Software Engineering
Field of Electronics and Computers
Electrical and Computer Engineering
AUTh

8th Semester
Spring 2022

MedBook

Searching for a doctor made easy.

User Requirements and System Specifications

Version 1.0

Gerogiannis Konstantinos 9638 konsgero@ece.auth.gr

Gerontopoulos Anastasios 9682 ganastas@ece.auth.gr

Mparmpounakis Konstantinos 9759 kmparmpou@ece.auth.gr

Antoniadis Prodromos 9911 piantoni@ece.auth.gr

25/04/2022

Table of Contents

List of Figures

1. Design Patterns

1.1 Proxy design pattern

In order to satisfy NFR-2 (i.e., NFR-2: The system must comply with the current data protection regulations set forth by GDPR), the Proxy design pattern is used. This pattern provides the required security and reliability to the system. Specifically, by implementing the Protection Proxy, the ProxyAppointment class allows access to RealAppointment objects only for authenticated users. A RealAppointment object is accessible only by system administrators and the doctor and patient associated with that specific appointment (for example, authentication can be achieved through the properties appPatientID, appDoctorID, and an isAdmin function for the mentioned user categories). Additionally, access rights vary depending on the type of user accessing the object. Furthermore, the Virtual Proxy can also be applied so that an image object is displayed until the RealAppointment is loaded from the database.

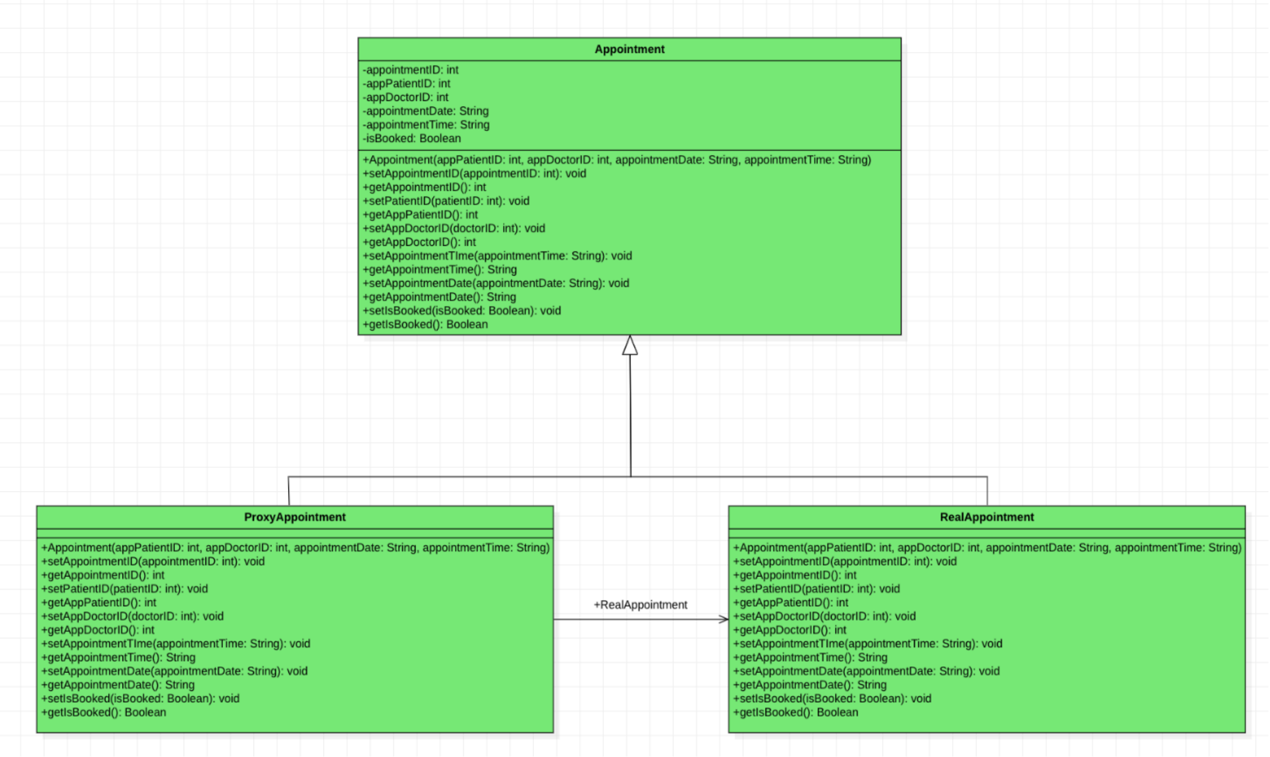

Appointment
-appointmentID: int
-appPatientID: int
-appDoctorID: int
-appointmentDate: String
-appointmentTime: String
-isBooked: Boolean
+Appointment(appPatientID: int, appDoctorID: int, appointmentDate: String, appointmentTime: String)
+setAppointmentID(appointmentID: int): void
+getAppointmentID(): int
+setPatientID(patientID: int): void
+getAppPatientID(): int
+setAppDoctorID(doctorID: int): void
+getAppDoctorID(): int
+setAppointmentTIme(appointmentTime: String): void
+getAppointmentTime(): String
+setAppointmentDate(appointmentDate: String): void
+getAppointmentDate(): String
+setIsBooked(isBooked: Boolean): void
+getIsBooked(): Boolean
ProxyAppointment
+Appointment(appPatientID: int, appDoctorID: int, appointmentDate: String, appointmentTime: String)
+setAppointmentID(appointmentID: int): void
+getAppointmentID(): int
+setPatientID(patientID: int): void
+getAppPatientID(): int
+setAppDoctorID(doctorID: int): void
+getAppDoctorID(): int
+setAppointmentTIme(appointmentTime: String): void
+getAppointmentTime(): String
+setAppointmentDate(appointmentDate: String): void
+getAppointmentDate(): String
+setIsBooked(isBooked: Boolean): void
+getIsBooked(): Boolean
+RealAppointment
RealAppointment
+Appointment(appPatientID: int, appDoctorID: int, appointmentDate: String, appointmentTime: String)
+setAppointmentID(appointmentID: int): void
+getAppointmentID(): int
+setPatientID(patientID: int): void
+getAppPatientID(): int
+setAppDoctorID(doctorID: int): void
+getAppDoctorID(): int
+setAppointmentTIme(appointmentTime: String): void
+getAppointmentTime(): String
+setAppointmentDate(appointmentDate: String): void
+getAppointmentDate(): String
+setIsBooked(isBooked: Boolean): void
+getIsBooked(): Boolean

Figure 1: Proxy design pattern

1.2 Abstract Factory design pattern

In order to satisfy NFR-2 (i.e., NFR-2: The system must operate without issues on every device running Windows 7 or a more recent version of the operating system and Android 8 or a more recent version), the Abstract Factory design pattern is employed. With this pattern, our system is capable of being supported by a family of products. The client software creates concrete implementations of the abstract class UserDeviceFactory, and depending on which implementation we have, a different version of createUserDevice() is called to create a concrete implementation of the abstract UserDevice. This pattern also offers us design extensibility so that if we want our system to be supported by a new device, this can be achieved by creating the corresponding concrete classes.

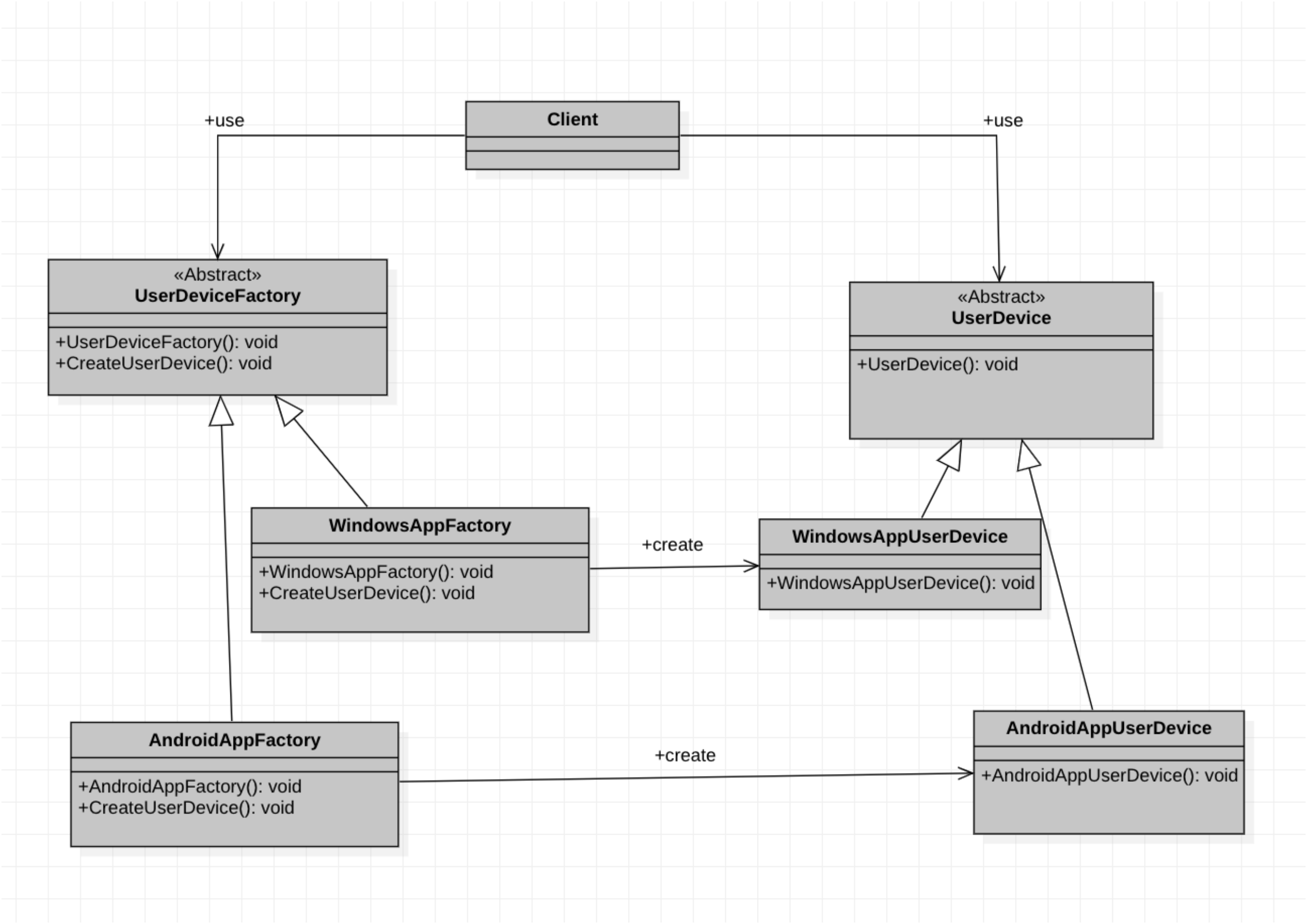

Client
+use
+use
«Abstract»
UserDeviceFactory
+UserDeviceFactory(): void
+CreateUserDevice(): void
«Abstract»
UserDevice
+UserDevice(): void
WindowsAppFactory
+WindowsAppFactory(): void
+CreateUserDevice(): void
+create
WindowsAppUserDevice
+WindowsAppUserDevice(): void
AndroidAppFactory
+AndroidAppFactory(): void
+CreateUserDevice(): void
+create
AndroidAppUserDevice
+AndroidAppUserDevice(): void

Figure 2: Abstract Factory Design Pattern

1.3 Observer design pattern

In this case, a behavioral design pattern is used to satisfy a functional requirement (i.e., FR-3: The notification system must be able to send notifications to the user). Specifically, whenever the status of an appointment is updated (e.g., change in time or date), the users (Patient and Doctor) involved in that appointment are notified. This is achieved through the `notifyUsers` function, which calls the `displayNotificationMessage` function for each user. The `displayNotificationMessage` function, in turn, creates the corresponding `PatientNotificationMessageGUI` or `DoctorNotificationMessageGUI` object and redirects the user to a screen where a relevant message about the appointment status change is displayed (the message varies depending on the user type). In the event that one of the involved users is offline, the `NotificationMessageGUI` screen (and consequently the notification message) will be displayed to the user as soon as they reconnect to the application.

Note: The classes mentioned above may also include other properties and methods, which are omitted as we focus on the functionality of the pattern.

