EB

**Übungen**

**zum Seminar SAPUI5 und Fiori  
  
Tag 05 – Übungen**

### Zusatzaufgabe 1a – Einfügen von Daten (mit verschachtelten Datensätzen) per Create\_deep\_entity mittels OData

Für diese Aufgabe in unserem SFlight Szenario benötigen wir Daten von 2 verknüpften Entitäten. Wir werden dafür die DB Tabelle Scarr nutzen und sie mittels einer 1..n Verknüpfung mit der DB Tabelle Spfli verbinden. Dadurch bilden wir für jede Luftfahrtgesellschaft bis zu n geplante Flugrouten ab. Ein Flugplan beschreibt u.a. den Start- und Zielort, sowie die Start- und Landezeit eines Fluges.

Gehen Sie also ins SAP Backend und mittels der Transaktion SEGW zu ihrem in den vorigen Übungen erstellten OData Service (ZZ\_SFLIGHT).

Fügen Sie eine Entität Scarr per Import aus dem ABAP Dictionary aus der DB Tabelle Scarr hinzu. Generieren Sie auch das EntitySet dazu.

Fügen Sie eine Entität Spfli per Import aus dem ABAP Dictionary der Tabelle Spfli hinzu. Generieren Sie auch das EntitySet dazu.

Legen Sie in der Entity Scarr eine Assoziation HeaderToItem an. Die erste Entität ist Scarr mit der Kardinalität 1 und einem zusätzlich anzulegenden Navigationsattribut HeaderToItem. Die Abh. Entität ist Spfli mit der Kardinalität 0..n. Verknüpfen Sie die Schlüssel Mandt und Carrid aus Scarr mit den entsprechenden Feldern in Spfli.

Die Assoziationsmenge kann HeaderToItemSet heißen.

Überprüfen Sie Assoziation, Navigationsattribut und Assoziationsmenge und aktivieren Sie den Service.

Jetzt sind noch zwei Änderungen an der Modelstruktur in der ModelproviderKlasse \*MPC\_EXT vorzunehmen. Der Typ für HeaderToItem ist in der Klasse zu definieren, damit aus Methoden auf ihn zugegriffen werden kann.

Danach die Methode „define“ redefinieren, um den Typ für HeaderToItem an die Scarr Entität zu binden.

Im nächsten Schritt in die DataProvider Klasse \*DPC\_EXT wechseln und dort die Methode /iwbep/if\_mgw\_appl\_srv\_runtime~create\_deep\_entity redefinieren.

Hier implementieren wir jetzt das Einfügen der später aus dem UI5 Frontend übermittelten Daten in die DB Tabellen Scarr und Spfli.

Lesen Sie die Kopfdaten für Scarr ein, schreiben Sie die Daten ggf. in die Tabelle Scarr und und fügen Sie den Inhalt der HeaderToItem Tabelle in die Tabelle Spfli ein (per Insert oder Modify).

Testen Sie per SAP Gateway Client (Transaktion /IWFND/GW\_CLIENT) ob das Einfügen von Datensätzen per Deep Entity funktioniert. Schicken Sie dazu einen POST Request an das EntitySet ScarrSet, welches die Daten per JSON übergibt.

### Zusatzaufgabe 1b – Erstellen einer UI5 App, die verschachtelte Daten per OData einfügt.Trans

Nutzen Sie die UI5 App aus der letzten Übung und entfernen Sie den Content im MainView.view.xml.

