OpenLegacy

PRODUCT DATASHEET

OpenLegacy API Integration & Management for Legacy or Core Mission-critical Systems

Legacy or core mission-critical systems are the foundation of most of the world's largest and most successful organizations. However, system complexity can make it time-consuming and costly to innovate and compete with younger, nimble organizations with modern technology.

Regardless of your long-term legacy strategy – whether it be to modernize or migrate – your target markets are demanding new, innovative digital services now and will not be patient while you work it out.

There are various ways to extend your legacy systems (mainframe/midframe UI, applications and databases) to the web, mobile or cloud. Our microservices-enabled approach has proven far faster and superior to traditional enterprise integration, SOA, manual API development or using API software that was not designed for legacy systems.

Key Benefits

OpenLegacy's API Software for API Integration & Management revolutionizes the way legacy or core mission-critical systems are extended to create web, cloud or mobile innovations. Our patented – yet open standards – approach cuts through layers of complexity to forever improve and accelerate the way API integration and management is done.

While there are many approaches to API development and management, OpenLegacy specializes in the ones originating from legacy systems.

SPEED

- Automated generation of internal and external APIs from various legacy sources
- Immediate external API publication
- Generation of API clients for over 20 types of front-end languages
- Custom templates for full control of code generation
- Can be quickly customized for proprietary back-ends using connectors SDK
- Faster development, testing, and deployment to get to market faster
- Get new digital projects into production in weeks versus months
- Automated, continuous testing which not only improves quality but speeds production

SIMPLICITY

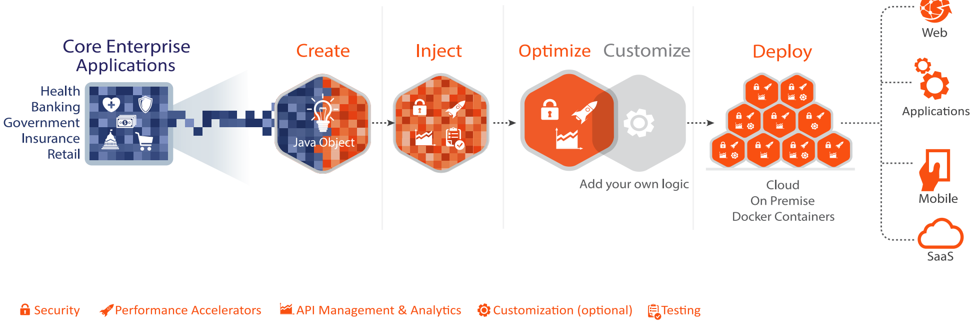
- Uses standard Java stack: Spring Boot, Maven, JPA, Swagger
- Flexible for any additional Java software stack. e.g. Jenkins, Nexus, any 3rd party libraries
- Easy, automated and standard deployment to J2EE, Docker, AWS, Bluemix, CloudFoundry etc.
- Out of the box publishing as REST, Java SDK, web services
- Designed for any skill level, standard Java skills needed for customization

SECURITY

- Compliant with all security standards
- Three types of security: architecture, secure layer and identity
- Java code is inherently strong-type security, no need for software JSON validations
- Supports OAath2 and LDAP
- Data-masking offers granular control over exposed data

OpenLegacy specializes in automating and accelerating the creation of APIs to extend legacy systems to digital services for the web, cloud or mobile innovations.

OpenLegacy API Integration & Management



OpenLegacy specializes in automating and accelerating the creation of APIs to extend legacy systems to digital services for the web, cloud or mobile. Our patented and automated approach, combined with pre-built connectors to most popular legacy systems, significantly reduces the time required to create APIs. A developer with standard java skills can create an API in several minutes, which can be deployed as a digital service usually in days or weeks.

Key Features

API CONNECTORS

In just minutes using pre-built legacy connectors, you can automatically extract strong-type metadata from any legacy or core mission-critical system (including data requiring business logic) to create a standard Java API. When deployed as either WAR or JAR files, our patented technology also enables these connectors to perform at extremely high performance levels, measured at up to 10X faster.

