

HealthClinicApi

U ovom dokumentu se nalazi detaljno objašnjenje zadatka. Zadatak je radjen u Visual studio, a potrebne pakete koje morate instalirati mozete naci u README fajlu.

Modeli u zadatku:

AdmissionRecord – prijem pacijenta

MedicalFindingRecord – nalaz pacijenta

Patient

Doctor

Veze u tabeli:

Doctor -> AdmissionRecord (1 na više); 1 doktor ima vise prijema, a 1 prijem pripada samo 1 doktoru

Patient -> AdmissionRecord (1 na više); 1 pacijent ima vise prijema, a 1 prijem pripada samo 1 pacijentu

AdmissionRecord -> MedicalFindingRecord (1 na više); 1 prijem ima vise nalaza, a 1 nalaz odgovara samo 1 prijemu

Patient -> MedicalFindingRecord (1 na više); 1 pacijent ima vise nalaza, 1 nalaz pripada 1 pacijentu

Radi lakseg testiranja programa, podaci su vec “seedani” u bazu.

Objasnit cu svaku funkcionalnosti na osnovu teksta zadatka..

PACIJENT

1. Evidenciju (unos, izmjena i brisanje) prozvoljnog broja pacijenata u sistemu, pri cemu je za svakog pacijenta neophodno unijeti sljedece: - Ime i Prezime pacijenta - Datum rođenja - Spol (lista vrijednosti: musko, zensko, nepoznato) - Adresa (opcionalno) - Broj telefona (opcionalno)

GET ALL PATIENTS:

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** /api/Patient
- Parameters:** No parameters
- Execute:** Button to execute the request
- Clear:** Button to clear the request
- Responses:** Section showing the response details
- Curl:**

```
curl -X 'GET' \
  'https://localhost:7092/api/Patient' \
  -H 'accept: */*'
```
- Request URL:** `https://localhost:7092/api/Patient`
- Server response:** Code 200
- Response body:**

```
{
  "birthdate": "1999-07-01T00:00:00",
  "gender": "Female",
  "adress": "New Jersey 22",
  "number": "062575685"
},
{
  "id": 4,
  "name": "Rachel",
  "lastname": "Green",
  "birthdate": "1978-05-21T00:00:00",
  "gender": "Female",
  "adress": "Ferde Hauptman 32",
  "number": "+4930901820"
},
{
  "id": 5,
  "name": "Phoebe",
  "lastname": "Buffay",
  "birthdate": "2002-10-10T00:00:00",
  "gender": "Female",
  "adress": "Iza sedan mora i gora 92",
  "number": "+3812509182509"
}
},
{
  "success": true,
  "message": ""
}
```
- Response headers:**

```
content-type: application/json; charset=utf-8
date: Tue, 03 Jan 2023 09:18:50 GMT
server: Kestrel
```

Vidimo da get metoda radi, da je Code 200, success je true, a prikazao je listu svih pacijenata u bazi (ovi pacijenti su vec seedani u bazu)

POST PATIENT (dodaj pacijenta)

Required polja su Name, Lastname, Birthdate I Gender, a Adress I Number su opcionalni. Probat cu unijeti pacijenta cije je ime prazan string I dobit cemo gresku:

Patient

GET /api/Patient

POST /api/Patient

Parameters

No parameters

Request body

application/json

```
{
  "name": "",
  "lastname": "Sabotic",
  "birthdate": "1998-11-05",
  "gender": "Female",
  "address": "Humska 234",
  "number": "+38761395783"
}
```

Execute Clear

Request URL

https://localhost:7092/api/Patient

Server response

Code Details

400

Undocumented Error: response status is 400

Response body

```
{
  "data": null,
  "success": false,
  "message": "Name and Lastname can't be empty or contain only whitespaces!"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 03 Jan 2023 09:29:41 GMT
server: Kestrel
```

Vidimo da je response 400, success je false I dobijamo odgovarajucu poruku. Greska se isto tako desi ako se nikako ne unese Ime I prezime pacijenta, ili kao se ne unese Gender ili Birthdate. U slucaju da je broj telefona nevalidan format, javit ce gresku:

POST /api/Patient

Parameters

No parameters

Request body

application/json

```
{
  "name": "Nejla",
  "lastname": "Sabotic",
  "birthdate": "1998-11-05",
  "gender": "Female",
  "address": "Humska 234",
  "number": "ja sam ekskluziva kazu da sam mrak"
}
```