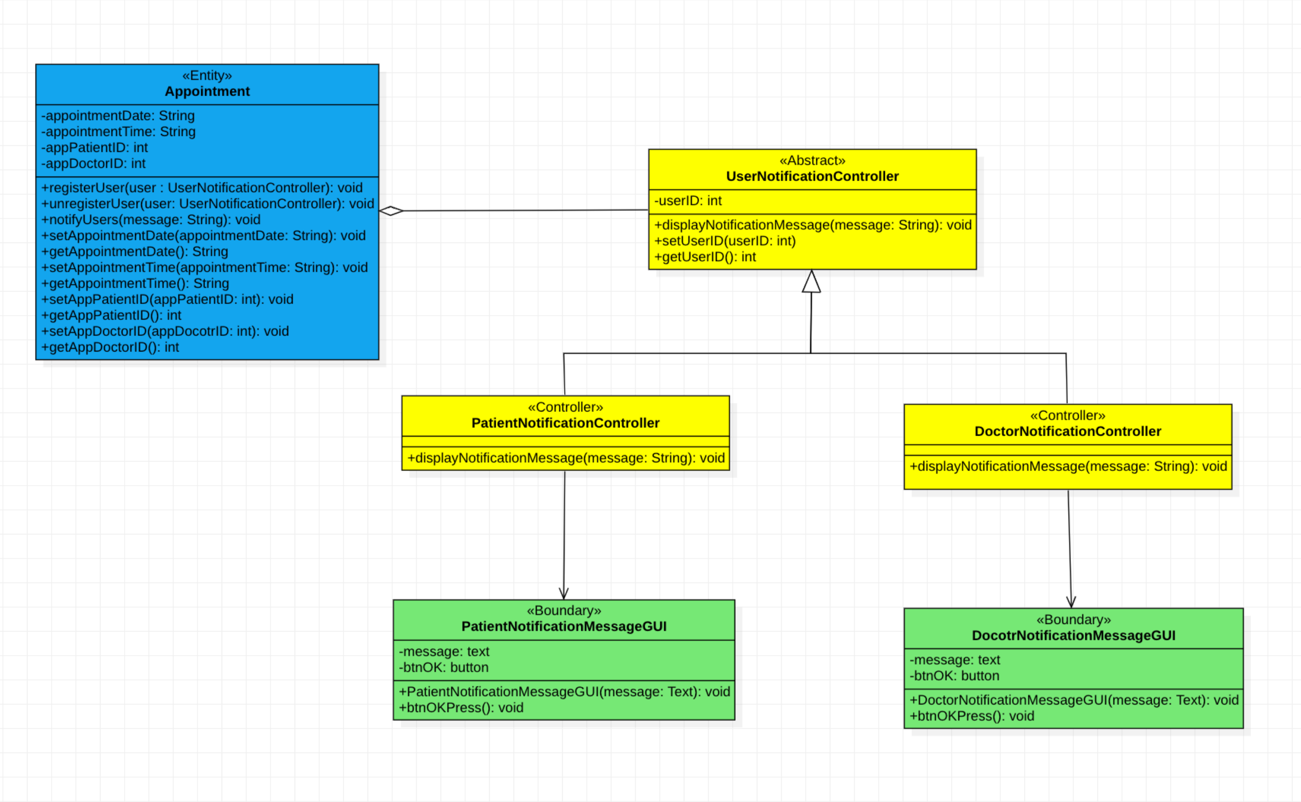

«Entity»
Appointment
-appointmentDate: String
-appointmentTime: String
-appPatientID: int
-appDoctorID: int
+registerUser(user : UserNotificationController): void
+unregisterUser(user: UserNotificationController): void
+notifyUsers(message: String): void
+setAppointmentDate(appointmentDate: String): void
+getAppointmentDate(): String
+setAppointmentTime(appointmentTime: String): void
+getAppointmentTime(): String
+setAppPatientID(appPatientID: int): void
+getAppPatientID(): int
+setAppDoctorID(appDocotrID: int): void
+getAppDoctorID(): int
«Abstract»
UserNotificationController
-userID: int
+displayNotificationMessage(message: String): void
+setUserID(userID: int)
+getUserID(): int
«Controller»
PatientNotificationController
+displayNotificationMessage(message: String): void
«Controller»
DoctorNotificationController
+displayNotificationMessage(message: String): void
«Boundary»
PatientNotificationMessageGUI
-message: text
-btnOK: button
+PatientNotificationMessageGUI(message: Text): void
+btnOKPress(): void
«Boundary»
DocotrNotificationMessageGUI
-message: text
-btnOK: button
+DoctorNotificationMessageGUI(message: Text): void
+btnOKPress(): void

Figure 3: Observer Design Pattern

2. System Architecture

- API in Swaggerhub

2.1 System resources detection

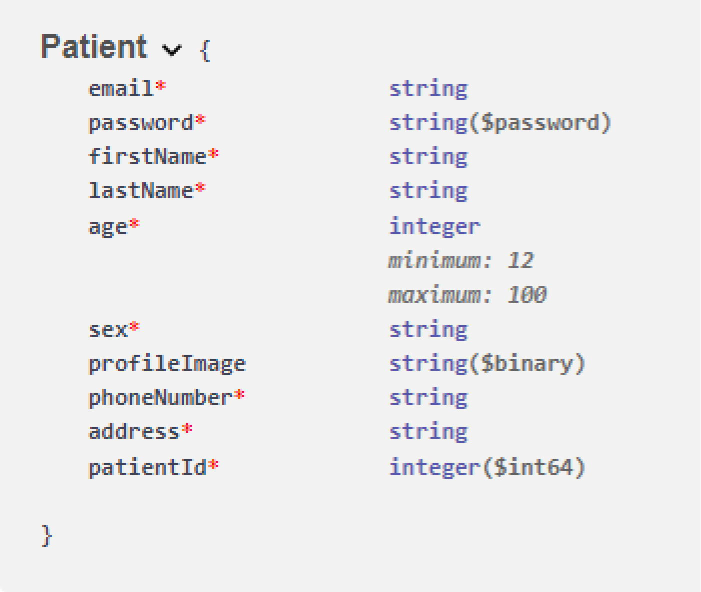
Based on the principles of REST architecture, identify the Resources of your system (one-off, data, and algorithmic - as defined in the lecture "Advanced Types of Client-Server Architecture: REST") using the classes you recognized in the previous deliverable and record them in a table format as follows (for the sake of summary).

BEC class	REST resource	Endpoints (HTTP Verbs)
Patient	/patient	POST
Patient	/patient/{patientId}	POST, PUT
Doctor	/doctor	POST
Doctor	/doctor/{doctorId}	POST, PUT
Appointment, (Patient)	/appointments/patient/{patientId}	POST, GET
Appointment, (Patient)	/appointments/patient/{patientId}/{appointmentId}	DELETE
Appointment, (Doctor)	/appointments/doctor/{doctorId}	PUT, GET
Appointment, (Doctor)	/appointments/doctor/{doctorId}/{appointmentId}	PUT
Review, (Doctor)	/reviews/{doctorId}	POST, GET
Report	/reports	POST
None	/search	GET
(Patient)	/notifications/{patientId}	POST, GET, DELETE
(Doctor)	/notifications/{doctorId}	POST, GET, DELETE

2.2 REST Interface Documentation

2.2.1 Resource Patient

2.2.1.1 Patient data model



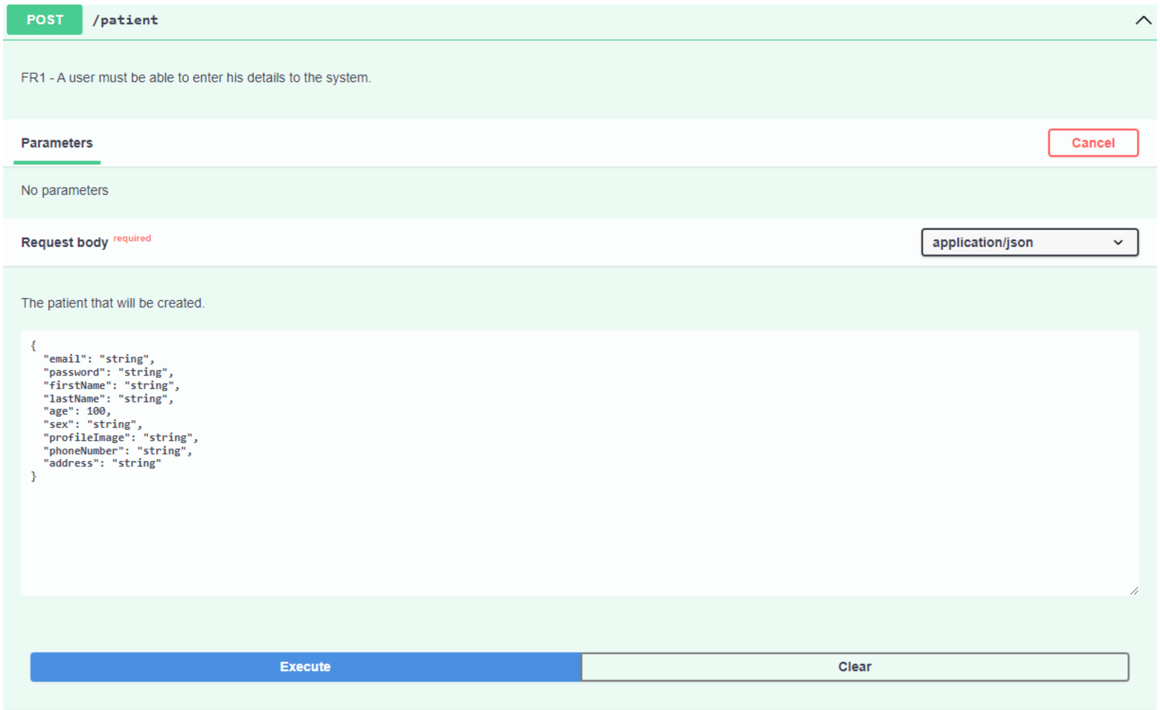
```
Patient ⌄ {
    email*                  string
    password*               string($password)
    firstName*              string
    lastName*               string
    age*                    integer
                            minimum: 12
                            maximum: 100

    sex*                    string
    profileImage            string($binary)
    phoneNumber*            string
    address*                string
    patientId*              integer($int64)

}
```

Figure 4: Patient data model

2.2.1.2 Endpoint POST of resource Patient



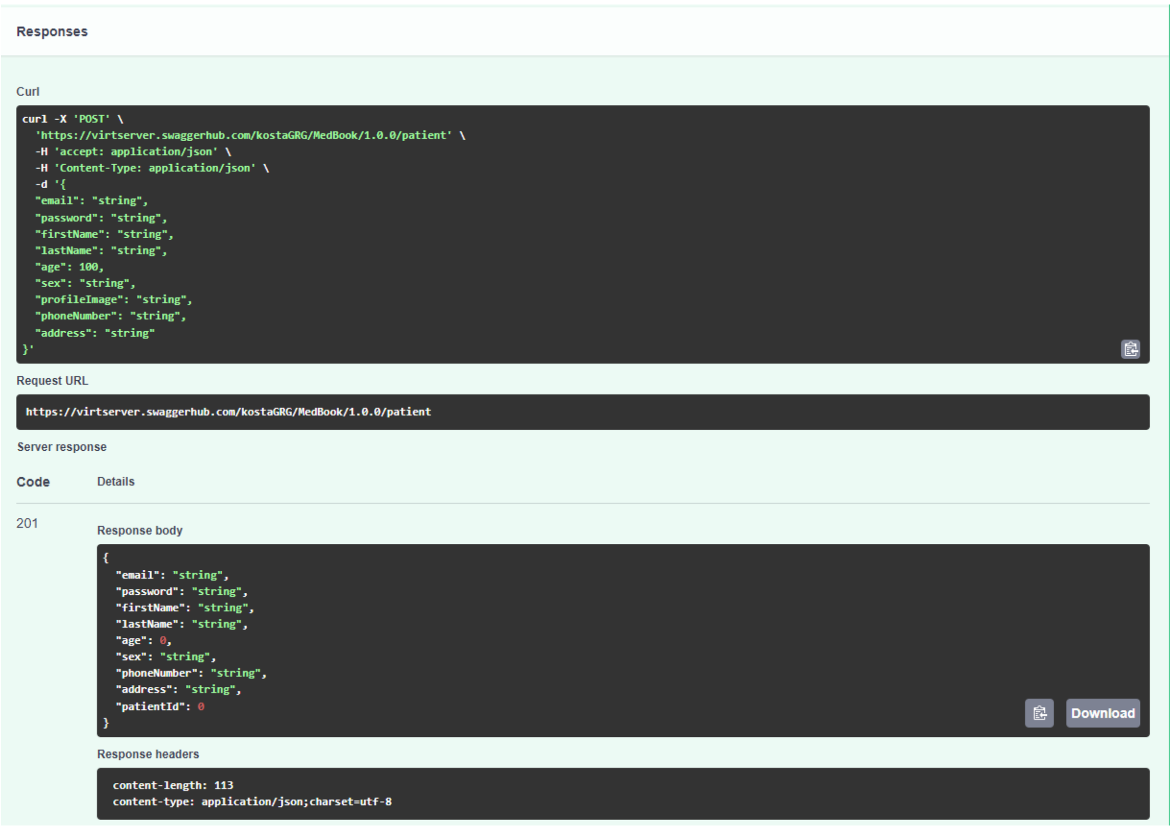
FR1 - A user must be able to enter his details to the system.

