Anubhav trainings

LIVE

www.anubhavtrainings.com

UI5 and Fiori Training

on Business Application Studio

04th Dec. 2023, 7 AM IST

Book your seat now!!

Agenda

01 Who is this course for

02 Overview of the course

03 Our use case for Demo

04 FAQs and Q&A

Who is this course for

UX Designer – Deny Joy

A UX Designer, she would like to start the development of UI5 applications

Fresher - Vishal Sharma

Fresher with no experience, and aspire to become a full-stack developer

Technical Advocate – Shanaya Khan

Working in an industry with other UI technology and want to be a UI developer with SAPUI5

Application Developer- Suhas Raane

Working as an ABAP consultant, want to build apps on ECC or S/4HANA

Overview of the course

1. FUNDAMENTALS

Learn the core concepts of HTML5, CSS, JS, jQuery
Setup BAS

2. SAPUI5

Understand SAP UI5 framework and SDK and build using MVC architecture.

3. FIORI UX

Create Fiori Applications using the Mobile library with the latest concepts

4. ODATA DEVELOPMENT

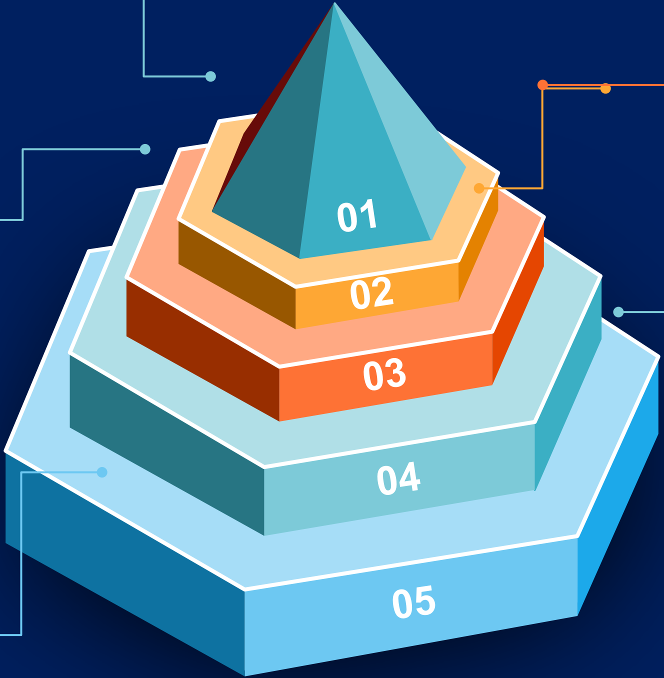
Build Services in the Gateway system and expose ERP Data to Fiori Apps

40 Hours Training

The Course is designed for absolute beginner/ABAP Developers who have never worked on UI5 and aspire to transform their Dialog/WD programs to Fiori UX Apps.

www.anubhavtrainings.com

What do we do today?



01 Launch Dev Tool

Start Business Application Studio in BTP

02 Create Project

Create New UI5 Project

03 MVC Architecture

Model, View, Controller

04 Test Our App

Run the App preview in BAS

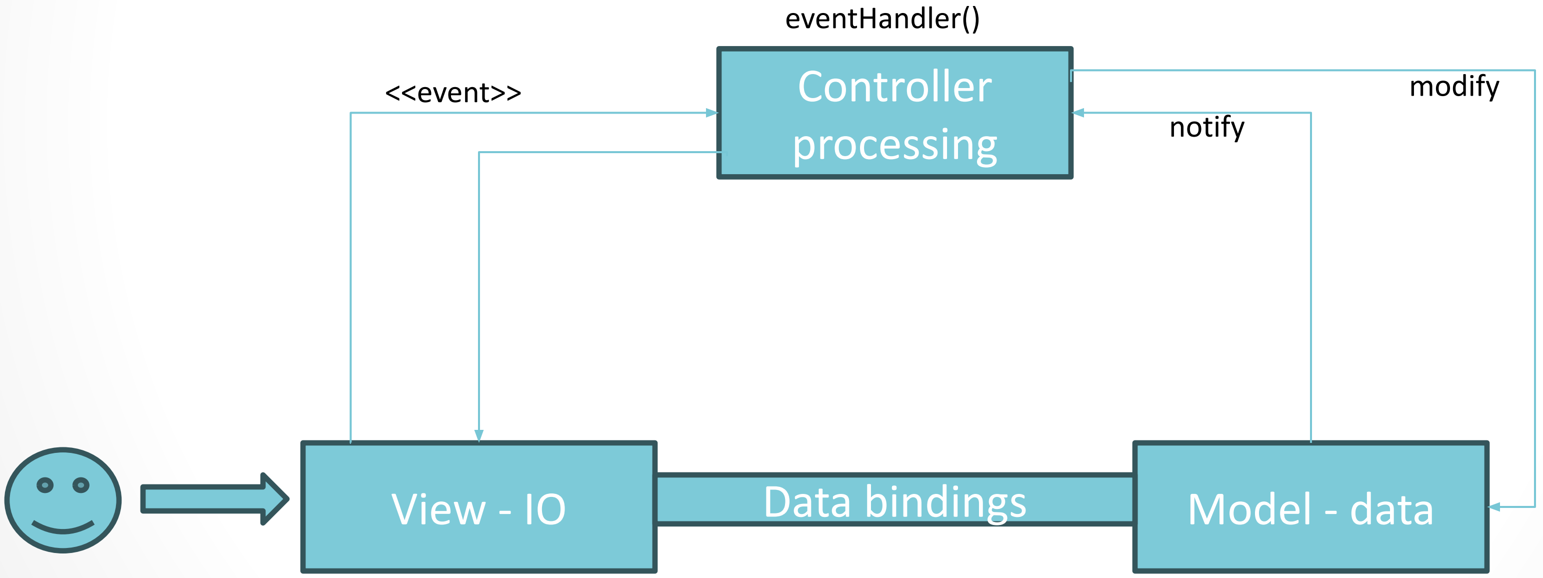
05 Run In Browser

Run the Application in the browser

What is SAP UI5

SAP UI5 is a **framework** to develop **responsive** web applications
Framework – collection of libraries
Library – Collection of classes
Class – Collection of methods, attributes, events and aggregation

Responsive – is a web application which adapt itself according the device.



Steps to create first UI5 project

- Create a new folder **ui5demo**
- We need to initialize this project, we use a command called **npm init**
- We need add a file called **webapp/manifest.json** file with the required properties – id, type
- Now mark this project as a SAP UI5, we use a command **ui5 init**
- Created a view file and a controller file with basic skeleton
- Add a Button to the view and add the click logic inside the controller
- Create the view object and place in the body
- Run the **ui5 serve** command to start and test the project

Dev tools to create Fiori App

2012
Eclipse

2015
WebIDE

2019
BAS

2019
VS Code

Factors	BAS	VS Code
Connectivity	Online	Offline
Cost	Cost of BTP for productive usage	Free
Company	SAP	Microsoft
Speed	Comparatively low	High
Experience	Same as VS Code as it’s a baby of vs code	Original
Type of development	Multi cloud dev tool	Fiori App
Integrations and plugin	Same	Same

VS Code tool

1. Download Node JS 18 and install - https://nodejs.org/en/blog/release/v18.12.0 You can check by running command **node -v**
2. Download git-scm tool for source code management - https://git-scm.com/downloads check **git --help**
3. Download the VS Code and install the same in laptop - https://code.visualstudio.com/download
4. We need to create a workspace (folder in our laptop) in our machine where we can start creating our project
5. Go to extensions section and install **Fiori Extension pack** and **Five Server (Live Server)**
6. SAP Provide a set of toolkit to make developer life easy for developing Fiori apps – SAP UI5 Tooling

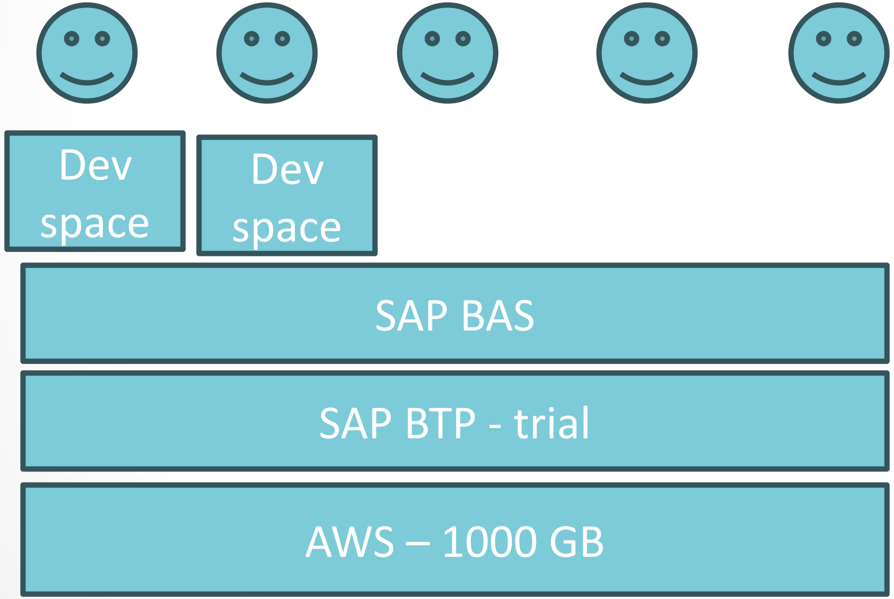
   ```
   npm install -g @ui5/cli
   npm install -g @sap/ux-ui5-tooling
   ```

What is BTP

Business Technology Platform – Platform as a Service provided by SAP for developers and development teams to build cloud native applications.

Setting up BAS tool on BTP

1. Create a free account with sap, you can reuse your existing P/S user ID or a SAP account which you might created in past. https://sap.com
2. BTP Cockpit to login and create first BTP account - https://account.hanatrial.ondemand.com/trial/#/home/trial
3. Choose data centre and Infrastructure provider AWS US (VA)
4. It Creates a global account (for billing purpose), a sub account for data center, dev space, also subscribe to BAS service
5. We need to create a dev space in the BAS tool



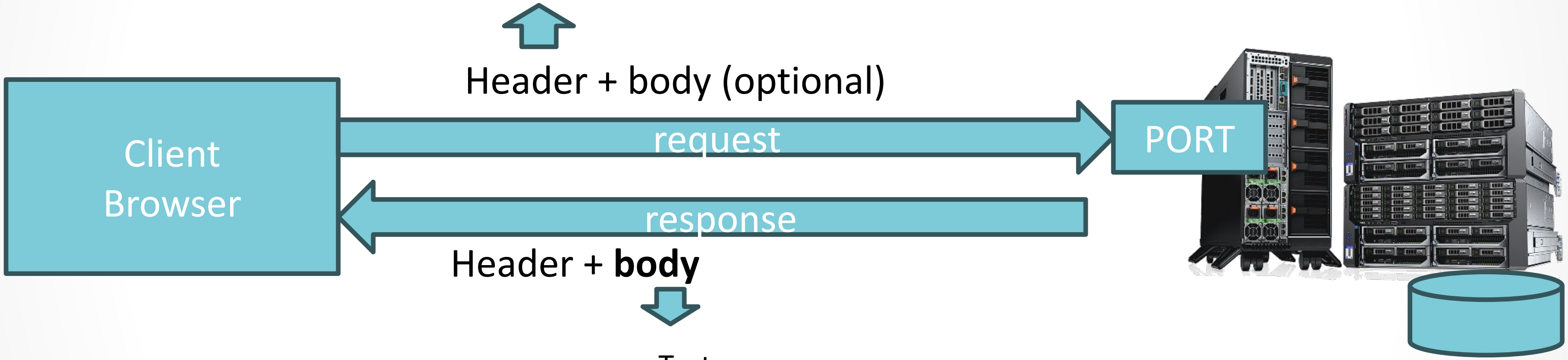
Dev space

- Provides a private development environment
- If a developer disrupt something, it wont affect other developers.
- Optimum utilization of memory, each dev space is a virtual machine with 4GB of memory allocated.
- SAP **pre-configure** the dev space according to kind of development we want

Client Server Architecture

GET – read data from server
POST – create data in server
PUT – update data
DELETE- remove data from server

- **Form Submit**
- **HTTP Request**
- **AJAX call**



Header + body (optional)

Header + **body**

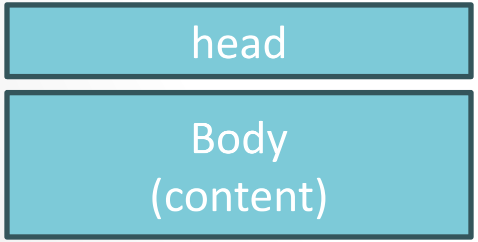
Text
Html
Xml
Json
Application/pdf
Media-content

What is HTML?

HTML stands for HyperText Markup Language, it is used to create static web content. Which means it does not have programming capability to manipulate data. W3C (world wide web consortium) is responsible to bring new versions of HTML and all the browsers (Edge,Chrome,Firefox..) comply to the HTML.

Year	Version
1991	HTML
1993	HTML 2.0
1995	XHTML
1999	HTML4.0
2011	HTML5

An html file always ends with .html extension. HTML is not a programming language. An HTML document contains the header and the body. Header is the brain and body is the content of the HTML.



head

Body (content)

Funda Fox

- HTML is not a programming language
- Every html document/file, ends with .html extension
- We should use the tags offered by W3C only
- Syntax = <tagName prop=val>CONTENT</tagName>
- We can learn HTML from official website of W3C
- We will have header which is the brain of html which contains title, meta tags which users wont see
- Whereas we have body, with the content our user will see
- **Every browser understand HTML directly**
- **HTML is a TREE data structure, and you can see this structure by pressing F12 on your browser. This is known as DOM (Document Object Model).**
- Every html element can have a unique identifier (id), apart from that we can put a name which is a semantic identifier.
- F12 known as browser developer toolbar. This is where we can also check the network calls which our client is sending to the server.

There is nothing like Magic!

Block level elements

These elements does not have their own footprint on the screen. But they are responsible to organize our html content on the page.

div – division tag

Tag which automatically starts from a new line. It is used to group multiple html elements on the page and optionally we can add the class name property. Class – classification.

span – inline tag

Inline element used to group elements in a straight line. We also have the class property.

Exercise:

Design a list tag with following topics

ABAP on HANA, SAP Analytics Cloud, SAP BTP, SAP UI5 and Fiori, SAP Datasphere

Make each item clickable and when we click each item, it should take to a introduction video of that topic from below channel

https://www.youtube.com/@AnubhavOberoy/videos

CSS – Cascading Style Sheets

CSS Stands for cascading style sheet, it is used to style, format, beautify HTML content of a page. Before CSS all the styling was done by tag names which become a nightmare for the developers.

<p>I love India</p>

Styled:

<p><color value="red"><b>I</b></color><font name="calibri><size value="10">Love</size></font><color value="blue">India</color></p>

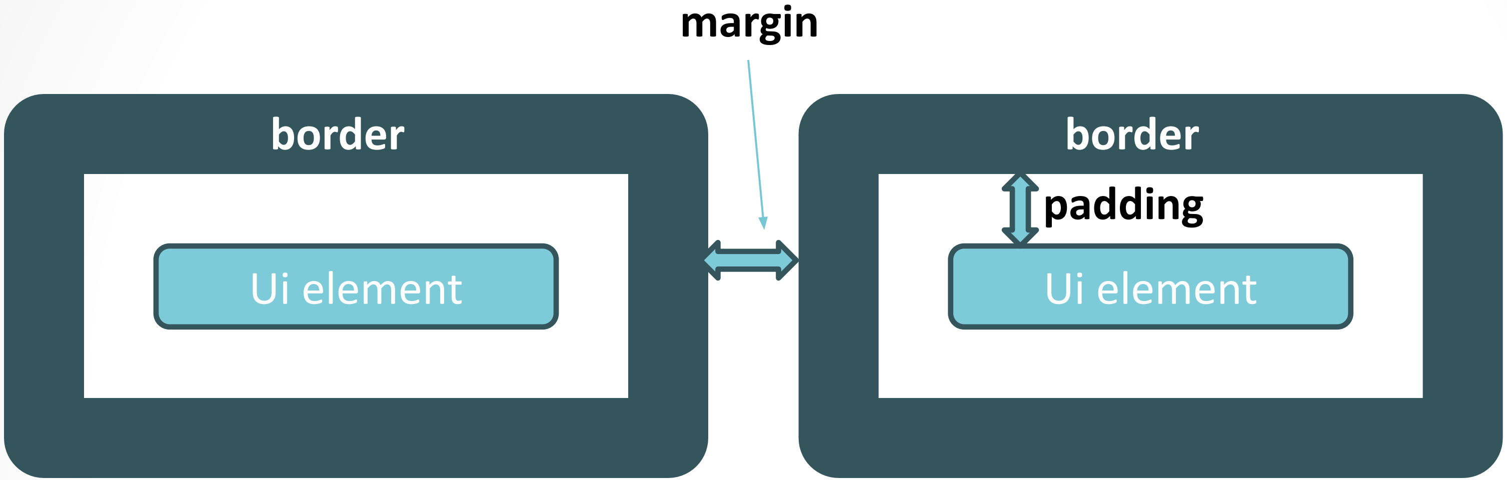
There are 3 ways to apply style on HTML

1. Inline Style – the style code will be written at html element level, this will lead to high maintenance again in future. To apply this, we use **style** property at tag level with syntax - <tagname **style="prop:val;prop:val"**>CONTENT</tag>
2. Internal Style – When the style code is applied at the page level using a **<style>** tag usually inside the header. This is fully reusable and lead to very less maintenance affecting multiple element(S) together.

```
<style>
        selector {
                prop: value,
                prop: value,
                …
        }
</style>
```

3. External Style – when the style code is written in a separate file with extension .css and reference the style code using a **link** tag.

css box model



Responsive web app – when an application adapt itself according to the device automatically.

