

Avaloq Banking Suite – Avaloq Front Rel. 2.4

Front Development Kit

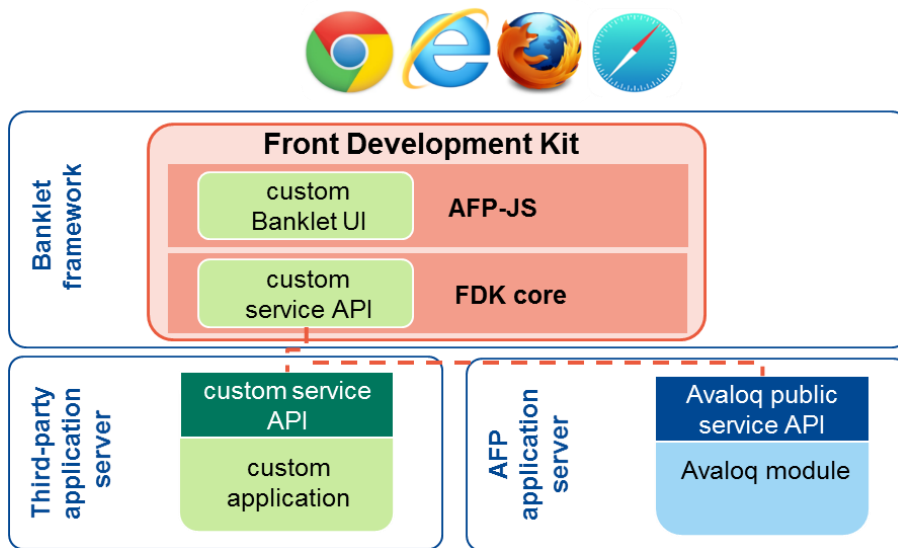
The Front Development Kit (FDK) makes it easier and more efficient to develop browser based Banklet user interfaces (UI). It includes:

- Tools to create Banklet UI projects
- Avaloq widgets and components
- Catalogue of Avaloq public application programming interfaces (API)
- Framework to expose SOAP and RESTservice
- Tools to support customisation of Avaloq Banklet UIs

The Front Development Kit provides the tools and software packaging conventions, and server-side resources to support the secure integration of applications into the Avaloq Banking Portal.

The Front Development Kit targets experienced web and backend developers as it builds on top of proven standards and technologies like AngularJS, SASS, Apache Maven, npm, Java, SOAP/REST.

System Overview



- AFP-JS: is the Avaloq Banklet UI framework. It provides a JavaScript (JS) based client-side environment which is embedded into portal environment. Banklet UIs are Java Enterprise Edition compliant web applications with runtime dependencies on the Banklet orchestration framework
- The FDK core is a collection of libraries and utilities to support the usage of the Avaloq Front Platform (AFP) services. It includes a list of the available public service API's accessible to developers

Key Facts

Increase Developer Productivity

The framework support the lifecycle of a Banklet UI:

- Create project: retrieve a Banklet-scaffold project and setup the initial project
- Creating mock data
- Customise (e.g. configurations, language translations)
- Test your Banklet without having to deploy it into a running portal. This means that during the development phase you can quickly iterate and stay focused on the Banklet functionality
- Serve – serve the Banklet in your development environment without the need of an Avaloq Front Platform server. Use mocked or proxied data combined with live-coding to iterate quickly
- Static build – build a static version of the Banklet for demos
- Deployable build – build a deployable component

Abstract the Complexity from Web Developers

Our service-oriented frameworks decouple the fast moving frontend development from the stable core banking services. This allows the UI oriented Banklet developer to focus on customer experience. The

FDK core encapsulates the server side API's and exposes them to the Banklet as REST services

Explore the Public Services of the Avaloq Front Platform

The Front Development Kit provides a catalogue with the description of all available public services

Reuse Avaloq UI Standards

Use the same building blocks that Avaloq uses. This enables a unified look and feel for both Avaloq Banklet UI's and the custom Banklet UI's

Under Development for Future Releases

- Additional widgets
- Improved documentation
- Additional tutorials

Details

Technically, the basic integration pattern for content into the Avaloq Banklet framework is via iFrame. This kind of integration provides security as well as hassle-free coexistence of different JavaScript libraries. While iFrame-based integration adds hurdles in terms of Banklet intercommunication or for managing the look and feel of external content, the Banklet framework provides patterns, tools, and services to overcome these obstacles.

The Avaloq Banklets UI framework AFP-JS is built using state-of-the-art web UI technologies.

AFP-JS

- AFP core delivers AngularJS bootstrapping and configuration of many AngularJS related features
- Translations: Banklets use Angular-gettext to support translations. AFP tool helps you extract and compile the texts
- AFP commons provide JS utility functions used across the framework
- Mock data: creating service requests will then return the configured mock data. This can be used to mock the backend services for development and testing. It is possible to mock more advanced services to simulate, for example filtering, sorting, or other methods
- AFP widgets deliver all the visual elements:
 - Action bar
 - Chart
 - Collapsible
 - Datepicker
 - Dropdown
 - Form
 - Input
 - Notification
 - Panel
 - Slider
 - Table
 - Tag
 - Tooltip

AFP components are special purpose UI elements:

- Action table
- Editable table
- Filter panel
- Spiffy table

FDK Core

- REST framework – utility classes to help building REST services
- SOAP framework – utility classes to help building SOAP services
- Security framework – utility classes to handle and manipulate the authentication data required by AFP
- Public service API catalogue: including WSDL files of public SOAP services

Documentation

- Blueprints and reference implementation of applications and Banklets
- Tutorials
- AFP-JS showcase: documentation on general styling, widgets, and components. Styling includes typography, icons, colours, and layout concerns
- Contextual documentation and live-coding features to test, play and investigate the AFP-JS UI features
- Portal integration and orchestrator configuration
- Description of Avaloq's approach to CSS styling: central CSS feature
- Deploying a Banklet to Liferay Portal

Technical Requirements

- To test and deploy a Banklet UI the developer needs access to an Avaloq Front Platform server instance and file-based remote access to the server
- The technical requirements installed on developer machine are:

Component	Product	Version
Java	Oracle Java	1.8
JavaScript	ECMA	>1.5
Multipurpose Editor	any	any
Build Tool	Apache Maven	3.3.x
Source Management/Control	any	any
JavaScript runtime	nodeJS	5.x
JavaScript package manager	npm	3.x

- We currently recommend using JFrog Artifactory
- AFP-JS requires standard open-source libraries that can be freely downloaded from the internet as dependencies
- Using an integrated development environment is optional with AFP-JS