Execute Clear

Request URL
`https://localhost:7092/api/Patient`

Server response

Code	Details
400	<p>Undocumented Error: response status is 400</p> <p>Response body</p> <pre>{ "data": null, "success": false, "message": "Invalid phone number!"}</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Tue, 03 Jan 2023 09:34:06 GMT server: Kestrel</pre>

Broj može početi sa +, može imati minimum 8 ili maksimalno 16 cifara. Sljedeći zahtjev je ispravan.

```
{  "name": "Nejla",  "lastname": "Sabotic",  "birthdate": "1998-11-05",  "gender": "Female",  "adress": "Humska 234",  "number": "+39761395783"}
```

Response:

Request URL
`https://localhost:7092/api/Patient`

Server response

Code	Details
200	<p>Response body</p> <pre>{ "data": { "id": 0, "name": "Nejla", "lastname": "Sabotic", "birthdate": "1998-11-05T00:00:00", "gender": "female", "adress": "Humska 234", "number": "+39761395783" }, "success": true, "message": ""}</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Tue, 03 Jan 2023 09:48:17 GMT server: Kestrel</pre>

Sada je oke. Kada opet pokrenemo GET all metodu, vidjet ćemo da je novi pacijent dodan u bazu.

Request URL
`https://localhost:7092/api/Patient`

Server response

Code	Details
200	<p>Response body</p> <pre>{ "birthdate": "1970-05-21T00:00:00", "gender": "Female", "address": "Ferde Hauptman 32", "number": "+4930901820" }, { "id": 5, "name": "Phoebe", "lastname": "Buffay", "birthdate": "2002-10-10T00:00:00", "gender": "Female", "address": "Iza sedam mora i gora 92", "number": "+3812509182509" }, { "id": 6, "name": "Mejla", "lastname": "Sabotic", "birthdate": "1998-11-05T00:00:00", "gender": "Female", "address": "Humaka 234", "number": "+39761395783" }], "success": true, "message": "" }</pre> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Tue, 03 Jan 2023 09:49:37 GMT server: Kestrel</pre>

Vidimo novog pacijenta :D

PUT PATIENT

Prilikom update pacijenta, sva polja koja nisu unesena ce ostati ista ko prije. U primjeru ispod, promijenit cemo adresu, broj i pol pacijenta. Vidjet cemo da ce se to promijeniti, a da ce netaknuta polja ostati ista. Prije toga, moramo unijeti id pacijenta kojeg zelimo izmijeniti. Ako unesemo id koji ne postoji, javit ce nam gresku.

PUT /api/Patient

Parameters

Cancel

Reset

Name	Description
id	
integer(\$int32)	6
(query)	

Request body

application/json

```
{
  "gender": "Neuter",
  "address": "Bogalajaka 333",
  "number": "062575685"
}
```

Execute

Clear

Responses

Response:

Request URL

https://localhost:7092/api/Patient?id=6

Server response


Code

Details

200

Response body

```
{
  "data": {
    "id": 6,
    "name": "Nejla",
    "lastname": "Sabotic",
    "birthdate": "1998-11-05T00:00:00",
    "gender": "Neuter",
    "adress": "Maglajska 333",
    "number": "062575685"
  },
  "success": true,
  "message": "Your patient has been updated !"
}
```

 Download

Vidimo da su netaknuta polja ostala ista, a promijenio se spol, adresa I broj telefona.

GET SINGLE PATIENT:

Unesemo zeljeni id I zahtjev nam vraca trazenog pacijenta. U slucaju da pacijent sa tim id-em ne postoji, javit ce gresku.

GET /api/Patient/{id}

Parameters

Cancel

Name	Description
id * required	
integer(\$int32)	5
(path)	

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7092/api/Patient/5' \
  -H 'accept: */*'
```

Request URL

https://localhost:7092/api/Patient/5

Server response


Code

Details

200

Response body

```
{
  "data": {
    "id": 5,
    "name": "Phoebe",
    "lastname": "Buffay",
    "birthdate": "2002-10-10T00:00:00",
    "gender": "Female",
    "adress": "Iza sedam mora i gora 92",
    "number": "+3812509182509"
  },
  "success": true,
  "message": ""
}
```

 Download

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 03 Jan 2023 10:06:36 GMT
server: Kestrel
```

Responses

Code	Description	Links
200	Success	No links

DELETE PATIENT

Unesemo Id pacijenta kojeg zelimo obrisati I zahtjev nam vraca listu ostalih pacijenata. U slucaju da pacijent sa tim Id-em ne postoji, javit ce gresku. Vazno je spomenuti da ne mozemo obrisati pacijenta koji ima neki AdmissionRecord ili MedicalFindingRecord, jer se FK nalazi u tim tabelama. Prvo moramo obrisati AdmissionRecord I MedicalFindingRecord pacijenta, pa tek onda mozemo pacijenta. Ako pokusamo obrisati takvog pacijenta, dobit cemo gresku. Ovdje cemo obrisati novododanog pacijenta Nejlju pod ID-em 6.