Pseudo-classes in CSS

A special CSS statement given after colon in selection. Selection:pseudoclass

It applies conditional styling. For example, when user focus, hover, click and element then only we want styling to be applied.

Java Script (JS)

Java Script is browser's programming language, it is a light weight programming language which all the browsers understand directly. Java and Java Script are not same, they are different in nature, purpose and usage.
There are 2 types of JS
Client side JS – used to create web applications, when JS code runs in your browser, it is client side JS. Because all the browsers directly understand it w/o need of any compiler/interpreter. This course mainly focus on client side JS only. Since the code runs in client(browser) it is extremely fast. It is used to build web applications (Fiori apps)
Server side JS – When we execute our code outside the browser in a server using **Node JS** framework, it is called server side js. There is no difference in syntax, your code is running outside the browser.

How to apply JS?

- **Inline JS –** When JS code is written at html element level against an event (when user does an activity like click on a button the onclick event will trigger). <tagName **event=**"eventHandler()"></tagName>
- **Internal JS –** When we keep our code in a special tag called **<script>** tag in the header of the page with multiple functions. Making the code reusable.
- **External JS –** When JS code is written in an external file with **.js** extension. We finally connect the file using <script> tag only.

You can check all the list of events on w3c - https://www.w3schools.com/jsref/dom_obj_event.asp

Java Script Funda Fox

- All the browser’s understand JS automatically
- Every Statement in JS ends with semi-colon (;)
- JS is a case-sensitive programming language – abc is not same as Abc or aBc
- We follow camel case naming convention in js – first letter of a variable or function name always starts with small letter and next consecutive word’s first letter will start from caps. E.g. Iloveindia□ iLoveIndia
- **Java Script is Asynchronous Non Blocking IO programming language**
- In JS when a variable is created and a value is assigned, the data type is automatically allocated.
- We can use **var OR let** keywords to create variables of different kind.
- When we create a variable a NULL value is assigned automatically if we miss to assign a value to it
- In JS = is considered as Assignment operator, But == and === are treated as comparison operators.
- **Console is your best friend. F12 – browser developer toolbar**
- Symbol □ ( ) – parenthesis { } – curly braces [ ] – bracket
- In JS we have 2 very powerful in built variables
    - **document –** it is the object of DOM (document object Model), it can be used to manipulate the HTML runtime structure. It offers functions (APIs) – *getElementsByTagName, getElementsByClassName, getElementById*
    - **window –** it is the object of our browser window object so with the help of that we can also communicate the browser window.

Output functions in JS

There are 4 output functions

- alert – to show a popup to the user
- console.log – to print code inside console
- document.write – to change the whole DOM
- element.innerText – to change text of a element

Usage of JS

- JS can do following
    - Validate user input
    - Show output to the user
    - Manipulate data and DOM
    - Change CSS at runtime
    - Add dynamic elements
    - Write business and validation logic

Variables in JS

Scalar Variables

- Holds single value
- Write a value it overwrites

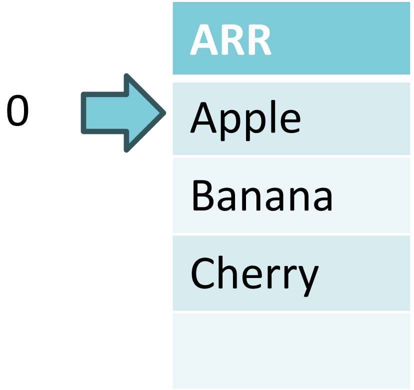
var x = 10;

x = 20;

Exercise (internal table):
Create a json with array of object for employees data in your company. Add 5 records of employee with each employee having json properties as *empId, empName, gender, salary, currency, smoker, rating*

Arrays

- Collection of multiple values of same data type
- They always processed via index
- The first item index starts with zero (0)



0

ARR

Apple

Banana

Cherry

Object

- Like structure in ABAP
- They hold multiple values of different kind of data
- Created using { } and refer properties using OBJECTNAME-PROPNAME
- Key value pairs

```
{
   "prop": value,
   "prop": value,
   …
}
```

Java Script Object Notation = JSON
https://jsonlint.com

Functions in JS

Block of JS code which is reusable

Anonymous Function – a function w/o a name

```
function (param1, param2){
   --code
}
```

```
(param1, param2,…) => {

}
```

Named Function

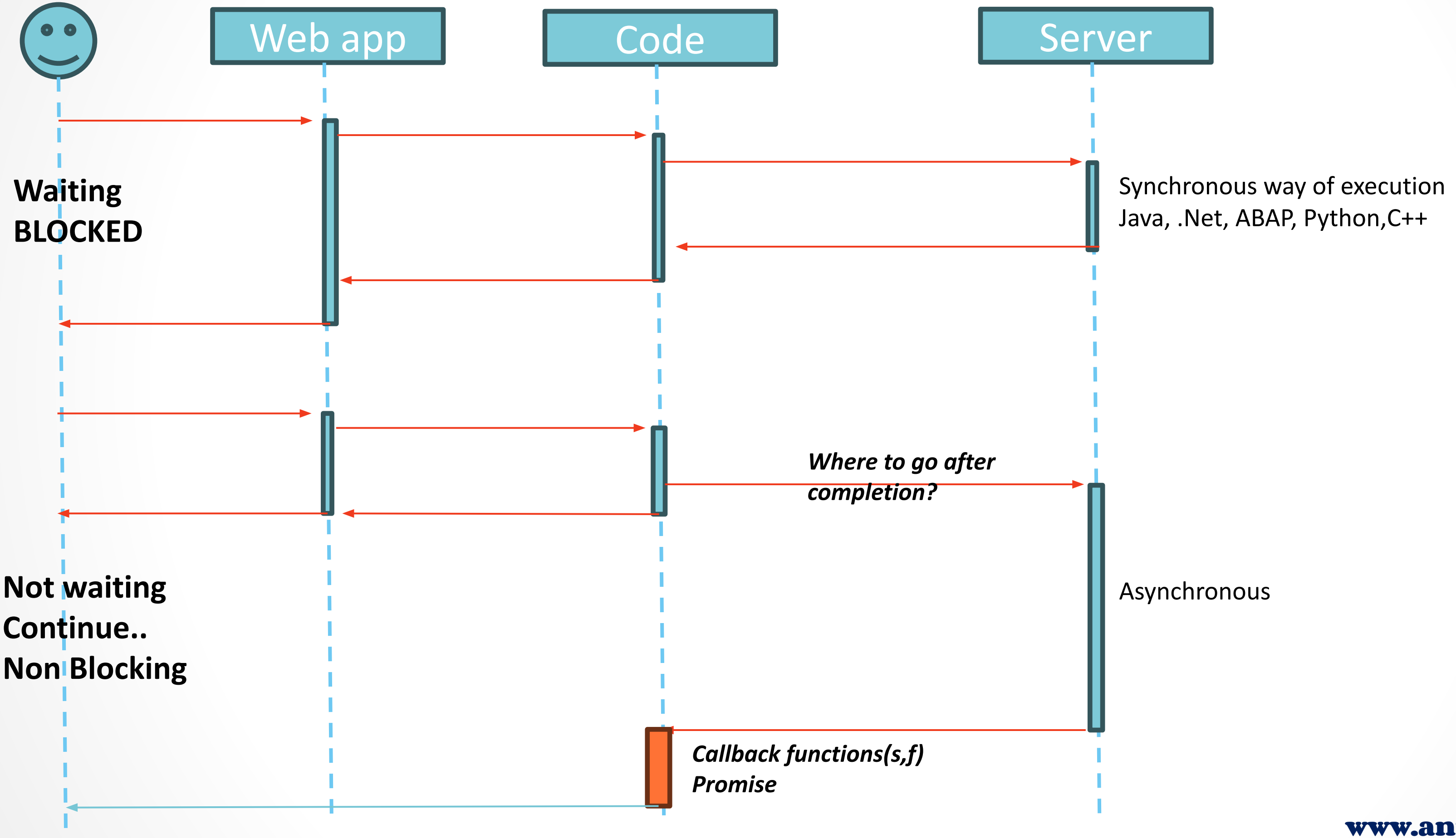
```
function myFunctionName(param1,param2){
  --code
}
```

```
var functionName = (param1, param2,…) => {

}
```

Debug	JS	ABAP
Step over	F10	F6
Step inside	F11	F5
Step Out	Shift+F11	F7
Execute	F8	F8

Asynchronous Non Blocking IO



Waiting

BLOCKED

Synchronous way of execution
Java, .Net, ABAP, Python,C++

Where to go after completion?

Not waiting

Continue..

Non Blocking

Asynchronous

Callback functions(s,f)
Promise

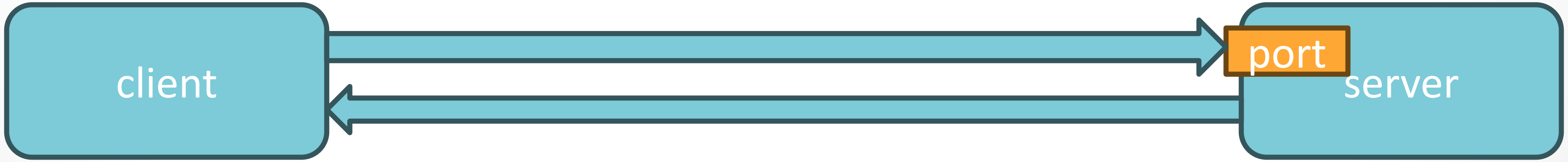
Node JS

Node JS is a framework which is capable of executing JS code outside the browser, leaving developers learning only one language. Node JS typically runs inside a server and can do followings

- Perform DB lookups
- Execute server side logic with manipulating data
- Writing business logic
- Perform the tasks like sending email, triggering APIs, integration logic

The main advantage of node js is to offer a very big eco system which allows developers to leverage modules built by community. We can consume the eco system using the npm (node package manager) for example creating excel, pdf, WhatsApp, sending mails.
https://npmjs.com

We need to install node js – node version 18
https://nodejs.org/en/blog/release/v18.18.0
https://nodejs.org/dist/v18.18.0/node-v18.18.0-x64.msi



How to create a node js project

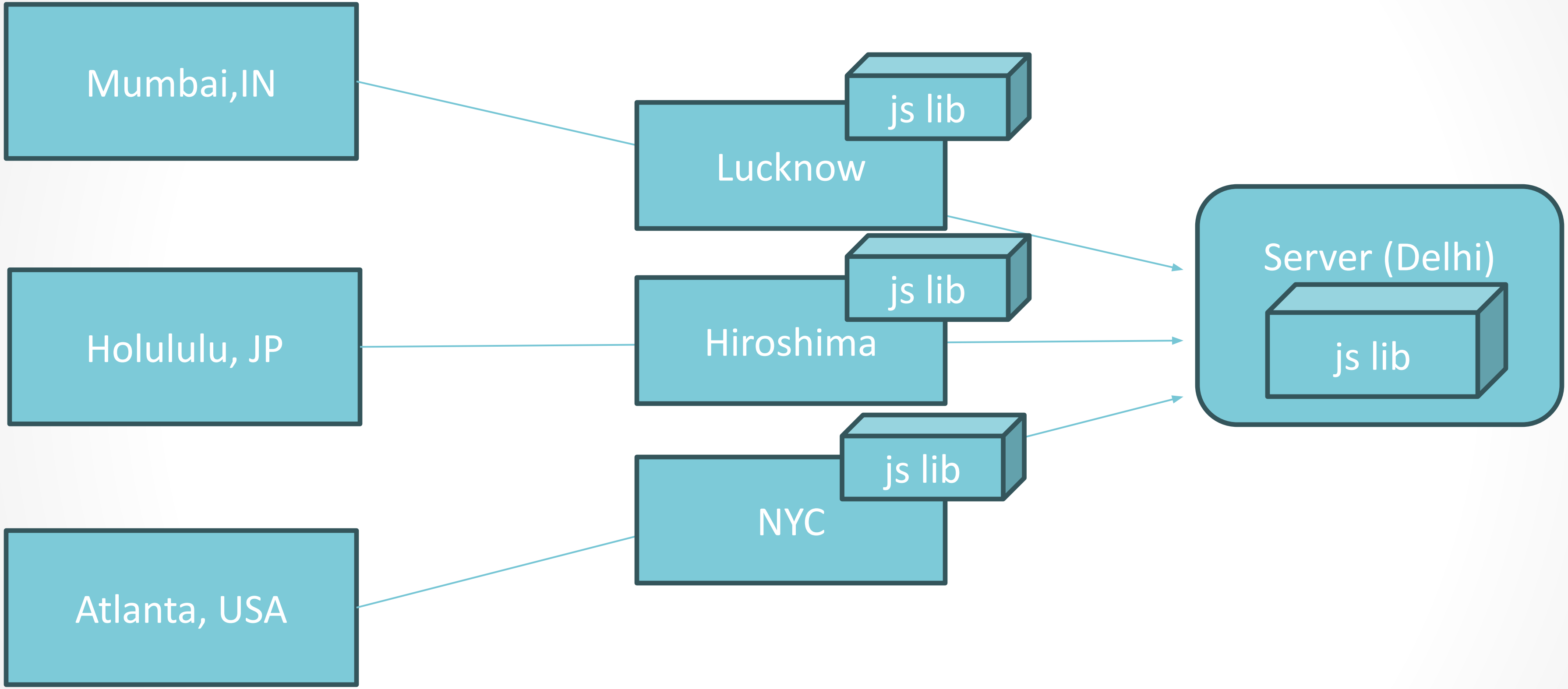
1. Create a project directory
2. Run command **npm init** to initialize a project which will create a file **package.json**
3. In this file we will have details about project name, dependencies, description, start script, etc. Later on when we share this project with other, they will also know what dependencies to install to be able to run our project
4. When we share project we never share the node_modules which are reusable, because they can run a command **npm install** which will read package.json file and automatically fetch the modules from npmjs.com
5. If you are inculcating new module inside your project from npmjs.com, you must install the node module using following commands, it will go to npmjs on internet and LOOK for that module
   1. Node module @ project level – **npm install <MODULE>**
   2. Node module @ system level – **npm install -g <MODULE>**

We can install SAP UI5 tooling which we will be using in our course

npm install -g @ui5/cli

npm install -g @sap/ux-ui5-tooling

Content Delivery Network (CDN)



jQuery

It has **nothing** to do with database SELECT queries. jQuery is a light-weight JS library which is used to simplify our JS work. The moto is **write less and do more.**

Why jQuery?

- Its Free
- It is used by top industry leader e.g. Google, Microsoft, Netflix
- Fast and speed up our JS work
- Perform multiple complex JS task with easy options
- It offers lots of free functions to speedup adaption of JS

How to use? – https://jquery.com

- Download the jquery and reference in local project
- Directly reference from CDN

What is the Syntax?

