State Funcții

~Documentatie proiect~

## Login (script1.js)

Returns json data about a single user.

* URL

'/api/Views/Login/SignIn1' – for user „Facultate”;

'/api/Views/Login/SignIn2' – for user „Universitate”

'/api/Views/Login/SignIn3' – for user „Departament”

* Method:

POST

* URL Params

No parameters required:

For each user, a different authentification method it’s called.

* Data Params

None

* Success Response:
  + User: Facultate
  + Content: { username: „AC”, password : none, user type: "Facultate" }
* Error Response:
  + Content: { error : „Utilizator sau parola incorecta”
* Sample Call:

$('#btnLogin').click(function () {

var user = { Scurt: $('#txtScurt').val(), Parola: $('#txtParola').val() };

var selectedValue = document.getElementById("list").value;

switch (selectedValue) {

case "Facultate": $.ajax({

url: 'http://localhost:50106/api/Views/Login/SignIn1',

type: "POST",

contentType: 'application/json',

data:

JSON.stringify(user),

success: function (response) {

if (status == "success")

sessionStorage.setItem('accesToken', response.access\_token);

//sessionStorage.setItem('Scurt', response.Scurt);

sessionStorage.setItem("id1", response);

window.location.href = "PageAdminPanel.html";

},

error: function (error) {

alert("Utilizator sau parola incorecte!");

}

});

break;

}

});

## User “Departament” - Data.html Page (data.js)

Returns json data for user „Departament”. Data about „CadreDidactice” because user „Departament” can perform CRUD only on these dates .

**READ - cadreList() function** – the method that returns data from the „CadreDidactice” table

* URL

'/api/CadreDidactice2/' + id3

* Method:

GET

* URL Params

Required:

id3=[integer]

* Success Response:
  + Content: { id : 1, nume : „Ignat”, prenume: „Iosif”, Departament: „Calculatoare”, Poziția: „Prof”, Titlu:” dr. ing.”, Titular: 1 }
* Error Response:
  + Content: { error : "Eroare la incarcare." }
* Sample Call:

$.ajax({

url: '/api/CadreDidactice1/' + id3,

type: 'GET',

dataType: 'json',

success: function (response) {

cadreListSuccess(response);

},

error: function () {

alert("Eroare la incarcare.");

}

});

**CREATE** – $('#btnSave') - the method that inserts new data in the „CadreDidactice” table

* URL

'/api/CadreDidactice2'

* Method:

POST

* URL Params

Required: None

* Success Response:
  + Content: { success : "Inserat" }
* Error Response:
  + Content: { error : "Nu s-a inserat. Acest cadru didactic exista deja. Pozitia cadrului didactic trebuie sa fie setata!" }
* Sample Call:

$.ajax({

url: '/api/CadreDidactice2',

type: "POST",

contentType: 'application/json',

data:

JSON.stringify(cadru),

success: function (response) {

if (status == "success")

sessionStorage.setItem('accesToken', response.access\_token);

$('#txtNume').val('');

$('#txtPren').val('');

$('#txtPoz').val('');

$('#txtTitlu').val('');

$('#txtTitular').val('');

alert("Inserat");

window.location.reload();

},

error: function (error) {

alert("Nu s-a inserat. \n" + "Acest cadru didactic exista deja. \n"+" Pozitia cadrului didactic trebuie sa fie setata!");

}

});

**UPDATE - $(**'#btnEdit') – the method that updates data from the „CadreDidactice” table

* URL

'/api/CadreDidactice2/'

* Method:

PUT

* URL Params

Required:

id = [integer]

* Success Response:
  + Content: { success: "Actualizat!" }
* Error Response:
  + Content: { error : "Eroare la incarcare." }
* Sample Call:

$.ajax({

url: '/api/CadreDidactice2/' + id,

type: "PUT",

contentType: 'application/json',

data: JSON.stringify(cadru),

success: function (response) {

if (status == "success")

sessionStorage.setItem('accesToken', response.access\_token);

alert("Actualizat!");

},

error: function (error) {

alert("Nu s-a actualizat." + "Nu s-a facut nicio modificare.");

}

});

**DELETE** – **cadruDelete(id) function** - the method that deletes the data from the „CadreDidactice” table

* URL

"/api/CadreDidactice2/" + id

* Method:

DELETE

* URL Params

Required:

id = [integer]

* Success Response:
  + Content: {success: "Sters cu succes!"}
* Error Response:
  + Content: { error : "Nu s-a sters!" }
* Sample Call:

$.ajax({

url: "/api/CadreDidactice2/" + id,

type: 'DELETE',

success: function (cadru) {

$(ctl).parents("tr").remove();

alert("Sters cu succes!");