DELETE /api/Patient/{id}

Parameters

Name	Description
id * required	integer(\$int32) (path)

Execute Clear

Responses

Curl

```
curl -X 'DELETE' \
  https://localhost:7052/api/Patient/6 \
  -H 'accept: */*'
```

Request URL

```
https://localhost:7052/api/Patient/6
```

Server response

Code Details

200

Response body

```
{
  "id": 6,
  "name": "Rachel",
  "lastname": "Green",
  "birthdate": "1970-05-21T00:00:00",
  "gender": "Female",
  "address": "Ferde Hauptman 32",
  "number": "+4930901826"
},
{
  "id": 5,
  "name": "Phoebe",
  "lastname": "Buffay",
  "birthdate": "2002-10-10T00:00:00",
  "gender": "Female",
  "address": "Ira sedon mora i gora 92",
  "number": "+3812509182509"
},
{
  "success": true,
  "message": "Your patient has been deleted!"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Tue, 03 Jan 2023 10:09:06 GMT
server: Kestrel
```

Vidimo da je u listi pacijenata obrisani pacijent sa Id-em 6.

DOKTOR

2. Evidenciju (unos, izmjenu i brisanje) proizvoljnog broja ljekara u sistemu, pri cemu je za svakog ljekara neophodno unijeti sljedece: - Ime ljekara - Prezime ljekara - Titula (lista vrijednosti: specijalista, specijalizant, med. sestra) – Sifra

U mom rjesenju, sifra doktora je **bilo koji validan int broj**

GET ALL DOCTORS

The screenshot shows a REST client window titled "Doctor". The method is GET and the URL is /api/Doctor. There are no parameters. The response status is 200. The response body is a JSON array of doctor objects.

```
curl -X 'GET' \
  'https://localhost:7892/api/Doctor' \
  -H 'accept: */*'

https://localhost:7892/api/Doctor

200

{
  "code": 3313,
  "id": 3,
  "name": "Sredko",
  "lastname": "Sredkić",
  "title": "Resident",
  "code": 4924,
  {
    "id": 4,
    "name": "Simonida",
    "lastname": "Puhalo",
    "title": "Nurse",
    "code": 8976,
    {
      "id": 5,
      "name": "Izet",
      "lastname": "Fazlinović",
      "title": "Specialist",
      "code": 8888
    }
  },
  "success": true,
  "message": ""
}
```

Response headers:

```
content-type: application/json; charset=utf-8
date: Tue, 03 Jan 2023 10:25:43 GMT
server: Kestrel
```

Zahtjev je vratio listu svih doktora u bazi.

POST DOCTOR (dodaj doktora)

Doktor ne može biti bez titule, sifra doktora ne može biti 0, ime i prezime doktora ne može biti prazno ili da se sastoji samo od razmaka. Svaki ovaj slučaj će vratiti grešku. Ispod ćemo precizirati slučajeve kada je sifra 0 i ime prazno

JSON koji saljemo:

```
{
  "name": "Bagi",
  "lastname": "Bagić",
  "title": "Specialist",
  "code": 0
}
```



```
}
```

Response:

Request URL

https://localhost:7092/api/Doctor

Server response

Code	Details
400	Error: response status is 400

Response body

```
{  "data": null,  "success": false,  "message": "Title and code can't be 0!"}
```

 Download

Sada saljemo prazno ime:

```
{  "name": "",  "lastname": "Bagic",  "title": "Specialist",  "code": 555}
```

Response:

Request URL

https://localhost:7092/api/Doctor

Server response

Code	Details
400	Error: response status is 400

Response body


```
{  "data": null,  "success": false,  "message": "Name and Lastname can't be empty"}
```

 Download

Sada saljemo validan JSON:

```
{  "name": "Bagi",  "lastname": "Bagic",  "title": "Specialist",  "code": 555}
```

Response:

Code	Details
200	<div>Response body</div> <pre>{ "data": { "id": 6, "name": "Bagi", "lastname": "Bagic", "title": "Specialist", "code": 555 }, "success": true, "message": ""}</pre> <div> Download</div>

Sada je sve oke I doctor je dodan u bazu.

