SAP Fiori - Development main principles

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Identification** | | | |
| **Author** | Amine LOUATI | **Document Location** |  |
| **Version** | **Status** | **Date (DD/MM/YYYY)** | **Classification** |
| 1.0 | Work in Progress | 22/11/2016 | Confidential |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| **Version** | **Date** | **Sections revised** | **Description** |
| 1.0 | 22/11/2016 |  | Initial creation |
|  |  |  |  |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Reference Documentation** | | |
| **Version** | **Document** | **Description** |
| 1.0 | Development environment |  |
| 1.0 | Development naming conventions |  |
| 1.0 | Development golden rules |  |
| 1.0 | Development check |  |

|  |  |  |
| --- | --- | --- |
| **Responsible** | | |
| **Interlocutor** | **Main** | **Backup** |
| **Business** |  |  |
| **Development Shared Services** |  |  |
| **Support** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Review and Approval** | | | |
| **Name Decathlon Approver** | **Approved (Yes/ No)** | **Comment** | **Date (DD/MM/YY)** |
|  |  |  |  |

**TABLE OF CONTENTS**

[**Purpose of this document**](#_30j0zll)

[**HTML5**](#_tyjcwt)

[**CSS3**](#_3dy6vkm)

[**Javascript**](#_4d34og8)

[**MVC (Model View Controller)**](#_csy0s12it9fl)

[**Data binding**](#_ri06tdrwpm7)

[**SAPUI5 Application initialization process**](#_3g6jb02do275)

[**Development environment**](#_w88g0yxa9j8w)

[**Architecture representation**](#_a92v5ihs3of4)

# Purpose of this document

This document provides the main principles of SAP Fiori development. It describes the main technical subjects and dev language required.

For convenience, SAP Fiori and SAP UI5 should be considered equivalent in this document.

**SAPUI5 (SAP user interface for HTML 5) is a collection of libraries that developers can use to build desktop and mobile applications that run in a browser.**

It is a client UI technology based on HTML5, CSS and JavaScript. Depending on the device the application is to run on (mobile, tablet or desktop PC), different UI libraries are used.

Applications and libraries can be stored on an SAP NetWeaver Application Server or an SAP HANA Cloud Platform. The servers also offer deployment, storage and connectivity tools.

## HTML5

HyperText Markup Language is the standard markup language for structuring and presenting content for the World Wide Web.

HTML5 is the successor of HTML 4.01, released for the first time in 1999. The internet has changed significantly since 1999 and it seemed like the creation of HTML5 was necessary. The new markup language was developed based on pre-set standards:

- New features should be based on HTML, CSS, DOM, and JavaScript.

- The need for external plugins (like Flash) needs to be reduced.

- Error handling should be easier than in previous versions.

- Scripting has to be replaced by more markup.

- HTML5 should be device-independent.

- The development process should be visible to the public.

**HTML5 was created to make the coding process easier and more logical.**

**All major browsers (Chrome, Safari, Firefox, Opera, IE) offer HTML5 support**

## CSS3

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language.

It is designed to separate the content from the presentation of documents, as the layout, colors, and fonts.

CSS3 is the latest standard for CSS and is completely backwards-compatible with earlier versions of CSS.

It has been split into "modules“. It contains the "old CSS specification“ and some new modules :

- Selectors

- Box Model

- Backgrounds and Borders

- Image Values and Replaced Content

- Text Effects

- 2D/3D Transformations

- Animations

- Multiple Column Layout

- User Interface

## Javascript

JavaScript is a scripting language. One of its main usages is to access and manipulate the elements created using a markup language

JavaScript support is built right into all the major web browsers, including Internet Explorer, Firefox and Safari. Provided that the visitors to your site are using web browsers that support JavaScript (most do) and have JavaScript enabled (it is by default), then your JavaScript will run when they visit the page.

JavaScript is an interpreted language, so no special program is required to create usable code. Any plain text editor such as Notepad is quite satisfactory to be able to write JavaScript.

JavaScript is classically used to

- Show / Hide text

- Form validation

- Dynamic and Highly responsive content

- Create a slideshow

- Display tooltips

## MVC (Model View Controller)

SAPUI5 uses the Model View Controller (MVC) concept to provide the following assets:

- Separation of the information representation and the user interaction

- file structure, naming, and usage patterns

- Support of development in distributed teams with different source locations

- UI declaration (in comparison to a programmatic construction)

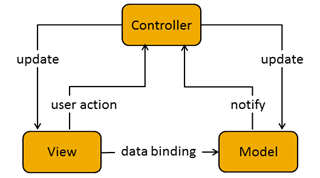
Model, View, and Controller can be associated as following

- The view is dedicated to the definition and the render of the UI.

- The model provides a management of data for the application.

- The controller modifies the view and model according to the view events and user interaction.