No parameters

The patient that will be created.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string"
}
```

Figure 5: Explanation and presentation of the resource parameters.



Responses

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string"
}'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient
```

Server response

Code | Details

201

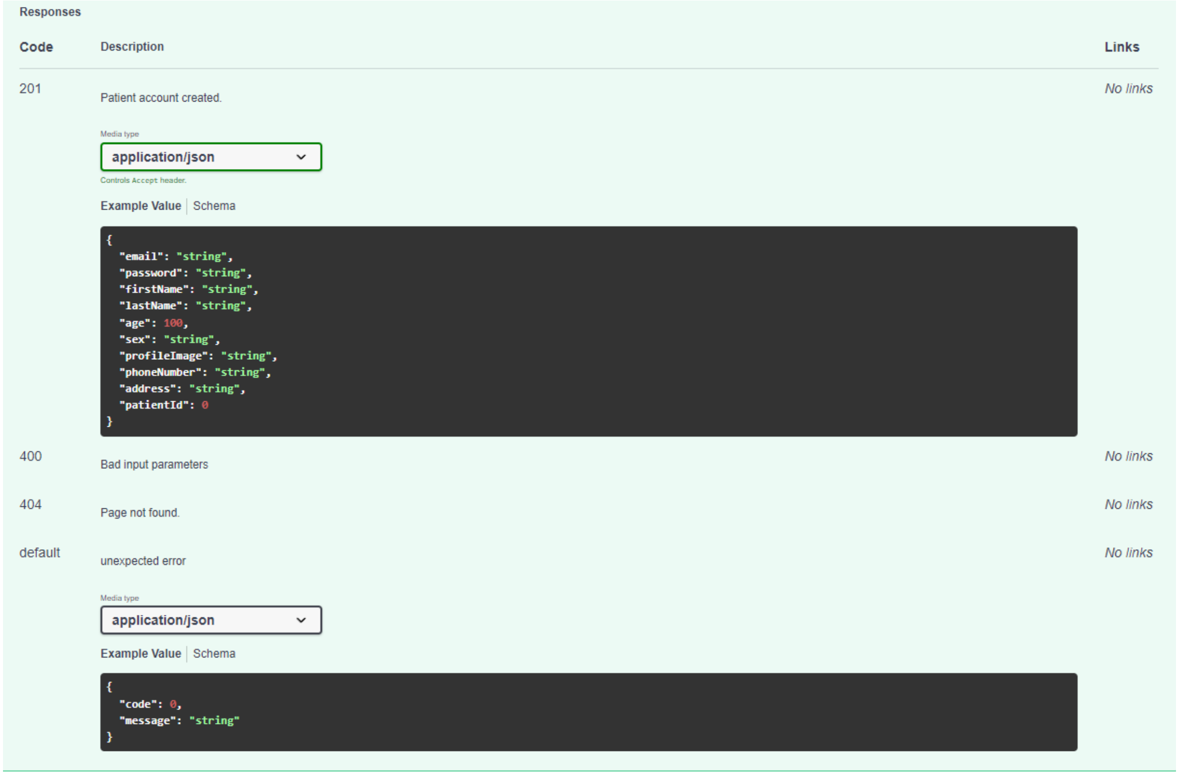
Response body

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 0,
  "sex": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 0
}
```

Response headers

```
content-length: 113
content-type: application/json;charset=utf-8
```

Figure 6: endpoint testing



Responses

Code | Description | Links

201

Patient account created.

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 0
}
```

No links

400 Bad input parameters — No links

404 Page not found. — No links

default unexpected error — No links

Media type

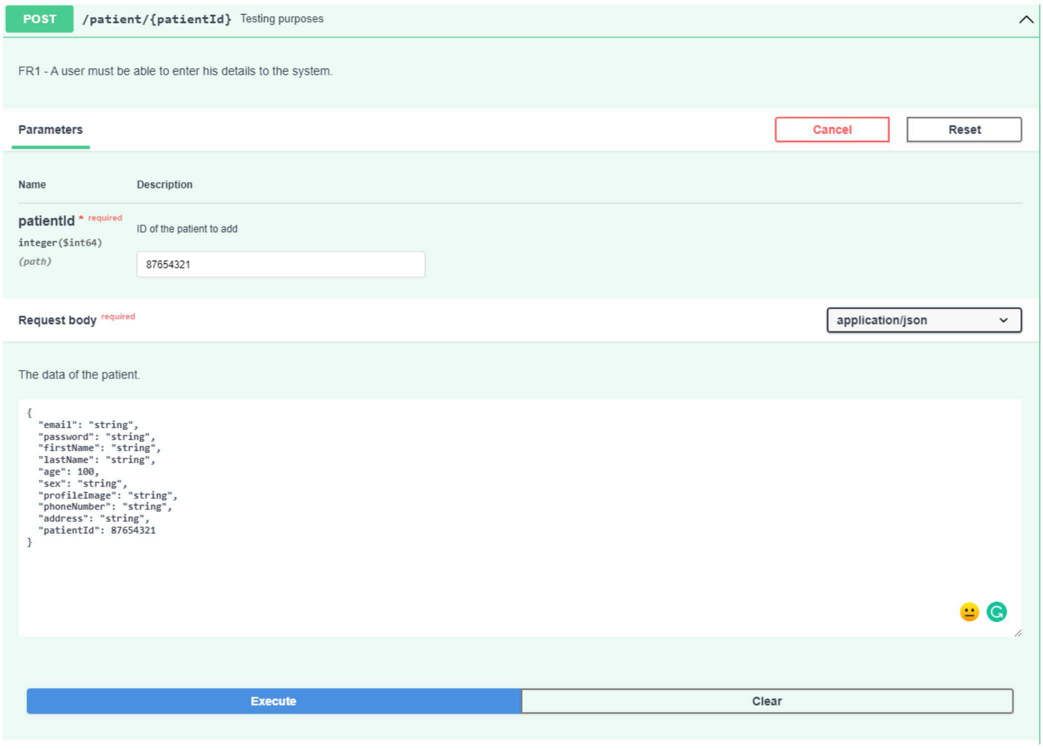
application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 7: System responses

2.2.1.3 Endpoint POST of resource Patient, for specific Patient Id



/patient/{patientId} Testing purposes

FR1 - A user must be able to enter his details to the system.

Parameters

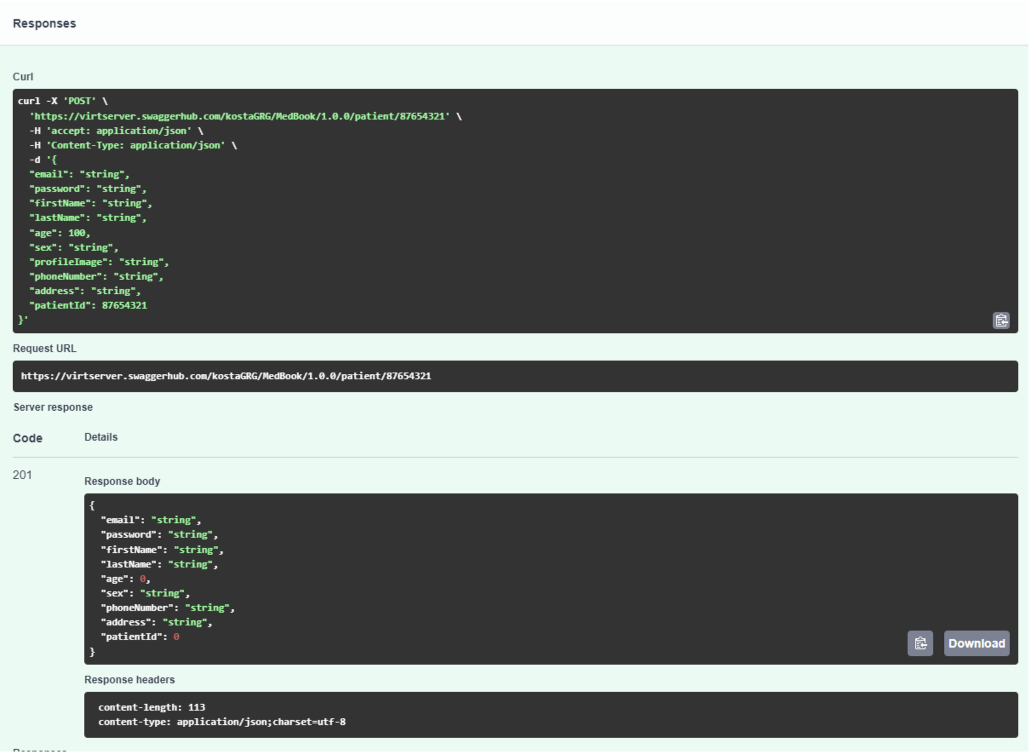
Name	Description
patientId * required integer($int64) *(path)*	ID of the patient to add 87654321

Request body required — application/json

The data of the patient.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 87654321
}
```

Figure 8: Explanation and presentation of the resource parameters.



Responses

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient/87654321' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 87654321
}'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient/87654321
```

Server response

Code | Details

201

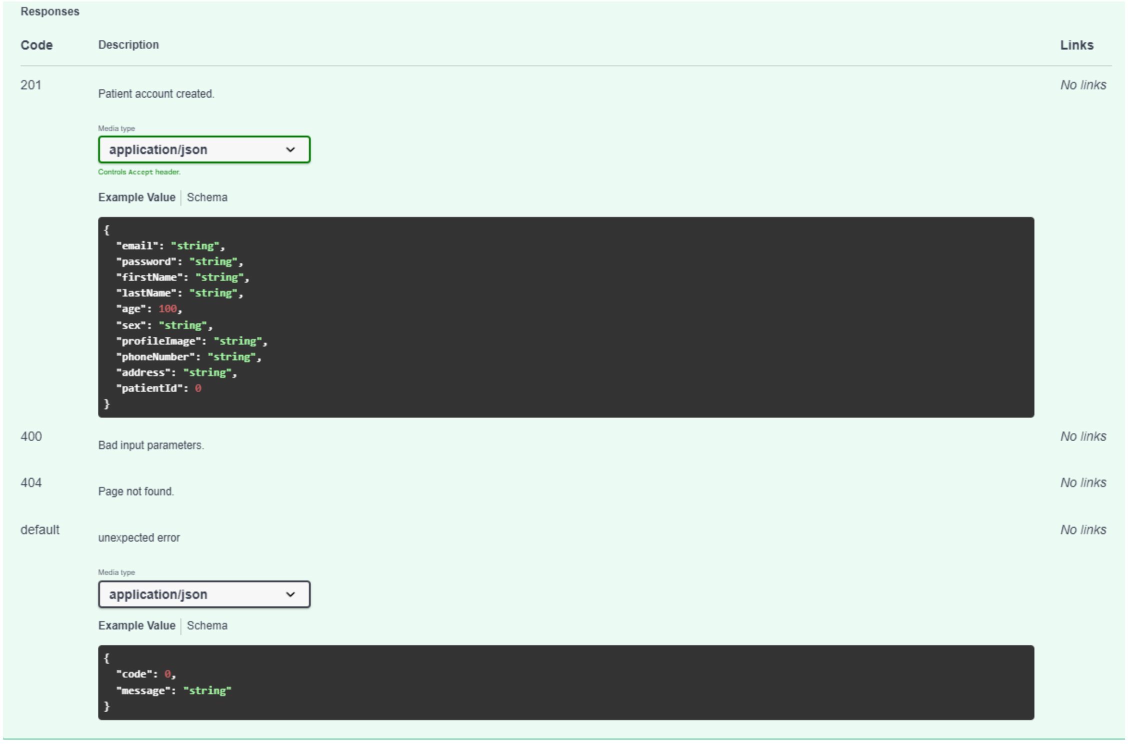
Response body

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 0,
  "sex": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 0
}
```

Response headers

```
content-length: 113
content-type: application/json;charset=utf-8
```

Figure 9: endpoint testing



Responses

Code | Description | Links

201

Patient account created.

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 0
}
```

No links

400

Bad input parameters.

No links

404

Page not found.

