| **Topic** | **Elementary** | **Basic** | **Advanced** | **Expert** |
| --- | --- | --- | --- | --- |
| **SAPUI5** | | | | |
| **Application structure** | 1. Know the standard **folder structure** of the SAPUI5 app. 2. Know the aspects of **Component** & **Manifest**:    1. Component definition.    2. Manifest structure. 3. Be aware of UI5 framework versioning system: major/minor/patch versions | 1. Know the different ways of **app bootstrapping**:    1. Standalone app.    2. flpSandbox app.    3. flpMockServer app.    4. unit/OPA test run app.    5. CDN/local sdk references. 2. Know the **configuration options** and **URL parameters** of the app. | 1. Know how the app gets **initialized**. Be able to explain each step in the init **flow**. | \_empty\_ |
| **Data Binding** | 1. Understand what is a **Data Binding**, it’s purpose and why it is so important. 2. Know the **Property Binding** mechanism. 3. Know the **Aggregation Binding** mechanism:    1. Know the **Filtering**, **Sorting**, **Grouping** capabilities:       1. Know the various ways of Filter instantiation and configuration       2. Be able to combine **multiple filters** in one **composition, filters nesting** 4. Know the **Element/Object binding** mechanism. 5. Know what is a **Binding Context.** 6. Know what is a **Formatter** function and when to use it. 7. Know what is a **DataType**, its purpose and how & when to use it:    1. Be aware of all the **Simple Data Types**       1. Know the **syntax while binding**    2. Be aware of the **Formatter Classes** and when to use them:       1. Date Format       2. Number Format       3. File Size Format       4. Unit Formatting       5. Currency Formatting 8. Know the 3 available **Binding Modes.** Be able to explain each of them, their purpose and when to use:    1. OneWay.    2. TwoWay.    3. OneTime. 9. Know the **data binding syntax**:    1. Binding path    2. Composite binding    3. Know the **expression binding** technique and be able to decide when to use the formatter function instead.    4. i18n binding, sap.base.strings.formatMessage 10. Know the **standard “binding” attribute** of a control. | 1. Understand the **Factory Function** concept. 2. Know the possibility of a **custom filtering function** definition and when it might be useful. 3. Know how to use the **grouper function.** 4. Know how to create a **Custom Data Type**. Understand the reason why it might be required. 5. Know the **binding techniques** of a **MetaModel**. | 1. Understand **templateShareable** concept. 2. Understand the **lifecycle of binding templates.** 3. Understand the **extended change detection** mechanism. | 1. Know how the **data binding** works on the level of **source code**. |
| **Controls** | 1. Understand the aspects of **controls usage in XML** format:    1. What is a **namespace**? Declare / import the libs through the namespaces.    2. Ability to use **default namespace** & **named namespace**. Prefixing controls in the view.    3. What is **XML Node**? Its definition syntax.    4. What are the **attributes** & **events**. Their purpose & syntax.    5. What is **aggregation**? Its purpose & syntax.    6. **Default aggregations** feature.    7. What is **association**? Its purpose & syntax.    8. What is **dependent**. Its purpose & syntax.    9. **Icons** definition syntax.    10. “**class**” attribute specifics, one time set-up. 2. Know the predefined CSS classes. 3. Know how to **instantiate control via JS**. 4. Be aware of guidelines from documentation for [control development](https://openui5.netweaver.ondemand.com/topic/4549da61e2d949d6a3d20ad8a9d17a6f) and [theming/CSS](https://openui5.netweaver.ondemand.com/topic/5e08ff90b7434990bcb459513d8c52c4) | 1. Know what is **Custom Data** and when to use it: 2. Syntax while binding. 3. Know how to get it in the controller. 4. Know how to write it to DOM and when it might be useful. 5. Know the **Field Groups concept**, their purpose & common use cases. | \_empty\_ | 1. Control’s functionality expansion via **Dependents** trick. |