Types of Core (Legacy) Connectors – The Java API is a generated code combining the connector and Java models (also called entities). An entity can describe the input and output parameters of legacy or core mission-critical systems such as applications, mainframe/midrange servers, databases and SOA/ESB services. Entities can invoke the program, knowing what to send and what we expect to receive.

Connector Automation – The software will analyze and generate entities automatically from the target system, whether it is legacy source code, a trail file describing screens, or even a direct connection to the database or application.

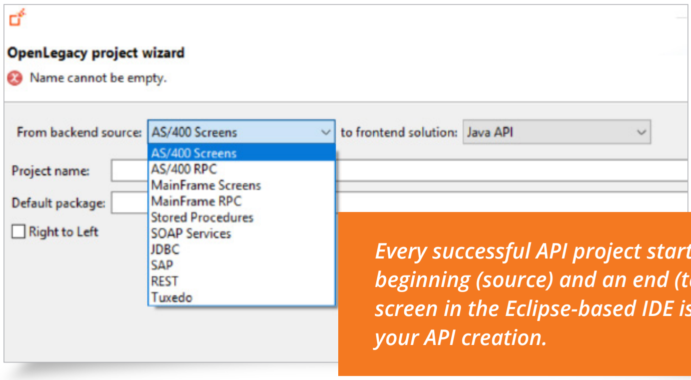
API DESIGN

Using built-in templates and wizards, automatically extend backend entity objects into REST APIs, SOA web services and deploy to containers or as microservices.

API Modeling & Configuration – Automatically create the external API in a contract first approach (top down) or using the existing models from the internal Java API. The API can be customized with graphical editors or modified directly.

This generated Java code can be deployed as a fully working API to any supported Java Web Server. The API includes an API catalog and test page allowing the API to be documented, tested and delivered.

Eclipse-based IDE – Available as a plug-in or stand-alone installation, not only does our IDE provide a graphical interface to create and configure APIs, but it also generates Java code behind the scenes. No special skills or training are required and it's easy to alter or add to your code for more complex or custom projects.



OpenLegacy project wizard

Name cannot be empty.

From backend source: AS/400 Screens   to frontend solution: Java API

Project name:

Default package:

□ Right to Left

Every successful API project starts with a beginning (source) and an end (target). This screen in the Eclipse-based IDE is the start of your API creation.

Samples of Connectors

Below is a sampling of our pre-built, automated connectors

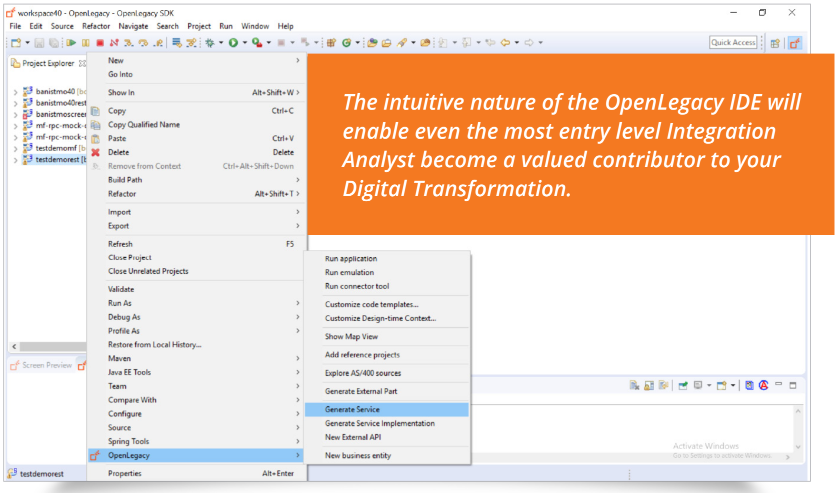
Platforms
Mainframe, AS400/IBMi, Unisys, Tandem

