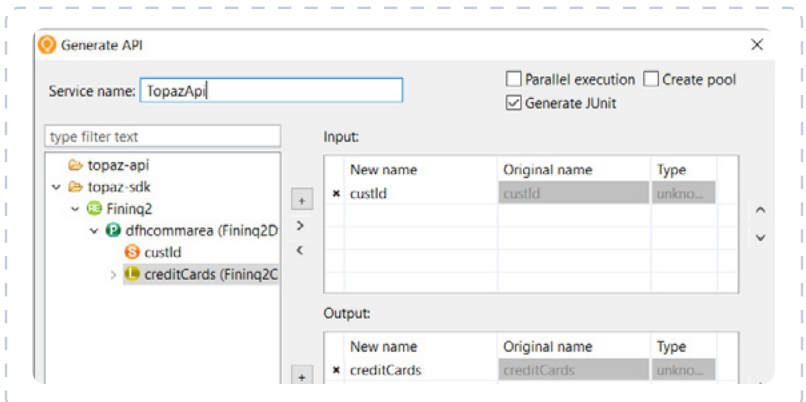
This is the project where the API will be generated into.



07

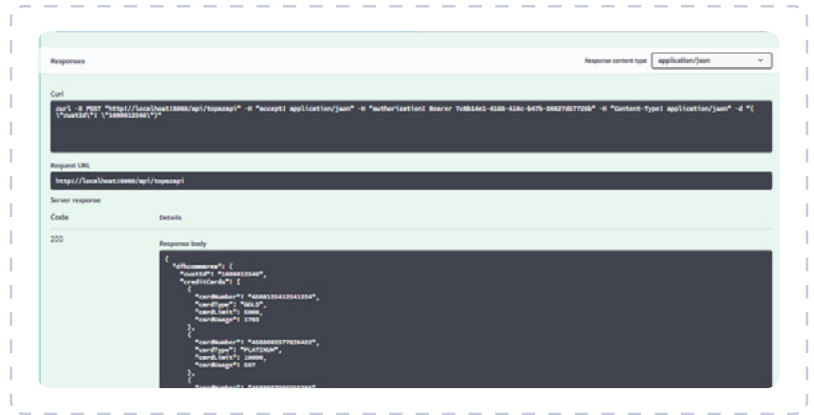
Generate Services (APIs) based on the SDK you generated previously.

The API includes the actual interface and the code to call any SDKs.



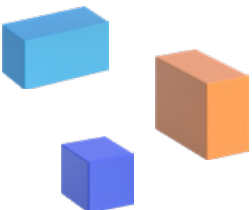
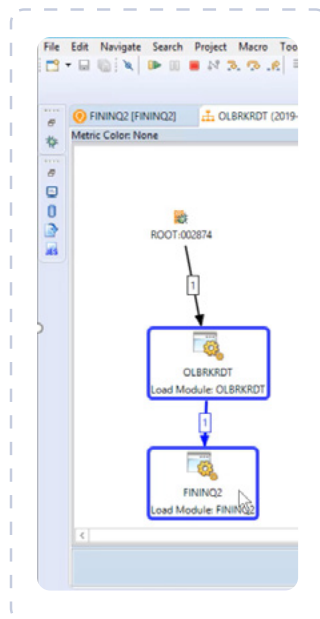
08

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



09

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



About OpenLegacy

OpenLegacy's Digital-Driven Integration enables organizations with legacy systems to release new digital services faster and more efficiently than ever before. It connects directly to even the most complex legacy systems, bypassing the need for extra layers of technology. It then automatically generates APIs in minutes, rapidly integrating those assets into exciting new innovations. Finally, it deploys them as standard microservices or serverless functions, giving organizations speed and flexibility while drastically cutting costs and resources. With OpenLegacy, industry-leading companies release new apps, features, and updates in days instead of months, enabling them to truly become digital to the core.

