SAP UI5 - MVC Concept

Model-View-Controller (MVC) concept is used in SAP UI5 development to keep the application data separate from the user interactions. This allows you to develop the web applications and make changes to the applications independently.

Model-View-Controller plays a different role in UI development −

* The **Model** is responsible for managing the application data in the database/backend.
* The **View** is responsible for defining the user interface to users. When a user sends a requests from his device, the view is responsible for data view as per the request submitted.
* The **Controller** is used to control the data and view events as per user interaction by updating the view and model.

You can define Model-View-Controller concept in SAPUI5 with the following features −

**Model**

* Model acts as a bridge between the view and the application data.
* Model is used to get the request from the view and respond as per the user’s input.
* Model doesn’t depend on classes.

**View**

* View is responsible to manage information display to the users.
* Views are based on Model.

**Controller**

* Controller is responsible for taking the input given by devices and communicates to model/view and to trigger correct action.
* Controllers are based on model.

SAP UI5 offers Views and Controllers in the form of single files −

* sap.ui.core.mvc.XMLView
* sap.ui.core.mvc.JSView
* sap.ui.core.mvc.Controller
* sap.ui.core.mvc.JSONView

**JSON Model**

* JSON model is a client-side model and is used for small data sets.
* JSON model supports two-way binding. Data binding concept is mentioned in the latter half of this tutorial.
* JSON model can be used to bind controls to JavaScript object data.

**XML Model**

* XML model can be used to bind controls to XML data.
* XML is also a client side model and hence is used only for small data sets.
* XML model doesn’t provide any mechanism for server-based paging or loading of deltas.
* XML model also supports two-way data binding.

**SAPUI5 Control Development Guidelines**

* Yesterday I told when I ever we develop any UI5 application we are going reuse standard template code for Controls
* But Reuse is Good. But carefully compare how many features of the reused control are needed in Our UI5 application. It will be impact on performance.
* Whenever developers developing SAPUI5 controls should follow the guidelines i.e APIs, behavior, and themes/CSS.

**API**

* Get the right API in first time
* Control names start with an **uppercase** letter **for Exp:** CamelCase.
* Property, event, aggregation, association, method, and parameter names start with a **lowercase** letter **for Exp:** camelCase.
* Properties, associations, and aggregations should be preferred over API methods due to data binding support and easier usage in XMLViews.

**Behavior**

For behavior-related development, the following guidelines apply:

* Do not use hardcoded IDs. When creating internal sub controls, their ID should be prefixed with **this.getId() + "-".**
* Use the SAPUI5 event handling methods when available instead of jQuery.bind()/.on(). When you use jQuery.bind() or jQuery.on(), always unbind them again**, for example** in onBeforeRendering() and in exit() and rebind after rendering.
* Keep in mind that a control can be used multiple times in a page.
* If an action takes a longer period of time, visualize this, **for example** by using a **BusyIndicator**.

**We discussed yesterday using Eclipse and WEBIDE we are going to develop the UI5 application?**

**What is difference between Eclipse and WEBIDE**

* Both are IDEs, it's your choice which IDE to use for development.
* The UI5 tools created for eclipse are now longer supported and the recommendation is to use Web IDE.
* Another difference is that SAP Web IDE is not free. Whereas Eclipse is free (Open Source)
* Whenever we created any UI5 application in WEBIDE. Some code will be generated automatically. Where as in Eclipse we need to write End-to-End Code.
* Whenever we create application WEBIDE. Its will ask service name(SAP NWGS). If you give proper service it will bind to UI elements and data will be populated. Later whatever you can change accordingly. Whereas Eclipse we need to write End-to-End Code for get the Model Service.