openlegacy

AWS DATA SHEET

Integration with AWS

Overview

OpenLegacy's API integration platform is the fastest and most standard way for legacy applications to be part of the AWS cloud. OpenLegacy quickly and efficiently generates APIs or serverless nodeJS functions for any legacy asset by connecting directly to the legacy system, automating code generation. With a couple of clicks, users can generate a consumable APIs inside containers of Lambda functions. There is no hand coding or additional configuration needed for use with AWS.

Key Benefits of OpenLegacy

- Generates APIs as:
  - Java code for flexibility to deploy in any AWS End-point service: Beanstalk, EC2 or others
  - ECF/Fargate ready microservices
  - Serverless NodeJS code Lambda functions
- Direct Connection to almost any core system
- Automatic code generation of APIs inside microservices
- Parses metadata and generates SDK that includes run-time connection to legacy system
- Easily deployed into any infrastructure (Docker, PCF, Tomcat, Kubernetes, OpenShift)

Key Benefits of AWS

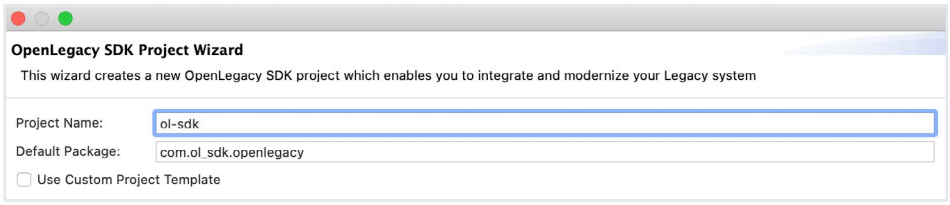
- Manage interaction with API consumers to optimize performance
- Wide deployment options for easy consumption, highly reliable Endpoint
- Security at the API and infrastructure layer to add protection to the legacy assets
- Monitoring, Multilayer Architecture, Auto Scaling (in and out), Load Balancing and several services are available to manage and control the APIs
- Services to produce Analytics at the API level

How OpenLegacy Works: OpenLegacy Can Deploy APIs to AWS in 3 Different Patterns

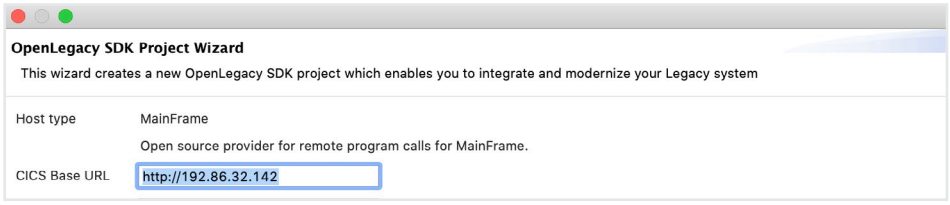
1. **Java POJO (Plain Old Java Object) API**
2. **As Microservice**
3. **Serverless Function**

For this example, we are showing how to deploy on Pattern 1 Java POJO API to EC2:

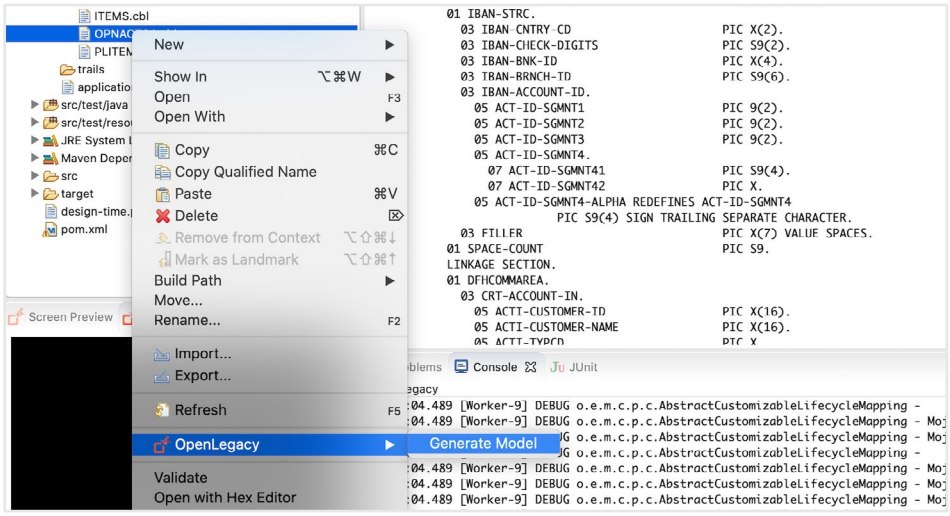
1. **Create an SDK project in the OpenLegacy IDE**