Programming Languages
COBOL, RPG, Assembler, Natural, PL1

Database
JDBS, VSAM, ADABAS, IMS, DB2, Datacom

Applications
SAP, Hogan, Altamira, Oracle Forms

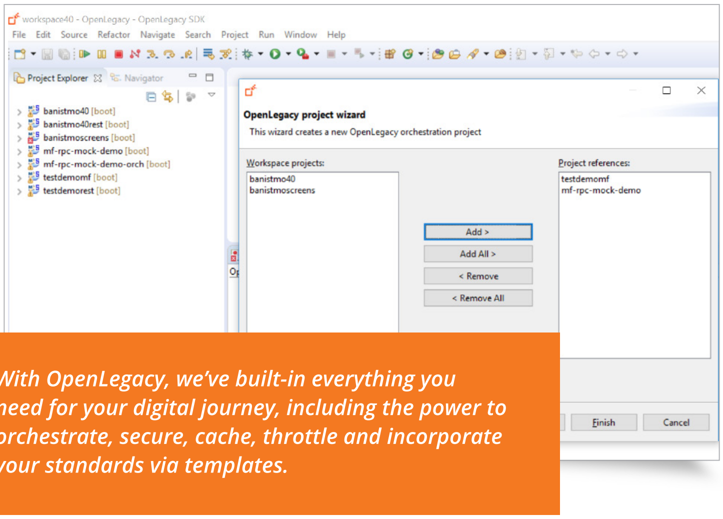
Other
ESB's, Tuxedo, CICs



The intuitive nature of the OpenLegacy IDE will enable even the most entry level Integration Analyst become a valued contributor to your Digital Transformation.

Orchestration – in addition to basic projects where there is one front-end and one back-end, you can orchestrate integrations to any number of back-ends. If the goal is to create an API that relies on Java without any integration, no back-ends are needed. Unlike traditional approaches, OpenLegacy uses dependency-based orchestration allowing much greater flexibility and simplicity in implementation.

To add multiple back-ends, an OpenLegacy orchestration project actually uses multiple basic projects with JAVA API front-ends. These projects are responsible for the integration with the relevant back-end in each, and exposes it as a JAVA API that OpenLegacy can later use when creating the services. These projects are added to the orchestration project as reference projects.



OpenLegacy project wizard

This wizard creates a new OpenLegacy orchestration project

Workspace projects:
banistmo40
banistmoscreens

Project references:
testdemomf
mf-rpc-mock-demo

With OpenLegacy, we've built-in everything you need for your digital journey, including the power to orchestrate, secure, cache, throttle and incorporate your standards via templates.

OpenLegacy Software Technologies

Zuul

Eureka

Sleuth

Hystrix

Spring Boot

Spring Cloud Foundry

Spring Security

Turbine

Zipkin

Elastic Search

Kibana

SonarQube

Nexus

Jenkins

Netflix Framework

Swagger

API ENHANCEMENT

Our API software enables you to enhance the security and performance of your APIs in a wide variety of ways. You can also add custom requirements as needed.

Security – You can inject three types of security into your API.

1. **Architecture Security –** Since artifacts are Java, they inherently feature strong-type security so no additional software-based validation layers are needed. In addition, OpenLegacy connectors can access their respective legacy data sources using secured connections. While certificates are supported in the host properties file for encrypted SSL/TLS communication, a VPN connection can also be established between the server and the legacy system, providing a highly secured communication method.

2. **API-specific security –** Sits on top of the API and provides granular control of who uses any specific API, plus the field and content of the API (for example, 'get customer details' - 1 see VIP customers; 1 only see account, 1 see financial info). OpenLegacy developed this security specifically for APIs and includes it in one convenient solution.

3. **Identity Security –** OAuth2 separates the roles of the client and resource owner, thus allowing secure token-based, profile-defined access to resources. A built-in data-masking wizard, can mask any core (legacy) data source to provide only specific data needed for a specific service. Other data will not be exposed. Identity management authentication supports LDAP and active directory.

