Awesome UI5 Features and why we use them in Fiori elements

Thorsten Erlewein, Bernhard Siewert, SAP

June 28, 2019

UI5con

learn.explore.connect.

Awesome UI5 Features

and why we use them in Fiori elements

Fiori elements is a metadata driven UI framework made with UI5. It provides consistent user interfaces for Fiori applications based on OData services.

Within the Fiori elements framework we use awesome UI5 features, that we want to introduce in this repository.

- Component Based Routing
- Transformation Using XML Preprocessor
- XML View Cache
- Declarative Event Handlers
- Controller Extensions

Disclaimer: Obviously, we use a lot more awesome UI5 features in Fiori elements than the ones mentioned in this session. We couldn't do what we do without the powerful OData support in UI5. However the features presented ***here*** *can be used with or without OData as they are model agnostic.*

Component Based Routing

Component Based Routing

Fiori elements transforms annotated OData metadata into UI5 XML views. That transformation needs to happen before the view is used. So we wrap those dynamic views inside of UI5 Components to be able to trigger the transformation when needed.

In addition UI5 Components have a built-in dependency management using the *manifest.json*. This enables an asynchronous library and component preload. Views don't offer a declaration of dependencies.

To be able to leverage such reusable components UI5 has introduced support for specifying components in the routing section of the *manifest.json*

see UI5 documentation

Transformation Using **XML Preprocessor**

Transformation Using XML Preprocessor

OData services provide access to data of many business objects with different structure. Fiori elements transforms these objects and their structures, plus semantic annotations on top into the appropriate controls and data bindings.

UI5 provides the XML preprocessor that allows to preprocess an XML view before it is turns into real UI5 controls. This method is called XML templating.

see UI5 documentation

XML View Cache

XML View Cache

As the definition of business objects and their structures rarely changes, it is not necessary to do the transformation (the XML templating) all the time.

Fiori elements leverages the XML View Cache of UI5 in order to ensure that the transformation is only done when there is a reason to do so.

see UI5 documentation

Declarative Event Handlers

Declarative Event Handlers

By now you most likely understand that Fiori elements framework developers think *declarative* meaning UI5 XML.

However when it comes to event handlers, before version 1.56 one always needed a controller function with a fixed API `myHandler: function(oEvent) {...}` and that function typically needs to know a lot about the view and the data model.

In Fiori elements we have templates that create the XML dynamically for the individual services. But we don't dynamically create controller code or reusable functions based on JavaScript. Instead we use the advanced syntax that allows us to pass the **right** data to a few generic functions. So rather than `press=".myHandler"` we can now create something like `press="some.reuse.fn(${someProperty})` during templating.

see UI5 documentation

Controller Extensions

Controller Extensions

SAP software of course comes with enterprise qualities, with extensibility being the one which is maybe the most appreciated by our customers. So when it comes to introducing reusable functions for event handlers, formatters and more, Fiori elements needs to provide means to extend these functions by customers.

UI5 has introduced controller extensions as a great way to provide dedicated hooks in reusable functions.

see UI5 documentation

Thank you.

Contact information:

Thorsten Erlewein
Development Architect Ux Engineering
thorsten.erlewein@sap.com

Bernhard Siewert
Development Architect Ux Engineering
bernhard.siewert@sap.com

UI5con
learn.explore.connect.