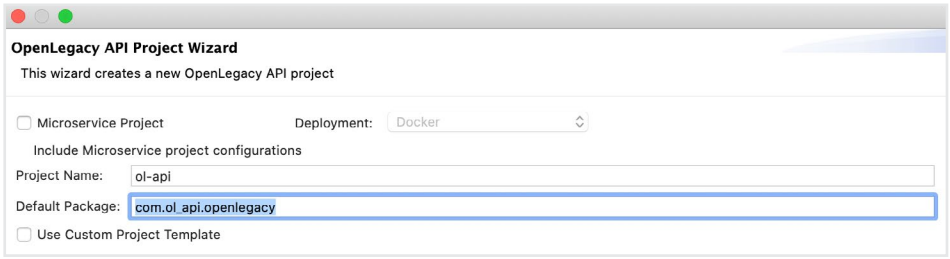
2. **Populate the connection configuration (host, port, username, pass, etc.).**
   This enables OpenLegacy to build the connection information about the backend into the SDK project. It also can retrieve any metadata from the legacy system for parsing.



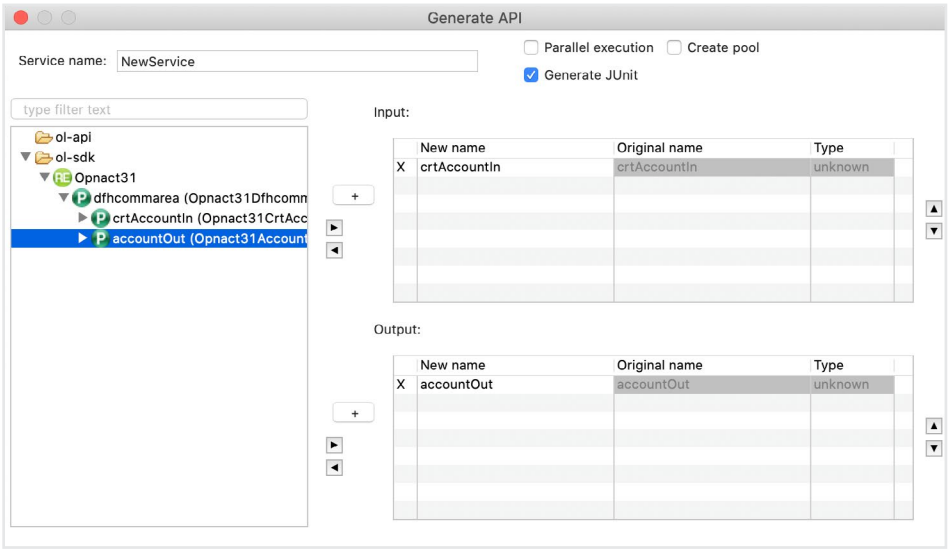
3. **Generate Java code based on metadata of back-end program.**
   The code goes into the SDK project for use by the APIs.



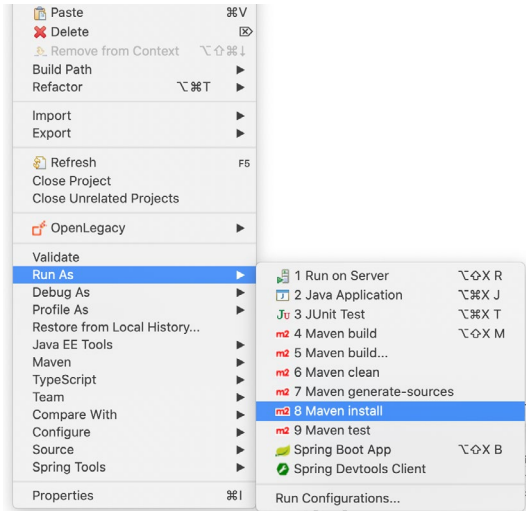
4. **Create an API project in the OpenLegacy IDE** – API data gets populated from the SDK