Fügen Sie dort einen Wizard ein. (Siehe SAPUI5 Doku z,B. hier:  
https://ui5.sap.com/1.71.61/#/entity/sap.m.Wizard/sample/sap.m.sample.Wizard),

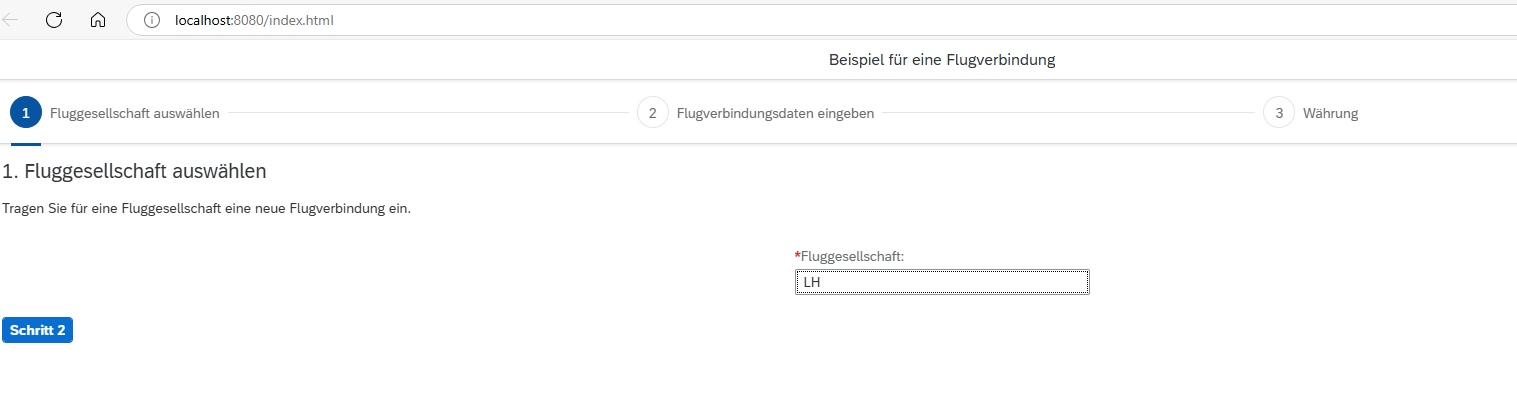
Der Wizard soll 3 Schritte enthalten.

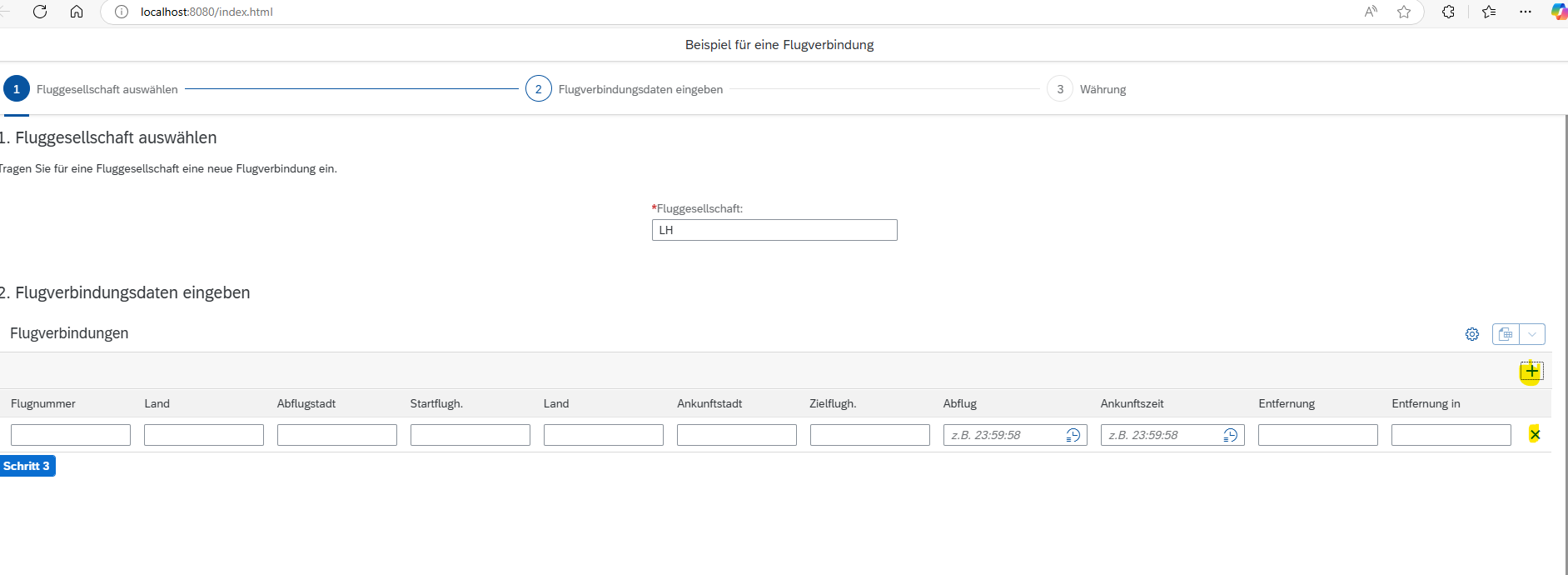
Im ersten Schritt ein Eingabefeld für eine Auswahl einer Luftfahrtgesellschaft. Im zweiten Schritt eine Tabelle (SmartTable) in die leere Zeilen eingefügt werden können (per + Button). Die Zeilen sollen Eingabefelder zur Eingabe der Daten für die Entity Spfli enthalten.Die SmartTable sollte an SpfliSet gebunden werden.