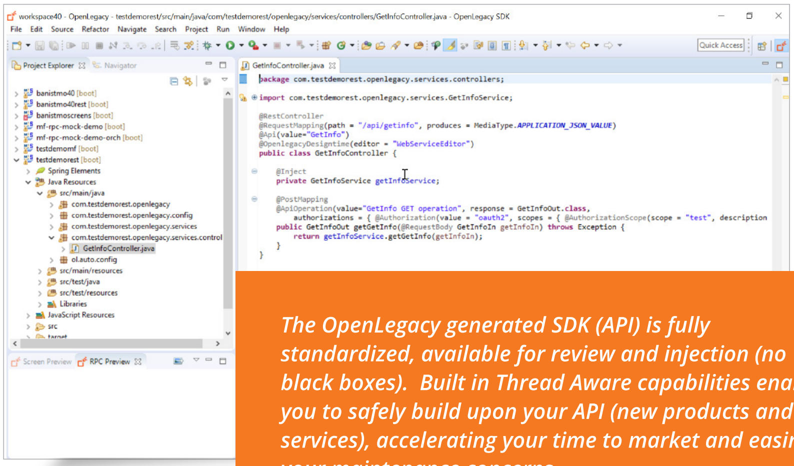
4. **API Bundles –** A bundle is a way to group similar APIs together and give them common configurations. A bundle can also be packaged and sold to API customers, such as third parties, partners, Fintech, etc.

Performance – Unlike other approaches, we're architected for speed. OpenLegacy's architecture flattens the entire integration process and provides a self-contained, automatically generated micro-application that encapsulates everything needed for the service. This approach dramatically reduces latency for both REST and SOAP services.

A huge global bank used openlegacy to implement six key global APIs and microservices in just two weeks

Unlike other vendor approaches, we're architected for speed. OpenLegacy's architecture flattens the entire integration process and provides a self-contained, automatically generated micro-application that encapsulates everything needed for the service.

Flexible Deployment Options – Including on-premise, Cloud, or any J2EE server or Docker container. Deploy to a developer portal using OpenAPI, the standard open source format to enable consumption of your APIs.



```java
package com.testdemorest.openlegacy.services.controllers;

import com.testdemorest.openlegacy.services.GetInfoService;

@RestController
@RequestMapping(path = "/api/getinfo", produces = MediaType.APPLICATION_JSON_VALUE)
@Api(value="GetInfo")
@OpenlegacyDesigntime(editor = "WebServiceEditor")
public class GetInfoController {

    @Inject
    private GetInfoService getInfoService;

    @PostMapping
    @ApiOperation(value="GetInfo GET operation", response = GetInfoOut.class,
        authorizations = { @Authorization(value = "oauth2", scopes = { @AuthorizationScope(scope = "test", description
    public GetInfoOut getGetInfo(@RequestBody GetInfoIn getInfoIn) throws Exception {
        return getInfoService.getGetInfo(getInfoIn);
    }
}
```

The OpenLegacy generated SDK (API) is fully standardized, available for review and injection (no black boxes). Built in Thread Aware capabilities enable you to safely build upon your API (new products and services), accelerating your time to market and easing your maintenance concerns.

OpenLegacy is microservices-enabled, combining the best of microservices with the best of API design and deployment.

Microservices – OpenLegacy is microservices-enabled so you can deploy APIs as microservices. A microservices architecture structures an application as a collection of loosely coupled services – an approach that leaves legacy systems intact for as long as you wish, yet offers freedom to use the microservices for rapid innovation and deployment anywhere, especially the Cloud. For more details, see www.OpenLegacy.com/API-Software/Microservices.

Hybrid Cloud Integration – Designed to connect anything, anywhere. Front-end SOAP and REST services generated from your legacy, on-premise systems are seamlessly integrated with cloud-based applications. Leverage existing data and functionality within new business models (SaaS, PaaS, BYOD, Bid Data, etc.) at a fraction of the cost and time required to develop new systems.

Since all APIs and microservices are built with standard Java frameworks, process automation and DevOps is easy and standard.