$(selector).action();

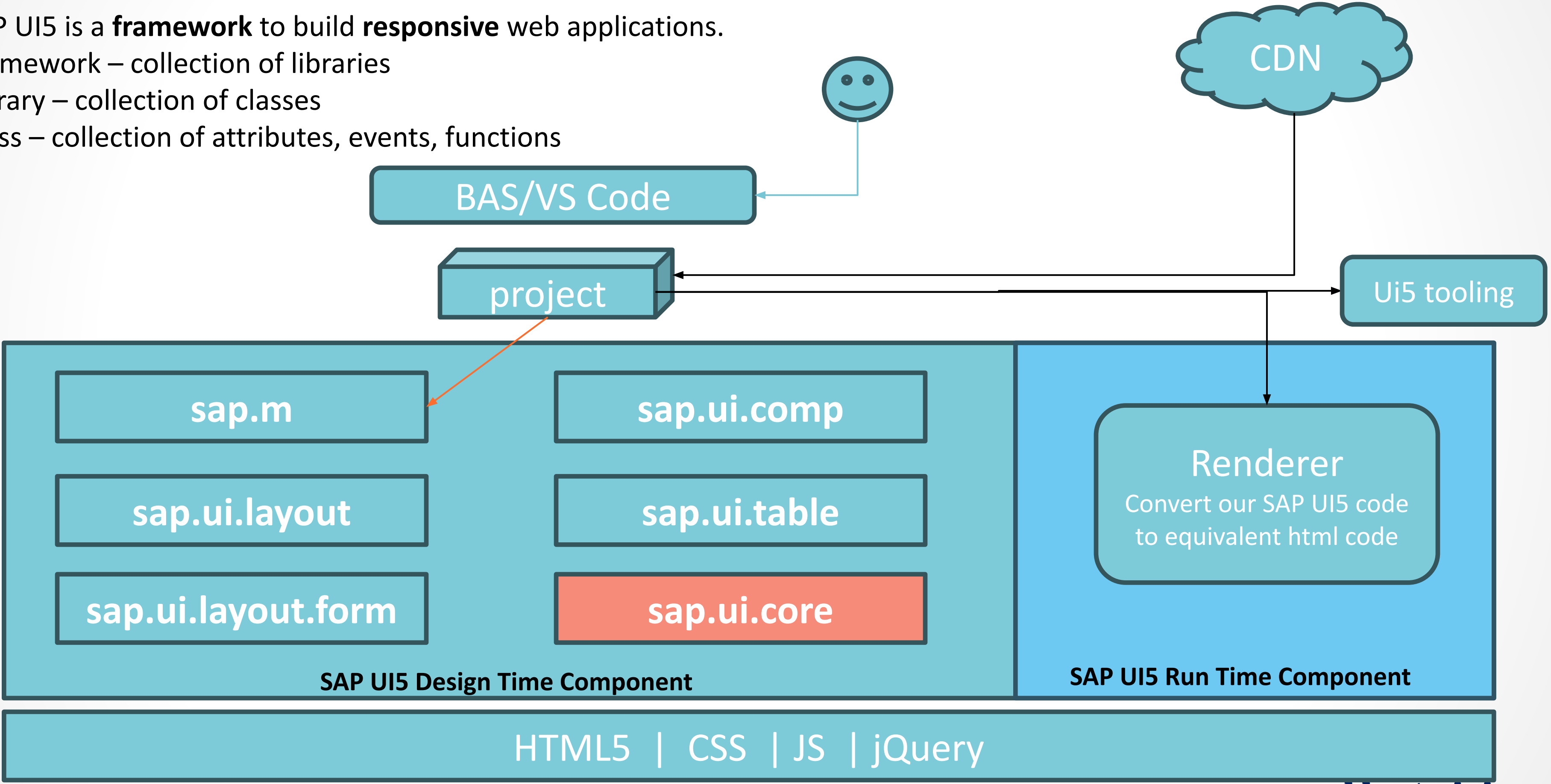
$ - indicates the browser that we want to use jQuery

Selector – *tagName, .className, #id*

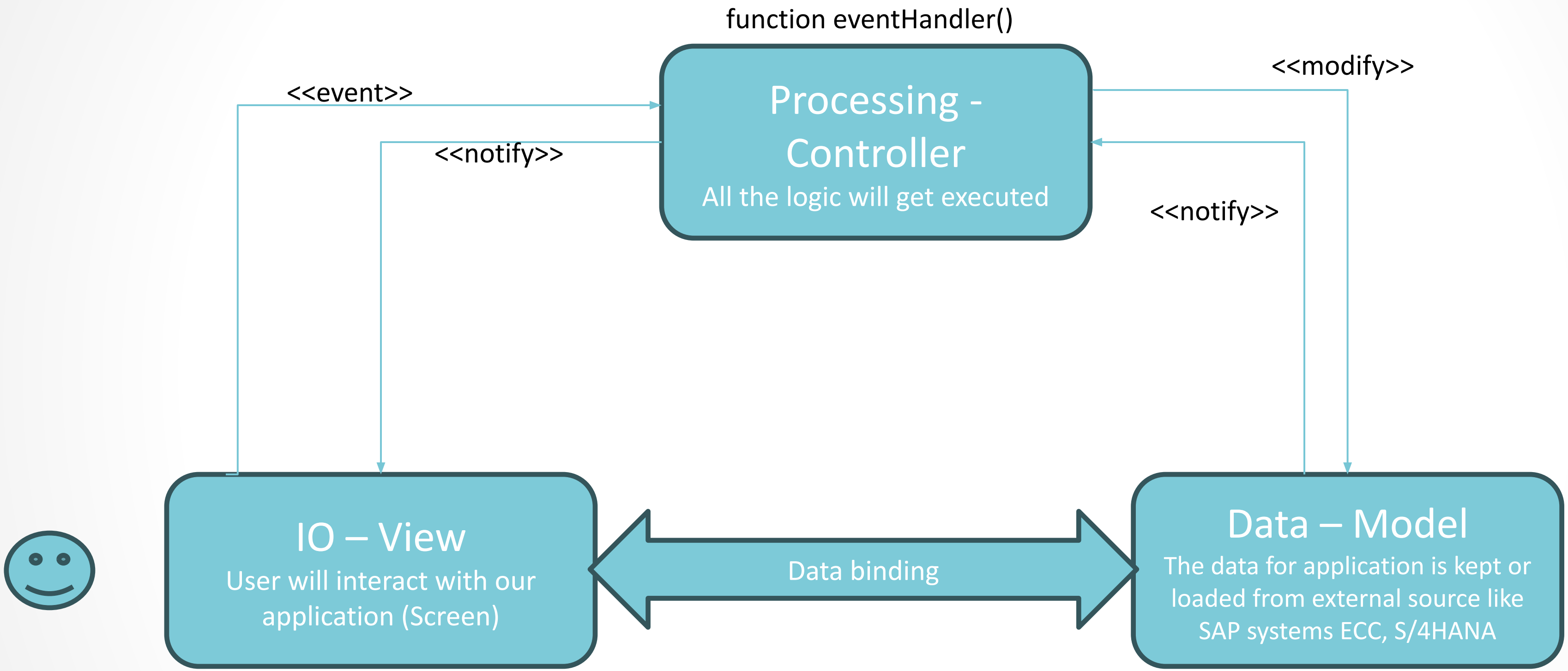
Action – functions which we can use

What is SAP UI5

SAP UI5 is a **framework** to build **responsive** web applications.
Framework – collection of libraries
Library – collection of classes
Class – collection of attributes, events, functions


CDN
BAS/VS Code
project
Ui5 tooling
sap.m
sap.ui.comp
sap.ui.layout
sap.ui.table
sap.ui.layout.form
sap.ui.core
SAP UI5 Design Time Component
Renderer
Convert our SAP UI5 code to equivalent html code
SAP UI5 Run Time Component
HTML5 | CSS | JS | jQuery

MVC Architecture – Any UI App


function eventHandler()
Processing - Controller
All the logic will get executed
<<event>>
<<notify>>
<<modify>>
<<notify>>
IO – View
User will interact with our application (Screen)
Data binding
Data – Model
The data for application is kept or loaded from external source like SAP systems ECC, S/4HANA

Data binding – process of connecting model and the view together. The model and view can exchange the data between themselves.

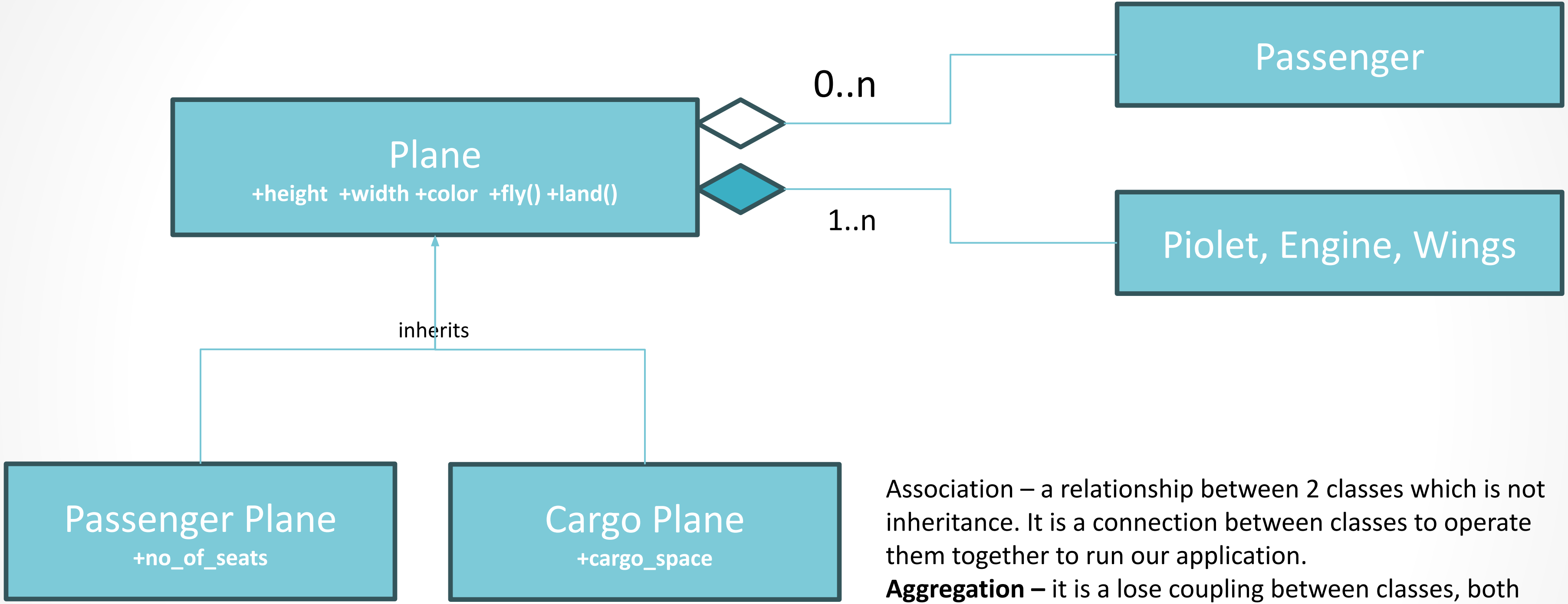
SAP UI5 SDK

- I want to read book on SAP UI5, can you recommend book/documentation
- Will you share some pdf document for study
- If ui5 brings new versions and features in future, after our training, how can I update myself
- My company want to use Fiori but do not have Fiori apps to experience right now, can you show free Fiori apps to my manager
- Anubhav you explained about the framework, libraries and classes, how can I view these classes to check what properties, events, methods these framework classes offer me to build applications

 SAP UI5 SDK (Software Development Kit)

https://ui5.sap.com

OOPS programming



Association – a relationship between 2 classes which is not inheritance. It is a connection between classes to operate them together to run our application.
Aggregation – it is a lose coupling between classes, both classes can function independently, should have relation.
Composition – it is a tight coupling between classes, both classes cannot function independently, must have relation.

Setting up first UI5 project

--initialization of a project

- Create a project folder
- Run **npm init** command to mark it as node project because we plan to use ui5 tooling which is a node module
- Create a webapp>manifest.json file to provide a unique ID of our UI5 application (mandatory), type of app
- Create a index.html file to start our code
- Initialize our project by UI5 tooling for that run **ui5 init** command (already installed @ui5/cli module)

--main course

- To run the project, simply command **ui5 serve**

Views

Views are the user interface for our applications. There are 4 types of views in UI5

- JS View
- XML View
- JSON View
- HTML View

Controller

Controllers are classes in SAP UI5, which are responsible to provide the processing logic.

Syntax to create a **class** in SAP UI5 – AMD (Asynchronous Module Loading) like syntax / Scaffolding syntax

```
sap.ui.define( [ dep1, dep2, dep3 ],
               function(oD1, oD2, oD3){

      });
```

Syntax to create a **controller class** in SAP UI5

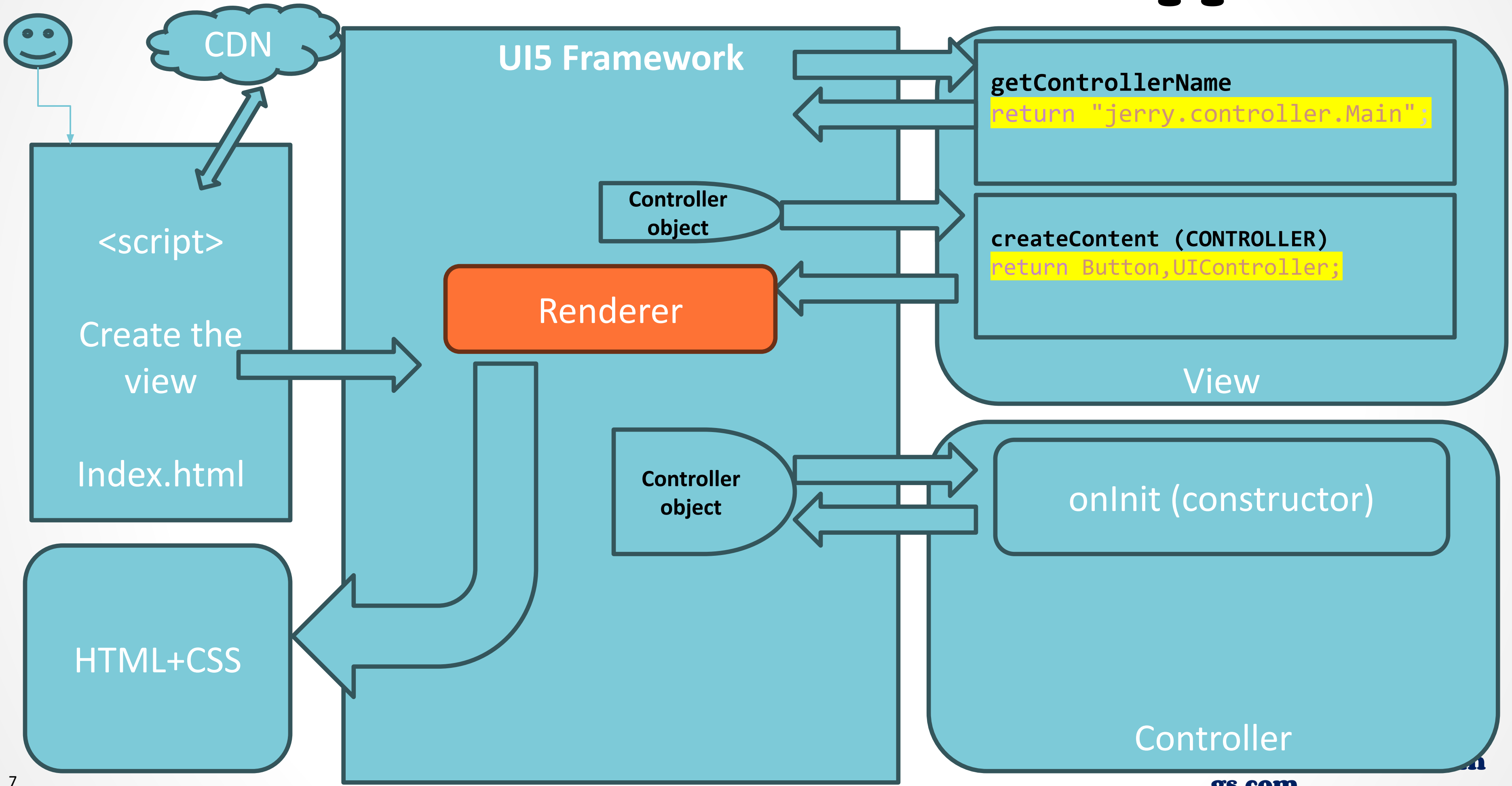
To make a class as controller class, sap says that you must inherit from a standard sap module called **sap/ui/core/mvc/Controller** module. In js the inheritance is indicated using a keyword called **extends**

A class in SAP UI5 can be referred as module just by replacing dot (.) with slash (/)

```
sap.ui.define( [ dep1, dep2, dep3 ],
               function(oD1, oD2, oD3){

      });
```

Under the hood – Runtime of our app



```
getControllerName
return "jerry.controller.Main";
```

```
createContent (CONTROLLER)
return Button,UIController;
```

Funda Fox

- When we want to interact with our UI elements, though we can use **document.getElementById** but we should never use it. Because it always provide us the ***HTML element object*** never the UI5 object, so with this we will limit ourself to use plain html. Hence we wont have the benefits of the framework.
- When we want to get the object of ***UI5 element object*** we can use the in-built UI5 api
- Get the object of running instance of our application – **sap.ui.getCore()**
- On top we can call the **byId(**controlId**)** function to obtain the UI5 control object
- The benefit here is that now we can use ALL the APIs of SAP UI5 framework for that control. To set/reset values, change behavior, change visibility etc.
- For every property of a control in SAP UI5, we have **setter** and **getter** methods (api). For example, if there is a value property for input field, we have **setValue** and **getValue** functions.
- When we create XML Views, the ID of the control will be changed to the **viewId—controlId** in the DOM hence when we use sap.ui.Core to get control object, we must use the complete id.
- In case of xml views, we want avoid VIEW ID in the byID function to get control object, hence we can first get the view object **this.getView()** and then on top get the control object using **byId()**
- Inside a controller, we can use **this** pointer, which points to current class object (our current controller).
- If there is a attribute, method, aggregation, event which you cant find inside a class, check the parent class.
- A library name is always small letters, a Control class name always starts capital letter and follow camel case.
- An aggregation always starts with small letter and follow camel case, an aggregation must follow namespace of the control class. If an aggregation is **default** we can skip writing it.

