# Modellierung Entität

entity Veranstaltung : cuid {

 name : String;

 ort : String;

 postleitzahl : String @assert.format : ‘[0-9]{5}?$‘ ;

}

entity Besucher : cuid, managed {

 name : String;

 age : Integer;

}

# Bereitstellung Service

# Deployment in der Cloud

service veranstaltungen {

 entity Veranstaltungen as projection on ns.Veranstaltungen;

 entity Besucher as projection on ns.Besucher;

}

## Fiori Elements App auf Basis von CAP hinzufügen

## Hinzufügen von app/package.json

## Hinzufügen von app/ui5.yaml

{

 "name": "fiori",

 "version": "0.0.1",

 "devDependencies": {

 "@sapui5/ts-types": "1.71.x",

 "@ui5/cli": "2.2.6",

 "@sap/ui5-builder-webide-extension": "1.0.x",

 "bestzip": "2.1.4",

 "rimraf": "3.0.2"

 },

 "scripts": {

 "build": "npm run clean && ui5 build --include-task=generateManifestBundle generateCachebusterInfo && npm run zip",

 "zip": "cd dist && npx bestzip ../veranstaltungen-content.zip \*",

 "clean": "npx rimraf veranstaltungen-content.zip dist"

 },

 "ui5": {

 "dependencies": [

 "@sap/ui5-builder-webide-extension"

 ]

 }

}

specVersion: '1.0'

metadata:

 name: fiori

type: application

resources:

 configuration:

 propertiesFileSourceEncoding: UTF-8

builder:

 resources:

 excludes:

 - "/test/\*\*"

 - "/localService/\*\*"

 customTasks:

 - name: webide-extension-task-updateManifestJson

 afterTask: generateVersionInfo

 configuration:

 appFolder: webapp

 destDir: dist

 - name: webide-extension-task-resources

 afterTask: webide-extension-task-updateManifestJson

 configuration:

 nameSpace: ns

 - name: webide-extension-task-copyFile

 afterTask: webide-extension-task-resources

 configuration:

 srcFile: "/xs-app.json"

 destFile: "/xs-app.json"

## Hinzufügen von app/xs-app.json

{

 "welcomeFile": "/index.html",

 "authenticationMethod": "route",

 "logout": {

 "logoutEndpoint": "/do/logout"

 },

 "routes": [

 {

 "source": "^(.\*)$",

 "target": "$1",

 "service": "html5-apps-repo-rt",

 "authenticationType": "none"

 }

 ]

}

## Anpassung von app/manifest.json

"dataSources": {

 "mainService": {

 "uri": "veranstaltungen/",

 "sap.app": {

 ...

 "crossNavigation": {

 "inbounds": {

 "fe-inbound": {

 "signature": {

 "parameters": {},

 "additionalParameters": "allowed"

 },

 "semanticObject": "veranstaltungen",

 "action": "display",

 "title": "Veranstaltungen anzeigen"

 }

 }

 }

 },

 "sap.app": {

 ...

 },

 "sap.cloud": {

 "public": true,

 "service": "app\_service"

 },

## Erstellung mta.yaml

\_schema-version: '3.1'

ID: CAP

version: 1.0.0

parameters:

 enable-parallel-deployments: true

build-parameters:

 before-all:

 - builder: custom

 commands:

 - npm install --production

 - npx -p @sap/cds-dk cds build --production

modules:

 # --------------------- SERVER MODULE ------------------------

 - name: app-srv

 # ------------------------------------------------------------

 type: nodejs

 path: gen/srv

 requires:

 # Resources extracted from CAP configuration

 - name: app-db

 provides:

 - name: srv-api # required by consumers of CAP services (e.g. approuter)

 properties:

 srv-url: ${default-url}

# ------------------------------------------------------------

# Destinations and Service Keys

# ------------------------------------------------------------

 - name: app-destination-content

 type: com.sap.application.content

 requires:

 - name: app\_html\_repo\_host

 parameters:

 service-key:

 name: app\_html\_repo\_host-key

 - name: app-destination-service

 parameters:

 content-target: true

 parameters:

 content:

 subaccount:

 destinations:

 - Name: app\_service\_app\_html\_repo\_host

 ServiceInstanceName: app-html5-app-host-service

 ServiceKeyName: app\_html\_repo\_host-key

 sap.cloud.service: app\_service

# ------------------------------------------------------------

# UI Deployer

# ------------------------------------------------------------

 - name: app\_ui\_deployer

 type: com.sap.application.content

 path: .

 requires:

 - name: app\_html\_repo\_host

 parameters:

 content-target: true

 build-parameters:

 build-result: resources

 requires:

 - artifacts:

 - veranstaltungen-content.zip

 name: veranstaltungen

 target-path: resources/

# ------------------------------------------------------------

# App Modules

# ------------------------------------------------------------

- name: veranstaltungen

 type: html5

 path: app

 build-parameters:

 builder: custom

 commands:

 - npm install

 - npm run build

 supported-platforms: []

resources:

# ------------------------------------------------------------

# Service Resources: Destination, html5-apps-repo

# ------------------------------------------------------------

 - name: app-destination-service

 type: org.cloudfoundry.managed-service

 requires:

 - name: srv-api

 parameters:

 service: destination

 service-name: app-destination-service

 service-plan: lite

 config:

 init\_data:

 subaccount:

 existing\_destinations\_policy: update

 destinations:

 - Name: app

 Description: CAP sample service

 Authentication: NoAuthentication

 ProxyType: Internet

 Type: HTTP

 URL: ~{srv-api/srv-url}

 HTML5.DynamicDestination: true

 ForwardAuthToken: true

 - name: app\_html\_repo\_host

 type: org.cloudfoundry.managed-service

 parameters:

 service: html5-apps-repo

 service-name: app-html5-app-host-service

 service-plan: app-host

## Durchführung des Deployments

mbt build

cf deploy mta\_archives/app\_1.0.0.mtar

# Definition UI mit Annotations

//Filterbar

SelectionFields: [

    incidentStatus\_code,

    category\_code,

    priority\_code

]

//Graph

Chart : {

 ChartType : #Column,

 Dimensions: [category\_code],

 DimensionAttributes : [{

 Dimension : category\_code,

 Role : #Series

 }],

 Measures:[IncidentsPerCategory],

 MeasureAttributes: [{

 Measure : IncidentsPerCategory,

 Role : #Axis1

 }]

//Table

LineItem: [

 {Value : priority\_code},

 {Value : category\_code},

 {Value : incidentStatus\_code},

 {Value : title}

 ]

# Freestyle App Development + Deployment

## Hinzufügen vom OData v2 Proxy

npm install @sap/cds-odata-v2-adapter-proxy -s

"use strict";

const cds = require("@sap/cds");

const proxy = require("@sap/cds-odata-v2-adapter-proxy");

cds.on("bootstrap", app => app.use(proxy()));

module.exports = cds.server;

## Installation Generator

## Generierung App

https://myservice.cfapps.eu10.hana.ondemand.com/v2/myMainEntity/

## Anpassung der App fürs Deployment

### Anpassung app/manifest.json

"dataSources": {

 "mainService": {

 "uri": "veranstaltungen/",

 "type": "OData",

 "settings": {

 "odataVersion": "4.0",

 "localUri": "localService/metadata.xml"

 }

 }

"models": {

 "": {

 "dataSource": "mainService",

 "preload": true,

 "settings": {

 "synchronizationMode" : "None"

 }

}

### Anpassung app/index.html

src=https://sapui5.hana.ondemand.com/resources/sap-ui-core.js

### Anpassung app/Component.js

sap.ui.define([

 "sap/ui/core/UIComponent",

], function (UIComponent) {

 return UIComponent.extend("myApp.Component", {

 metadata: {

 manifest: "json"

 },

 init: function () {

 UIComponent.prototype.init.apply(this, arguments);

 }

 });

});