PUT DOCTOR (update doktora)

Prilikom update doktora, sva polja koja nisu unesena ce ostati ista ko prije. U primjeru ispod, promijenit cemo titulu I šifru doktora. Vidjet cemo da ce se to promijeniti, a da ce netaknuta polja ostati ista. Prije toga, moramo unijeti id doktora kojeg zelimo izmijeniti. Ako unesemo id koji ne postoji, javit ce nam gresku.

Saljemo ovakav zahtjev:

```
{  "title": "Resident",  "code": 333}
```

Response:

Request URL	
https://localhost:7092/api/Doctor?id=6	
Server response	
Code	Details
200	<div>Response body</div> <pre>{ "data": { "id": 6, "name": "Bagi", "lastname": "Bagic", "title": "Resident", "code": 333 }, "success": true, "message": "Your doctor has been updated !"}</pre>

Vidimo da je ime I prezime ostalo kao prije, a da su se titula I šifra promijenile.

GET SINGLE DOCTOR

Unesemo zeljeni id i zahtjev nam vraća traženog doktora. U slučaju da doktor sa tim id-em ne postoji, javit će grešku. Posto kod pacijenta nisam pokazala kako greška izgleda, ovdje ćemo unijeti id 9 i vidjet ćemo sljedeći response:

Request URL

```
https://localhost:7092/api/Doctor/9
```

Server response

Code	Details
404 <i>Undocumented</i>	Error: response status is 404

Response body

```
{
  "data": null,
  "success": false,
  "message": "Doctor with that id doesn't exist!"
}
```

 Download

Sada ćemo unijeti doktora sa id-em 6, a zahtjev će nam vratiti tog doktora

Request URL


```
https://localhost:7092/api/Doctor/6
```

Server response

Code	Details
200	

Response body

```
{
  "data": {
    "id": 6,
    "name": "Bagi",
    "lastname": "Bagic",
    "title": "Resident",
    "code": 333
  },
  "success": true,
  "message": ""
}
```

 Download

DELETE DOCTOR

Unesemo Id doktora kojeg zelimo obrisati I zahtjev nam vraca listu ostalih doktora. U slucaju da doktor sa tim Id-em ne postoji, javit ce gresku. Vazno je spomenuti da ne mozemo obrisati doktora koji ima neki AdmissionRecord, jer se FK nalazi u toj tabeli. Prvo moramo obrisati AdmissionRecord tog doktora, pa tek onda mozemo samog doktora. Ako pokusamo obrisati takvog doktora, dobit cemo gresku. Ovdje cemo obrisati novododanog doktora sa Id-em 6, jer on nema nikakav AdmissionRecord.



Vidimo da je doktor Bagi obrisan.

PRIJEM PACIJENATA (ADMISSION RECORD)

3. Evidenciju proizvoljnog broja prijema pacijenata, pri cemu je za svaki prijem potrebno navesti: - Datum i vrijeme prijema - Pacijenta - Nadleznog Ljekara - Hitni prijem (Da/Ne)

Prilikom evidencije prijema (zahtjev br. 3) datum i vrijeme prijema nije moguće unijeti iz proslosti tj minimalni datum i vrijeme prijema mora biti danasnji datum.

Evidentirani prijemi iz zahtjeva br. 3 trebaju biti dostupni za pregled u vidu liste (tabele) na kojoj se vidi:

- ime i prezime pacijenta
- datum i vrijeme prijema
- ime i prezime nadleznog Ljekara u formatu („IME PREZIME – SIFRA“)
- akcije za: izmjenu prijema i otkazivanje prijema

Moguće je filtrirati prikaz liste (tabele) po datumu prijema u formatu „od – do“ tj moguće je specificirati datum od kojeg i do kojeg se trebaju prikazivati prijemi.

Potrebno je prikazati indikator da se radi o hitnom prijemu

Prilikom evidentiranja prijema iz zahtjeva br. 3 moguće je odabrati samo ljekare koji su specijalisti.

Svi ovi zahtjevi su implementirani I pokazat cu to kroz primjere.

GET ALL ADMISSION RECORDS

Ovaj zahtjev vraća sve prijeme svih pacijenata.