},

error: function () {

alert("Nu s-a sters!");

}

});

**Adding options in the combo box**

– **selectPozitia(value) function** - the function that return the options aviable for each combo box

* URL (exemple for position combo box – same function for title combo box, but with different url - '/api/Titlu/')

'/api/Pozitia/'

* Method:

GET

* URL Params

Required: None

* Success Response:
  + Content: {success: options += '<option value="' + response[i] + '">' + response[i] + '</option>'}
* Error Response:
  + Content: { error : "Valorile pozitiilor nu pot fi returnate." }
* Sample Call:

$.ajax({

url: '/api/Pozitia/',

type: 'GET',

dataType: 'json',

success: function (response) {

options = ' <option selected="selected">' + "Select position" + '</option>';

for (var i = 0; i < response.length; i++) {

options += '<option value="' + response[i] + '">' + response[i] + '</option>';

}

$("#selectPozitia").html(options);

},

error: function () {

alert("Valorile pozitiilor nu pot fi returnate.");

}

});

**Getting the id of the options from the combo box** – **secondFunction(value) function** - the function that returns the id of the selected position in the specific combo box

* URL (exemple for position combo box – same function for title combo box, but with different url - '/api/Titlu1?nume=' + value)

'/api/Pozitia1?nume=' + value

* Method:

GET

* URL Params

Required:

value = [string]

* Success Response:
  + Content: {success: idPozitia } ( the needed value)
* Error Response:
  + Content: { error : "Id unei pozitii nu este obtinut." }
* Sample Call:

$.ajax({

url: '/api/Pozitia?nume=' + value,

type: 'GET',

dataType: 'json',

success:function (data)

{

result = data;

},

error: function () {

alert("Id unei pozitii nu este obtinut.");

}

});

## User “Facultate” - PageAdminPanel.html Page

## Grupe Facultate Page (grupe.js)

Returns json data for user „Facultate”. Data about „Grupe” because user „Facultate” can perform CRUD operations only on the characteristics of the groups.

**READ - grupeList() function** – the method that returns data from the „Grupe” table.

* URL

'/api/CrudGrupeAn/' + id1

* Method:

GET

* URL Params

Required:

id1 = [integer]

* Success Response:
  + Content: { id : 1, cod\_pi : ‘0000003569’, An: 1, nr\_grupe: 2, nr\_subgr: 4, Facultate:”Facultatea de Automatica si Calculatoare “ }
* Error Response:
  + Content: { error : "Grupele facultatii nu pot fi obtinute!" }
* Sample Call:

$.ajax({

url: '/api/CrudGrupeAn/' + id1,

type: 'GET',

dataType: 'json',

success: function (response) {

grupeListSuccess(response);

},

error: function () {

alert("Grupe facultatii nu pot fi obtinute!");

}

});

**CREATE** – $('#btnSave') - the method that inserts new data in the „Grupe” table

* URL

'/api/CrudGrupeAn'

* Method:

POST

* URL Params

Required: None

* Success Response:
  + Content: { success: "Inserat"}
* Error Response:
  + Content: { error : "Nu s-a inserat. Inregistrarea exista deja. Faceti o alta introducere sau editati inregistrarea existenta!" }
* Sample Call:

$.ajax({

url: '/api/CrudGrupeAn',

type: "POST",

contentType: 'application/json',

data:

JSON.stringify(cadru),

success: function (response) {

if (status == "success")

sessionStorage.setItem('accesToken', response.access\_token);

$('#txtCod').val('');

$('#txtAn').val('');

$('#txtNrg').val('');

$('#txtNrsg').val('');

alert("Inserat");

window.location.reload();

},

error: function (error) {

alert("Nu s-a inserat.\n"+"Inregistrarea exista deja.\n"+"Faceti o alta introducere sau editati inregistrarea existenta!");

}

});

**UPDATE**– $('#btnEdit') - the method that updates data in the „Grupe” table

* URL

'/api/CrudGrupeAn/' + id

* Method:

PUT

* URL Params

Required:

id = [integer]

* Success Response:
  + Content: { success: "Actualizat!"}
* Error Response:
  + Content: { error : "Nu s-a actualizat." }
* Sample Call:

$.ajax({

url: '/api/CrudGrupeAn/' + id,

type: "PUT",

contentType: 'application/json',

data: JSON.stringify(grupa),

success: function (response) {

if (status == "success")

sessionStorage.setItem('accesToken', response.access\_token);

$("#txtCod1").val('');

$("#txtAn1").val('');

$("#txtNr1").val('');

$("#txtNr\_sub1").val('');

alert("Actualizat!");

window.location.reload();

},

error: function (error) {

alert("Nu s-a actualizat.");

}

});