No links

default

unexpected error

No links

Media type

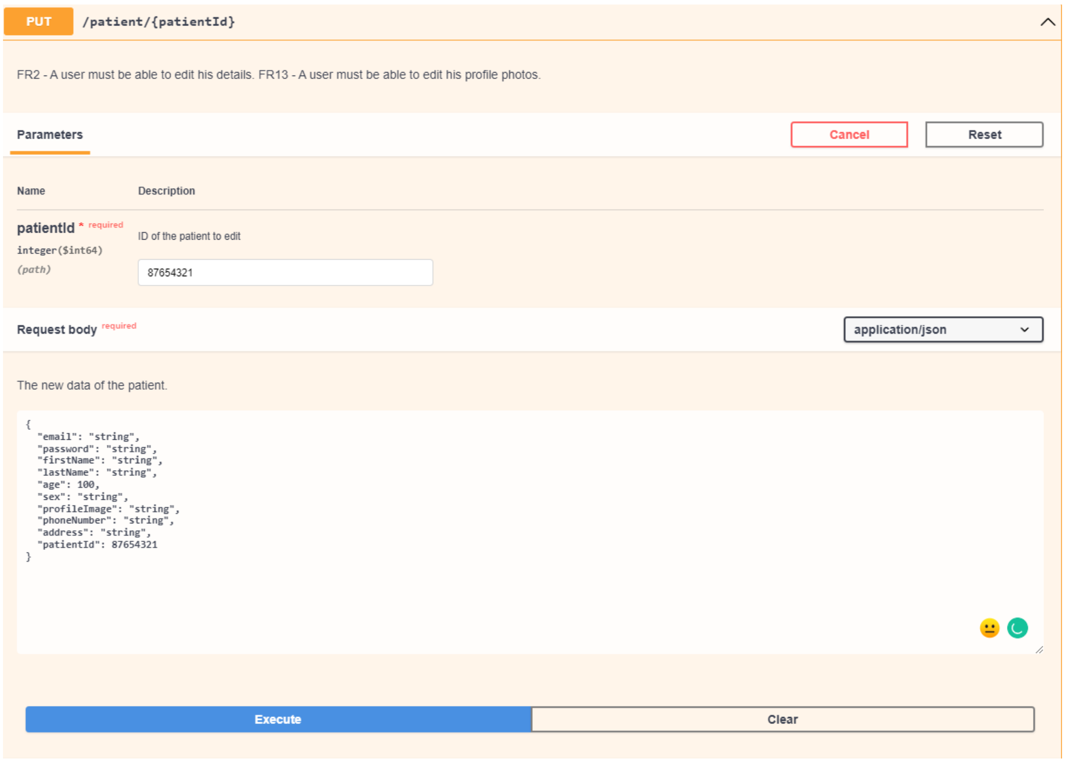
application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 10: System responses

2.2.1.4 Endpoint PUT of resource Patient, for specific Patient Id



FR2 - A user must be able to edit his details. FR13 - A user must be able to edit his profile photos.

patientId * required
integer($int64)
(path)

ID of the patient to edit

87654321

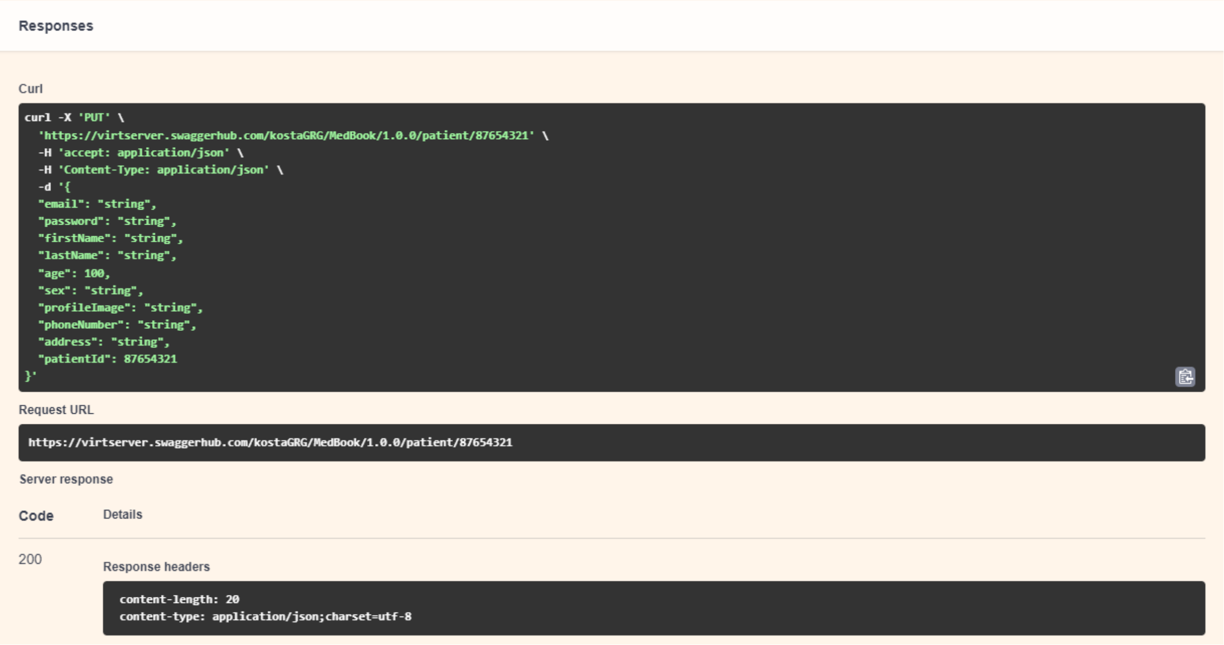
Request body required

application/json

The new data of the patient.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 87654321
}
```

Figure 11: Explanation and presentation of the resource parameters



Curl

```
curl -X 'PUT' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient/87654321' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "patientId": 87654321
}'
```

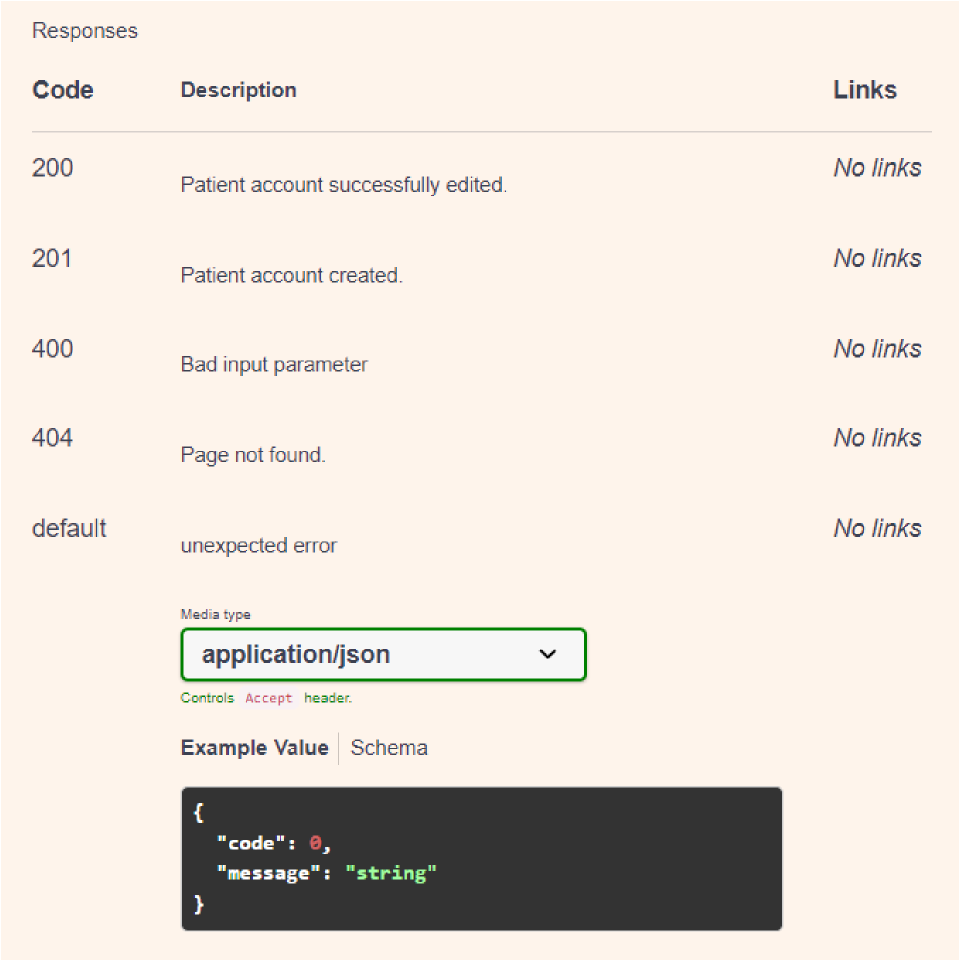
Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/patient/87654321
```

Server response

Code	Details
200	Response headers content-length: 20 content-type: application/json;charset=utf-8

Figure 12: endpoint testing



Responses

Code	Description	Links
200	Patient account successfully edited.	*No links*
201	Patient account created.	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

Media type

application/json

Controls `Accept` header.

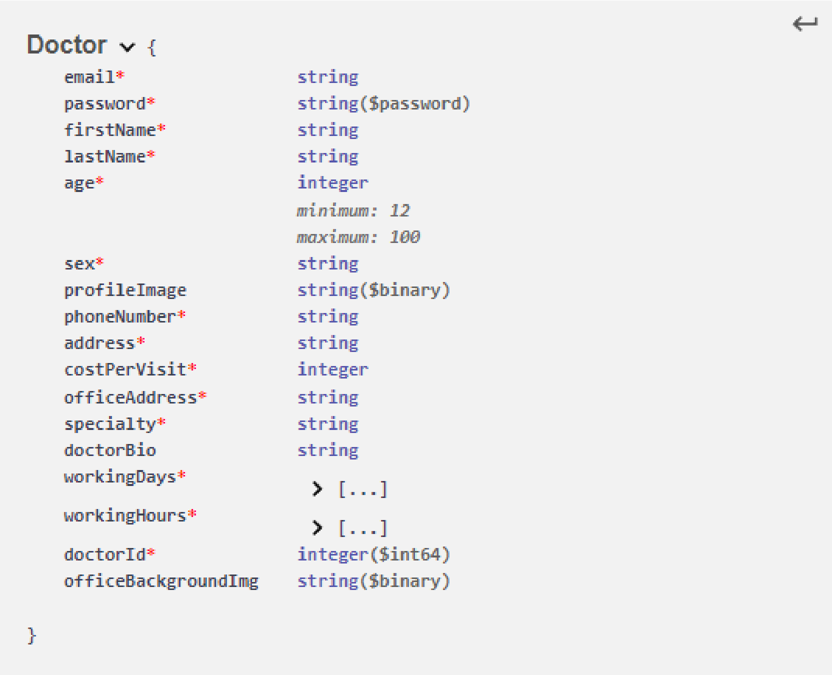
Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 13: System responses

2.2.2 Resource Doctor

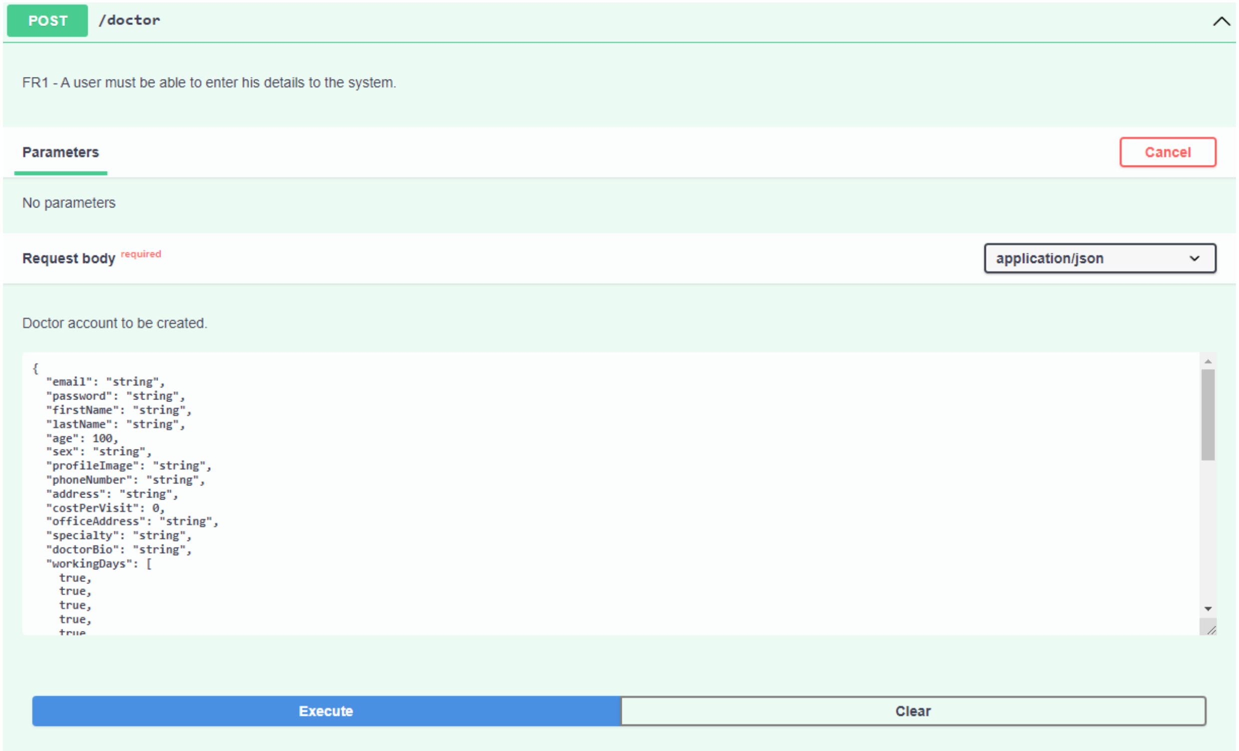
2.2.2.1 Doctor data model



```
Doctor ⌄ {
    email*                 string
    password*              string($password)
    firstName*             string
    lastName*              string
    age*                   integer
                           minimum: 12
                           maximum: 100
    sex*                   string
    profileImage           string($binary)
    phoneNumber*           string
    address*               string
    costPerVisit*          integer
    officeAddress*         string
    specialty*             string
    doctorBio              string
    workingDays*             > [...]
    workingHours*            > [...]
    doctorId*              integer($int64)
    officeBackgroundImg    string($binary)
}
```

