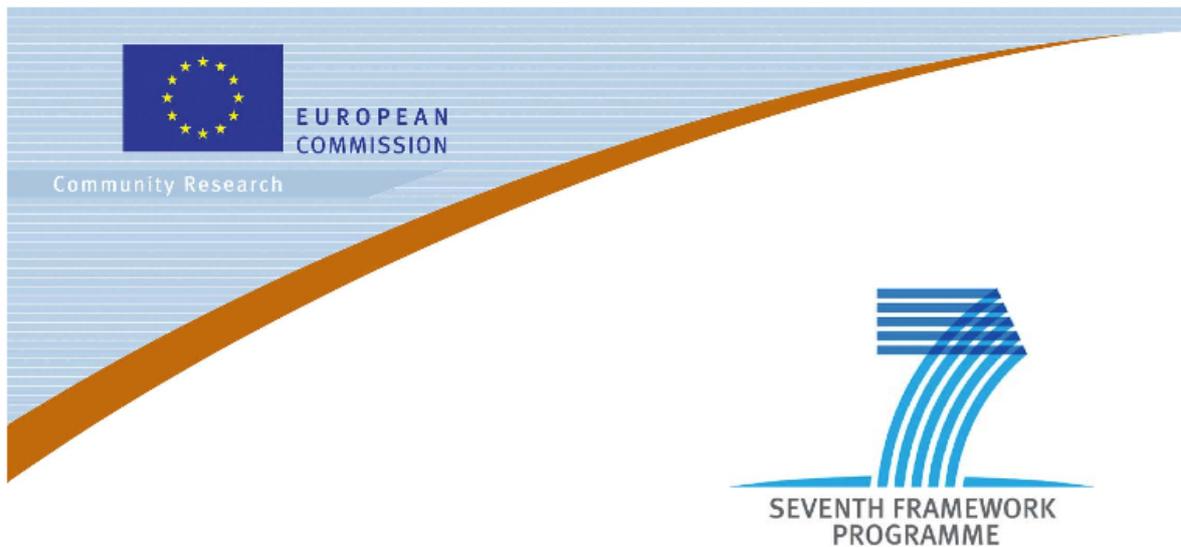


API IDE Support front page



Large-scale Integrated Project (IP)



D9.3: API IDE Support

Project acronym: FI-WARE

Project full title: Future Internet Core Platform

Contract No.: 285248

Strategic Objective: FI. ICT-2011.1.7 Technology foundation: Future Internet Core Platform

Project Document Number: ICT-2011-FI-285248-WP9-D9.3a

Project Document Date: 30 May 2012

Deliverable Type and Security: Public

Author: Matteo Melideo (ENG)

Contributors: Davide Dalle Carbonare (ENG), Fredrik Pettersson(EAB), John Sandberg (EAB)

Abstract: This deliverable provides a subset of reference tools to store/manage/publish the information regarding Generic Enablers and to make them available directly into the tools used for the Future Internet application development.

Keyword list: FI-WARE, PPP, DevComE, Catalogue, Drupal, Generic Enabler, Library Management, Dependency Management, Maven, Eclipse

Contents

Articles

API IDE Support front page	1
API IDE Support	2
Catalogue	2
Library and Dependency Management	5

References

Article Sources and Contributors	9
Image Sources, Licenses and Contributors	10

API IDE Support

Software Components

This deliverable is mainly composed by software prototypes that are listed here and briefly described. More information on the tool, together with how to obtain it, are available through the reported links.

Catalogue

The catalogue is a container for assets (e.g. GEs) published by asset providers; developers can browse, search and get access to those assets.

Library and Dependency Management

The proposed composition of tools provides a comprehensive support to the developer on managing the project's libraries and dependencies in general.

Catalogue

Introduction

The catalogue is a container for assets published by asset providers and made available to developers that can browse the catalogue to find the specific assets they are interested in and download or use those assets. It provides features for asset publishing, browsing of assets, community for discussions around these assets and references to the appropriate support functionalities available for each asset.