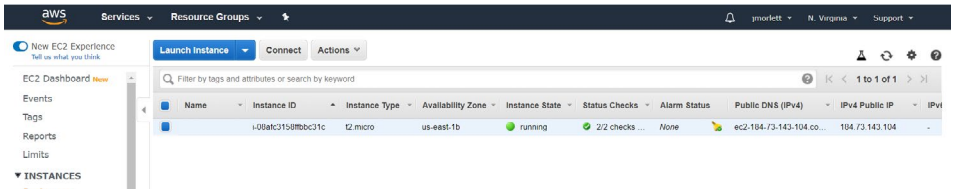
5. **Create API inputs and outputs based on the back-end asset generated into the SDK**



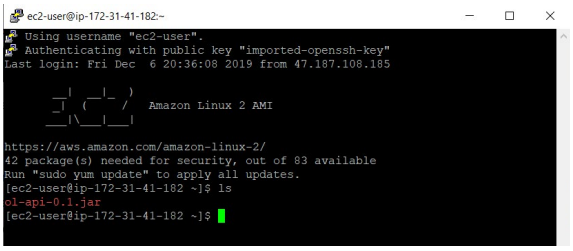
6. **Put the microservice-based API into a JAR file by choosing "Maven Install"**



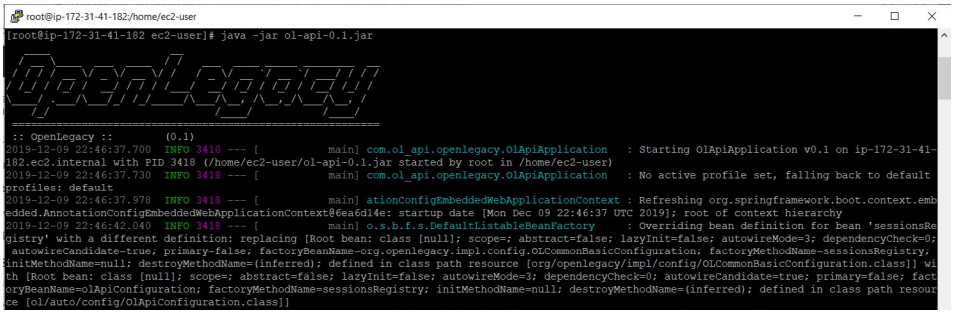
7. **Goto AWS and configure and Launch an EC2 Instance to deploy**



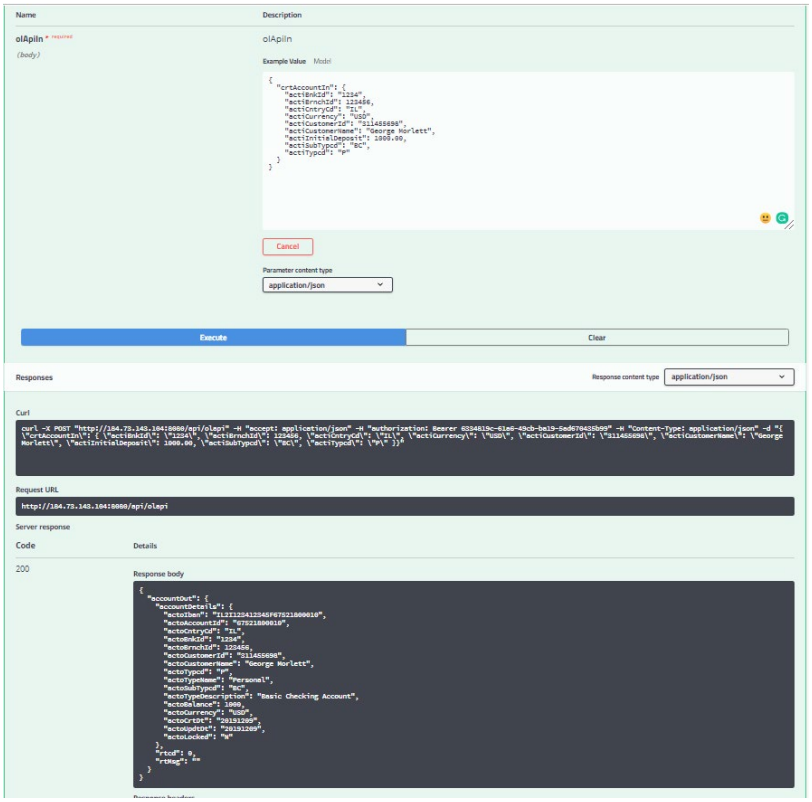
8. **Copy the Java POJO Jar file you created in OpenLegacy into the EC2 Instance**



9. **Execute the API inside the EC2 instance**



10. **Test the API using the OpenLegacy generated Swagger page**



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.