Figure 14: Doctor data model

2.2.2.2 Endpoint POST of resource Doctor



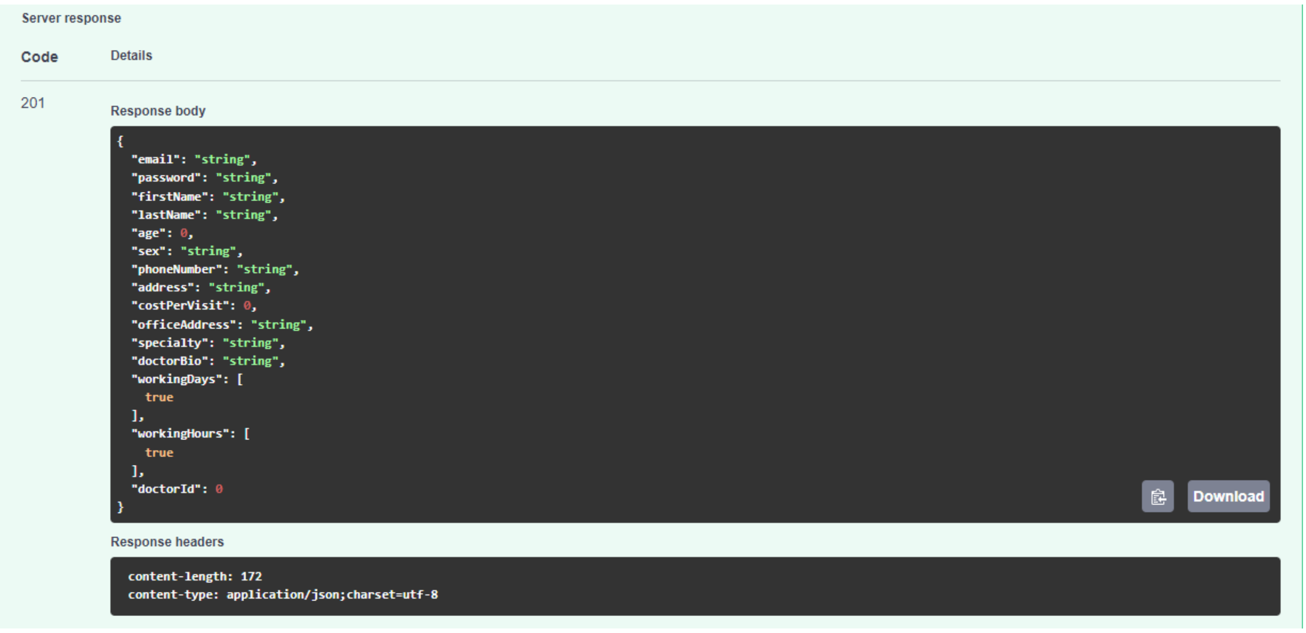
FR1 - A user must be able to enter his details to the system.

No parameters

Doctor account to be created.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true
```

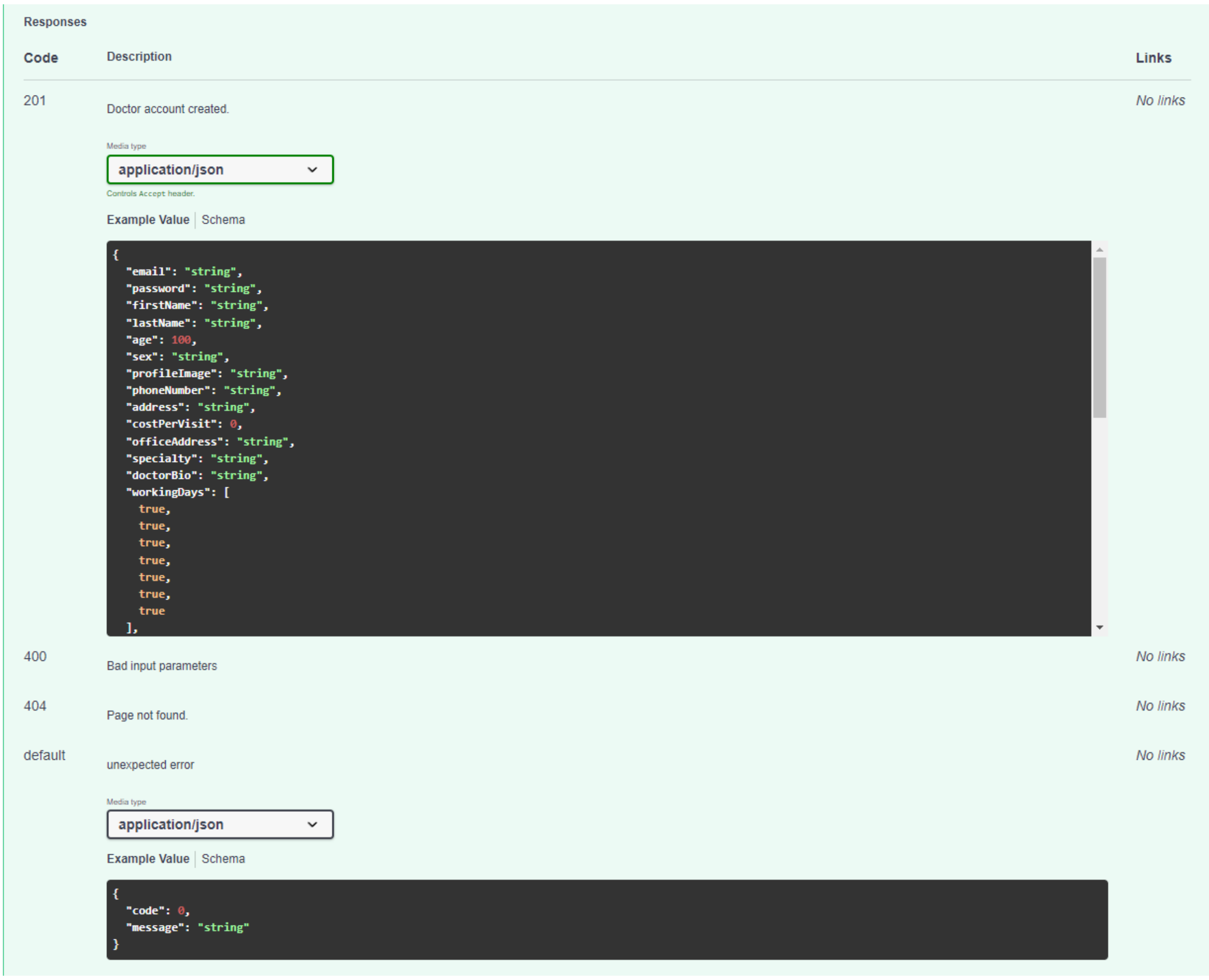
Figure 15: Explanation and presentation of the resource parameters.



```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 0,
  "sex": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true
  ],
  "workingHours": [
    true
  ],
  "doctorId": 0
}
```

```
content-length: 172
content-type: application/json;charset=utf-8
```

Figure 16: endpoint testing



Responses

Code	Description	Links
201	Doctor account created.	*No links*

Media type
application/json
Controls Accept header.

Example Value | Schema

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true,
    true,
    true
  ],
```

Code	Description	Links
400	Bad input parameters	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

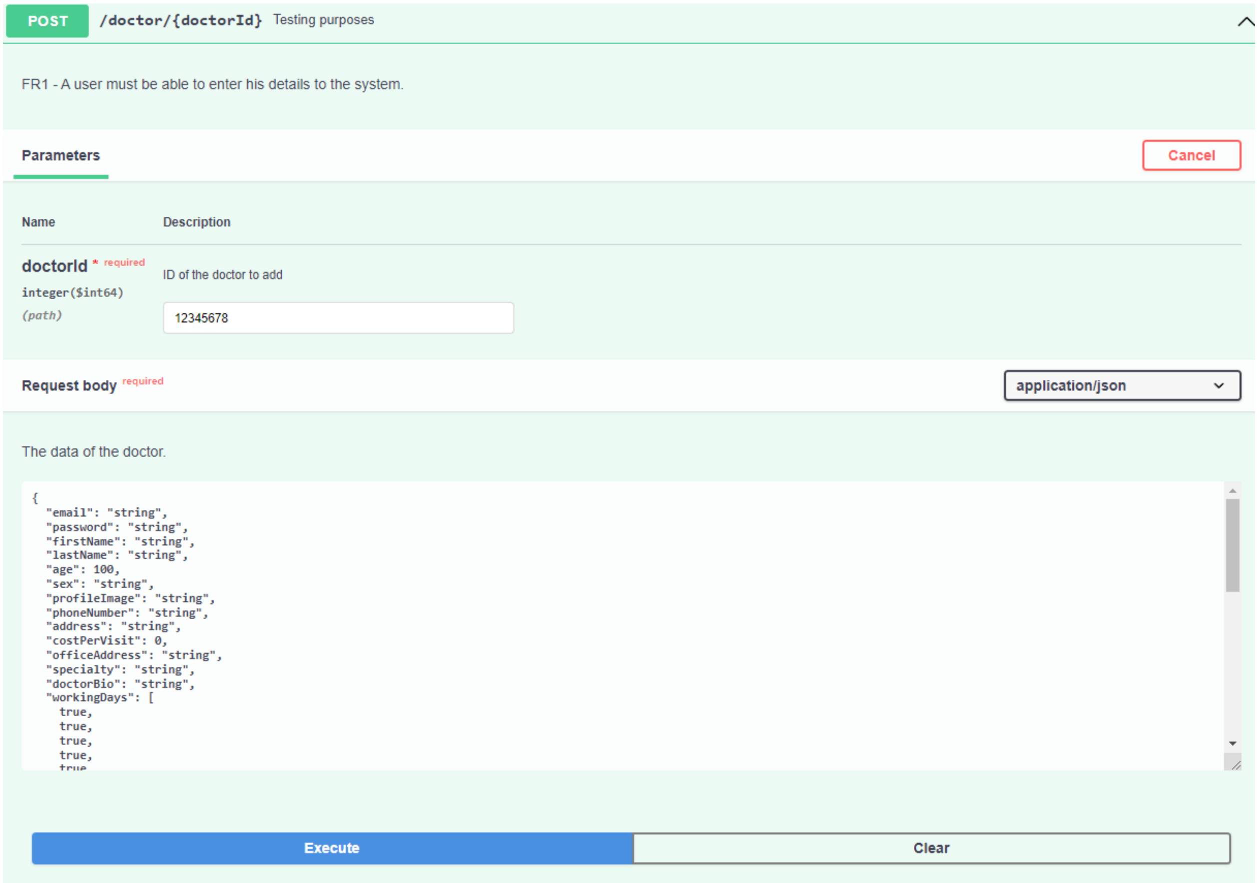
Media type
application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 17: System responses

2.2.2.3 Endpoint POST of resource Doctor, for specific doctorId



FR1 - A user must be able to enter his details to the system.

doctorId * required
integer($int64)
(path)

