1. Creiamo un modello json (tiles.json) dentro la cartella model

[

    {

      "title": "SAP Teched 2020 Countdown",

      "route": "first"

    },

    {

      "title": "Covid 19",

      "route": "second"

    },

    {

      "title": "Vaccination Tracker",

      "route": "third"

    },

    {

      "title": "Pollution Tracker",

      "route": "EXT-third",

      "url":"http://localhost:8081/index.html"

    }

  ]

1. Creo TilesView e TilesController

3)Voglio generare le mie tiles dinamicamente, come faccio? Le metto dentro un layout

Che dipendenza per layout?  xmlns:l="sap.ui.layout"

4)Imposto tiles.json come modello della view : “Tiles”

 function (Controller , JSONModel) {

        "use strict";

        return Controller.extend("tilesproject.controller.Tiles", {

            onInit: function () {

                let myTylesModel = new JSONModel("../model/tiles.json")

                this.getView().setModel(myTylesModel , "tiles")

            }

5)Imposto la view:  
 <l:HorizontalLayout id="hLayout1" allowWrapping="true" content="{tiles>/}">

        <GenericTile id="tileItem" class="sapUiTinyMarginBegin

        sapUiTinyMarginTop tileLayout" header="{tiles>title}"

        press="press(${tiles>route})">

        </GenericTile>

   </l:HorizontalLayout>

press="press(${tiles>route})"> cosa stiamo facendo qui?

Stiamo passando alla funzione press il route dell’item del modello di riferimento

5)Creiamo la view e controller countdown.

6)Implementiamo il routing per la view countdown:

nei "routes": [

e nei "targets": {

 "routes": [

        {

          "name": "RouteTiles",

          "pattern": ":?query:",

          "target": [

            "TargetTiles"

          ]

        },

        {

          "name": "first",

          "pattern": "first",

          "target": [

            "Countdown"

          ]

        }

      ],

      "targets": {

        "TargetTiles": {

          "viewType": "XML",

          "transition": "slide",

          "clearControlAggregation": false,

          "viewId": "Tiles",

          "viewName": "Tiles"

        },

           "Countdown": {

          "viewType": "XML",

          "viewId": "countdown.Countdown",

          "viewName": "countdown.Countdown"

        }

5)Implementiamo funzione routing:

press:function (oRoute) {

                this.getOwnerComponent().getRouter().navTo(oRoute);

            }

6) SVILUPPIAMO VIEW COUNTODOWN

1b) usiamo css grid come layout

<https://sapui5.hana.ondemand.com/#/entity/sap.ui.layout.cssgrid.CSSGrid/sample/sap.ui.layout.sample.GridTemplateRows/code> (aiuta per impostare numero colonne , rige e larghezza di entrambe

CountdownView:

<mvc:View controllerName="tilesproject.controller.countdown.Countdown"

    xmlns:mvc="sap.ui.core.mvc" displayBlock="true"

    xmlns:l="sap.ui.layout"

    xmlns:grid="sap.ui.layout.cssgrid"

    xmlns="sap.m">

   <Page id="pageCountdown">

   <grid:CSSGrid id="grid1"

     gridTemplateRows="1fr"

     gridGap="1rem"

   >

   <Image id="sapImg" src="https://upload.wikimedia.org/wikipedia/commons/8/8f/SAP-Logo.svg"/>

     <grid:CSSGrid

               id="grid2"

               gridTemplateColumns="1fr 1fr 1fr 1fr"

               gridGap="1rem">

                    <GenericTile id="tileDay" class="sapUiTinyMarginBegin sapUiTinyMarginTop tileLayout" header="Days"  press="press">

          <TileContent id="tileC1">

               <NumericContent id="numericC1" value="{timer>/days}"  withMargin="false" valueColor="Critical" />

          </TileContent>

     </GenericTile>

     <GenericTile id="tileHours" class="sapUiTinyMarginBegin sapUiTinyMarginTop tileLayout" header="Hours"  press="press">

          <TileContent id="tileC2">

               <NumericContent id="numericC2" value="{timer>/hours}"  valueColor="Error" withMargin="false" />

          </TileContent>

     </GenericTile>

     <GenericTile id="tileMinutes" class="sapUiTinyMarginBegin sapUiTinyMarginTop tileLayout" header="Minutes"  press="press">

          <TileContent id="tileC3">

               <NumericContent id="numericC3" value="{timer>/minutes}" valueColor="Good" withMargin="false" />

          </TileContent>

     </GenericTile>

     <GenericTile id="tileSeconds" class="sapUiTinyMarginBegin sapUiTinyMarginTop tileLayout" header="Seconds"  press="press">

          <TileContent id="tileC4">

               <NumericContent id="numericC4" value="{timer>/seconds}" withMargin="false" />

          </TileContent>

     </GenericTile>

          </grid:CSSGrid>

   </grid:CSSGrid>

   </Page>

</mvc:View>  
  
creo model per countdownview:  
 onInit: function () {

                this.timer = {

                    "days":0,

                    "hours":0,

                    "minutes":0,

                    "seconds":0

                  }

                let myTilesModel = new JSONModel(this.timer);

                this.getView().setModel(myTilesModel,"timer");

                setInterval(this.calculateTime.bind(this),1000);

            },

Creo funzione calcolo time:

calculateTime:function(){

                let techedDate = new Date("Dec 8 2023");

                let currentDate = new Date();

                let diff = techedDate.getTime() - currentDate.getTime();

                this.timer.days = Math.floor(diff / ( 1000 \* 60 \* 60 \* 24 ));

                this.timer.hours = Math.floor((diff % ( 1000 \* 60 \* 60 \* 24 )) / ( 1000 \* 60 \* 60 ));

                this.timer.minutes = Math.floor((diff % ( 1000 \* 60 \* 60 ))/ ( 1000 \* 60  ) );

                this.timer.seconds = Math.floor((diff % ( 1000 \* 60  ))/ ( 1000  ) );

                this.getView().getModel("timer").setData(this.timer);

              },