Information

Name: Catalogue

Version: 1.0

License: Drupal part: GPLv2 User Management System: TBD

Source Code: Contact Ericsson <mailto:john.sandberg@ericsson.com>

Website:

- FI-Ware Catalogue ^[1]

Scope: The main goal for the catalogue is to provide a common place (website) where users can browse, search and get access to Generic Enablers.

Components

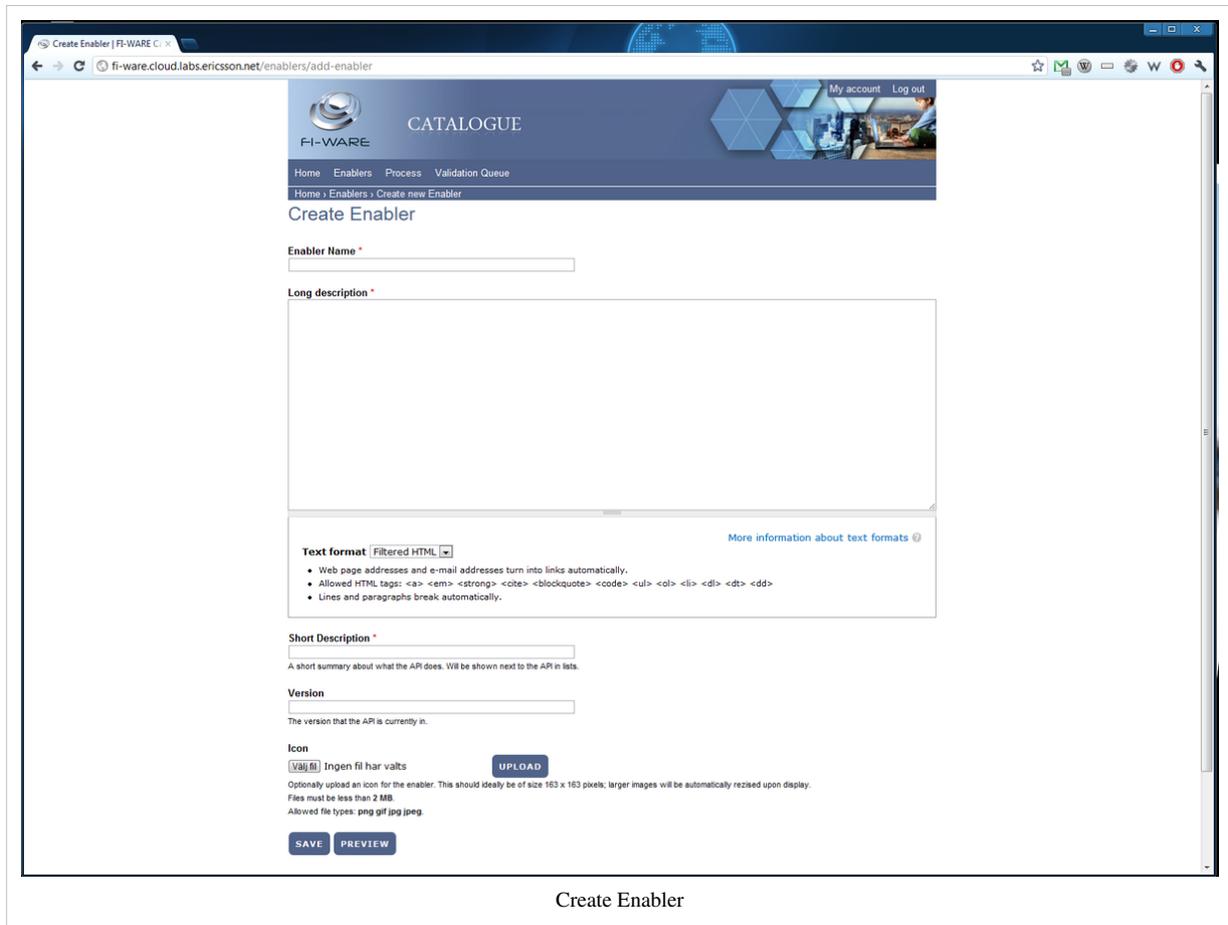
The catalogue is based on the Drupal content management system and an add-on system for user management. It is currently running on a server hosted by Ericsson at <http://fi-ware.cloud.labs.ericsson.net>. This site is live and publicly available and you can register a user and start browsing for or adding content.