| **Custom controls** | 1. Be aware of the Custom Controls creation possibility and why it might be required. | 1. Know the various **types of custom controls**. Ability to decide in which case what type to use. 2. Know the composite control aspects:    1. Syntax.    2. Know **Standard Composite** syntax & purpose..    3. Know **XML Composite** syntax & purpose. 3. Know the **control extension** approach (via inheritance from a standard). 4. Know how to create a **true custom control**:    1. Understand **control metadata**:       1. **Properties** & settings       2. **Aggregations** & settings       3. **Associations** & settings       4. **Events** & settings    2. Ability to implement own **custom renderer**.    3. Understand the **Render Manager** mechanism.    4. Know the approach of **reusing renderer** from existing control.    5. Know the approach of **extending renderer** from existing control.    6. Know how to declare **custom CSS** for the new control, guidelines, naming, placement.    7. Know the **i18n** for control aspects. | 1. Know the **accessibility aspects** to stick to. | \_empty\_ |
| **Smart Controls** | 1. Understand **Smart Table** concepts:    1. Annotations:       1. Line Item.       2. Presentation variant..       3. Data field.       4. Sort / Filter restrictions.       5. Semantic Object.       6. Criticality.       7. Text Arrangement.    2. Know how to declare **сustom columns**.    3. **Personalization** (column selection, grouping, sorting, filtering).    4. **Reference to Smart Filter Bar** awareness    5. **Variant Management**.    6. Know **onBeforeRebind** event main use cases. 2. Understand **Smart Filter Bar** concepts:    1. Annotations:       1. FilterRestrictions.       2. FieldGroup.       3. ValueList.       4. Hidden.    2. Know how to declare **сustom filter items**.    3. “**Adapt Filters**” feature awareness.    4. Variant Management..    5. “Basic Search” feature awareness. 3. Understand **Smart Field** concepts:    1. We aware of **annotations.**    2. Know available **field types.**    3. Know the **Value Help** concept, its configuration.    4. Know the **field control** concept, ability to describe the main use cases. | 1. Understand **Smart Form** concepts:    1. Annotations. 2. Understand **Dynamic Field Control** concept. Understand **UI.Hidden** annotation. 3. Know the **Side Effects** concept of Smart Field control. | 1. Understand **Smart Chart** concepts    1. Annotations 2. Understand **Smart Link** concepts:    1. Annotations.    2. **Semantic Object Controller.**    3. Know the **intent based navigation** concept. | \_empty\_ |
| **MVC** | 1. Know the various types of **Views**:    1. XML view syntax.    2. JS View syntax.    3. Know the **Fragments** concept, its purpose. 2. Know the **Controllers** aspects:    1. Definition syntax.    2. Know how the controller gets **attached to a view**.    3. Know the **lifecycle methods**.    4. Understand the **inheritance** concept. Know the syntax, **super methods** call.    5. Know the **base controller** best practice.    6. Know the **dialogs reuse approach** via custom controller. 3. Know the **Model** aspects:    1. Know the **available models’** basics:       1. JSON, OData, Resource.       2. Know the purpose of each model.    2. Know **where the model can be declared**:       1. In manifest       2. In controller       3. In component    3. Know the concept of **named** and **default models**. Syntax of referencing to named/default models, syntax for the relative binding in case of named model. | 1. Understand how the **id’s for the controls** are generated in a view. 2. Adding controls as **dependent**. 3. Be able to **compare JS & XML** view approaches. Ability to identify the use cases when each of them could be useful or harmful. 4. Understand the basic concept of **models propagation** through the views and controls tree. 5. Know the **main model events**, derived from sap.ui.model.Model:    1. propertyChange.    2. requestCompleted.    3. requestFailed.    4. requestSend.    5. updateBindings. | 1. Understand the mechanism of **views nesting.** 2. Understand the mechanism of **view destruction**. 3. Understand the mechanism of **controllers destruction**. | \_empty\_ |