What is XML and XML Views

XML stands for eXtensible Markup Language, which is also a tree data structure but unlike HTML, the XML does not have standard tags, all the tags are custom tag.

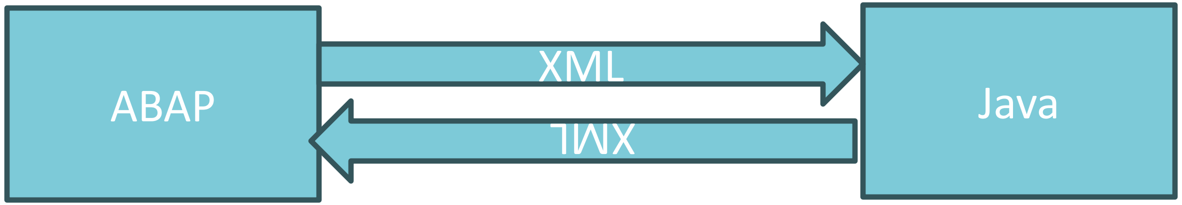
Root node

```
<data>
        <empStr>
                <empId>100</empId>
                <empName>Anubhav</empName>
                <salary>5000</salary>
        </empStr>
</data>
```

The origination of xml was to share/transfer data between heterogeneous technologies.

Internal table □ XML : Rendering
XML □ Internal Table : Parsing

XML □ Collections
Collections □ XML



Why XML views over JS Views

- XML views are industry standards all the top companies
- XML processing is much faster than JS views, parsing of XML to HTML by renderer is much faster
- XML views follow pure MVC architecture where NO ONE can ever put processing logic inside an XML
- No one can do JS injections inside the view

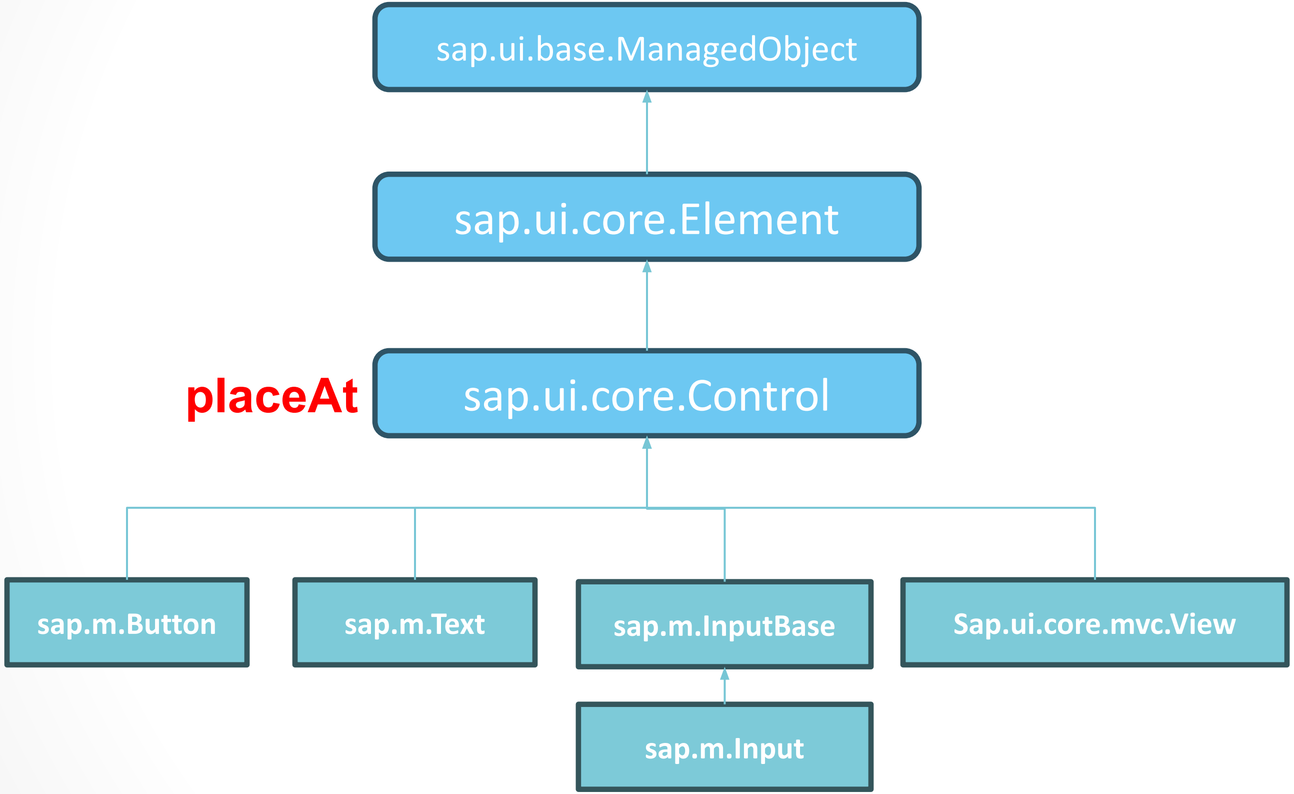
Syntax

```
<satish:View xmlns:satish=“sap.ui.core.mvc” xmlns=“sap.m” controllerName=“controllerAddress”>
        <Button></Button>
        <Button />
</satish:View>
```

- Can 2 views points to same controller - Yes
- Is vice-versa possible that 1 view can point to 2 controllers - No
- If I have multiple views can we write processing logic for all in one controller or what’s the best-practice – Ideally have ONE Controller for ONE View

Control Hierarchy

It is the representation of objects arrangement inside SAP UI5



Title

Design a form which shows a list of label and input fields for empId, empName, salary, smoker, currency fields.
sap.ui.layout.form.SimpleForm

Model

Model is a ***object of data***. It hold and control all the data which we need in our applications. It also used to manipulate the data like in ABAP we have internal tables. There are 4 **kind** of models in SAP UI5

- JSON Model
- XML Model
- Resource Model

Client-Side Models – These models keeps all the data in client side(browser) when our app starts. Best suited for small size of data which require frequent manipulation of our data. Like internal tables in abap. They get destroyed when we close application.

- OData Model

Server-Side Models – These models keeps all the data in the server. Best suited for large size of data which require to be stored in SAP/Other servers. They will be used to interact with our server to load and perform CURDQ (Create, Update, Read, Delete, and Query) operations on our data.

Steps to create and use model in our app

1. Create a brand new object of the model

   var oModel = ***new** sap.ui.model.json.JSONModel();*

   var oModel = ***new** sap.ui.model.xml.XMLModel();*

   var oModel = ***new** sap.ui.model.resource.ResourceModel();*

   var oModel = ***new** sap.ui.model.odata.v2.ODataModel();*

2. Set or load data inside the model

   oModel.setData(sData);

   oModel.loadData(filePath);

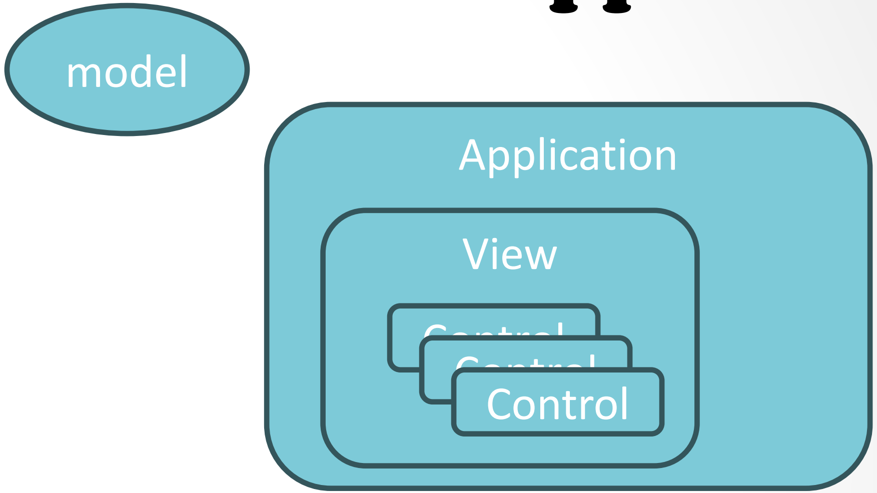
3. Make the model aware to the application/view/control

   sap.ui.getCore().setModel(oModel)

   this.getView().setModel(oModel)

   this.getView().byId("controlId").setModel(oModel)

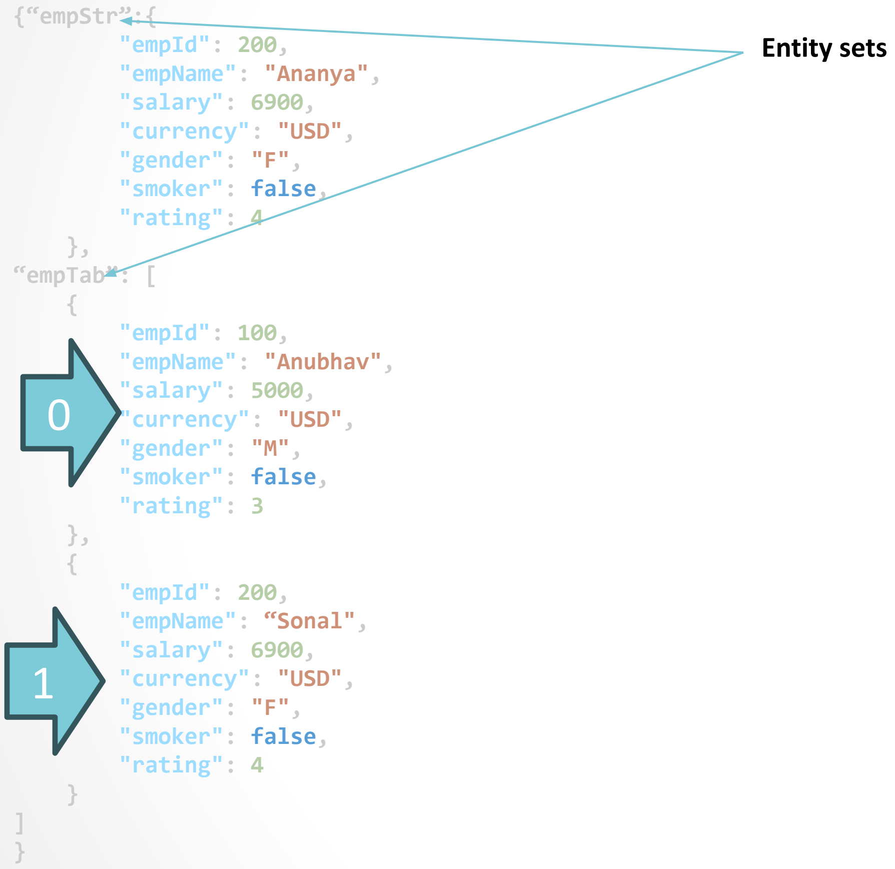
4. Perform binding so that the data exchange can happen between the view and the model – xPath



Exercise: Kindly Complete the binding for remaining 4 fields.

XPath of Model data

Every curly braces is a slash(/) and every array item is a (/index).



```
{“empStr”:{
        "empId": 200,
        "empName": "Ananya",
        "salary": 6900,
        "currency": "USD",
        "gender": "F",
        "smoker": false,
        "rating": 4
    },
“empTab”: [
    {
        "empId": 100,
        "empName": "Anubhav",
        "salary": 5000,
        "currency": "USD",
        "gender": "M",
        "smoker": false,
        "rating": 3
    },
    {
        "empId": 200,
        "empName": “Sonal",
        "salary": 6900,
        "currency": "USD",
        "gender": "F",
        "smoker": false,
        "rating": 4
    }
 ]
}
```

Which currency **ananya** is paid off
/empStr/currency == xpath
What is the xpath for employee id of ananya
/empStr/empId
What is the Rating of Sonal
/empTab/1/rating
Address of salary of Anubhav
/empTab/0/salary

Binding Types

SAP UI5 supports 4 types of binding types

- Property Binding – when we bind property of a control to the xpath, it is called property binding. For example **value** is the property of **Input** control hence when we bind with it, it's a property binding.
- Expression Binding – when we bind an expression with the property of a ui element instead of a xpath, it is called expression binding. The use case is to dynamically determinate the state or the value of a control.

  Syntax:

  {= *condition ? True : false*} in the condition we can use value of xpath using ${xpath}

  e.g. if an employee is smoker, make the employee name field grayed out.

  source = smoker   target = employee name   condition □ smoker == true

- Aggregation Binding – When we bind aggregation of a control with the xpath of the model, it is called aggregation binding. For example, we have **rows** aggregation of a table control which we bind to display data in a table.
- Element Binding

Different Syntax for binding

To do the ***same thing***, SAP provide **different syntaxes (options)**

1. Direct xpath address **value="{xPath}"**
2. Path keyword syntax **value="{path: xPath}"** (*complex binding syntax*)
3. Dynamic binding using **bindValue(xPath)**
4. Dynamic binding with generic method **bindProperty(name_of_prop, xPath)**

How to play with Model and its data

Step 1: in the controller we can obtain the model object any time – **sap.ui.getCore().getModel()**

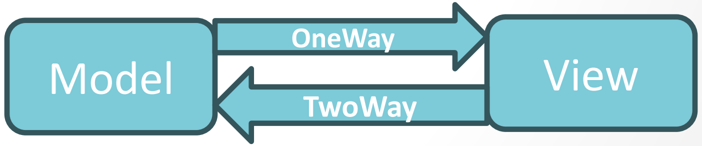
Step 2: We can use the methods(API) provided by SAP to get or set the data using functions

oModel.getProperty(xPath)

oModel.setProperty(xPath, VALUE)

Binding Modes – it controls how the data exchange will happen between the model and the view

- OneWay – When we change the data in the model, if it reflect in the view, its one way binding mode.
- TwoWay – When we change the data in the model, it reflects on UI, and when we change data in UI, it reflects back into the model.
- OneTime – The data exchange happens only at the start of application, after that the model and the ui are completely decoupled.

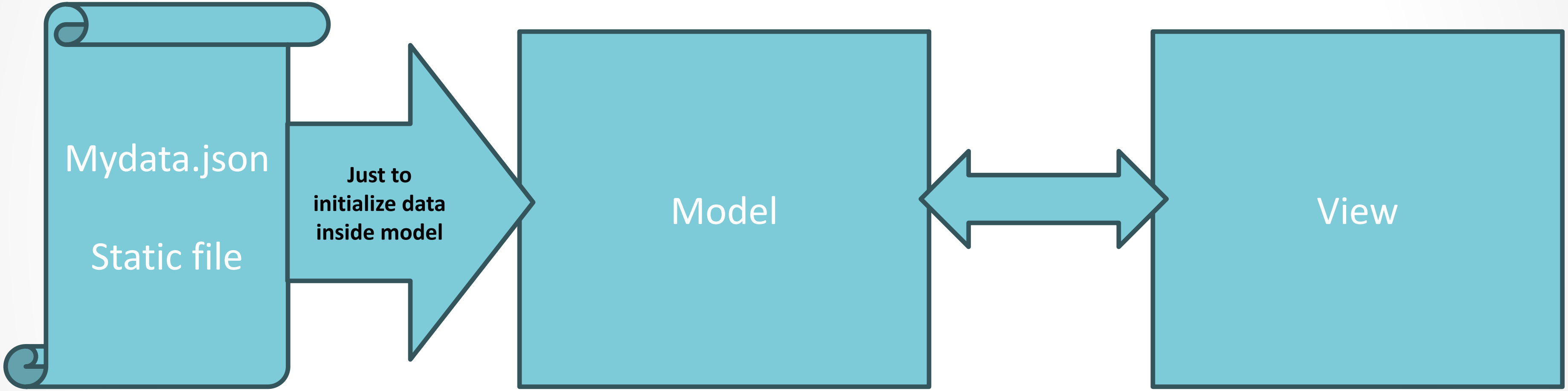


What is the default binding mode for JSON model?

TwoWay

Title

<content>



Mydata.json

Static file

Just to initialize data inside model

Model

View

Hook Functions

Are the functions which gets called by SAP UI5 at certain point in application lifecycle, a developer can use them to trigger their code at certain point. For example at the start of the app, at the end of the app, after the page is displayed to the user.

