Intel Cloud Integrity Technology 3.0

**HTML5**

# Background

The KMS uses Jetty9 to serve static HTML5 content and JAX-RS APIs.

# Architecture

## Splash Screen

The root path to the application retrieves a “splash screen” that displays the Intel logo and then redirects the browser to the main page of the application.

The splash screen’s style should be consistent with the application style and it contains some elements of the application main page for continuity.

The splash screen is provided by mtwilson-core-html5.

## Main Application

The mtwilson-core-html5 project contains the static HTML5 resources for the main application. Other features are loaded and inserted to the UI by this code.

## UI Plugins

There are two methods for providing static HTML5 content to the UI.

First, any .jar file can add static HTML5 content to the UI by placing it under a “publicResources” folder. In the source tree this would be under the project’s src/main/resources/publicResources. A JAX-RS class in mtwilson-core-html5 provides read access to all files under “publicResources” in the classpath. However, the UI has to already “know” to load those resources from a specific URI relative to the application URI.

Second, any feature can add static HTML5 content to the UI by placing it under a src/main/html5 folder in the Maven project layout and inheriting from mtwilson-core-feature-zip. See blueprint for mtwilson-core-html5 for more details. This content can then be discovered by and automatically integrated into the UI by including extensions for known extension points.

## Resource Loader

A resource loader javascript class in the UI is in charge of loading HTML, CSS, and JS files, ensuring that each file is loaded only once, and ensuring that a Javascript snippet that declares a dependency on external Javascript files is not run until all those files have been loaded. This resource loader is provided by mtwilson-core-html5.