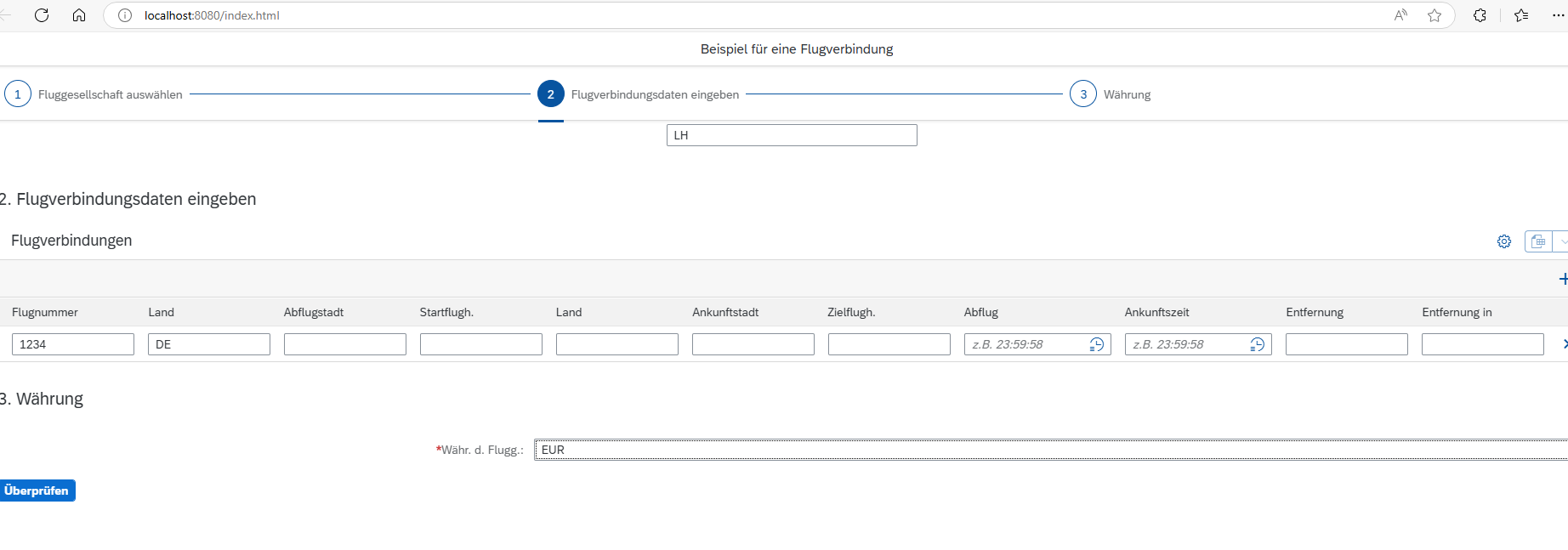
D.h. es soll für jede eingefügte Zeile mit ODataModel.createEntry eine Eintrag für SpfliSet erzeugt werden.