There are 4 hook functions in SAP UI5

onInit – like the constructor of a class, it gets called once the object of the class (controller) is initialized by the framework. We use this function to write initialization code. The function gets called only **once** in the life of a controller. E.g. we have initialized our model objects inside the init function.

onExit - like the destructor of a class, it gets called once the object of the class (controller) is destroyed by the framework. We use this function to write clean-up code. The function gets called only **once** in the life of a controller. E.g. we have created a connection to external system and at the time application close, we want to release those resources.

onBeforeRendering - like the PBO in ABAP, it gets called again and again everytime just before the corresponding view is displayed to the user. We use this function to write pre-processing code. The function gets called **multiple times** in the life of a controller. E.g. we need to hide a button or make a field Gray out based on some data.

onAfterRendering - like the DOMODIFYVIEW in Webdynrpo ABAP, it gets called again and again everytime just after the corresponding view is displayed to the user. We use this function to write post-processing code. The function gets called **multiple times** in the life of a controller. E.g. we want to apply jquery affect on our ui element(s).

Exercise:

Add multiple records (min 7) inside empTab entityset in json. It should have empid, empname, salary, currency, mStat, gender, smoker, rating etc.

Send your json to Anubhav.abap@gmail.com

Within next 16 h.

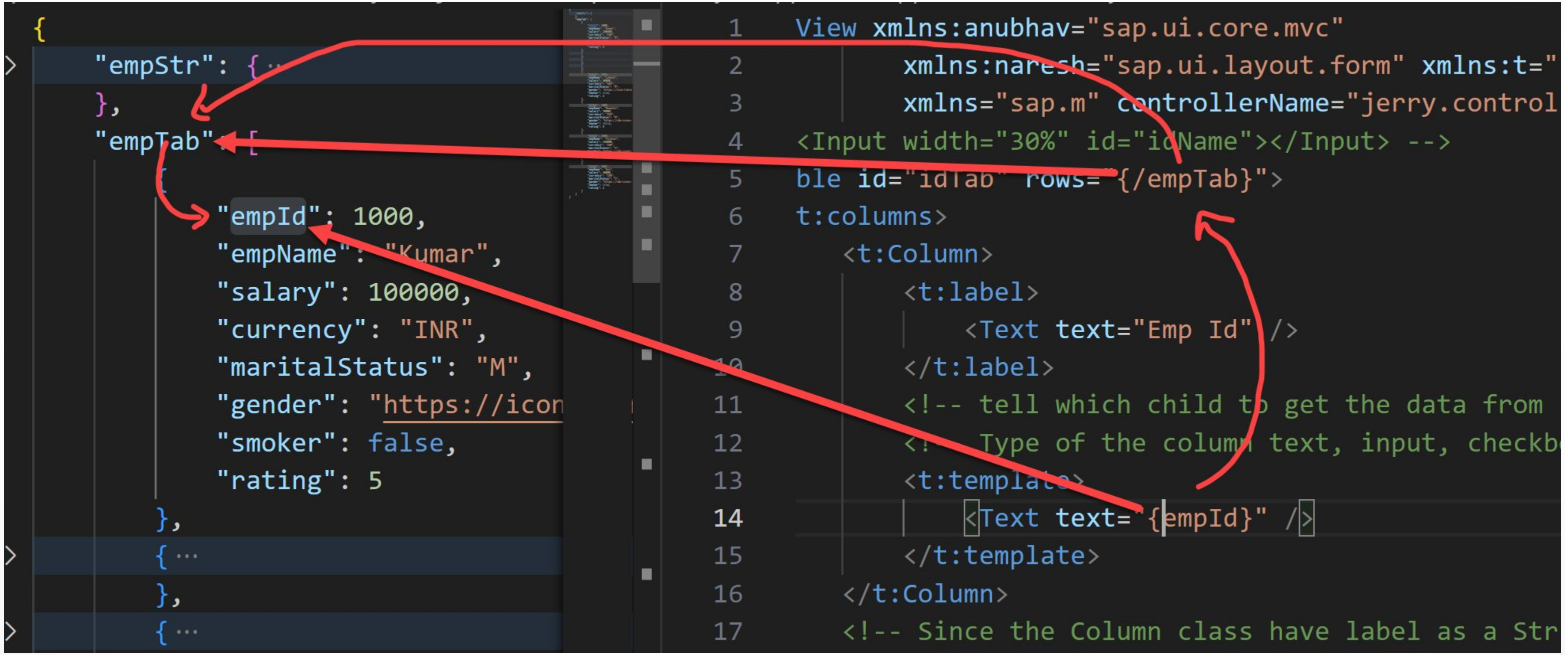
Reusable Variables

When we have a reuse API or a custom variable which is declared, initialized and used in many functions at many places. It will in future lead to high maintenance. We can create global variable.

Aggregation Binding

Absolute Path – When we bind a control with its aggregation, the xpath of the root entity is called Absolute path. In this example the /empTab is a absolute path.

Relative Path – The path which we provide at the child level control which points to the child of absolute xpath in json model, is called relative path. Because its relative to the parent.



```
{
    "empStr": {···
    },
    "empTab": [
        {
            "empId": 1000,
            "empName": "Kumar",
            "salary": 100000,
            "currency": "INR",
            "maritalStatus": "M",
            "gender": "https://icon
            "smoker": false,
            "rating": 5
        },
        {···
        },
        {···
```

```
View xmlns:anubhav="sap.ui.core.mvc"
        xmlns:naresh="sap.ui.layout.form" xmlns:t="
        xmlns="sap.m" controllerName="jerry.control
<Input width="30%" id="idName"></Input> -->
ble id="idTab" rows="{/empTab}">
t:columns>
    <t:Column>
        <t:label>
            <Text text="Emp Id" />
        </t:label>
        <!-- tell which child to get the data from
        <!-- Type of the column text, input, checkb
        <t:template>
            <Text text="{empId}" />
        </t:template>
    </t:Column>
    <!-- Since the Column class have label as a Str
```

Question

1. Can we have multiple models in a single application - YES
2. If yes, how will the system differentiate between them – We can give different names to different models hence the system will be able to recognize between them. This concept is called **named model** in sap ui5
3. When we do the binding, we should use the model name in the binding using named model. **Model>xPath**
4. Is it also possible to set xml, resource, json, all kinds of models together in a single app – **YES, if we use unique name for each model, we can do that**

Internationalization – i18n

A resource model is used for internationalization, which allows us to support for multiple languages. Using this approach we can extend our app in different language. We need to create a file i18n.properties where we maintain all the text labels.

Developer

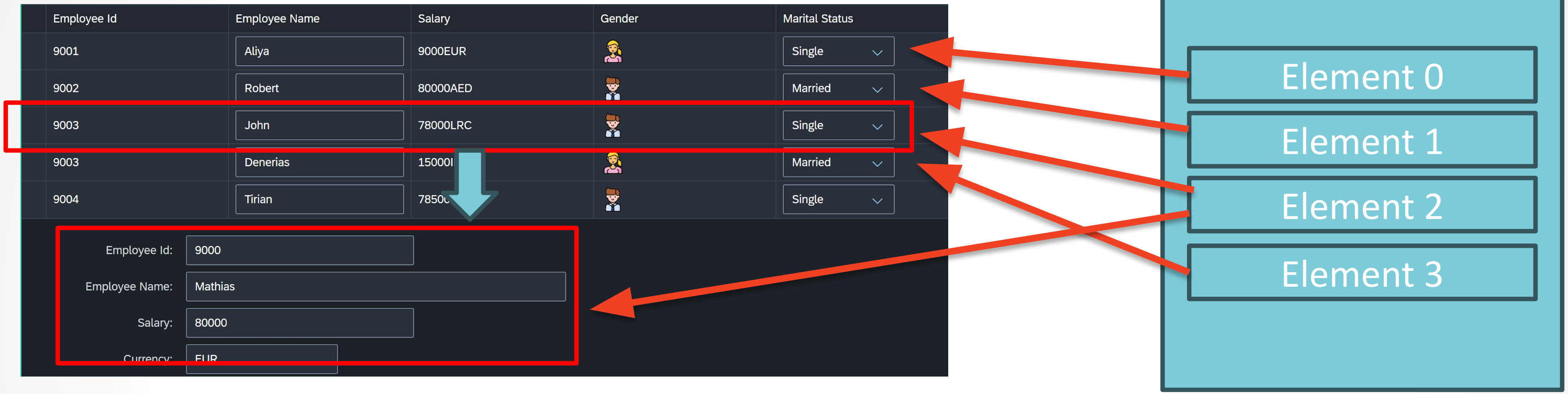
- Create the i18n resource file with text labels
- Create model
- Bind this named model with all static texts

KM Manager

- Copy the i18n file to language specific file
- Rename with language code
- Maintain language labels

For testing we can use ?sap-ui-language=CODE

Title



	Employee Id	Employee Name	Salary	Gender	Marital Status
	9001	Aliya	9000EUR		Single
	9002	Robert	80000AED		Married
	9003	John	78000LRC		Single
	9003	Denerias	15000I		Married
	9004	Tirian	7850		Single

Employee Id:	9000
Employee Name:	Mathias
Salary:	80000
Currency:	EUR

Element – is the memory allocated by model. So from empTab, these elements are created by model in the memory.

Element Binding – when we bind an element of a aggregation binding with another control, in order to get the same data which is available at the level of table.

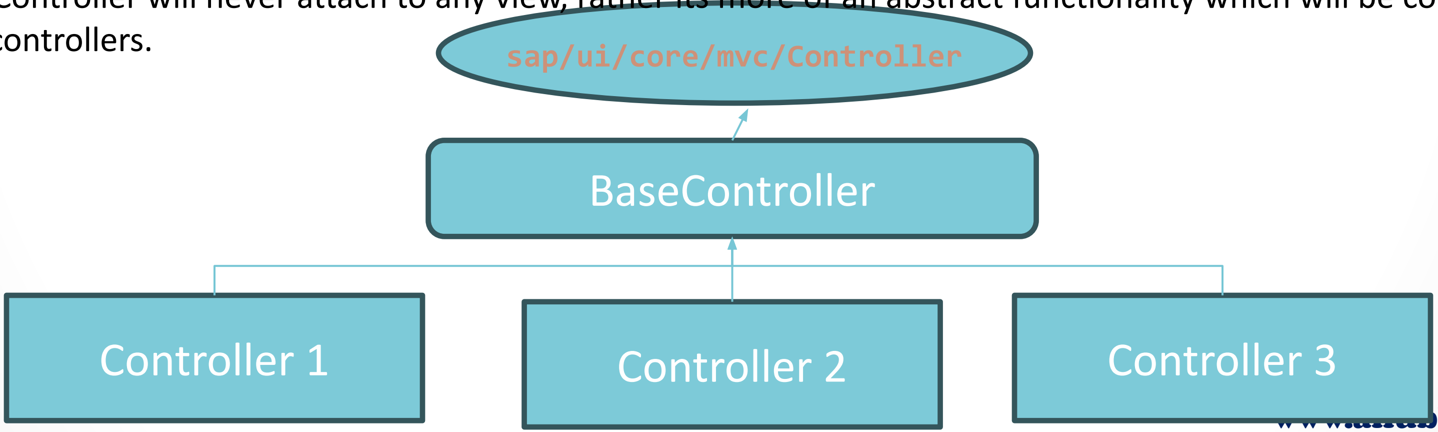
Steps:

1. When we choose a row, we will determine the address of the element **/empTab/index**
2. Bind the simple form with same element address (absolute path for simple form)
3. All the child of simple form should be bound to element's children's as relative path

Funda Fox

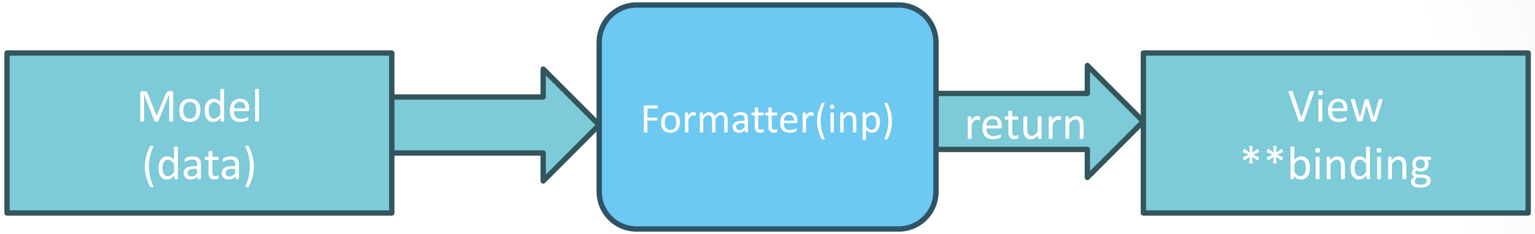
- In SAP UI5, whenever an event is triggered, it will pass a default event object to the event handler method
- This event object will carry lots of information with regard to the event
- This information can be used to extract the details of the event. We can also check the event parameters by checking SDK for that event
- SAP offers 3 APIs for the event object
    - getParameters – to know all the event parameters
    - getParameter – to know single parameter details
    - getSource – to get the object of source control

- At times we need some common variables and functions inside all our controllers, so instead of creating these variables and functions in each controller, we can define a parent controller (BaseController)
- This BaseController will never attach to any view, rather its more of an abstract functionality which will be consumed by other controllers.



Formatter

Formatter is a function which used to format the data in sap ui5. At times we have requirement to format and then show the data to the UI. Instead of changing data inside the original dataset (contamination) we can apply formatter. Formatter will format our data @ Runtime hence the original source is never modified.



Use cases: time formatting, date formatting, text formatting, currency formatting.

1. Create a formatter module (util/Formatter.js) and add functions inside
2. Import the dependency in controller classes
3. Call the formatter in the view

Difference between UI5 and Fiori

https://experience.sap.com/fiori-design-web/sap-fiori/

UI v/s UX

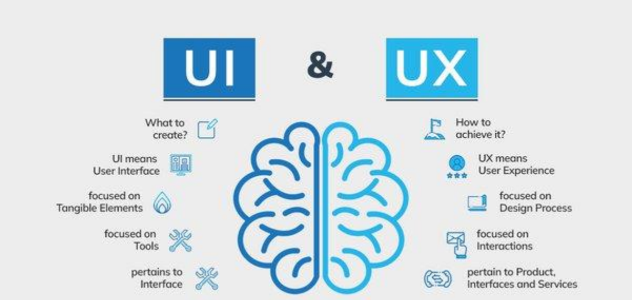
UI stands for User Interface and UX stands for User Experience

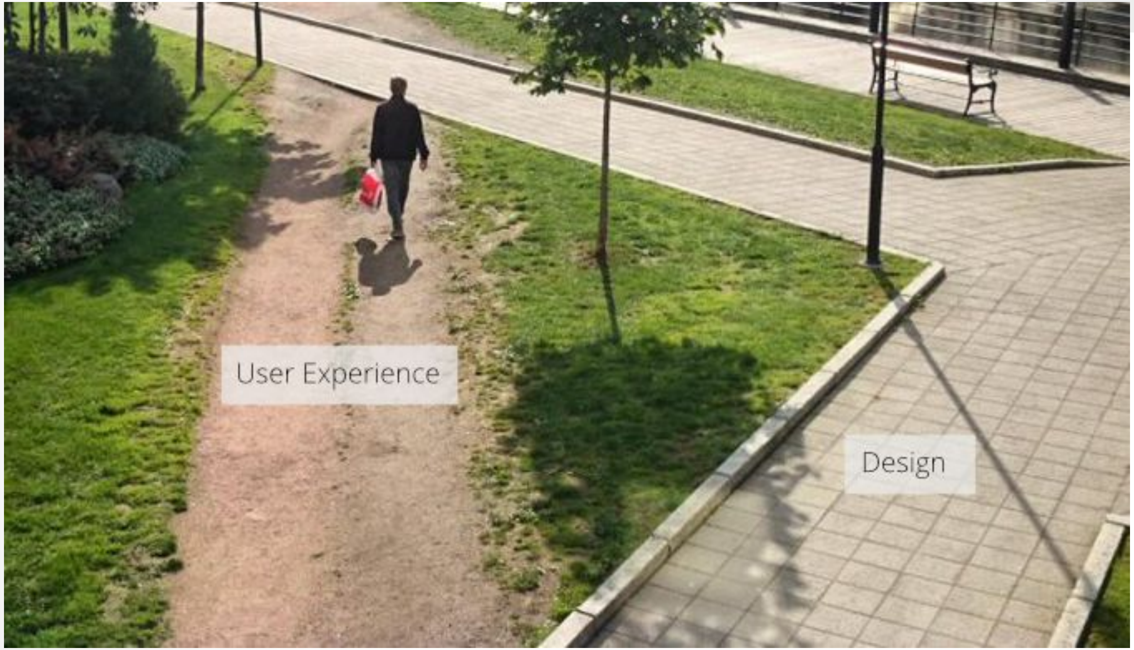
UI is a developer's point of view but the UX is end user point of view