Request URL
`https://localhost:7092/api/AdmissionRecord`

Server response

Code Details

200

Response body

```
{
  "records": [
    {
      "id": 3,
      "admittedAt": "2023-06-30T00:00:00",
      "patientName": "Wanda Vision",
      "doctorName": "Izet Fazlinović - 8888",
      "urgent": "Yes"
    },
    {
      "id": 4,
      "admittedAt": "2023-01-05T00:00:00",
      "patientName": "Rachel Green",
      "doctorName": "Teufik Arslanagić - 1221",
      "urgent": "No"
    },
    {
      "id": 5,
      "admittedAt": "2023-02-02T00:00:00",
      "patientName": "Phoebe Buffay",
      "doctorName": "Amira Dizdarević - 3313",
      "urgent": "Yes"
    }
  ],
  "success": true,
  "message": ""
}
```

Download

Vidimo sve prijeme, datum prijema je ovdje seedan u bazu, ali cemo vidjeti kasnije da datum prijema ne moze biti stariji od danasnjeg datuma. Isto tako, vidimo pacijenta, nadležnog ljekara I da li se radi o hitnom slucaju. Primjecujemo da je ime ljekara napisano onako kako se u zadatku trazi (IME PREZIME – SIFRA)

Moguće je filtrirati prijeme po datumima. Prilikom filtriranja, format datuma neka bude YYYY-MM-DD ili YYYY/MM/DD ILI MM/DD/YYYY ILI MM-DD-YYYY

Sada cemo filtrirati po datumu (od 1. Maja 2023. Do 1. Jul 2023.):

AdmissionRecord

GET /api/AdmissionRecord

Parameters

Cancel

Name	Description
startDate	05/01/2023
endDate	07/01/2023

Execute Clear

Rezultat zahtjeva ce biti svi prijemi izmedju ta dva datuma (ukljucujuci I te datume):

```
Server response
Code    Details
200
Response body
{
  "data": [
    {
      "id": 1,
      "admittedAt": "2023-05-21T00:00:00",
      "patientName": "Ted Mosby",
      "doctorName": "Teufik Arslanagić - 1221",
      "urgent": "Yes"
    },
    {
      "id": 2,
      "admittedAt": "2023-05-18T00:00:00",
      "patientName": "Barney Stinson",
      "doctorName": "Amira Dizdarević - 3313",
      "urgent": "No"
    },
    {
      "id": 3,
      "admittedAt": "2023-06-30T00:00:00",
      "patientName": "Manda Vision",
      "doctorName": "Izet Fazlinović - 8888",
      "urgent": "Yes"
    }
  ],
  "success": true,
  "message": ""
}
```

POST ADMISSION RECORD (dodaj prijem)

Ovdje je vazno naglasiti da prijem ne moze biti stariji od danasnjeg datuma. Za provjeru, unijet cemo datum prijema 12.12.2022.

Poslat cemo zahtjev sljedeceg oblika:

```
{
  "admittedAt": "2022-12-12",
  "patientId": 5,
  "doctorId": 5,
  "urgent": false
}
```

Response:

```
Server response
Code    Details
400
Undocumented Error: response status is 400
Response body
{
  "data": null,
  "success": false,
  "message": "Admitted date can't be older than today!"
}
```


Sada cemo unijeti doktora koji nije specijalista I dobit cemo gresku:

Saljemo:

```
{
  "admittedAt": "2022-12-12",
```

```
"patientId": 5,  
"doctorId": 4,  
"urgent": false  
}
```


Response:

Code	Details
400 <i>Undocumented</i>	Error: response status is 400 Response body <pre>{ "data": null, "success": false, "message": "The doctor must be a specialist!" }</pre>  Download

Sada cemo unijeti validan objekat:

```
{  
  "admittedAt": "2023-01-05",  
  "patientId": 5,  
  "doctorId": 5,  
  "urgent": false  
}
```

Response:

Server response	
Code	Details
200	Response body <pre>{ "data": { "id": 5, "admittedAt": "2023-01-05T00:00:00", "patientName": "Phoebe Buffay", "doctorName": "Izet Fazlinovic - 8888", "urgent": "No" }, "success": true, "message": "" }</pre>  Download

Vidimo da je Phoebe Buffay (pacijent sa id-em 5) bila na prijemu 05.01.2023. godine u 00:00:00, nadležni doktor je Izet Fazlinovic (specijalista sa id-em 5), te da se ne radi o hitnom prijemu.