We can sum up the interaction between the different components as following:



The relation between views and controllers are generally 1:1.

The instantiation of controller without relation with a view can be done and in the same way, view can be associated to no controller.

A view is a SAPUI5 control and can have or inherit a SAPUI5 model.

Views and controllers are reusable and can be developed in distributed way.

## Data binding

The purpose of data binding in the UI is to separate the definition of the user interface (view), the data visualized by the application (model), and the code for the business logic for processing the data (controller). The separation has the following advantages:

- It provides better readability, maintainability, and extensibility,

- it allows the change of the view without touching the underlying business logic,

- it allows to define several views for the same data.

The UI is updated automatically thanks to the data binding which allows a link between UI controls and the model.

SAPUI5 also uses data binding to bind two data sources or information sources together to keep them synchronized: a change done in one data source is also reflected in the other data source.

A “model instance” and a “data binding instance” are required for data binding:

- The model instance holds the data and provides methods to set the data or retrieve the data from a server. It also provides a method for creating bindings to the data.

- It contains the binding information and provides an event triggered each time the data is changed and also the linked visual element. The data binding instance is created by a method provided by the model instance.

SAPUI5 offers several binding mode for model implementation:

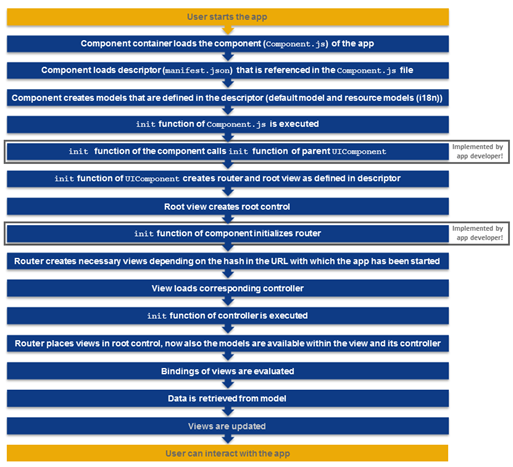
|  |  |
| --- | --- |
| **One-way binding** | Data is transported in one direction only: from the model, through the binding instance to the consumer (usually the property of a control), but never in the other direction. |
| **Two-way binding** | When properties in the model is updated, so it changes in the UI and reciprocally when UI elements is updated, the changes is propagated to the model. |
| **One-time binding** | The data is loaded in the model once. Usually used for static texts example for language purposes. |

In SAPUI5, the default binding mode for model implementations is two-way binding and can be changed by the application for each model declaration.

## SAPUI5 Application initialization process

When a user accesses to a SAPUI5 application, a request is sent to the server to load the application into the browser. Also views and the required libraries are loaded. The model is also instantiated and business data is fetched from the database.

Here the initialization process of a SAPUI5 application :



## Development environment

UI development toolkit for HTML5 is the referent for SAPUI5 developments. It can be provided by 2 tools:

- **SAP Web IDE** in 2 versions:

- **Cloud:**

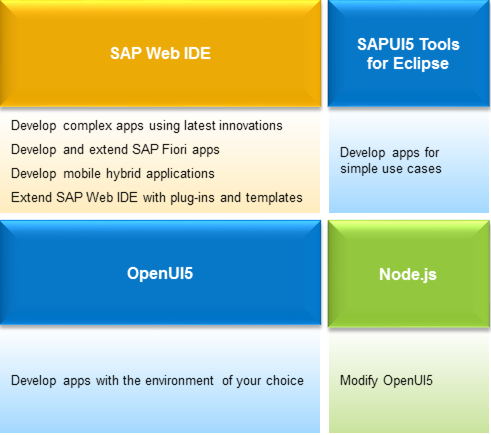
This version allows developing directly on SAP HANA Cloud Platform without any development infrastructure on client side and using a web interface. All developments are hosted by SAP which offers a shared repository.

- **Local:**

It is downloadable from SAP web site and usable in local on user infrastructure. The main inconvenient is that the version doesn’t follow the versioning strategy of SAPUI5 version in contrast to the SAP Web IDE cloud version.

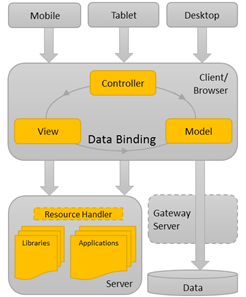
- **SAPUI5 Tools for Eclipse**:

It is a set of plugins for Eclipse downloadable directly from Eclipse update tool which allows development thanks to wizard, skeletons and structures in symbiosis with Eclipse functionalities.



## Architecture representation

Here, an easy representation of the different components and interactions for a SAPUI5 application



The devices use the client or browser to retrieve and load resources needed by the applications hosted by the Web application server. Data needed by the application are retrieved through a gateway server and hosted on a data server.

Following, a draw of all components and interactions according the Runtime, Design Time phase and Actors