Documentation

Usage of the catalogue should be self explanatory to anyone used to interacting with a web portal and normally you don't have a manual for usage of a website. More help texts will be added on the site during the project according to feedback of possible unclarities. Here is though an overview about one key part of using the catalogue, the publication process.

Publication Process

1. Make sure you have a user account and is logged in to the Catalogue
2. Click the "Add new Enabler" link
3. Fill in all the required information and submit your new GE (see example below)
4. The GE will be reviewed by the Catalogue administrators and if approved you will be notified by email



The screenshot shows a web browser window with the URL `fi-ware.cloud.labs.ericsson.net/enablers/add-enabler`. The page title is "Create Enabler" and the breadcrumb is "Home > Enablers > Create new Enabler". The form contains the following fields and sections:

- Enabler Name ***: A text input field.
- Long description ***: A large text area.
- Text format**: A dropdown menu set to "Filtered HTML". A link "More information about text formats" is provided.
- Short Description ***: A text input field with a note: "A short summary about what the API does. Will be shown next to the API in lists."
- Version**: A text input field with a note: "The version that the API is currently in."
- Icon**: A section for uploading an icon. It includes a "UPLOAD" button and instructions: "Optionally upload an icon for the enabler. This should ideally be of size 163 x 163 pixels; larger images will be automatically resized upon display. Files must be less than 2 MB. Allowed file types: png gif jpg jpeg."
- Buttons**: "SAVE" and "PREVIEW" buttons are located at the bottom of the form.

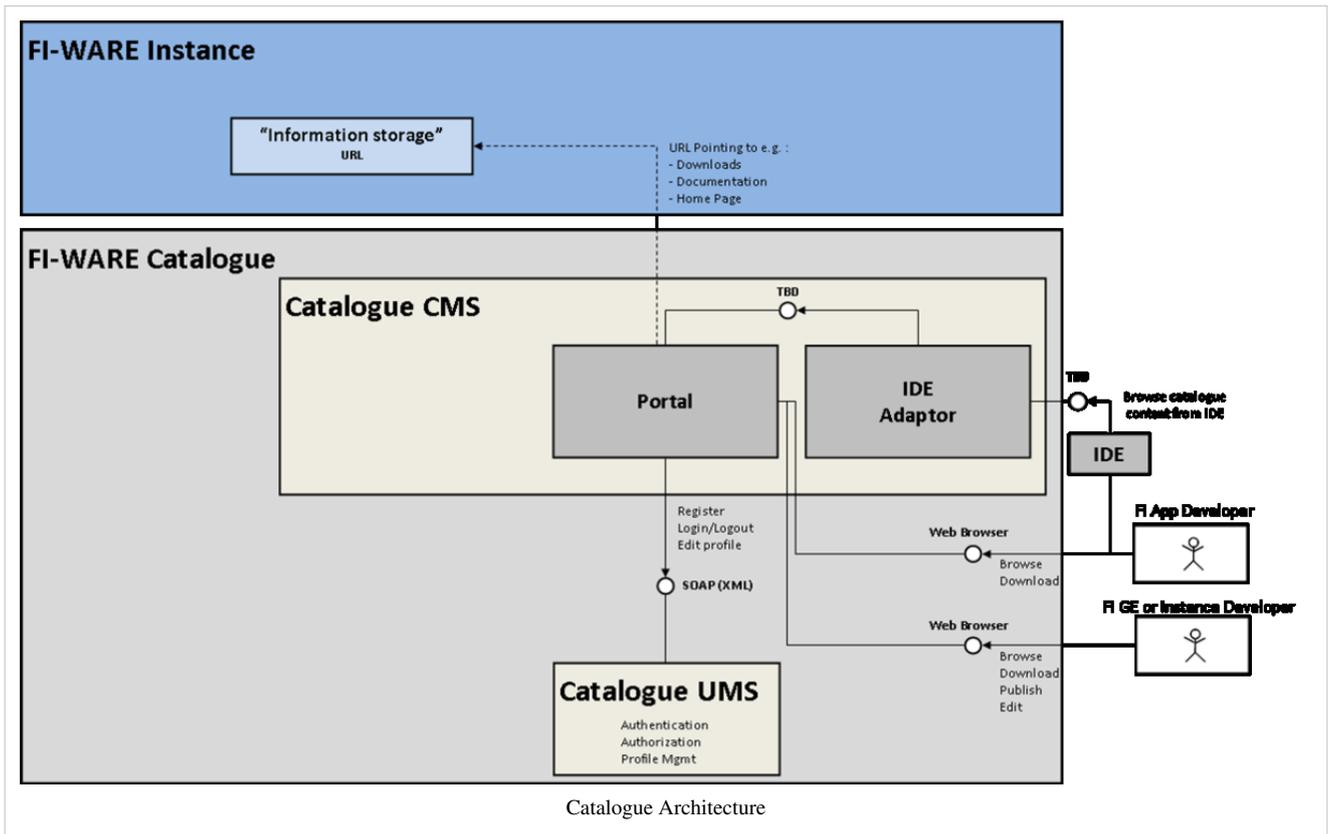
You then get access to edit the full GE information and you should provide all the documentation relevant to someone reading the GE description in the Catalogue. When ready for publication you publish your GE and it will then be available to all users.