PUT ADMISSION RECORD

Prikom update prijema, sva polja koja nisu unesena ce ostati ista ko prije. U primjeru ispod, promijenit cemo id doktora (promijenio se nadležni ljekar) I hitnost prijema. Vidjet cemo da ce se to promijeniti, a da ce netaknuta polja ostati ista. Prije toga, moramo unijeti id prijema kojeg zelimo izmijeniti. Ako


unesemo id koji ne postoji, javit ce nam gresku. Mijenjat cemo prijem pod id-em 6 (to je prijem koji smo unijeli POST metodom)

Saljemo objekat:

```
{  
  "doctorId": 1, //zelimo promijeniti nadleznog ljekara  
  "urgent": true //mijenjamo hitnost prijema  
}
```

Response:

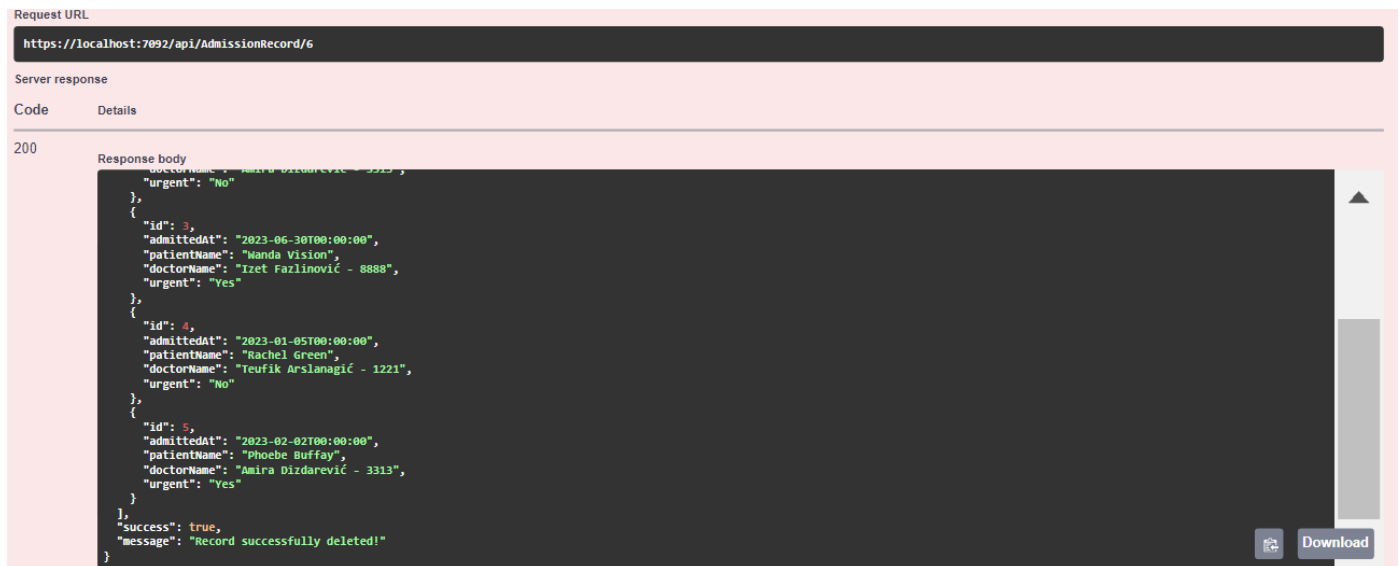
Server response

Code	Details
200	<div>Response body</div> <pre>{ "data": { "id": 6, "admittedAt": "2023-01-05T00:00:00", "patientName": "Phoebe Buffay", "doctorName": "Teufik Arslanagic - 1221", "urgent": "Yes" }, "success": true, "message": "" }</pre> <div> Download</div>

Vidimo da je nadlezni ljekar sada Teufik, a da je hitnost "Yes", a ostala netaknuta polja su ostala ista kao prije.

DELETE ADMISSION RECORD

Unesemo Id prijema kojeg zelimo obrisati I zahtjev nam vraca listu ostalih prijema. U slucaju da prijem sa tim Id-em ne postoji, javit ce gresku. Ovdje cemo obrisati novododani prijem pod ID-em 6



Request URL
`https://localhost:7092/api/AdmissionRecord/6`

Server response

Code	Details
200	<p>Response body</p> <pre>{ "doctorName": "Amira Dizdarević - 3313", "urgent": "No" }, { "id": 3, "admittedAt": "2023-06-30T00:00:00", "patientName": "Wanda Vision", "doctorName": "Izet Fazlinović - 8888", "urgent": "Yes" }, { "id": 4, "admittedAt": "2023-01-05T00:00:00", "patientName": "Rachel Green", "doctorName": "Teufik Arslanagić - 1221", "urgent": "No" }, { "id": 5, "admittedAt": "2023-02-02T00:00:00", "patientName": "Phoebe Buffay", "doctorName": "Amira Dizdarević - 3313", "urgent": "Yes" }], "success": true, "message": "Record successfully deleted!" }</pre>

Vidimo da je prijem sa id-em 6 uspjesno obrisani

NALAZI (MEDICAL FINDING RECORD)

4. Evidenciju nalaza za evidentirani prijem pacijenta. Prilikom unosa nalaza potrebno ja za dati prijem unijeti sljedece podatke:

- Tekstualni opis nalaza
- Datum i vrijeme kreiranja nalaza

Potrebno je osmisliti na koji nacin se vrsi pisanje nalaza za dati prijem i isti implementirati. Datum i vrijeme nalaza ne unosi korisnik vec se sistemski dodjeljuje.