UI can be changed but UX cannot be changed

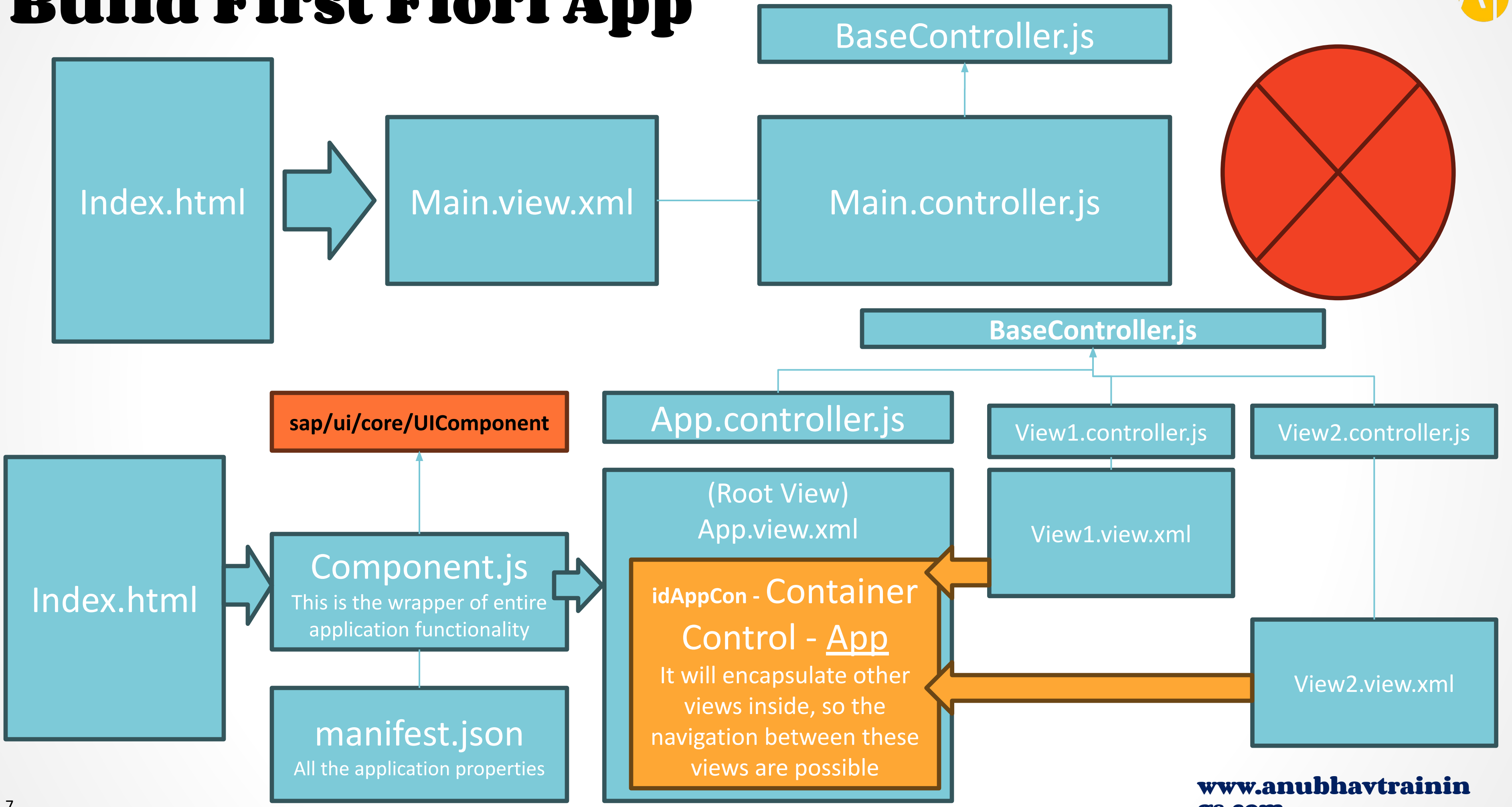
Our goal will always be to make our UI as close as possible to UX

UI is technical but the UX is psychological





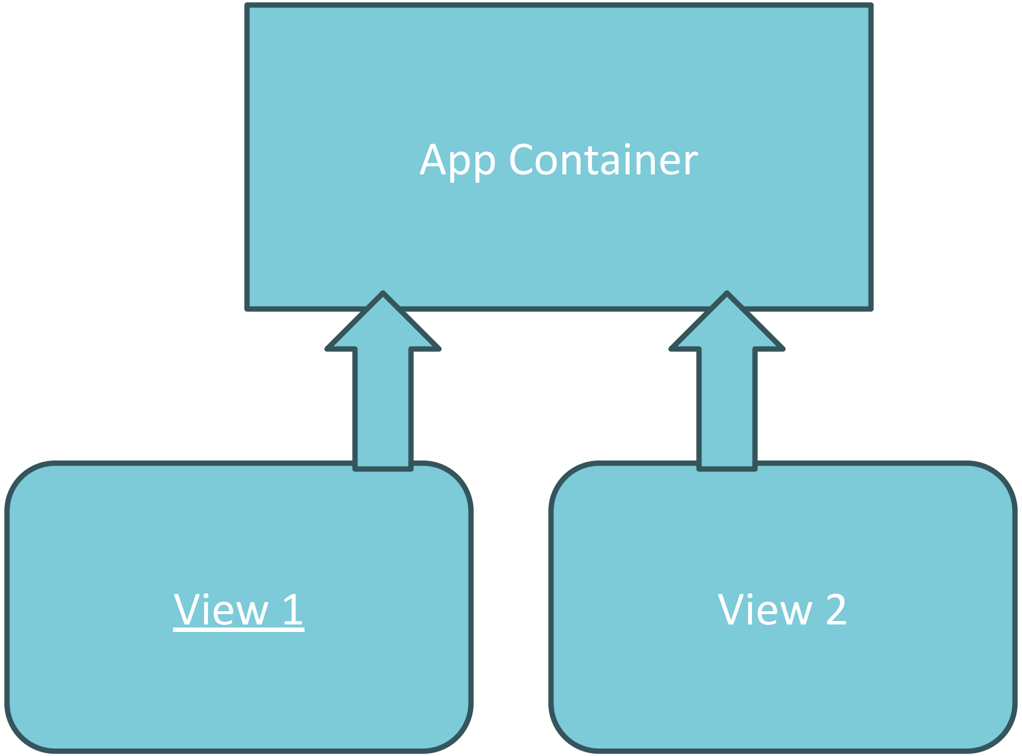
Build First Fiori App



Component.js

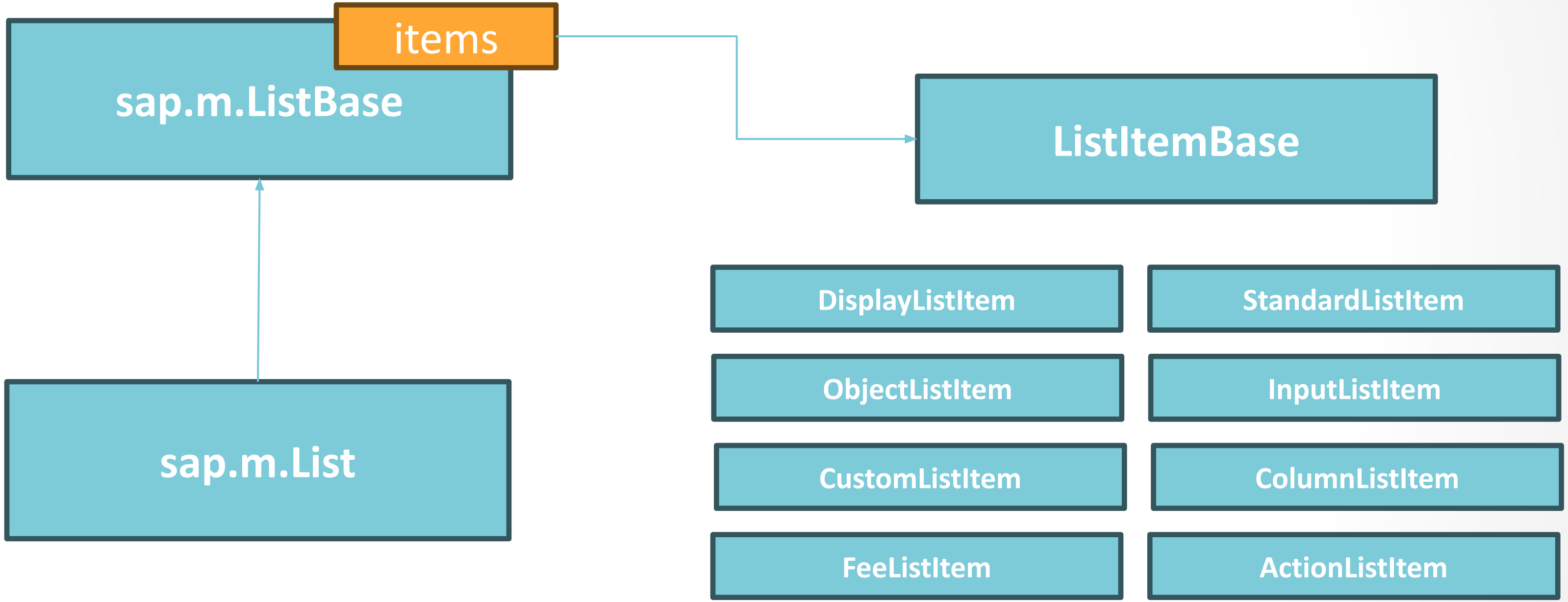
- It is the wrapper of entire application functionality in SAP system
- It inherits from **sap/ui/core/UIComponent** module, So that we can get all the in-built SAP Functionality from this class
- It is the starring point of every Fiori application
- An application which contains **Component.js**, is called **Fiori like app**
- Usually the Component.js exist in the root directory (webapp) of our application
- There only and only ONE Component.js per application, the name MUST be same **Component.js**
- If our application does not have Component.js, we will NEVER be able to integrate it will launchpad
- It comes with a companion file called manifest.json, which has all the application properties
- All the important application properties like name, id, version, description, docu, dependencies, lib, components etc. are documented inside manifest.json. Many of these manifest properties are merely documentation, some of them will also drive application functionality
- We should never have productive application logic inside index.html like earlier we kept view instantiation in index file. The index file should just start Component.js only
- Component.js is the LAUNCHER/Starring point of ALL Fiori apps
- A Component.js file consists of following sections
  - ❑ Metadata – where we link our manifest.json file
  - ❑ Init() – will write the initialization code, in addition, we will call the parent class constructor
  - ❑ createContent() – this is where we initialize our root view (App.view.xml)
  - ❑ Destroy() – here is where we write clean up code

Navigation between views



List Control

A list control is a single column table. The data inside the list is called **items.** Almost every Fiori app has list controls. The List class inherits from the ListBase class and has items aggregation which points to ListItemBase class.



Purpose of list item

- DisplayListItem – used to create Display only data on screen, shows data left and right aligned
- StandardListItem – used to create data top and bottom with an icon
- InputListItem – used to add input control to the list (standard css and margin classes)
- CustomListItem – if none of the list item fulfil our purpose, we can opt for Custom List Item
- ObjectListItem – it used to display the business object which contains the title, info, icon, number, numberUnit e.g. Sales order, purchase order, a/c document, material, BoM, Service contract

Exercise:
Prepare a json of dummy fruits data as below. We need 3 entity sets
fruits - 15 – name, color, taste, season, price, currency, imageUrl, status(A,O,D)
suppliers - 15 – name, sinceWhen, city, contactNo, contactPerson, email, fruits: [“”,””]
cities - 10 – name, state, famousFor, population
With in 12H – Anubhav.abap@gmail.com

Manifest.jsonApplication Descriptor file

Every Fiori application contains many different project properties as well as declarations for the application code, these things can be kept inside a special file called manifest.json file (application descriptor file).

Why ?

- When we join a new project in any company, at times there is No one who provide us the KT about the existing application, so how we know the details of an app.
- If I build a project in my company and planning to leave the team, how do I make sure that the handover is properly done for the existing application.
- Once upon a time, I was on vacation and enjoying my drink on goa beaches, I get a call on my cell phone, its my manager, they ask me to come back to hotel and connect for sometime to explain other colleague about the application I built before my vacation.
- When SAP deliver a standard sap application to us, how do we know the details about that app as SAP do not share the mobile number of the developer in the files.
- What is the gold standard/best practice to create a Fiori app.
- I had built an app around a year ago, today we have an issue in the app, I forgot what was that application all about, how do I maintain a silent documentation of my app.
- When a fresher joins my team, how can I bring down the time to handover my app.

manifest.json – application descriptor file

It contains most of the documentation properties of our app. Every Fiori app must have the manifest.json file. The application properties like name, id, version, type, dependencies, libraries, models, connections, external, components, reuse componts, devices, themes etc.

https://ui5.sap.com/#/topic/7701636d088147569d99b4f08d418bd9

Exercise

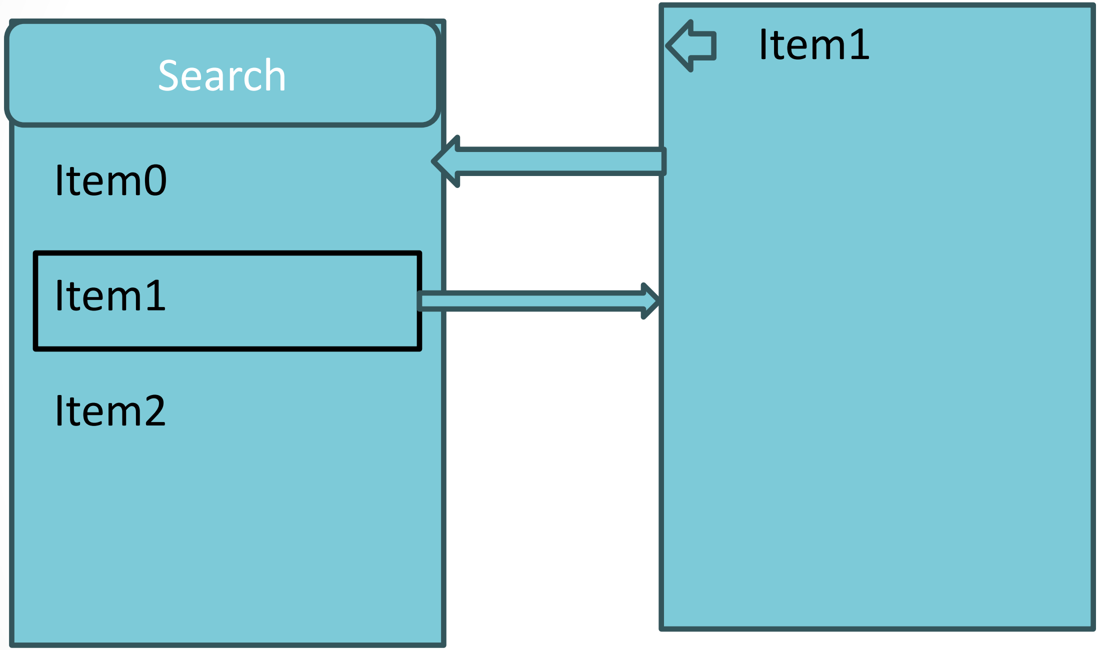
Display the text for the availability and color coding for the status.