Architecture

The catalogue will be deployed as part of the FI-WARE Testbed and expose the assets available in this testbed. The catalogue enables users (such as e.g. FIApp or GE implementations developers) to browse and interact with content as well as with FI-Ware GE and instances providers. The users can, in addition, also publish new GE implementations, info about GE Implementations and edit them. The catalogue is based on two main components:

- Catalogue Content Management System (CMS) - a portal including an adaptor for integrating with an IDE
- Catalogue User Management System (UMS)

The CMS/portal includes a publishing system and stores published content making it available for browsing. The IDE adaptor provides an interface to the IDE that enables browsing of catalogue content. The UMS takes care of registration, authentication, authorization and profile management. It is role based and it is possible to configure different roles and provide them different rights to the catalogue.



References

- [1] <http://fi-ware.cloud.labs.ericsson.net/>

Library and Dependency Management

Introduction

The proposed composition of tools provides a comprehensive support for the library and dependency management topic. The selected reference tools are the Maven integration for Eclipse (m2e), on the user side, and Nexus for as a repository management, server, solution. With m2e, Maven is used within Eclipse in a natural and intuitive interface. A Nexus installation at FI-WARE Instance level facilitates the adoption by the developers of Future Internet applications of the artifacts made available by that FI-WARE Instance. This document highlights the most relevant features and benefits in the context of Future Internet application development based on the FI-WARE environment. Additional and detailed documentation on m2e and Nexus are available from the official web sites.

Information

Maven Integration for Eclipse

Name: Maven Integration for Eclipse ^[1]

Version: 1.0.x

License: EPL v1.0

Download: Eclipse Marketplace

Documentation: Official Website ^[1]

Scope: Dependency Management plug-in

Nexus

Name: Nexus ^[2]

Version: 2.0.x

License: EPL v1.0

Download: Official Website ^[3]

Documentation: Official Website ^[3]

Scope: Artifact Repository Management

Apache Maven

Name: Apache Maven ^[4]

Version: 3.0.x

License: Apache Licence v2

Download: Official Website ^[5]

Documentation: Official Website ^[6]

