OpenLegacy
The Business End of Integration

Sales Battle Card

API Management

How API can help transform your business

- For organizations who are ready to transform into fast-paced, customer-centered businesses, having mission-critical data and processes available when and where needed is critical.
- APIs offer a strategic technology because they are not sensitive to where those data or processes come from – they focus on the interface.
- The stability, maintainability and security of utilizing risk-free open standard technology for exposing the required data and processes, in standard JSON and REST protocol.

Planning & Design

The OpenLegacy API integration platform is primarily intended to expose existing backend assets as APIs. For this end, OpenLegacy provides a Modeller tool which can be used to record existing business process interactions in order for them to be automatically delivered as APIs.

Implementation

Once the user imports a backend asset such as source code, WSDL file, Database table etc. into the IDE, the OpenLegacy platform automatically parses the backend asset and creates a strong typed API based on it. Considerations such as field names, data types etc. are also resolves automatically. The result is an API fully designed and usable based on an existing business process.
Since the result is a completely standard Java code, the user can further customize and introduce additional logic to the API. Note that these customizations can be pre-configured by using templates.

Deploy and Run

Once the API is generated in the IDE successfully, it can then be deployed to the a server.

Deployment options are very flexible:

- OpenLegacy on-prem server - This is OpenLegacy's server, based on Apache Tomecat technology
- Multiple Java application servers - Jboss, IBM Liberty Profile and many other Java app servers are also supported as deployment options
- Public Cloud Deployments - OpenLegacy supports deployments to IBM Bluemix and Amazon AWS, as well as OpenLegacy's own hosting solution
- Cloud Containers - OpenLegacy supports deployment to a Docker based container for private or public clouds
- Hybrid Deployments - With strong hybrid features such as the OpenLegacy Secure Gateway, hybrid deployments are automatic and can include on-prem, private and public cloud

Runtime features include:

- oAuth 2.0 key generation for securing applications and servers using the APIs
- A secure layer which provides granular control over specific users's access to the APIs, including filters on the data itself
- A deep caching solution which allows for the caching of the backend assets for better performance. This can be used with most available caching engines
- Analytic data is gathered both on the API endpoint and the backend asset allowing for insights on both the consumer and the provider of the data
- API publishing and promotion is done using Swagger

Versioning and Retirement

Versioning of an OpenLegacy API can be regarded as a twofold feature:

- Pre-deployment versioning is done using integration with major source-control solutions (Github, ClearCase, Subversion etc.)
- Post-deployment, each deploy process creates a new version for the deployed API, based on the pre-deployed version name, which can be managed in the management console.

Retirement is done via the lifecycle management feature which allows for each API to have one of the following statuses:

- Active: API is valid, available and ready for use

OpenLegacy
The Business End of Integration

Sales Battle Card

API Management

- Paused: API is valid but has been made intentionally unavailable
- Invalid: API has been made unavailable due to errors
- Deprecated: API is available but is being phased out. Any use of API will be logged and reported
- Retired: API is no longer available as it has been rendered obsolete

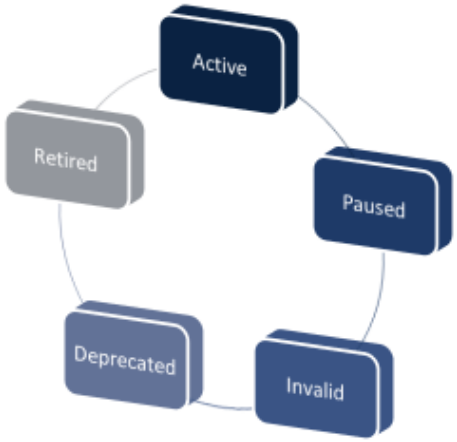
Automated Testing

Changes in status can be made manually using the management console or programmatically using the management API.

in addition, for each API generated, a Junit testing module is also created. This testing module can be used with automation servers such as jenkins to provide automatic testing and status setting for each API.

API Lifecycle Management

- **Active**: API is valid, available and ready for use
- **Paused**: API is valid but has been made intentionally unavailable
- **Invalid**: API has been made unavailable due to errors
- **Deprecated**: API is available but is being phased out. Any use of API will be logged and reported
- **Retired**: API is no longer available as it has been rendered obsolete



OpenLegacy

API Lifecycle Management

- OpenLegacy API projects are in fact standard JAVA projects developed in our Eclipsed based IDE.
- All major version control products are supported, including Git, Subversion and Clearcase.
- Deployment is very simple and streamlined – the integration developer can deploy to the server of their choice via a menu option in the IDE.
- The deployed artifact is a WAR file, which can easily be incorporated in any deployment management solution available
- Because of OpenLegacy's flexible architecture – the platform can be utilized to auto generate the critical legacy application API and seamlessly transition it to the API management tool of choice.

OpenLegacy