09-01-2024 :

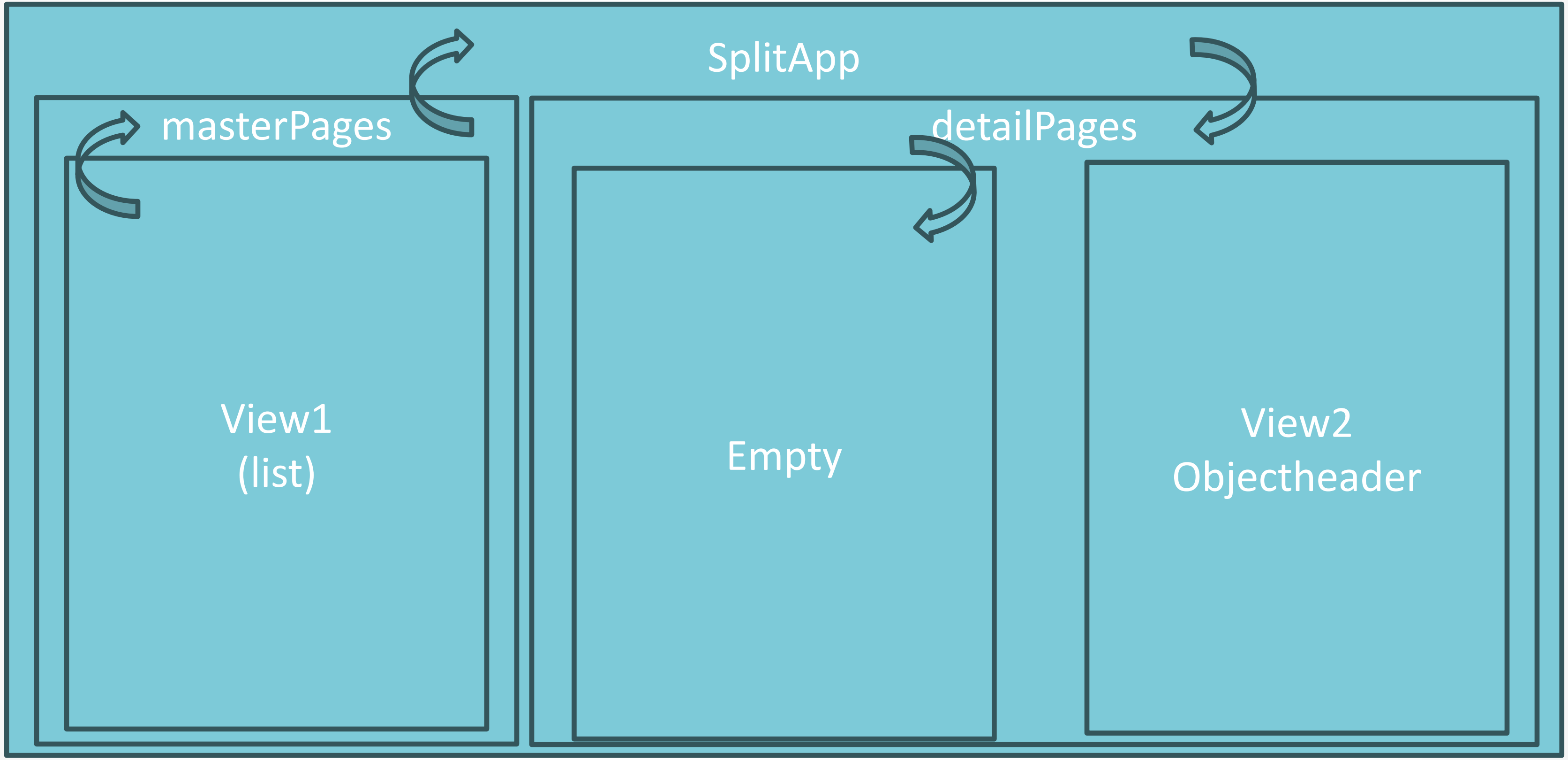
- Enable the multi select mode for list control
- Add a toolbar in the list with a delete button
- On click of delete, remove all the selected items

2. Display the data of selected item on next screen
Hint: Use element binding concept

Worklist Fiori App



SplitApp Container control



Issues with Current Application

- The browser back and forward buttons are not behaving correctly.
- Our application is unable to restore the last active state, if we compare with amazon, ebay, Netflix, the app is not behaving like the one it should behave.
- Also if you observe, every product on amazon provide a **unique end point**, which allows us to restore the state of UI in case of a network failure. This also enable sharing of url easier hence load the same state in your colleagues, family’s system.
- We do not have every product marked with unique end point/hashtag/route/pattern.
- We are creating all our view objects in Component.js right at the beginning when app starts, this will consume memory and pile up view objects even there is no surety that our user will ever navigate to the view. Ideally the views should be created when they really needed by user @runtime on-demand.
- We are writing lots of code to create view objects which is duplicate code. In a project, one developer write view code in Component file and another developer writes it in App.controller, this becomes a situation when someone leave the team and makes handovers very hard to make.
- What the best practice/gold standard to create view objects in Fiori apps.
- We are tightly coupling one view code (view2) with another view’s controller (view1), this will cause high maintenance in the future.
- What is the best way to navigate from one view to another as per standard guidelines.

→ Router

What is Router?

Router is an object in SAP UI5 which is available out-of-box from UIComponent class of SAP UI5. This is the exact class we used to build our Component.js. It is responsible for creating our view objects as well as navigation between the views. So no more need of our parent control to perform navigation. This is the best practice.

- The router would need routing configuration, which we must define in manifest.json file. If we don't maintain this, it will cause dump.
- Get the router object from base class (UIComponent)
- Call the initialize function for router (it will look for routing configuration in manifest)
- The Syntax is like below

```
rootView: {},          --specify the root of our app App.view.xml
routing: {
        config: {}     --the type of views we have, path etc
        routes: [],    --end points like amazon
        target: {}     --view configurations which view to load when
}
```

- Route matched handler is a special method provided by Router which gets triggered every time the route change in the URL. The Route in URL can change because of

❑ When we press back and forward button

❑ When we manually change the root

❑ When we load app first time

❑ When we enter the url shared by someone

❑ When user click the item and router object trigger the route change

Difference between m.Table and ui.table.Table

- Both have columns aggregation but for creating a m.Table we use **items** instead of rows
- A mobile table is a list control so all the properties, aggregations we studied about the list applies to m table
- Mobile table is responsive, it can adapt according to device
- Even we not bind the records (items), still the columns will display in empty table

Exercise:

✔ Create new view (supplier.view – view3)

✔ On click of table item, navigate to this new view

Display the selected supplier data

Hint: between view1 and view2, we need to do same between view2 and view3

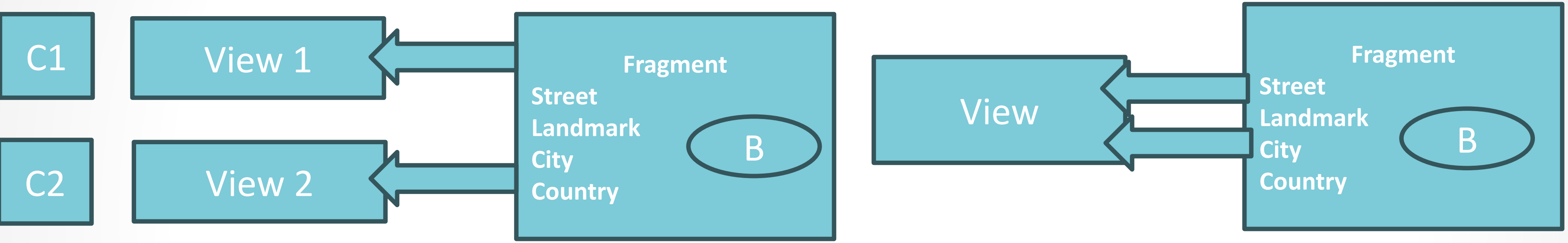
Charts in SAP Fiori

Dimension – dimension is a category based on which we construct our chart, it is typically master data of our application which is a string/char

Measure – measures are the values and they are always numeric

Fragments

Fragments are light weight UI parts which are used to modularize the View. Unlike the views they don't have their own controllers, rather they rely on the controller of the HOST view. We can reuse a fragment multiple times in multiple views.



Fragments are like parasite; they don't cook their own food rather depends on the food cooked by host. We have 2 types of fragment

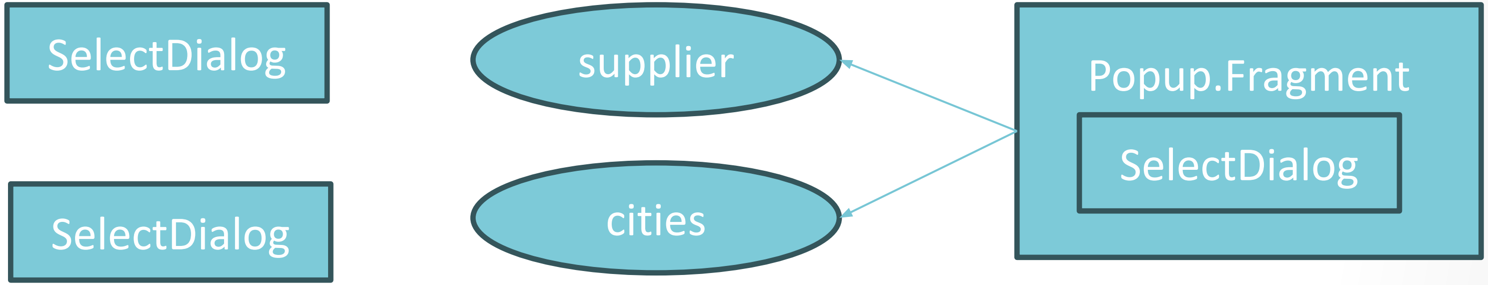
Static – added in the view statically at design time (<core:Fragment>)

Dynamic – added @ runtime using controller code with JS. An object of the fragment needs to be created.

Use case

My manager asked me to implement following requirements for the supplier table

1. Add a toolbar above the table with a filter icon, when we click on the icon, we will display a ***popup*** showing all the list of suppliers.
2. Enable F4 help (value help) for every city field inside the table. On F4, a ***popup*** display the list of cities to choose from and when we select a city, we put the value back on the table cell.



Title

<content>

OData Introduction

OData stands for Open data protocol, it was initially introduced by Oasis and later acquired by Microsoft. It is one of the best way to expose data as a web service on internet. OData is a stateless protocol to expose our ERP data as well also known as REST – Representational State Transfer.

Working with OData Services

A service always starts with a service document, a service document is obtained using the definition of the service and contains entitysets. An entityset represents an End point to perform CURDQ (Create, Update, Read, Delete and Query) An entityset is defined using Entity type. An entity type represent the skeleton of the data which contains the properties like keys, length, data type, max length, decimal places for all the fields. E.g. we can have an entity type for Order which has orderId, soldToParty, amount, curr, disCh, division, SalesOrg and then have OrderSet as entityset for end point to perform operations.

Microsoft provides a FREE odata service for us to learn and experience the Odata.

1. Service document – list of entity sets

https://services.odata.org/northwind/northwind.svc/

2. Service metadata document to display all the entity types inside the service (skeleton of our data)

https://services.odata.org/northwind/northwind.svc/$metadata

3. How to read data for all customers (SELECT * FROM customers)

https://services.odata.org/northwind/northwind.svc/Customers?$format=json

4. Read single customer data (SELECT * FROM customers WHERE customerId = SOMEID)

https://services.odata.org/northwind/northwind.svc/Customers('ANATR')?$format=json

5. Read only few properties to minimize volume of data (SELECT col1, col2 FROM customer)

https://services.odata.org/northwind/northwind.svc/Customers?$format=json&$select=CustomerID,CompanyName,Country

6. Count the total records inside the customer system

https://services.odata.org/northwind/northwind.svc/Customers/$count

OData GET Calls

1. Load data chunk by chunk – Pagination – (SELECT * UP TO n ROWS FROM customers)

https://services.odata.org/northwind/northwind.svc/Customers?$format=json&$top=3&$skip=2

2. Load data with Filter (SELECT * FROM customers WHERE Country = Germany)

https://services.odata.org/northwind/northwind.svc/Customers?$format=json&$filter=Country%20eq%20%27Germany%27

3. Filter based on a pattern search (SELECT * FROM customers WHERE Name LIKE ‘R*’)

https://services.odata.org/northwind/northwind.svc/Customers?$format=json&$filter=substringof(%27RU%27,CompanyName)%20eq%20true

4. Getting data from multiple tables (SELECT * FROM tab1 INNER JOIN tab2 ON joincondition)

In odata services we first create association which define the relationship between entities, its like a foreign key relationship and then we define navigation property using that association so that we can load the data @ Runtime based on the relationship (Navigation property) for dependent data. E.g. load orders of our customer.

https://services.odata.org/northwind/northwind.svc/Customers('ANTON')/Orders?$format=json

https://services.odata.org/northwind/northwind.svc/Customers('ANTON')?$format=json&$expand=Orders

Use $expand to get the dependent data together with the main entity data.

- Install a free tool called POSTMAN tool – https://www.postman.com/downloads/
- Need to have a real SAP System access to practice and build OData services

OData URL Format

http(s)://hostname:port/sap/opu/odata/SERVICENAME/EntitySet?p1=val&p2=val&p3=val

Options for Real SAP System access

1. [FREE] Use your company SAP System

SAP_BASIS – 740 and above

SPRO and check if SAP Gateway is active or not

You must also have S_DEVELOP, S_SERVICE authorization to be able to build programs and services.

2. For those who are working as independent consultant, don't have project, not want to use client system – We provide under one roof all services and server access which is a cost (on-demand)

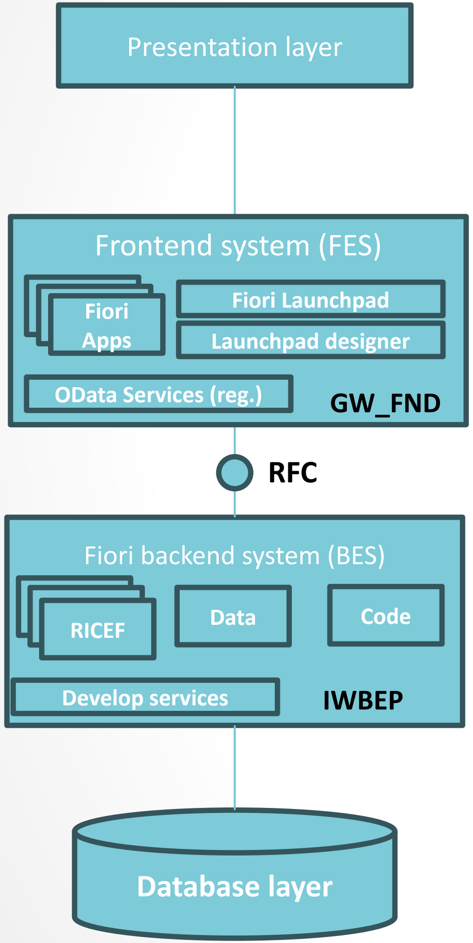
Cost – 2500 INR per user per month / 40 USD per user per month

Process – Hello team, I want to take server access for s/4hana from Anubhav trainings, please tell me the process to make payment in INR currency. Thank you. --- server@anubhavtrainings.com

What to expect?

- Latest S/4HANA on-premise system, fully activated
- We provide all development rights to our users so they don't waste time with system
- Get documentation and videos to know how to connect to our server
- 24X7 uptime for the server, 100 GBPS lease line
- No VPN required from your country to connect to our server
- We provide also all the data pre-configured inside server so you can practice All Anubhav's courses
- We have 2 h of SLA with our working time 9 AM to 9 PM
- It can connect to SAP BAS in BTP and also to VS Code
- You will get GUI Credentials from your local machine, build and deploy odata and connect to BAS and VS Code with Fiori Apps.
- Anubhav will showcase everything in same server to you.
- **If you opt today, we will provide FREE SAPGUI installation to you.**

Fiori Deployment options



SAP Fiori Gateway Central HUB architecture

Innovation w/o disruption suitable for existing ECC or Business suite customers

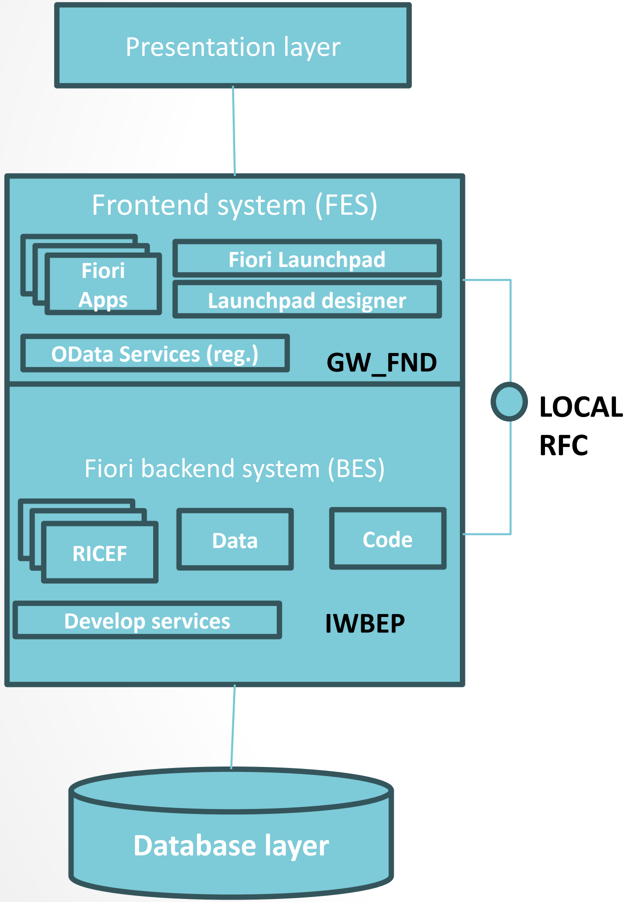
Pros

- Customers do not need to upgrade
- The FES system can work like a switch (HUB) allows us one FES over multiple BES like CRM, SRM
- Double down on security
- If one system is down, it won't affect the other system
- Allows innovation w/o customer worry about upgrading old system

Cons

- Cost goes up
- Manage multiple systems hence multiple teams and hardware
- Confusion
- Double maintenance of users

SAP Fiori Embedded Deployment Architecture



Now we technically have just one hardware and one system which has both backend and frontend components installed and contain a RFC called LOCAL to connect with itself.