Scope: Dependency Management

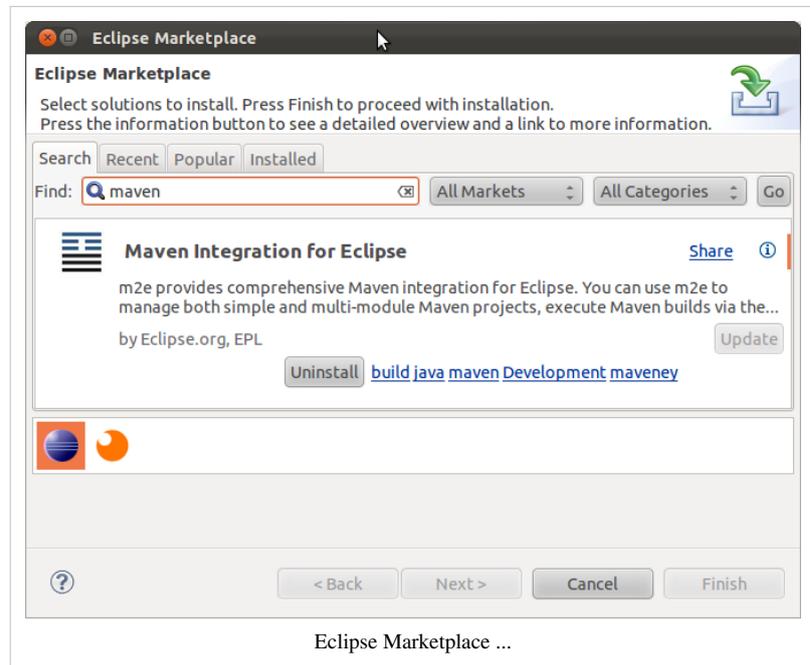
Installation

m2e

This plug-in can be installed directly from the Eclipse IDE opening the wizard at *Help -> Eclipse Marketplace ...* and searching for *Maven* (see image). Install the "Maven Integration for Eclipse" component following the guided steps. All the requirements are resolved automatically.

Nexus

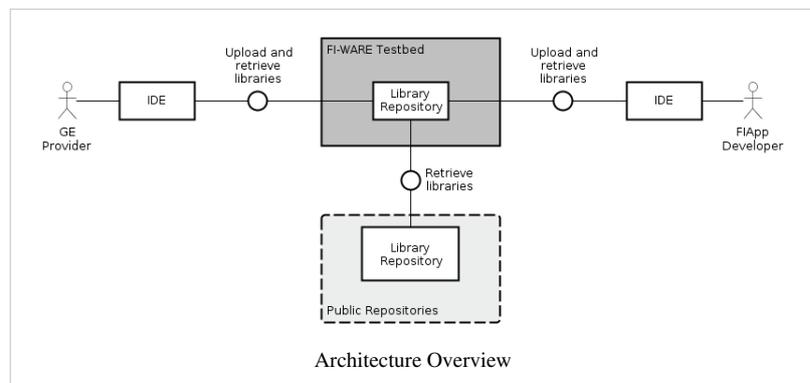
To download and install the Nexus solution follow the instructions described on the official web site, the dedicated page is Download and Install Nexus ^[3].



Architecture

In this part it's described how the two solutions, m2e and Nexus, works together, highlighting the development use cases aspects. The proposed solution aims to take advantage from the most relevant features of both of them.

In a complete and integrated scenario, it's feasible to have GE/SE Providers that contribute libraries for a specific



FI-WARE Instance (Testbed in the specific case). These assets, developed for that FI-WARE Instance, are stored into that instance, thanks to the adoption of a service for library and dependency management service. The reference implementation for this service is the Nexus solution. The choice of this one among the others is due to the fact the client component, installed into the Eclipse IDE, is the m2e (Maven integration for Eclipse) solution that are developed and maintained by the same company. Both of them represent a *standard de facto* for this sort of tools at support of software development. The GE/SE Provider thanks to the m2e plug-in is able to upload the assets into the repository directly from the Eclipse IDE.

Once the assets are made available into the Nexus service for that FI-WARE Instance, a FIApp developer that wants to build an application on top of that FI-WARE Instance is able to quickly and easily obtain all the assets of interests. This operation can be done browsing the repository directly from its native web interface or from the Eclipse IDE using the m2e plug-in.