| **Modules** | 1. Know the **syntax of a module definition** in SAPUI5. 2. Know how the module can be consumed.    1. sap.ui.define at the beginning of a file    2. sap.ui.require in the custom place of a file | 1. Understand the **AMD concept**. 2. Know what is a **\*-dbg.js version** of a module. | 1. Know how the **circular references** are resolved. 2. Understand what is a **SAPUI5 library**, its definition in the consuming app, its syntax and how it’s downloaded during the app initialization. 3. Usage of **core:require** in XML view. | \_empty\_ |
| **Routing** | 1. Understand the purpose of **# sign** in the route and how the browser reacts on its change. 2. Know the router configuration in manifest:    1. **“config”** property. Know that it’s a common config for all the targets.    2. “**routes**” property:       1. **Mandatory parameters** in route pattern syntax and usage.       2. **Optional parameters** in route pattern syntax and usage.       3. Query parameters.    3. “**targets**” property:       1. Understand what is **controlId** and **controlAggregation** properties.       2. Know the different types of **transitions**/animations 3. Know how to organize the **manual routing** in the app's logic. “**navTo”** method syntax and usage. 4. Know the main **Router events** and their use cases:    1. routePatternMatched.    2. routeMatched.    3. patternMatched & match (**on Route**). 5. Understand how the **nav back** & **history** work.. 6. Know how to catch **invalid hashes**, not found page. | 1. Understand how **routing** works in general **SPA**. 2. Understand what happens after the **initialize** method is called on the Router instance. | 1. Understand how to implement the **bookmarking** possibility of: 2. Search. 3. Sorting. 4. Dialogs. 5. Know how to implement **lazy loading** of some part of a screen. | 1. Know the **multiple targets** in a single route mechanism. 2. Know the **components nesting through the routing** approach. |
| **ODataModel** | 1. Know how the **data** are **stored** in the model **internally**. Understand what is a **key of a property** and which parts it consists of. Know how to construct the key manually to access the OData property in a model. 2. Know how to perform **CRUD** operations. 3. Understand **function import,** know the ways of calling it. 4. Know the **possible parameters** that can be used in **aggregation binding specifically with ODataModel**. 5. Know OData query options 6. Know how to **filter/sort** a collection (using server or local data). 7. Understand the **CountMode** property and its each value. | 1. Know what is a **GroupId** and **deferred groups**. 2. **Optimizing** dependent bindings. 3. Be aware of the **main events**:    1. Annotations loading events.    2. Metadata loading events.    3. Batch loading events. 4. Be aware of **metadataLoaded promise**. 5. Know what is **MetaModel**, its purpose, main use cases, dynamic syntax. 6. Understand the binding path syntax, passing **extra binding parameters**. 7. Understand when the data binding **produces a BE request** and when it doesn’t produce it. 8. Understand the **pending changes queue** mechanism. 9. Understand a “**search focus**” and **fuzziness** concept. 10. Understand the **XSRF** Token. | 1. Know how to set **custom HTTP headers** and the main use cases. 2. Know how the **concurrency control** is managed. Understand **ETags** concept. 3. Understand the **batch processing** mechanism. 4. Understand what is a **change set**. 5. Know what is an **OperationMode** and where it can be used. | \_empty\_ |
| **OData Metadata** | 1. Know what an **EntityType** is. Its features:    1. Keys.    2. Properties.    3. NavigationProperties / associations. 2. Know what an **EntitySet** is. 3. Know what a **Function Import** is**.** 4. Know the main **data types**:    1. Decimal.    2. DateTime.    3. Int.    4. String. | 1. Know what a **ComplexType** is. 2. Know the **non-standard attributes** that were **defined by SAP**. | \_empty\_ | \_empty\_ |
| **Localization** | 1. Know the aspects of the **ResourceModel** (binding type, where it takes the data from, how multilang is supported). 2. Know how to use the i18n string with **placeholders** | 1. Know the **i18n bundle enhancing** mechanism. 2. Be aware of the purpose of **text classification**, **text types**. | 1. Know the **translation flow in SAP** through the translation system. 2. Know what is “**Terminologies**” concept | \_empty\_ |