Ideja za pisanje nalaza je sljedeca:

Korisnik unese ID pacijenta I ID prijema (pod uslovom da taj pacijent mora pripadati tom prijemu, ako nije tako dobit cemo gresku). Korisnik unese opis nalaza, te se prikaze taj opis I podaci o prijemu trazelog pacijenta.

GET ALL MEDICAL FINDING RECORDS

Vraca sve nalaze za sve pacijente. Ispod vidimo kako ti objekti izgledaju.

Server response

Code	Details
200	<div>Response body</div> <pre>{ "id": 3, "description": "The patient complains of back pain. it is necessary to take a spine scan", "createdAt": "2023-01-03T10:00:17.5086964", "admissionRecord": { "id": 3, "admittedAt": "2023-06-30T00:00:00", "patientName": "Wanda Vision", "doctorName": "Izet Fazlinović - 8888", "urgent": "Yes" } }, { "id": 4, "description": "Sore throat for 10 days. Drink as much tea as possible and Tylool hot.", "createdAt": "2023-01-03T10:00:17.5086965", "admissionRecord": { "id": 4, "admittedAt": "2023-01-05T00:00:00", "patientName": "Rachel Green", "doctorName": "Teufik Arslanagić - 1221", "urgent": "No" } } }, "success": true, "message": "" }</pre> <div>Download</div>

POST MEDICAL FINDING RECORD

Dodaje nalaz za određenog pacijenta. Korisnik unese ID pacijenta I ID prijema (pod uslovom da taj pacijent mora pripadati tom prijemu, ako nije tako dobit cemo gresku). Korisnik unese opis nalaza, te se prikaze taj opis I podaci o prijemu traženog pacijenta.

Probat cemo unijeti nalaz, ali da ovdje pacijent ne pripada datom prijemu:

Objekat:

```
{
  "description": "Ma hej odlican ti je nalaz",
  "patientId": 1,
  "admissionRecordId": 5
}
```

Pacijentu Ted Mosby smo pokusali dodijeliti prijem od pacijenta Phoebe Buffay, te nam Response javlja gresku:

Server response


Code	Details
400 <i>Undocumented</i>	<div>Error: response status is 400</div> <div>Response body</div> <pre>{ "data": null, "success": false, "message": "That admission record doesn't belong to that patient!" }</pre> <div>Download</div>

Bitno je naglasiti da su u ovoj POST metodi sva polja obavezna za unijeti. Dakle, opis nalaza ne može biti prazan string ili da sadrži samo razmake, niti može biti null.

Sada ćemo poslati ispravan objekat. Napisat ćemo nalaz za Rachel Green (id pacijenta je 4), a njen prijem je kod doktora Teufika (Id prijema je 4):

```
{  
  "description": "Ma hej odlican ti je nalaz",  
  "patientId": 4,  
  "admissionRecordId": 4  
}
```

Response:



The screenshot shows a REST client interface with the following details:

- Request URL:** `https://localhost:7092/api/MedicalFindingRecord`
- Server response:**
 - Code:** 200
 - Details:** Response body
- Response body (JSON):**

```
{  
  "data": {  
    "id": 7,  
    "description": "Ma hej odlican ti je nalaz",  
    "createdAt": "2023-01-03T12:39:03.9699529Z",  
    "admissionRecord": {  
      "id": 4,  
      "admittedAt": "2023-01-05T00:00:00",  
      "patientName": "Rachel Green",  
      "doctorName": "Teufik Arslanagic - 1221",  
      "urgent": "No"  
    }  
  },  
  "success": true,  
  "message": ""  
}
```
- Buttons:** Copy icon and Download button.

Vidimo da je nalaz uspješno napisan, te da je datum i vrijeme nalaza danasnji datum i trenutno vrijeme(to sistem ispise).


PUT MEDICAL FINDING RECORD

Kod update nalaza, može se promijeniti samo opis samog nalaza. I taj opis je obavezno polje. Isto tako, opis ne može biti prazan ili da sadrži samo razmake. Prije toga, moramo unijeti id nalaza koji želimo prepraviti. Unijet ćemo id novog nalaza (id mu je 7).