How To

The sections of this How To consider mainly the developer point of view and the interaction within the Eclipse IDE environment. For a complete and comprehensive documentation on how to use and configure this set of tools refer to the official documentations:

- [Nexus: Repository Management with Nexus](#) ^[7]
- [m2e: FAQ](#) ^[8]
- [Maven: Maven Users Centre](#) ^[9]

Configure the Repository

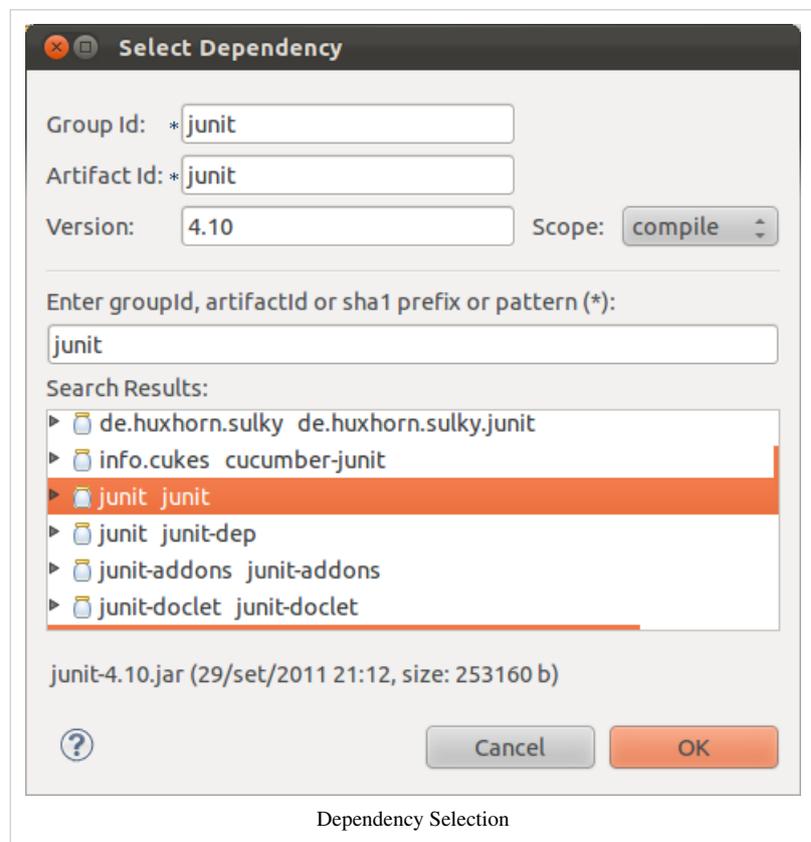
To install and configure the Nexus repository for a FI-WARE Instance it's useful to follow the indications reported in Installation manual ^[10].

Add a library to the project

This is the typical usage of the Apache Maven tool. It can be done manually editing the POM files or more easily, and graphically, from the user interface provided by the m2e plug-in. The screen-shot here aside, shows an example of a library search results. In fact from the m2e plug-in it's possible to query the repository for the desired library, get the list of the available results, select the one to add to the current project (POM).

Deploy an artifact to the Repository

To deploy an artifact to a remote repository is a typical Maven operation that is well documented on the official Getting Startd Guide ^[11]. This operation allows developers to make their assets available to all the other developers connected to that FI-WARE Instande Repository.



Improvements

Thanks to a traditional feature of maven, the archetypes ^[12], it will be possible to define some of those, in the context of the FI-WARE project, that will be useful for starting a new project for a FIApp or for a Generic Enabler. An archetype is a template that is used to setup the structure of a new project. This template structure can cover aspects that range from the nature of the project (web app, jar library, java application, mobile app) to predefined dependencies definitions.

References

- [1] <http://eclipse.org/m2e>
 - [2] <http://nexus.sonatype.org>
 - [3] <http://www.sonatype.org/nexus/go>
 - [4] <http://maven.apache.org/>
 - [5] <http://maven.apache.org/download.html>
 - [6] <http://maven.apache.org/guides/index.html>
 - [7] <http://www.sonatype.com/index.php/Support/Books/Repository-Management-with-Nexus>
 - [8] http://wiki.eclipse.org/M2E_FAQ
 - [9] <http://maven.apache.org/users/index.html>
 - [10] <http://www.sonatype.com/books/nexus-book/reference/install.html>
 - [11] http://maven.apache.org/guides/getting-started/index.html#How_do_I_deploy_my_jar_in_my_remote_repository
 - [12] <http://maven.apache.org/guides/introduction/introduction-to-archetypes.html>
-