**DELETE** – **grupaDelete(id) function** - the method that deletes the data from the „Grupe” table

* URL

"/api/CrudGrupeAn/" + id

* Method:

DELETE

* URL Params

Required:

id = [integer]

* Success Response:
  + Content: {success: "Sters cu succes!"}
* Error Response:
  + Content: { error : "Nu s-a sters!" }
* Sample Call:

$.ajax({

url: "/api/CrudGrupeAn/" + id,

type: 'DELETE',

success: function (cadru) {

$(ctl).parents("tr").remove();

alert("Sters cu succes!");

window.location.reload();

},

error: function () {

alert("Nu s-a sters");

}

});

**Adding options in the combo box**

– **selectCod(value) function** - the function that return the options aviable for each combo box

* URL (exemple for code combo box – same function for year combo box, but with different url - '/api/An/' + id1)

'/api/CodPi/' + id1

* Method:

GET

* URL Params

Required: id1 = [integer]

* Success Response:
  + Content: {success: options += '<option value="' + response[i] + '">' + response[i] + '</option>'}
* Error Response:
  + Content: { error : "Valorile codurilor planurilor de invatamant nu pot fi obtinute." }
* Sample Call:

$.ajax({

url: '/api/CodPi/' + id1,

type: 'GET',

dataType: 'json',

success: function (response) {

options = ' <option selected="selected">' + "Select code" + '</option>';

for (var i = 0; i < response.length; i++) {

options += '<option value="' + response[i] + '">' + response[i] + '</option>';

}

$("#in1").html(options);

},

error: function () {

alert("Valorile codurilor planurilor de invatamant nu pot fi obtinute");

}

});

**Getting the id of the options from the combo box** – **jsfunction1(value) function** - the function that returns the id of the selected position in the specific combo box

* URL (for code combo box there is no need of this ajax call, the value is taken directly from the combo box)

'/api/An?an=' + value – for year combo box

* Method:

GET

* URL Params

Required:

value = [string]

* Success Response:
  + Content: {success: result3} ( the needed value)
* Error Response:
  + Content: { error : "Valoare unui anumit an nu este obtinuta!" }
* Sample Call:

$.ajax({

url: '/api/An?an=' + value,

type: 'GET',

dataType: 'json',

success: function (data) {

result3 = data;

},

error: function () {

alert("Valoare unui anumit an nu este obtinuta!");

}

});

## Planuri Invatamant Page (planInvatamant.js)

Returns json data for user „Facultate”. Data from „PlanInv” because user „Facultate” can see the data stored in this database table.

**READ - planuriList() function** – the method that returns data from the „PlanInv” table.

* URL

'/api/PlanInvatamant/' + id1

* Method:

GET

* URL Params

Required:

id1 = [integer]

* Success Response:
  + Content: { unitpos: ‘0000004084’, cod : ‘0000000714’, sectiune: ‘0000000001’, unitform: ‘0000004084’, secom: NULL, nume: Calculatoare (lb.eng.)-lic, Facultate: ”Facultatea de Automatica si Calculatoare “, Scurt: C&TI(e) }
* Error Response:
  + Content: { error : "Planurile de invatamant nu pot fi obtinute" }
* Sample Call:

$.ajax({

url: '/api/PlanInvatamant/' + id1,

type: 'GET',

dataType: 'json',

success: function (response) {

planuriListSuccess(response);

},

error: function () {

alert("Planurile de invatamant nu pot fi obtinute");

}

});

## Alocare Planuri Invatamant Page (alocare1.js)

Returns json data for user „Facultate”. Data from „PIAlocare” because user „Facultate” can perform CRUD operations only on the characteristics of these allocations.

**READ - alocariList() function** – the method that returns data from the „PIAlocare” table.

* URL

'/api/PIAlocare/' + id1

* Method:

GET

* URL Params

Required:

id1 = [integer]

* Success Response:
  + Content: { id: 1, PiDetail: 1, Departament: Calculatoare}
* Error Response:
  + Content: { error : "Alocarile nu pot fi afisate." }
* Sample Call:

$.ajax({

url: '/api/PIAlocare/' + id1,

type: 'GET',

dataType: 'json',

success: function (response) {

alocariListSuccess(response);

//alert(id);

},

error: function () {

alert("Alocarile nu pot fi afisate.");

}

});

**CREATE** – $('#btnSave') - the method that inserts new data in the „PIAlocare” table

* URL

'/api/PIAlocare'

* Method:

POST

* URL Params

Required: None

* Success Response:
  + Content: { success: "Inserat"}
* Error Response:
  + Content: { error : "Inregistrarea exista!" }
* Sample Call:

$.ajax({

url: '/api/PIAlocare',

type: "POST",

contentType: 'application/json',

data:

JSON.stringify(alocare),

success: function (response) {

if (status == "success")

sessionStorage.setItem('accesToken', response.access\_token);

$('#txtId').val('');

$('#txtDep').val('');

alert("Inserat");

window.location.reload();

},

error: function (error) {

alert("Inregistrarea exista!");

}

});

**UPDATE**– $('#btnEdit') - the method that updates data in the „PIAlocare” table

* URL

'/api/PIAlocare/' + id

* Method:

PUT

* URL Params

Required:

id = [integer]

* Success Response:
  + Content: { success: "Actualizat"}
* Error Response:
  + Content: { error : "Nu s-a actualizat." }
* Sample Call:

$.ajax({

url: '/api/PIAlocare/' + id,

type: "PUT",

contentType: 'application/json',

data: JSON.stringify(alocare),

success: function (response) {

if (status == "success")

sessionStorage.setItem('accesToken', response.access\_token);

$("#txtId1").val('');

$("#txtDep1").val('');

alert("Actualizat");

window.location.reload();

},

error: function (error) {

alert("Nu s-a actualizat.");

}

});

**DELETE** – **alocareDelete(id) function** - the method that deletes the data from the „PIAlocare” table

* URL

"/api/PIAlocare/" + id

* Method:

DELETE

* URL Params

Required:

id = [integer]

* Success Response:
  + Content: {success: "Sters cu succes!"}
* Error Response:
  + Content: { error : "Nu s-a sters!" }
* Sample Call:

$.ajax({

url: "/api/PIAlocare/" + id,

type: 'DELETE',

success: function (alocare) {

$(ctl).parents("tr").remove();

alert("Sters cu succes!");

},

error: function () {

alert("Nu s-a sters");

}

});

**Adding options in the combo box**

– **selectDepartament(value) function** - the function that return the options aviable for departament combo box

* URL (exemple for departament combo box)

'/api/SelectDepartament/' + id

* Method:

GET

* URL Params

Required: id = [integer]

* Success Response:
  + Content: {success: options3 += '<option value="' + response[i] + '">' + response[i] + '</option>'}
* Error Response:
  + Content: { error : "Departamentele nu pot fi incarcate" }
* Sample Call:

$.ajax({

url: '/api/SelectDepartament/' + id,

type: 'GET',

dataType: 'json',

success: function (response) {

for (var i = 0; i < response.length; i++) {

options3 += '<option value="' + response[i] + '">' + response[i] + '</option>';

}

$("#selectedDepartament1").html(options3);

},

error: function () {

alert("Departamentele nu pot fi incarcate");

}

});

**Getting the id of the options from the combo box** – **selectDepartament(value) function** - the function that returns the id of the selected position in the specific combo box

* URL

'/api/SelectDepartament?departament=' + value

* Method:

GET

* URL Params

Required:

value = [string]

* Success Response:
  + Content: {success: idDep} ( the needed value)
* Error Response:
  + Content: { error : "Id-ul departamentului selectat nu poate fi obtinut." }
* Sample Call:

$.ajax({

url: '/api/SelectDepartament?departament=' + value,

type: 'GET',

dataType: 'json',

success: function (response1) {

idDep = response1;

},

error: function () {

alert("Id-ul departamentului selectat nu poate fi obtinut.");

}

});

## Detalii Planuri Invatamant Page (detaliiPlanInv.js)

Returns json data for user „Facultate”. Data from „PIDetali” because user „Facultate” can see the data stored in this database table.

**READ - detaliiList() function** – the method that returns data from the „PIDetali” table.

* URL

'/api/DetaliiPlanInv/' + id1

* Method:

GET

* URL Params

Required:

id1 = [integer]

* Success Response:
  + Content: { id: 2, cod: ‘0000001189’, formainv: ‘0000000008’, an: 3, semestru: 2, tipdiscip:’0000000003’, cod\_disc:’0000000001’, limbapred: ‘0000000001’, formapred:’0000000001’, numarore: 2, tipunist: ‘0000000015’, ordine: 0, catedra: ‘0000008251’, CicluStudii : 1, TipDisciplina: ‘O’, LimbaPredare: ‘Engleza’ , CSLP: ‘C’, FormaStudiu: ‘A’}
* Error Response:
  + Content: { error : "Detaliile planurilor de invatamant nu pot fi obtinute!!" }
* Sample Call:

$.ajax({

url: '/api/DetaliiPlanInv/' + id1,

type: 'GET',

dataType: 'json',

success: function (response) {

detaliiListSuccess(response);

//alert(id);

},

error: function () {

alert("Detaliile planurilor de invatamant nu pot fi obtinute!!");

}

});