Napomena: Vidjela sam da se prilikom update nalaza ne promijeni datum kreiranja nalaza, pa sam to naknadno uradila. Ispod ću pokazati novi nalaz, prvo kako je izgledao prije PUT metode, a onda nakon nje.

Prije put metode:

Server response


Code	Details
200	<div>Response body</div> <pre>{ "data": { "id": 6, "description": "Nalaz je ocaj", "createdAt": "2023-01-03T13:19:46.4417654Z", "admissionRecord": { "id": 2, "admittedAt": "2023-05-18T00:00:00", "patientName": "Barney Stinson", "doctorName": "Amira Dizdarević - 3313", "urgent": "No" } }, "success": true, "message": "" }</pre> <div> Download</div>

Nakon put metode (mijenjamo opis nalaza, a datum se automatski mijenja):

Request URL

https://localhost:7092/api/MedicalFindingRecord?id=8

Server response

Code	Details
200	<div>Response body</div> <pre>{ "data": { "id": 8, "description": "ODLICAN NALAZ", "createdAt": "2023-01-03T13:21:08.6712013Z", "admissionRecord": { "id": 2, "admittedAt": "2023-05-18T00:00:00", "patientName": "Barney Stinson", "doctorName": "Amira Dizdarević - 3313", "urgent": "No" } }, "success": true, "message": "" }</pre> <div> Download</div>

Vidimo da je promijenjen opis nalaza I datum nalaza.

GET MEDICAL FINDING RECORD FOR SINGLE PATIENT

Ovdje unosimo id pacijenta, a zahtjev nam vraća sve nalaze vezane za tog pacijenta. Unijet ćemo id pacijenta Rachel Green, te će nam vratiti sve njene nalaze. U slučaju da unesemo id pacijenta koji ne postoji, dobit ćemo gresku.

Request URL

https://localhost:7092/api/MedicalFindingRecord/4

Server response

Code Details

200

Response body

```
{
  "data": [
    {
      "id": 4,
      "description": "Sore throat for 10 days. Drink as much tea as possible and Tylo1 hot.",
      "createdAt": "2023-01-03T10:00:17.5086965",
      "admissionRecord": {
        "id": 4,
        "admittedAt": "2023-01-05T00:00:00",
        "patientName": "Rachel Green",
        "doctorName": "Teufik Arslanagić - 1221",
        "urgent": "No"
      }
    },
    {
      "id": 7,
      "description": "Ipak možda ima malih komplikacija",
      "createdAt": "2023-01-03T12:39:03.9699529",
      "admissionRecord": {
        "id": 4,
        "admittedAt": "2023-01-05T00:00:00",
        "patientName": "Rachel Green",
        "doctorName": "Teufik Arslanagić - 1221",
        "urgent": "No"
      }
    }
  ],
  "success": true,
}
```

Download

Vidimo sve nalaze za pacijenta Rachel Green.

DELETE MEDICAL FINDING RECORD

Brise nalaz ciji id unesemo. Ako unesemo id nalaza koji ne postoji, dobit cemo gresku.

Request URL

https://localhost:7092/api/MedicalFindingRecord/10

Server response

Code	Details
404	Error: response status is 404

Response body

```
{
  "data": null,
  "success": false,
  "message": "Medical record with that id doesn't exist!"
}
```

Download

Sada cemo obrisati nalaz koji postoji (nalaz sa id-em 7):

Request URL

https://localhost:7092/api/MedicalFindingRecord/7

Server response

Code	Details
200	

Response body

```
{
  "id": 7,
  "description": "Sore throat for 10 days. Drink as much tea as possible and Tylo1 hot.",
  "createdAt": "2023-01-03T10:00:17.5006965",
  "admissionRecord": {
    "id": 4,
    "admittedAt": "2023-01-05T00:00:00",
    "patientName": "Rachel Green",
    "doctorName": "Teufik Arslanagic - 1221",
    "urgent": "No"
  }
},
{
  "id": 6,
  "description": " oke ",
  "createdAt": "2023-01-03T12:37:49.1083495",
  "admissionRecord": {
    "id": 5,
    "admittedAt": "2023-05-21T00:00:00",
    "patientName": "Ted Mosby",
    "doctorName": "Teufik Arslanagic - 1221",
    "urgent": "Yes"
  }
}
},
{
  "success": true,
  "message": "Record deleted!"
}
```

Download

Vidimo da je nalaz uspješno obrisani.