Standard Java Frameworks – Since all APIs and microservices are built with standard Java frameworks (such as Spring, Maven, Git, Jenkins, and CloudFoundry), process automation and DevOps is easy and standard. No proprietary deployment, testing or versioning solutions are needed.

OpenLegacy legacy API software is microservice-enabled

IT Benefits

- Improve modularity making an application easier to understand, develop and test
- Reduce interdependencies, so updates don't have unintended consequences
- Enables small teams to develop, deploy and scale services independently
- Supports agile, DevOps, continuous delivery and deployment
- Enables large enterprises to create a global API strategy
- Allows more efficient and cost-effective scaling of specific services
- Accelerates future legacy migration projects since APIs are already microservices

OpenLegacy

Business Benefits

- Build new products and offerings at a fraction of the time & cost
- Quickly iterate and evolve to changing market needs
- No more house of cards where one innovation leads to other delays
- Transform IT from a business anchor to an engine accelerating forward
- Reduce the cost & time required for larger legacy migration initiatives

API TESTING

Keep your pace and avoid delays with built-in JUnit testing and continued testing in run-time.

Accelerated, risk-free testing – since OpenLegacy APIs do not interfere with the underlying legacy or core mission-critical system, changes can be made easily and quickly without effecting the legacy or core mission-critical system.

Since OpenLegacy APIs do not interfere with the underlying legacy or core mission-critical system, changes can be made easily and quickly.

Continuous improvement – Automatically perform testing in both build-time and run-time for quality assurance and continuous improvement before, during and after deployment. This is especially important for legacy or core mission-critical system APIs supporting mission critical processes and millions of transactions.

Automated testing – OpenLegacy can use automation servers such as Jenkins to automatically run tests to proactively alert you if a run-time API is not functioning as planned.

API management – Easily manage your APIs, monitor who is using them, how they are used and where they are used.

Management console – the console enables real-time control over the performance and behavior of the API.

The console provides users to manage:

- Roles and role reports
- Properties
- Logs and Analytics
- Services usage
- RESTful services
- Microservices
- API usage
- Caching

Analytics – Follow the entire 'back and forth' journey of each message, from endpoints to back-end. Observe internal and external usage, including the number of calls, duration, date etc.

Analytics and usage data on both the API and the back-end SDK levels are available for viewing in real time through a rich dashboard with advanced search and filter capabilities.

Analytics and usage data on both the API and the back-end SDK levels are available for viewing in real time through a rich dashboard with advanced search and filter capabilities.

Need help getting started?

OpenLegacy offers a wide variety of services engagements to help jump-start your initiative and address common questions such as how to deal with a current API vendor that is not providing the speed and convenience you need, identifying and delivering the first key APIS, staff training and more.
www.openlegacy.com/services

Need proof?

Please check out our many detailed case studies spanning multiple industries. Since many of our clients are global mega companies, we are not allowed to mention them by name. However, reference calls can be arranged when the time is right. **www.openlegacy.com/case-studies**

About Us

OpenLegacy helps organizations quickly launch innovative digital services by extending their legacy or core mission-critical systems to the web, mobile and cloud in days or weeks versus months. Our API software quickly reduces project backlog by automating and accelerating API creation, deployment, testing and management from core applications, mainframes and databases. Together, business and IT teams can quickly, easily and securely meet consumer, partner or employee demands for digital services without modernizing or replacing legacy or core mission-critical systems, and without special programming skills or invasive changes to existing systems and architectures. Learn why leading companies choose OpenLegacy at **www.openlegacy.com**.

OpenLegacy Inc.
11921 Freedom Drive, Suite 550
Reston, Virginia, 20190

Phone: 609-608-0556
Email: info@openlegacy.com

Switzerland (Europe)
Rue Etienne Dumont 1
Geneva, 1204
Switzerland

Mexico
Av. Insurgentes Sur #730,
Col. Del Valle,
Delegación Benito Juárez, Piso 2
México, DF. CP 03104

Israel
3 Mota Gur
Olympia Park
Petah Tikva , Israel

OpenLegacy