**Step 1. Hello World**

Index.html: Contine 2 sectiuni principale: Head si Body.

Head: e folosit de browser pt a procesa documentul, prin intermediul tagurilor meta

Body: aici se descrie aspectul paginii (putem pune paragrafe poze etc) tot cu ajutorul tagurilor(p, img)

*Pt rulare folosim: ui5 serve*

Port:8080 si trb sa avem “/document”

**Step 2. Bootstrap**

Adaugam framework ul SAPUI5 si modulele de baza.

Pt asta folosim tag-ul script in Head. In interiorul tag-ului folosim attribute precum:

<script>

src->spune browserului unde gaseste libraria SAPUI5

theme-> alegem tema

libs-> alegem libraria care are controle UI de un anumit tip(noi avem m)

compactVersion-> pt cea mai recenta versiune de SAPUI5, folosim “edge”

async-> setam pe true ca sa ruleze asincron( adica sa faca lucrurile in paralel, mai multe fire de executie)-> incarcarea simultana a resurselor pe ecran !!!

onInit-> modul prin care se Evita scrierea de cod js executabil direct in fisierul HTML(Pt Securitate) II SPUNEM UNDE SA GASEASCA FISIERUL ( II DAM RUTA) ULTIMUL E FISIERUL NOSTRU

resourceroots-> specificam ca resursele sunt localizate in acelasi folder ca index.html prin “./”

Cream fisierul index.js-> in el se gaseste logica. Este apelat in scipts

Sap.ui.define-> defineste un module cu un propriile id si dependinte care vor fi incarcate async

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

**Step 3. Controls**

**Controlele-folosite pt a define aspectul si comportamentul unor parti ale paginii**

In index.html la body stergem textul si punem sapUiBody-> adauga diferite teme pt stil

Dam id=”content”

A screenshot of a computer

Description automatically generated with medium confidence

**Step 4. XML Views**

**XML VIEWS- ofera modalitate prin care sa modularizam controalele folosite( adica sa separam)**

**Text

Description automatically generated**

App.view.xml-responsabil pt instantierea lui text control(la pasaul 3 era instantiate in index.js)

Referentiaza sap.m-> adica libraria.. sap.m e default namespace astfel putem folosi Text control

Text

Description automatically generated

Text

Description automatically generated

**Step 5. Controllers**

**Controllers-js files in ui5 app in care pui logica pt ui … practic fisierele controllers implementeaza logica pt XML VIEW**

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

controllerName->ce controller sa foloseasca.. sap.ui….-> controller e numele folderului iar App numele controllerului

Button-> ce text sa afiseze si event press

Text

Description automatically generated

onShowHello-handler function

Text

Description automatically generated

**Step 6. Modules**

**Modules- orice js file care poate fi incarcat si executat de browser**

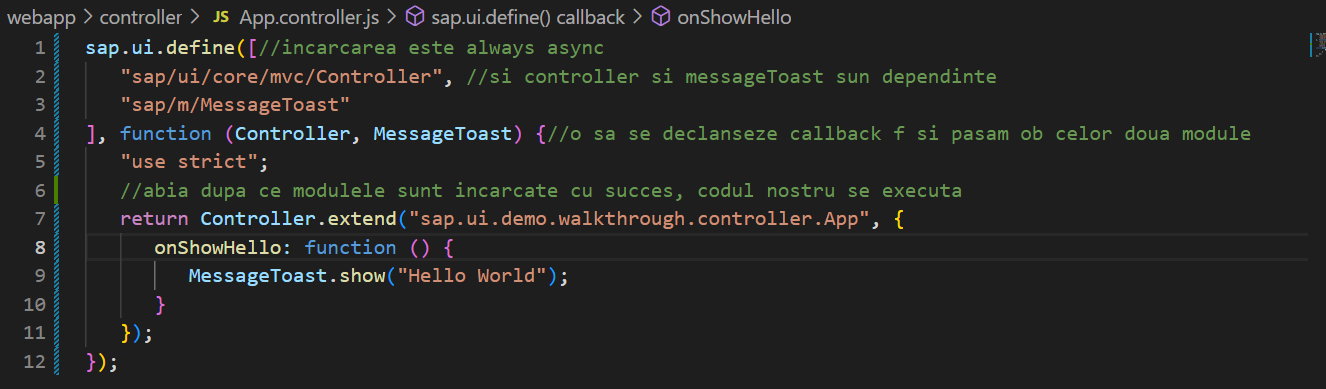
* **Poate continue orice tip de cod, nu are restrictii de genul**

**SE REFERA SA IMPARTIM CODUL IN MAI MULTE FISIERE(MODULE), SA NU AVEA UN SG FISIER CU O TONA DE COD)**

DE RETINUT!!