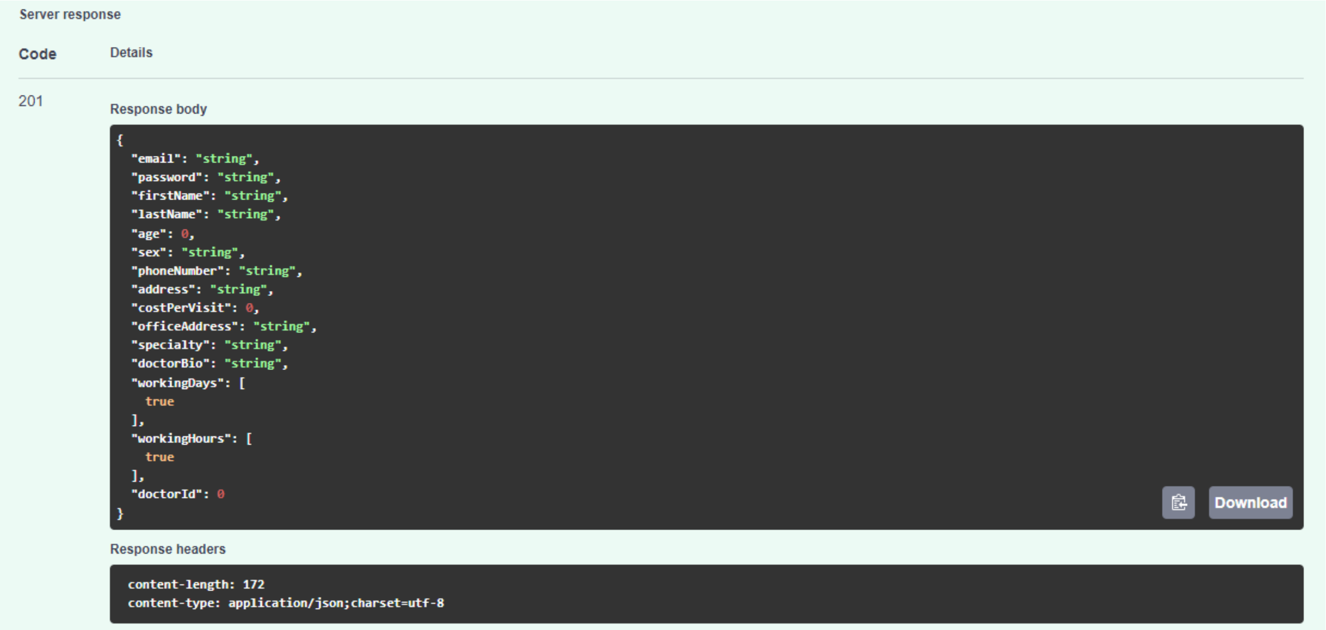
ID of the doctor to add

12345678

The data of the doctor.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true
```

Figure 18: Explanation and presentation of the resource parameters.



201

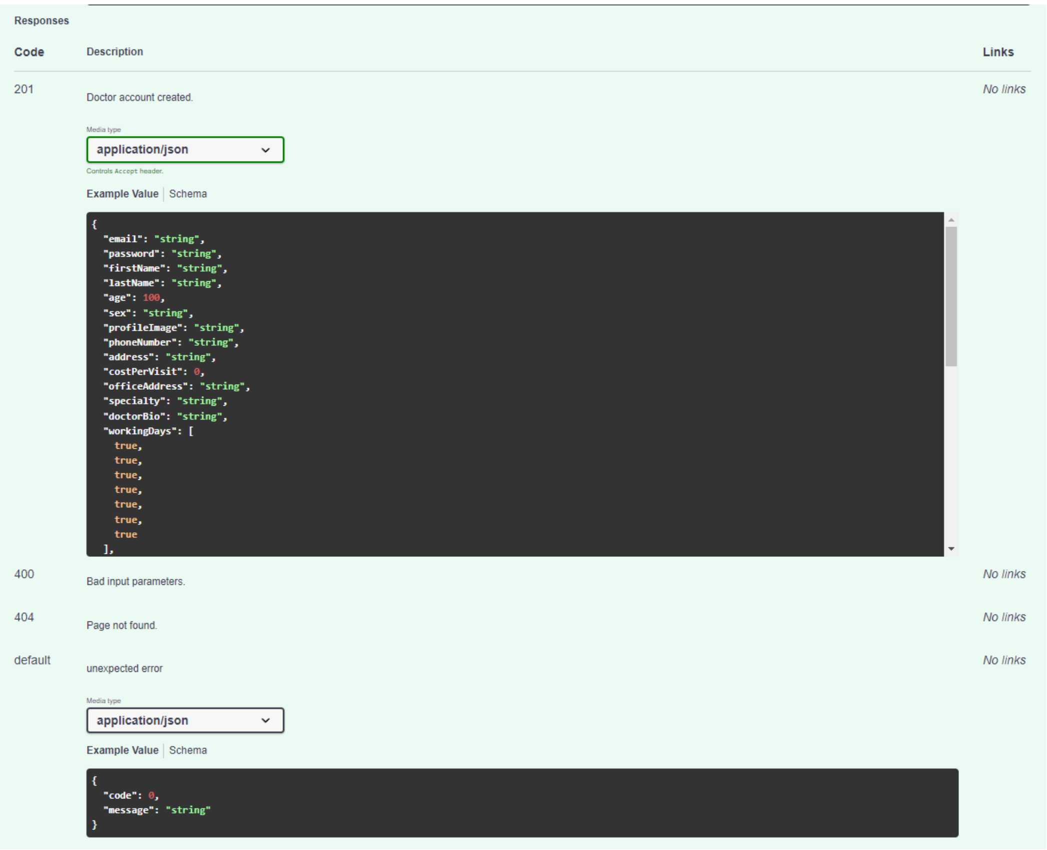
Response body

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 0,
  "sex": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true
  ],
  "workingHours": [
    true
  ],
  "doctorId": 0
}
```

Response headers

```
content-length: 172
content-type: application/json;charset=utf-8
```

Figure 19: endpoint testing



Responses

Code Description Links

201 Doctor account created. No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true,
    true,
    true
  ],
```

400 Bad input parameters. No links

404 Page not found. No links

default unexpected error No links

Media type

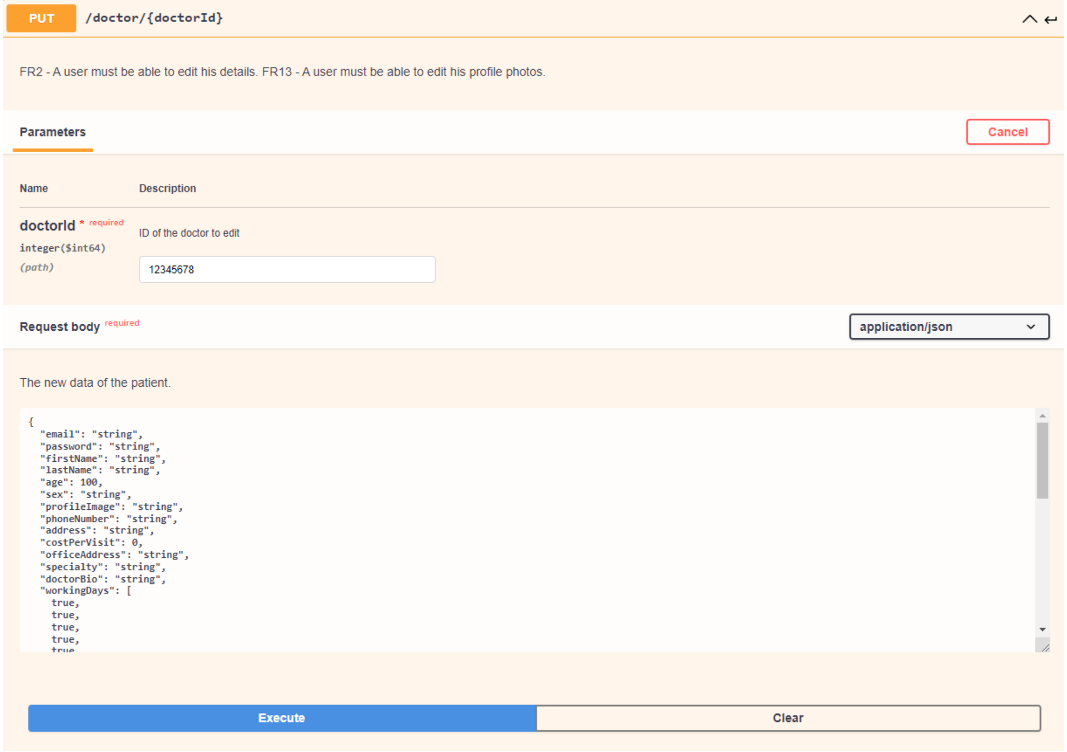
application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 20: System responses

2.2.2.4 Endpoint PUT of resource Doctor, for specific doctorId



PUT /doctor/{doctorId}

FR2 - A user must be able to edit his details. FR13 - A user must be able to edit his profile photos.

Parameters

Cancel

Name	Description
doctorId * required integer($int64) (path)	ID of the doctor to edit 12345678

Request body required

application/json

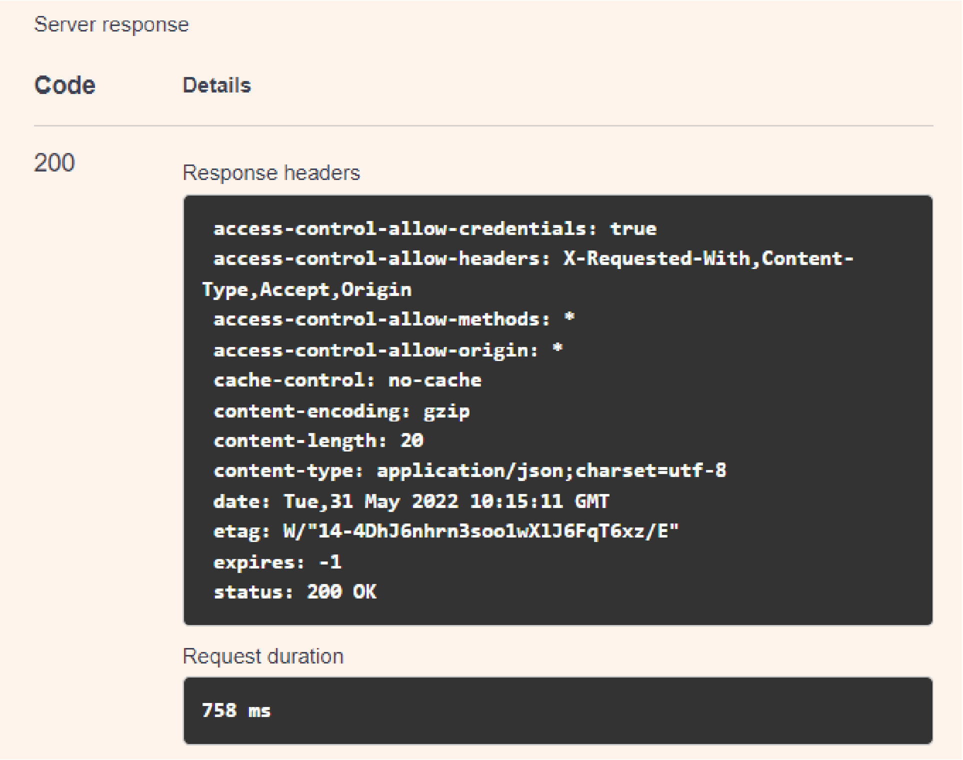
The new data of the patient.

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string",
  "age": 100,
  "sex": "string",
  "profileImage": "string",
  "phoneNumber": "string",
  "address": "string",
  "costPerVisit": 0,
  "officeAddress": "string",
  "specialty": "string",
  "doctorBio": "string",
  "workingDays": [
    true,
    true,
    true,
    true,
    true
```

Execute

Clear

Figure 21: Explanation and presentation of the resource parameters



Server response

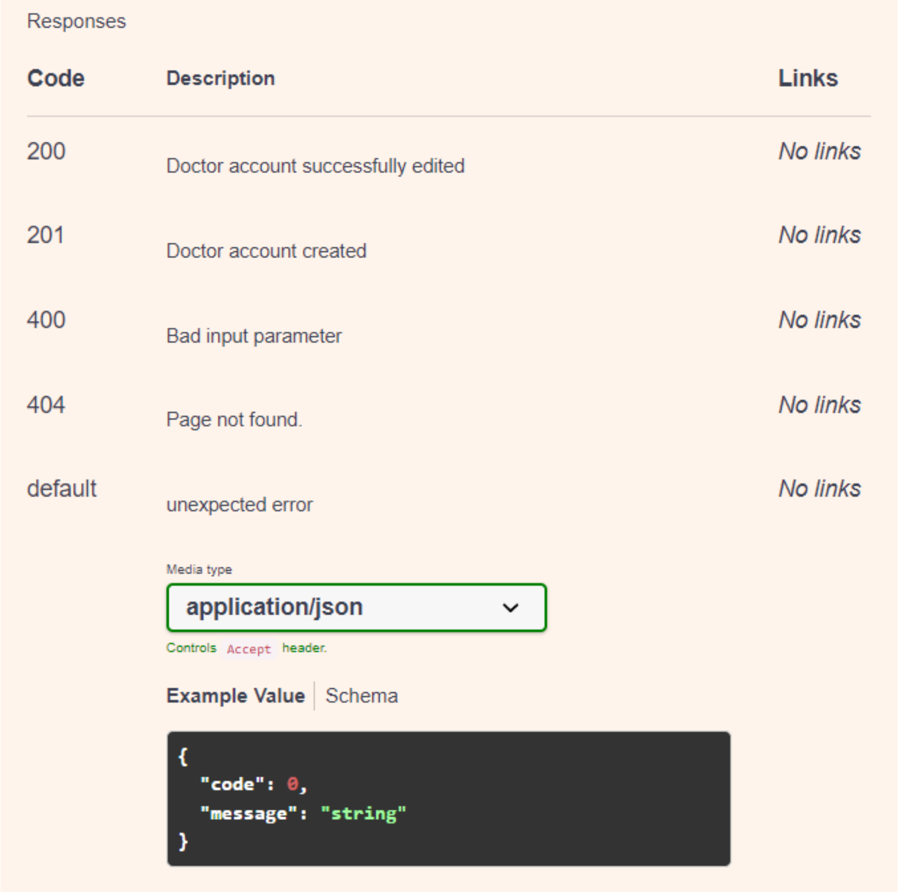
Code	Details
200	Response headers

```
 access-control-allow-credentials: true
 access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin
 access-control-allow-methods: *
 access-control-allow-origin: *
 cache-control: no-cache
 content-encoding: gzip
 content-length: 20
 content-type: application/json;charset=utf-8
 date: Tue,31 May 2022 10:15:11 GMT
 etag: W/"14-4DhJ6nhrn3soo1wX1J6FqT6xz/E"
 expires: -1
 status: 200 OK
```

