OpenLegacy
Digital to the Core

Datasheet

CICS API Caller: Now Your CICS Mainframe Can Call an External API

Overview

The OpenLegacy platform already generates APIs from your mainframe to be consumed by cloud, web and mobile applications. Now with the OpenLegacy API Caller, you can natively call an external API from a CICS mainframe application. By using standard APIs to directly connect legacy systems to the digital world, OpenLegacy empowers the mainframe to be fully participate in any modern digital architecture.

Key benefits of API Caller

- Bypass complex middleware and get access to REST APIs
- Automate API initiation using pre-built connectors
- Reduce technical debt and total cost of ownership
- Automates the generation of mainframe code to call an API based on an API Specification
- At run-time, the entire transaction is a direct connection between the API CICS application
- The system is secure and requires authentication
- Users are managed via the application's internal user database
- No configuration needed on the legacy side
- Not locked into a single vendor solution

CICS applications need to contact the outside world

CICS applications are designed to deal with rapid high volume transaction processing and sometimes need to initiate a connection to other systems to gather more data or complete a transaction. In a world where most integrations use APIs, this has made it hard for mainframes to leverage external resources.

The challenge is in fully realizing your digital transformation while making the most of your existing CICS investment. That's exactly why OpenLegacy built an approach that helps companies like yours deliver seamless functionality to modern digital services, without complexity and middleware.

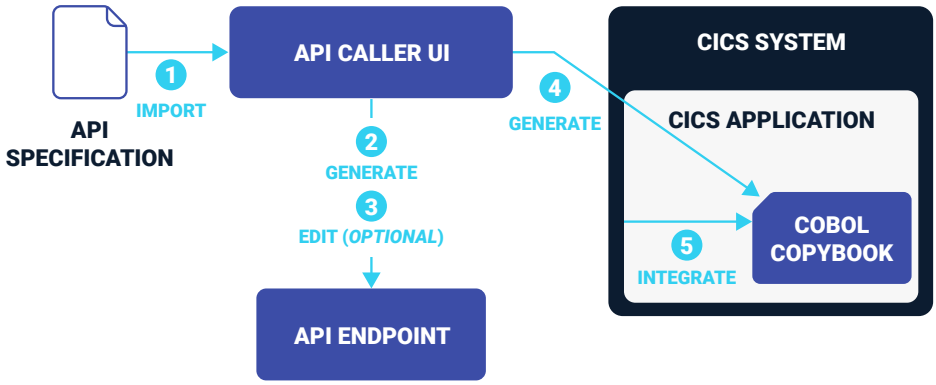
Your familiar system. And a whole new approach

The OpenLegacy platform doesn't require platform-specific expertise to generate COBOL code for use on the back-end. We simplify the process by analyzing the Swagger API definition and automatically generating the COBOL copybooks. Your developers integrate the code into their COBOL applications and the code takes care of initiating the call to the external API.

How it works: two main flows

Design time: retrieve information from the API and generate integration code

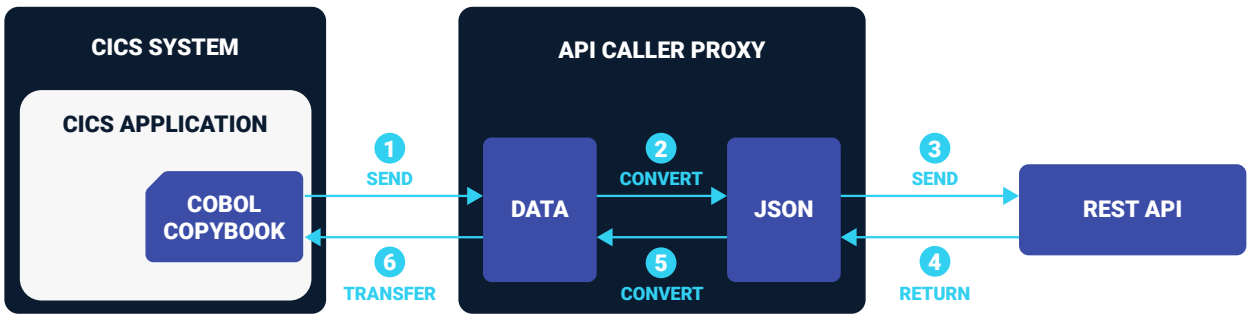
1. The user imports API specifications
2. The system generates API end-points from the Swagger specification
3. The user can edit the end-points
4. The user generates COBOL Copybooks from the end-points
5. The user integrates the Copybooks into their legacy applications



Design time flow importing the API spec and turning it into a COBOL Copybook

Run time: to connect from the CICS system through the API

1. The CICS system sends data to the API Caller proxy
2. The proxy converts data to a JSON request
3. The proxy sends the JSON to the REST API
4. The REST API returns a response and passes to it to the proxy
5. The proxy converts the response back into data
6. The proxy transfers the response back to the CICS application



Run-time flow for CICS Application call and response with a REST API

This table shows the Integration abilities of OpenLegacy's platform vs a traditional approach:

	API Caller	Traditonal Approach
Configuration		
Mainframe	Basic CICS resource definition	Requires dedicated LPARs, set up through the USS side of the mainframe and latest versions of all software components
Design		
IAM—authentication manager	Direct integration to the IAM authentication manager	Have to configure LDAP to get access to the CICS User Management
Front-end designer	Simple GUI for design	Requires use of complex command line instructions
Integration creation	Automatically generate COBOL copybook from swagger page	More files to integrate. Write a *.properties* file, generate and deploy data translation as a COBOL copybook, and server.*xml* files
Hand-coding	Not required	Need to write a communications stub inside the CICS system to handle the calls to the API
Runtime		
Proxy	Automatic data conversions handled outside the legacy system; scalable by adding another instance	Data conversions are done inside the CICS applications, adding an extra load to the system; solution doesn't scale since it is inside the CICS application

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.