Fisierul App.controller.js este in sine un modul pt ca avem define

Si de asemenea ne folosim de modulele Controller si MessageToast(linia 2 si 3)



Text

Description automatically generated

1. Se refera ca putem folosi fisierul(modulul) peste tot in app

**Step 7. JSON Model**

**JSON Model- client side model, doar tine date, nu poate trimite date updatate, trb ca diferite metode din app sa faca asta**

Text

Description automatically generated

IN INPUT

Text

Description automatically generated

**Step 8. Translatable Texts**

**Avem nev de fisierul i18n**

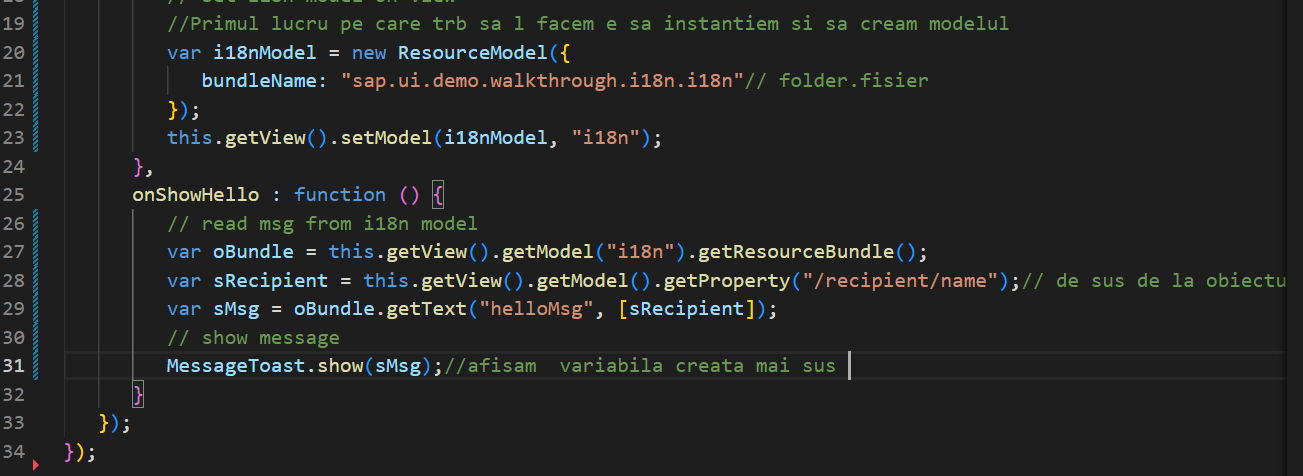
**Graphical user interface, text, application, chat or text message

Description automatically generated**

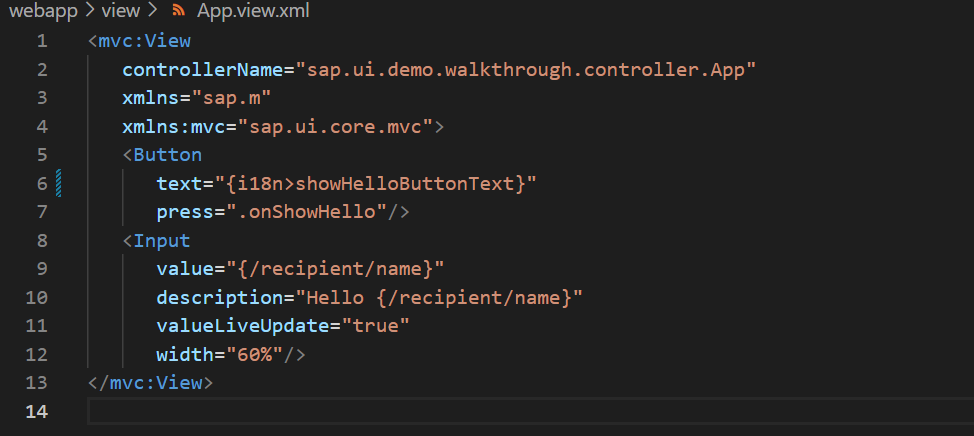
**Apoi in controller.. adaugam un nou modul ResourceModel**

****

**Facem ca la celelalte**

****

**Ca si butonul sa aiba un text din fifsierul i18n**

****