Request duration

```
758 ms
```

Figure 22: endpoint testing



Responses

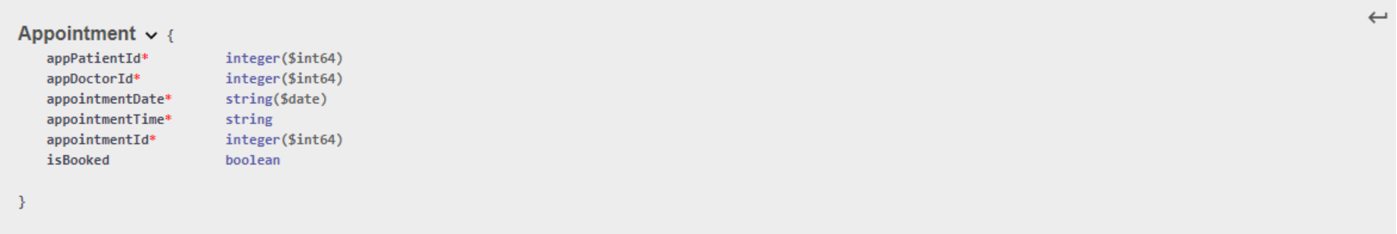
Code	Description	Links
200	Doctor account successfully edited	*No links*
201	Doctor account created	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

```
{
  "code": 0,
  "message": "string"
}
```

Figure 22: System responses

2.2.3 Resource Appointment

2.2.3.1 Appointment data model

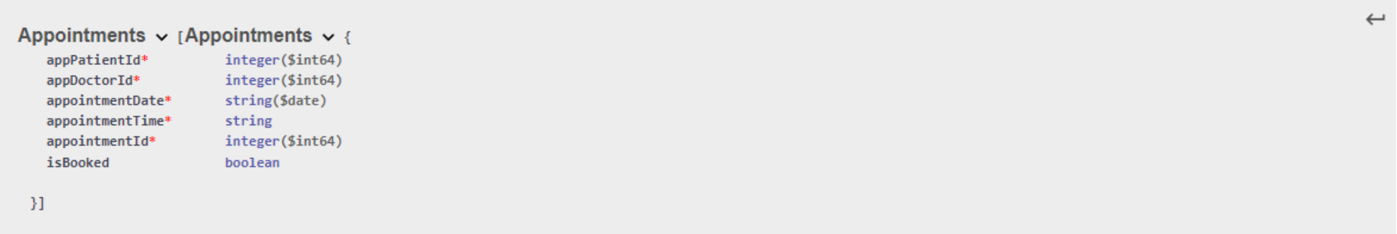


```
Appointment ⌄ {
    appPatientId*        integer($int64)
    appDoctorId*         integer($int64)
    appointmentDate*     string($date)
    appointmentTime*     string
    appointmentId*       integer($int64)
    isBooked             boolean

}
```

Figure 23: Appointment data model

2.2.3.2 Appointments data model (List of objects Appointment)

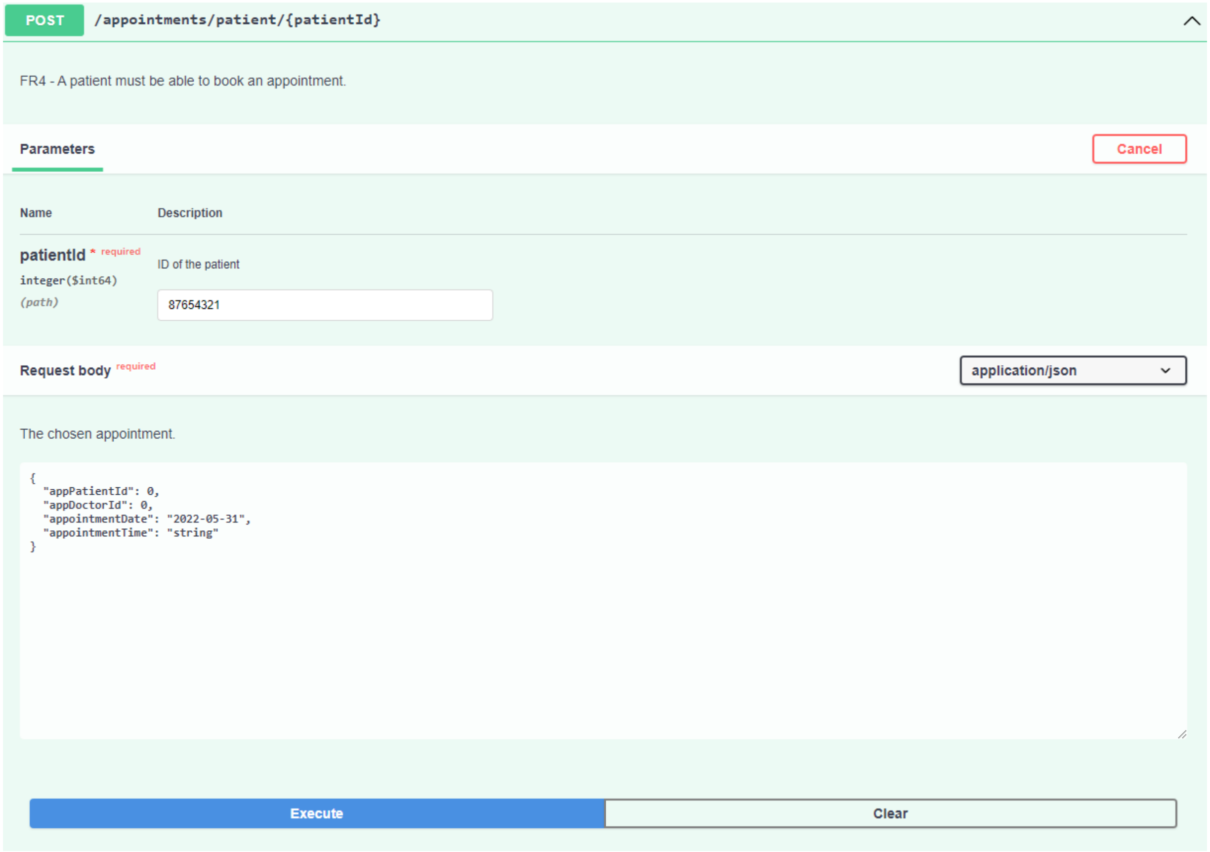


```
Appointments ⌄ [Appointments ⌄ {
    appPatientId*        integer($int64)
    appDoctorId*         integer($int64)
    appointmentDate*     string($date)
    appointmentTime*     string
    appointmentId*       integer($int64)
    isBooked             boolean

 }]
```

Figure 24: Appointments data model

2.2.3.3 Endpoint POST of resource appointments, for patient and specific patientId



FR4 - A patient must be able to book an appointment.

Name	Description
patientId * required integer($int64) (path)	ID of the patient 87654321

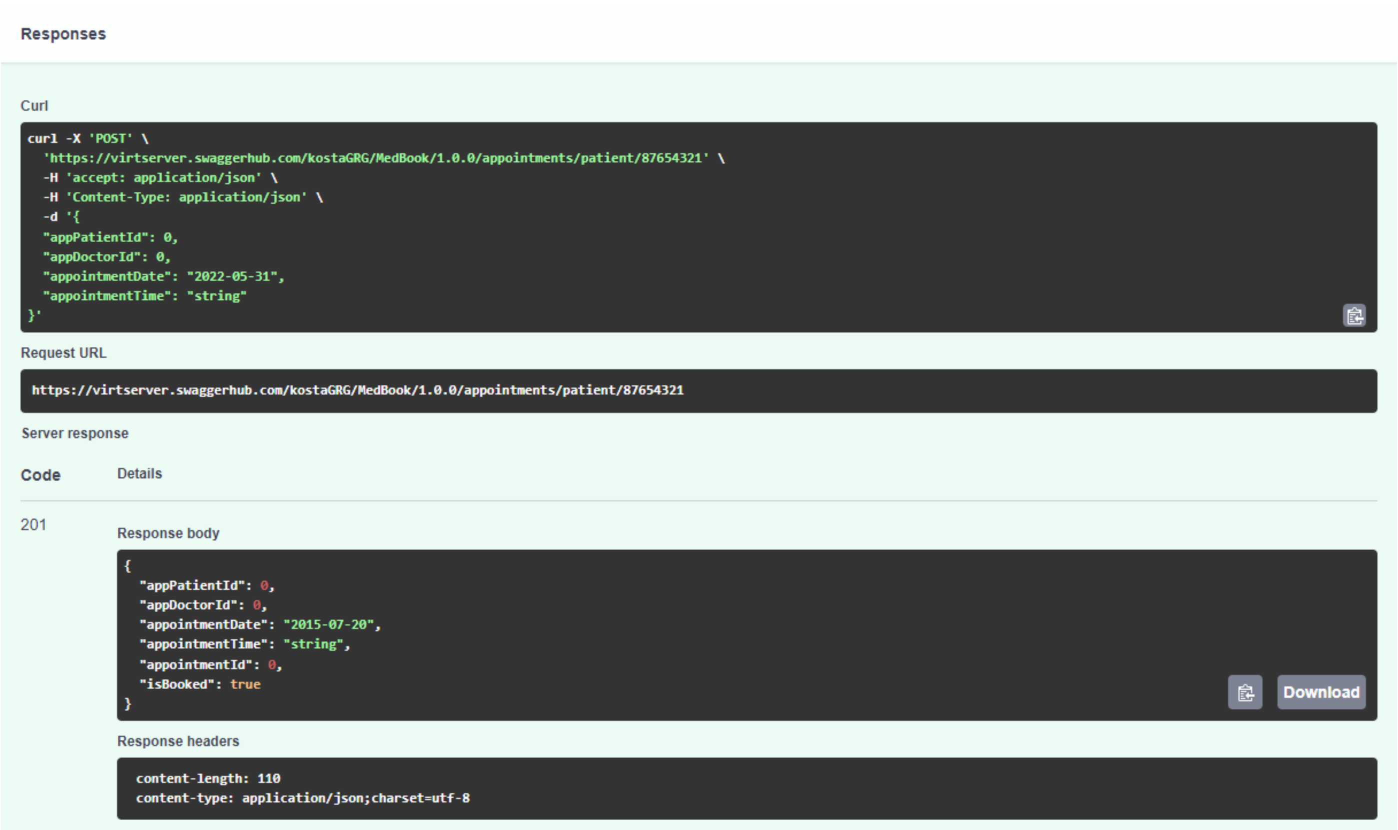
Request body required

application/json

The chosen appointment.

```
{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2022-05-31",
  "appointmentTime": "string"
}
```

Figure 25: Explanation and presentation of the resource parameters



Responses

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2022-05-31",
  "appointmentTime": "string"
}'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321
```

Server response

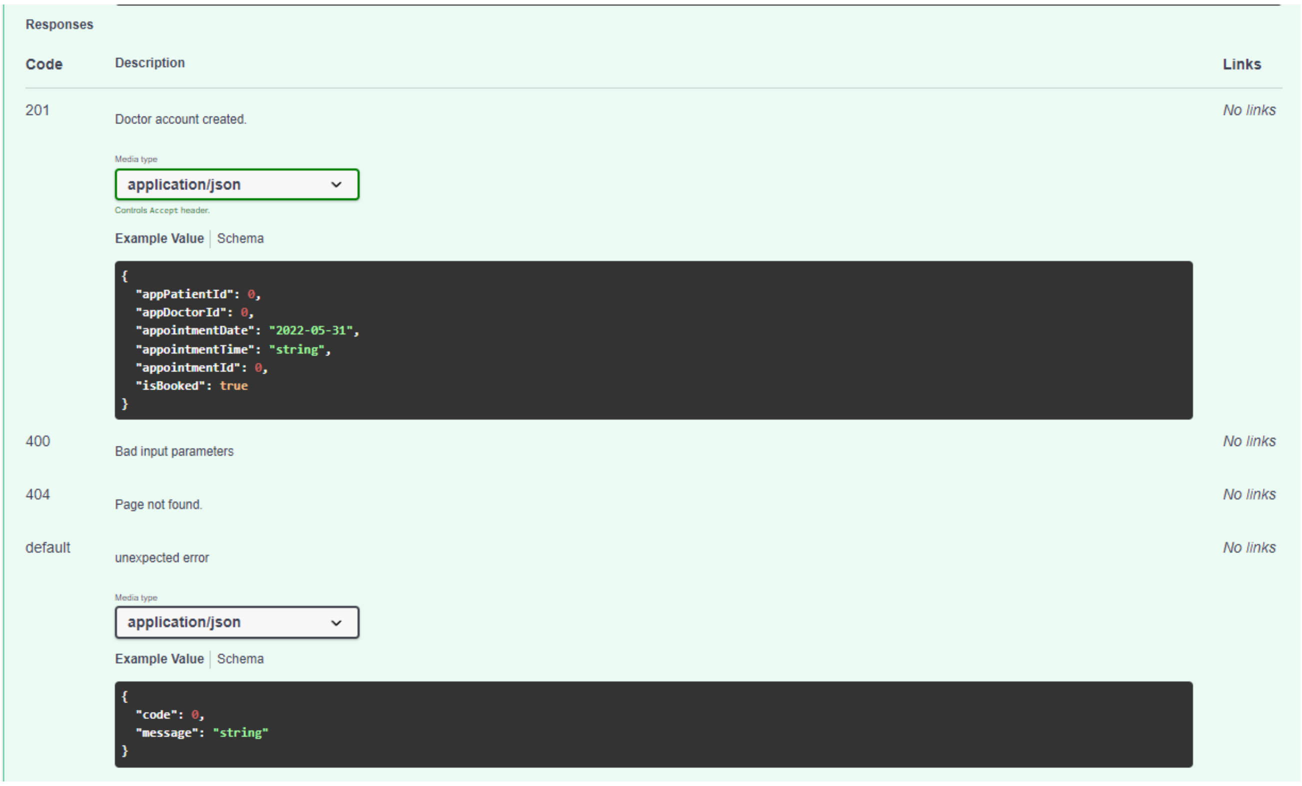
Code	Details
201	Response body