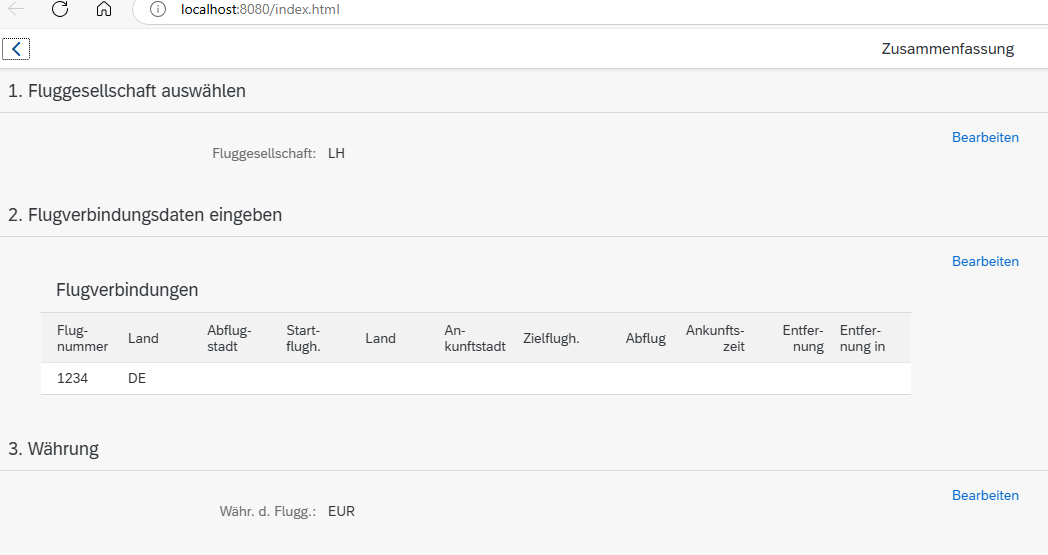
Im dritten Schritt soll man noch eine Währung auswählen können.

Nach dem dritten Schritt geht man auf eine Übersichtsseite. Dort sind nochmal alle eingegeben Daten in Textform aufgelistet und man kann per Speichern Button die Daten an den Odata Service schicken. Mit dem Cancel Button kann man abbrechen, d.h. einen Schritt zurücknavigieren.









Hinweis: Nutzen Sie im Controller des Views die onAfterRendering Funktion, um dort einen neuen Eintrag im ODataModel mit createEntry zu erzeugen und binden Sie den Context den sie von createEntry bekommen dann dort an den View um das Binding zwischen Model und View für denm neuen Datensatz zu „aktivieren“. In der onInit Funktion funktioniert es nicht wie

gewünscht, weil dort u.U. das ODataModel noch nicht komplett (Metadaten, Annotation) geladen ist.

Für die Tabelle nutzen Sie eine SmartTable und setzen deren Property: editable auf “true“, dadurch lassen sich die Zeilen bearbeiten.

Im beforeRebindTable Event der SmartTable müsste dann noch so eingeschränkt (per Filter) werden, dass die Tabelle nicht Datensätze aus dem Backend lädt, alternative das enableAutoBinding der SmartTable entfernen und das Binding selber setzen.

Auf der Übersichtsseite dann beim Speichern Button einen Event hinzufügen der die Daten aus dem Odata Model Datensatz der an ScarrSet gebunden wurde ausliest und die Datensätze die an Spfli gebunden wurden.

Daraus dann ein geschachteltes Objekt erzeugen. Dieses dann per oDataModel.create ans Backend schicken undd im Erfolgfall und Fehlerfall reagieren.