Pros

- Save the cost
- No need of redundant data
- No need of double maintenance
- Confusion goes away

Cons

- One single bulky system so need to handle with care
- In case the system is down, everything is down
- Have more memory with more powerful hardware

All SAP S/4HANA projects usage Embedded Deployment only.
https://www.sap.com/documents/2018/02/f0148939-f27c-0010-82c7-eda71af511fa.html

SAP Getway T Code

Anubhav training system is S/4HANA system so no confusion as we have embedded deployment.

SEGW – Gateway builder – to create odata services
/n/IWFND/MAINT_SERVICE – to register the odata services
/n/IWFND/GW_CLIENT – to test our odata services
/n/IWFND/ERROR_LOG – to check the logs of failures
SE11 – DDIC
SE24 – Class Builder
SE37 - Function modules
SE80 – ABAP Workbench
/n/IWFND/CLEAR_CACHE – to clear gateway cache

EPM Data model – Enterprise procurement Model

It is a demo data model provided by SAP in every sap system for learning, training and demo purpose. Using this data model, we can explore and create the data and sample modules which help us learning and building applications and services.

SNWD_PD – products, SNWD_TEXTS – product texts, SNWD_BPA – business partners SNWD_AD – address

Function modules (RFC enabled)

BAPI_EPM*

BAPI_EPM_PRODUCT_GET_LIST – Read all products

BAPI_EPM_PRODUCT_GET_DETAIL – Read single product data

BAPI_EPM_PRODUCT_CREATE – create new product

BAPI_EPM_PRODUCT_CHANGE – update product data

BAPI_EPM_PRODUCT_DELETE – remove from database

Business partners

BAPI_EPM_BP_GET_LIST – Read all business partner

BAPI_EPM_BP_GET_DETAIL – Read single business partner

BAPI_EPM_BP_CREATE – create new business partner

BAPI_EPM_BP_CHANGE – update business partner data

BAPI_EPM_BP_DELETE – remove from database for business partner

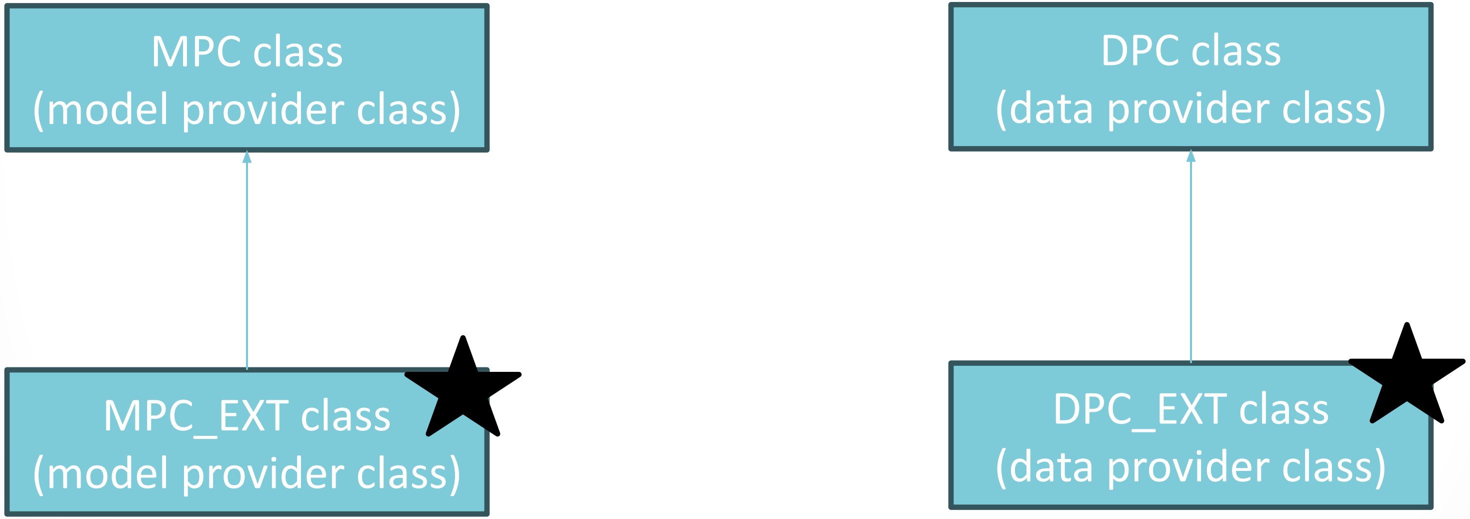
Runtime Artefacts

When we generate the service, multiple classes gets generated automatically. Since there is nothing like a magic, SAP system will create code automatically based on the definition of our service.

MPC – contains all the source code generated by system which serve the metadata of the service.

DPC – contain the code which is responsible to provide ***CURDQ implementation***

How can a developer like us modify the code according to our requirements. SAP provide also the child classes, for MPC they offer MPC_EXT and for DPC a DPC_EXT class is offered. All the custom code which we want, we write in child classes.



For entity 5 functions gets created inside the DPC class.

Title

<content>

Title

<content>

Title

<content>

Title

<content>

Title

<content>

Connection to SAP system which is On-premise

VS code

https://www.youtube.com/watch?v=6jk9G4vaFjQ&t=1931s&pp=ygUac2FwIHN5c3RlbSBjbG91ZCBjb25uZWN0b3I%3D

Steps

Starters:

1. If you using Anubhav training system direction connection is allowed with IP as our system is publically available but if you use your company system which is protected by firewall and usually used with VPN, we need to install and configure cloud connector. Check this video
2. We can add connection in BAS using service center and VS code using Add SAP system options
3. We need to add Advance sap Fiori tooling at project level and at system level – **npm install -g @sap/ux-ui5-tooling npm install @sap/ux-ui5-tooling**
4. Check if tooling is added using **fiori --help**

Main Course:

1. To connect our Fiori app with SAP system we have add a configuration so that the connection can work.
2. We need to add the proxy configuration from the node module in the ui5.yaml file to connect to our sap system check the documentation
3. Next, open the manifest.json file and add the odata service configuration to it as dataSources property
4. We now add a odata model as default model instead of local fruit model. This new default odata model will connect to our service.
5. Lets bind the entityset with the list control. Instead of loading the fruits data we will load the product data in our fiori app.

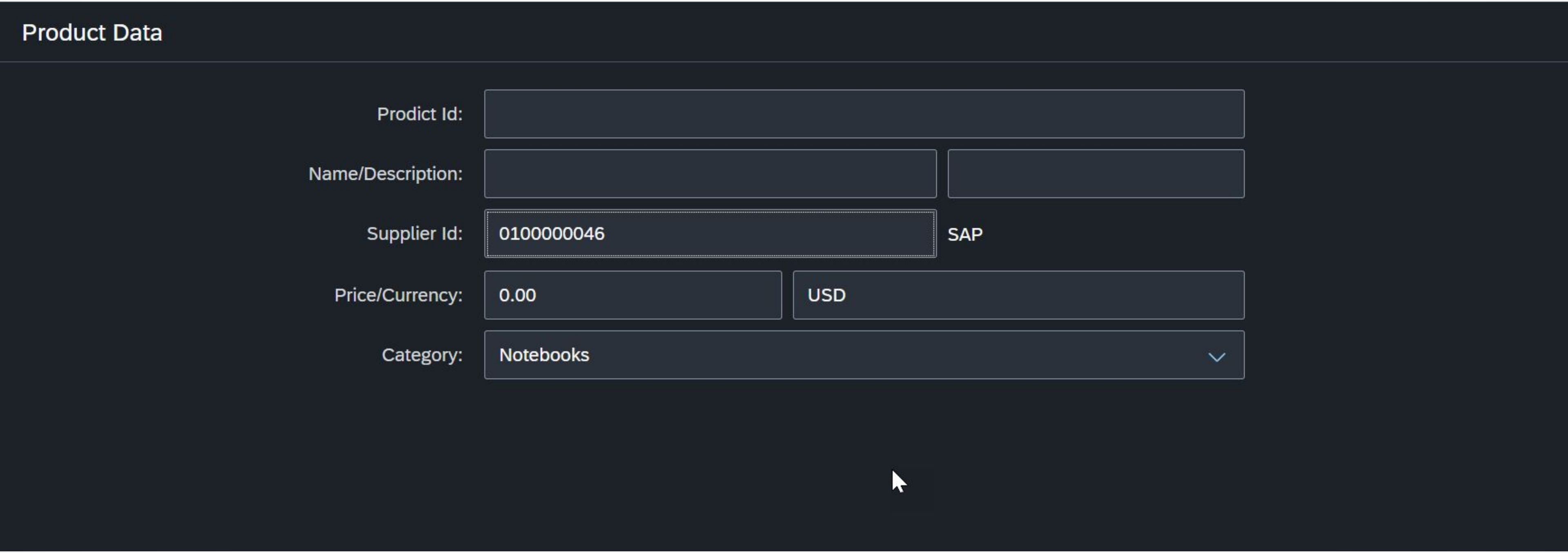
CORS – Cross Origin Resource Sharing

This is a browser security policy which will prevent us making a call to our odata service from our fiori app running in local machine. According to browser security, if you try to load data from a server, inspite running application in browser connected to another domain, the browser will block your call.

http://s4dev.st.com:8021/sap/opu/odata/sap/ZJAN_ODATA_SRV_01/ProductSet
?$format=json&$top=1

Title

<content>



Product Data

Prodict Id:		
Name/Description:		
Supplier Id:	0100000046	SAP
Price/Currency:	0.00	USD
Category:	Notebooks	

Title

<content>

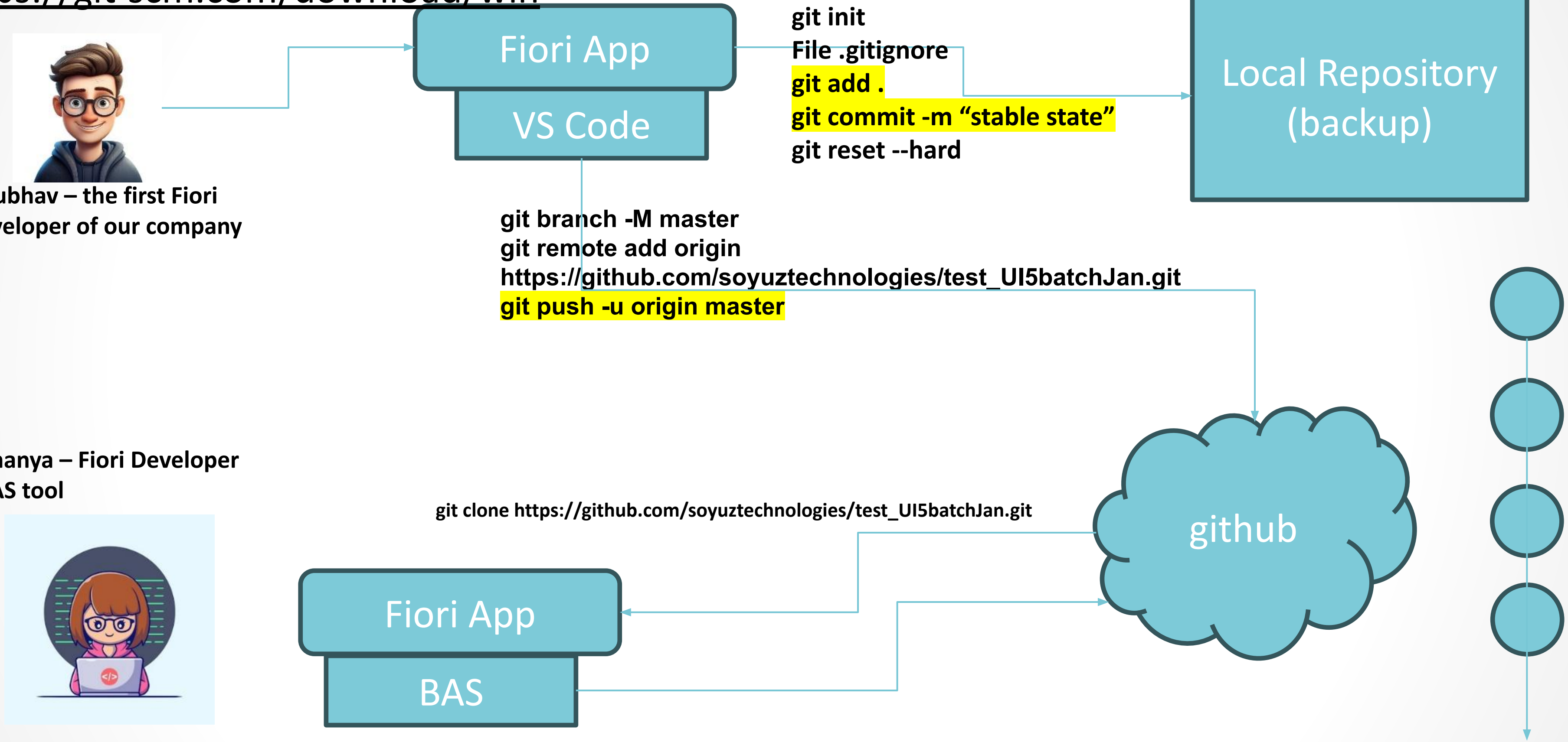
Challenges with our code

- Last night my code was stable, this morning I came to office and made changes to my code. Now I disrupted my application, I pray to god to roll back to last night state. So we need a local version management
- When someone leave the team, how can we make sure that they share the latest code with us.
- When someone on holiday, how can I have always the latest code to work upon.
- Our team is located in India, Denmark, Scotland, USA, Canada and Singapore. How can we share the code with everyone at same time so that all the colleagues working on different timezones and location can sync the changes.
- Until now all the code was shared as ZIP file, it is almost impossible to merge 40 zip files manually to get the final code.
- What is the best practice to share the code of Fiori apps between multiple developers in a company
- When code crash in production, how I know who has made the last change at what time so we can blame them.
- One developer using BAS, another using WebIDE (obsolete) and another using VS Code, how to effectively interchange the source code between these IDEs.

Git and git HUB
Git is a decentralized repository management system which allows manging the source code in our own machines. The laptop of a developer will work like a local code repository where we can have version management of code.

Scenario

https://git-scm.com/download/win



Anubhav – the first Fiori developer of our company

```
git init
File .gitignore
git add .
git commit -m “stable state”
git reset --hard
```

```
git branch -M master
git remote add origin https://github.com/soyuztechnologies/test_UI5batchJan.git
git push -u origin master
```

Ananya – Fiori Developer BAS tool

```
git clone https://github.com/soyuztechnologies/test_UI5batchJan.git
```

Lifecycle of a Fiori App

Title

<content>

ZBC_FEB

ZBCG_FEB

Fiori Launchpad

Fiori launchpad in SAP S/4HANA system is a single entry point to all Fiori apps in secure way, it offers

- Single access point for the apps in form of tiles
- Ensure security
- Provide default login experience so each Fiori developer do not need to provide own login screen
- Harmonize user experience
- Allows personalization – change theme, coloes, settigs, default values etc
- Every SAP user is trained on Fiori launchpad - /n/UI2/FLP

Title

<content>

Title

https://www.youtube.com/watch?v=hC_EWlF3MZE&pp=ygULZGVlcCBlbnRpdHk%3D

FAQs

1. What will be the duration of the training, and what pre-requisite?
The duration of the course is 40 hours, and there is no prerequisite other than basic programming skills.

2. The timing does not suit me so can I go for video mode?
Yes, you can opt for video mode training for our most recent batch and ask your question Over the mail.
This option also applies. Those who want fast-track learning and do not want to be with the existing batch.

3. What tool do I need to learn and where do I get it?
We will build all the apps on Business Application Studio and VS Code tool.
In our next class, we will see how to install the tool in your system or get cloud access.
VS Code is absolutely free for a single developer.

4. What about the server to practice Odata?
We need the Odata server only in the last 10 sessions for which we will guide you in session no. 29,
where, how to get the OData Server, and also maybe you can use co.
Server all the required components will be explained.

FAQs

5. Will you cover real-time scenarios?

Yes, we will learn end to end process of Developing the Fiori apps and finally connecting with real SAP data, deploy and deliver apps to the Production system.

6. I am new to SAP UI5, can I learn?

Yes, this course is designed in a way that a fresher can learn because we will cover all basics like HTML, CSS, JavaScript and JQuery.

7. If I miss a class how to refer?

Each class will be recorded, and you can watch is once done. So, even if you miss or do not miss the class, you can also watch recordings.

8. Process to enroll for this course?

Please mail us at **contact@anubhavtrainings.com**

www.anubhavtrainings.com

www.anubhavtrainings.com +91 8448454549 contact@anubhavtrainings.com

THANK YOU

“Teachers can open the door, but you must enter it yourself.” —Chinese proverb

www.anubhavtrainings.com