| **Layouting & Responsiveness** | 1. Understand the aspects and how the **main layouting controls** work:    1. ObjectPageLayout.    2. Dynamic Page.    3. TabBar.    4. Form.    5. Page.    6. FlexBox, HBox, VBox.    7. Horizontal/Vertical layouts.    8. NavContainer.    9. Split App. 2. Understand the **Grid System** mechanism. Know which controls possess the grid system. 3. Know the **standard CSS classes**. 4. Responsiveness:    1. Responsive CSS classes.    2. Responsive table, pop-in.    3. Form grid.    4. Grid control.    5. Text wrapping.    6. Floating footer. | 1. Understand the aspects and how the **main layouting controls** work:    1. FlexibleColumnLayout.    2. Semantic Page (.f, .m).    3. Grid.    4. Tnt. 2. Responsiveness:    1. FlexBox, HBox, VBox responsive setup.    2. Toolbar elements collapsing.    3. Flexible column layout setup.    4. Dynamic Page sticky header. | \_empty\_ | \_empty\_ |
| **Testing** | 1. Be aware of **why** the **testing** is so **important**. 2. Be aware of the **testing options** in SAPUI5:    1. Unit tests (qUnit).    2. Integration tests (OPA5). 3. Be aware of **TDD/BDD** approaches. | 1. Know how to organize the **unit testing** via qUnit:    1. Be able to set up a Test environment.    2. Know the unit test guidelines (from UI5 documentation).    3. Know the syntax of qUnit library, main assertions. 2. Know how to organize the **integration testing** via **OPA**:    1. Be able to set up a **test environment**.    2. Know how to find a needed control in the tested app to manipulate with.    3. Understand how the **navigation testing** is performed.    4. Know how to **test a user’s input**.    5. Know how to **test a user's interaction**. 3. Know how to deal with **mockserver**:    1. Know the **configuration aspects**, metadata, annotations retrieval.    2. Know the **fake data**: JSONs & automatically generated data on the fly.    3. Know how to implement the **custom CRUD middlewares** / **hooks**.    4. Know how to **handle a Function Import**.    5. Be able to **handle custom URL** parameters.    6. Be aware of the possibility of dealing with **multiple mockservers**. 4. Be aware of **SAP’s standard requirements** regarding **test coverage**. | 1. Know the **test doubles**:    1. Mocks    2. Stubs    3. Spies 2. Know how to implement the **BDD** approach with **Gherkin**. | 1. Know the **UI Veri5** framework. 2. Know the other SAP-related test systems:    1. SUPA.    2. START. |
| **Security** | 1. Be aware of the **standard browser security vulnerabilities**:    1. Cross-Site Scripting.    2. Clickjacking. 2. Know the **Server Security aspects**:    1. Cross-Origin Resource Sharing. | 1. Be aware of **secure programming aspects**:    1. Input validation.    2. Output encoding.    3. User Management / Authentication.    4. Local storage. 2. Be familiar with **Checkmarx** tool. | 1. Know the aspects of **Transport Security**:    1. Data encryption.    2. Session Security. 2. Know the attention points when **Third-Party libraries are** used. 3. Be aware of **URL Whitelist** filtering. 4. Be aware of **frame options**. 5. Be aware of **Content Security Policy**. | \_empty\_ |
| **Accessibility** | 1. Understand why accessibility is important with regards to UI/UX in enterprise companies. Ability to give some examples of the real use cases where accessibility matters. | 1. Understand what is **ARIA**? 2. Understand the purpose of **Labeling & Tooltips**. 3. Understand how the **keyboard handling** is implemented and how to customize it (allowing the user to operate with the app using keyboard-only). 4. Know the **general recommendations** from **UI5 docu**. | 1. Know the **UI5 Controls for ARIA**:    1. landmarks.    2. accessibleRole attribute. 2. Be aware of **Colors & Theming** (HCB, HCW), CSS variables in the context of the current theme. 3. Understand how the **screen reader support** is handled. 4. Understand the **Text Size** and **Fonts** aspects. 5. Understand how the **fast navigation** is handled (fast-navgroup, F6). 6. Understand the **RTL support** capabilities. | \_empty\_ |