```
{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2015-07-20",
  "appointmentTime": "string",
  "appointmentId": 0,
  "isBooked": true
}
```

Response headers

```
 content-length: 110
 content-type: application/json;charset=utf-8
```

Figure 26: endpoint testing



Responses

Code	Description	Links
201	Doctor account created.	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2022-05-31",
  "appointmentTime": "string",
  "appointmentId": 0,
  "isBooked": true
}
```

Code	Description	Links
400	Bad input parameters	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

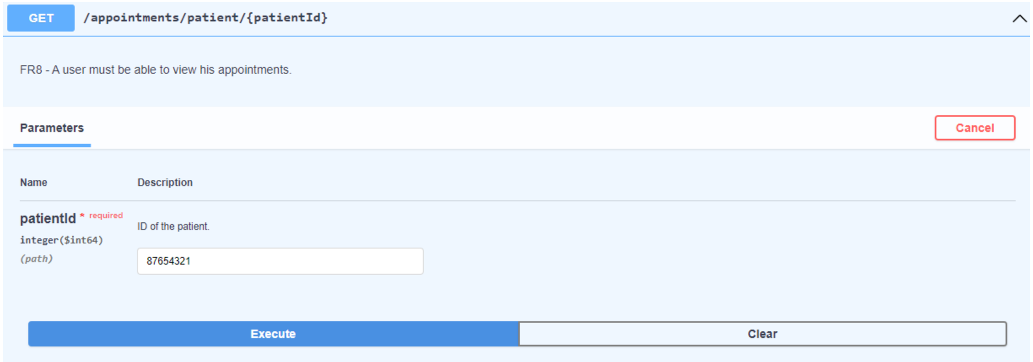
application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 27: System responses

2.2.3.4 Endpoint GET of resource appointments, for patient and specific patientId



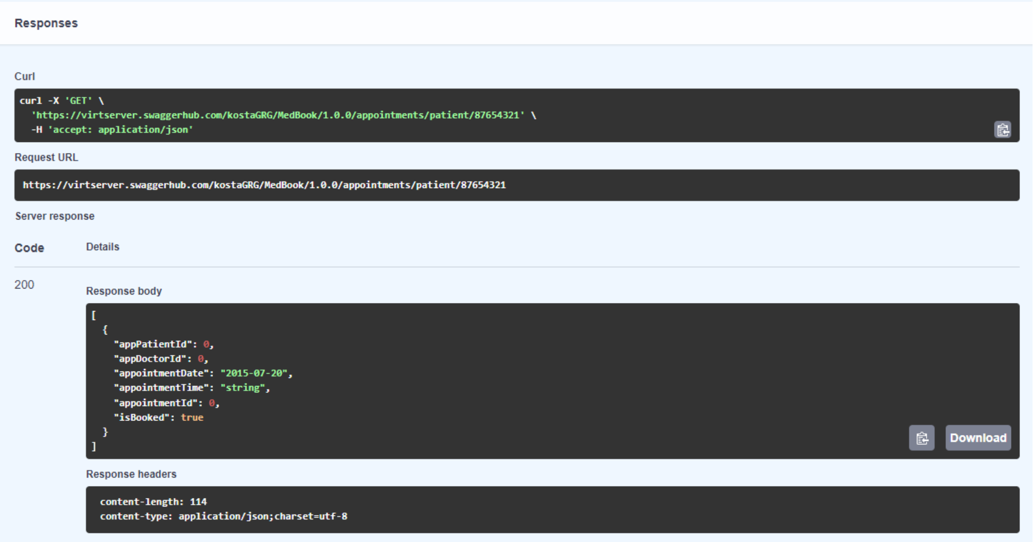
GET `/appointments/patient/{patientId}`

FR8 - A user must be able to view his appointments.

Parameters

Name	Description
patientId * required integer($int64) (path)	ID of the patient. 87654321

Figure 28: Explanation and presentation of the resource parameters



Responses

Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321
```

Server response

Code | Details

200

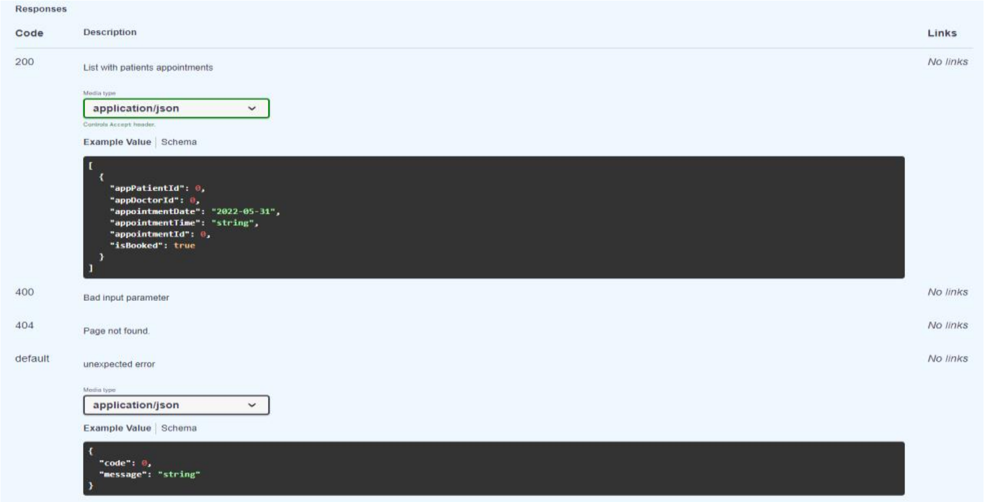
Response body

```
[
  {
    "appPatientId": 0,
    "appDoctorId": 0,
    "appointmentDate": "2015-07-20",
    "appointmentTime": "string",
    "appointmentId": 0,
    "isBooked": true
  }
]
```

Response headers

```
content-length: 114
content-type: application/json;charset=utf-8
```

Figure 29: endpoint testing



Responses

Code	Description	Links
200	List with patients appointments	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

200 — Media type: application/json (Controls Accept header.)

Example Value | Schema

```
[
  {
    "appPatientId": 0,
    "appDoctorId": 0,
    "appointmentDate": "2022-05-31",
    "appointmentTime": "string",
    "appointmentId": 0,
    "isBooked": true
  }
]
```

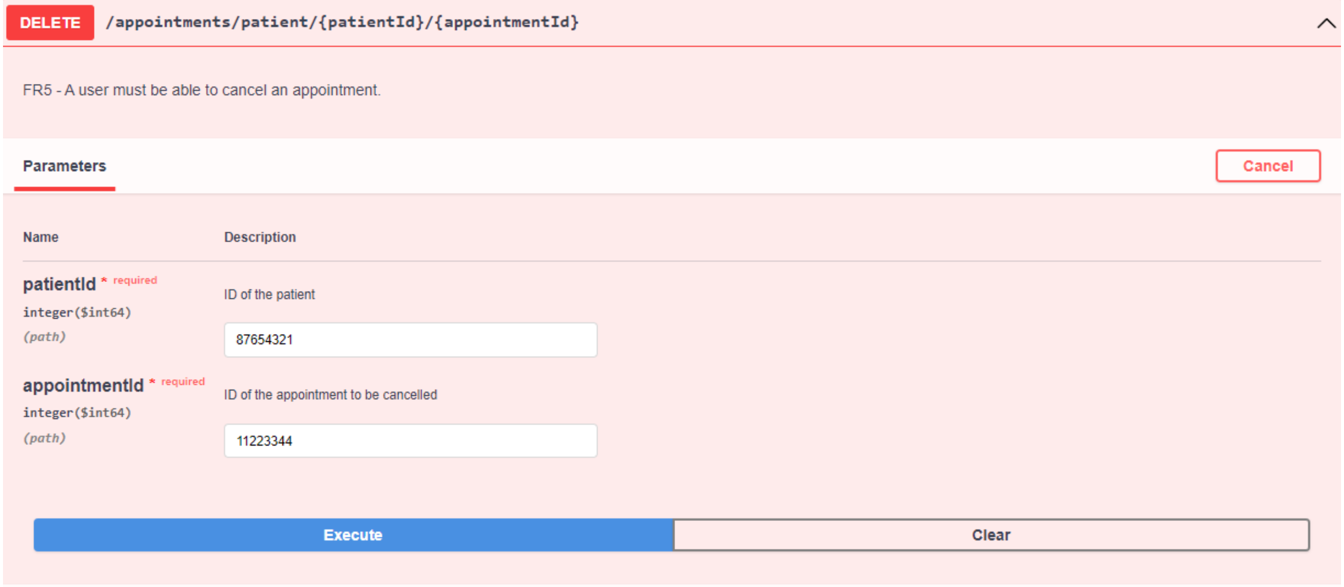
default — Media type: application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 30: System responses

2.2.3.5 Endpoint DELETE of resource appointments, for patient and specific patientId, appointmentId.

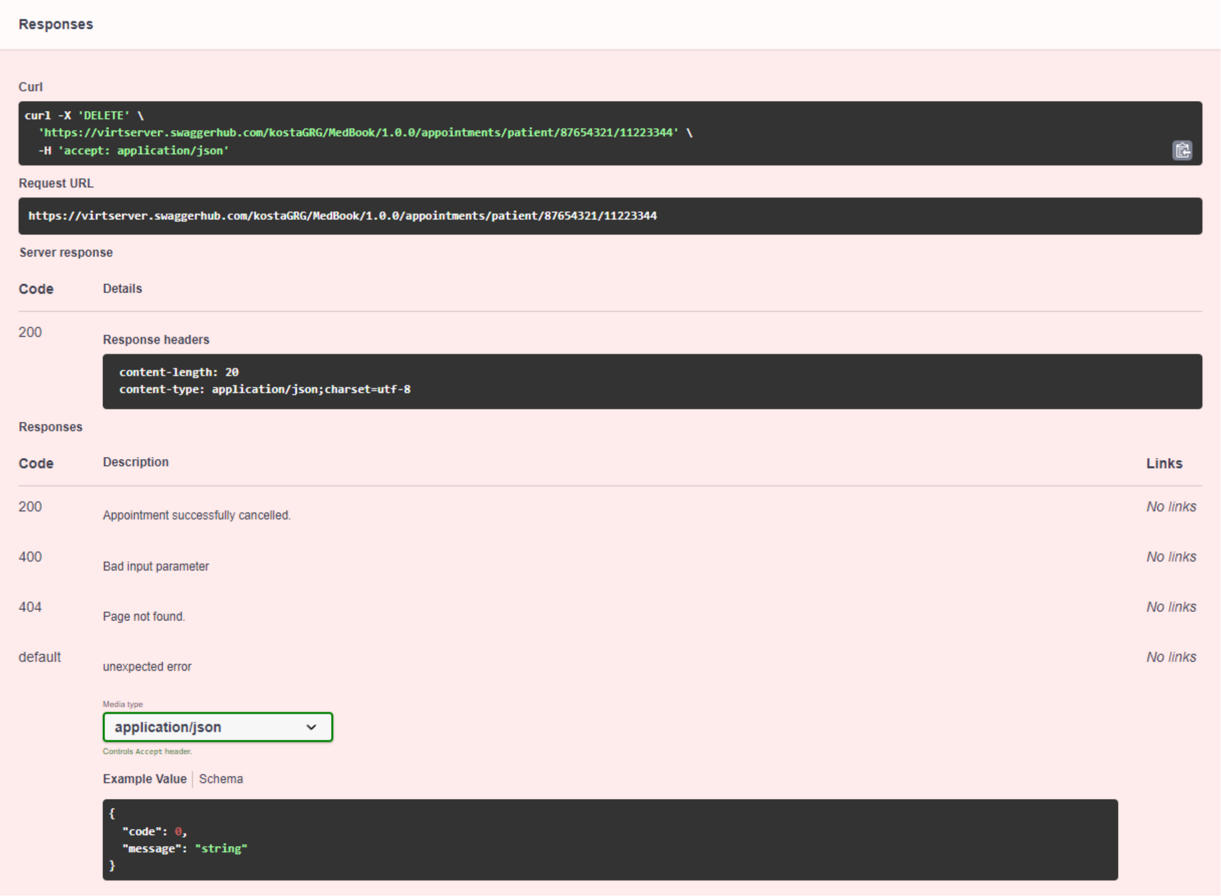


DELETE /appointments/patient/{patientId}/{appointmentId}

FR5 - A user must be able to cancel an appointment.

Name	Description
patientId * required integer($int64) (path)	ID of the patient 87654321
appointmentId * required integer($int64) (path)	ID of the appointment to be cancelled 11223344

Figure 31: Explanation and presentation of the resource parameters



Responses

Curl

```
curl -X 'DELETE' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321/11223344' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/patient/87654321/11223344
```

Server response

Code Details

200 Response headers

```
content-length: 20
content-type: application/json;charset=utf-8
```

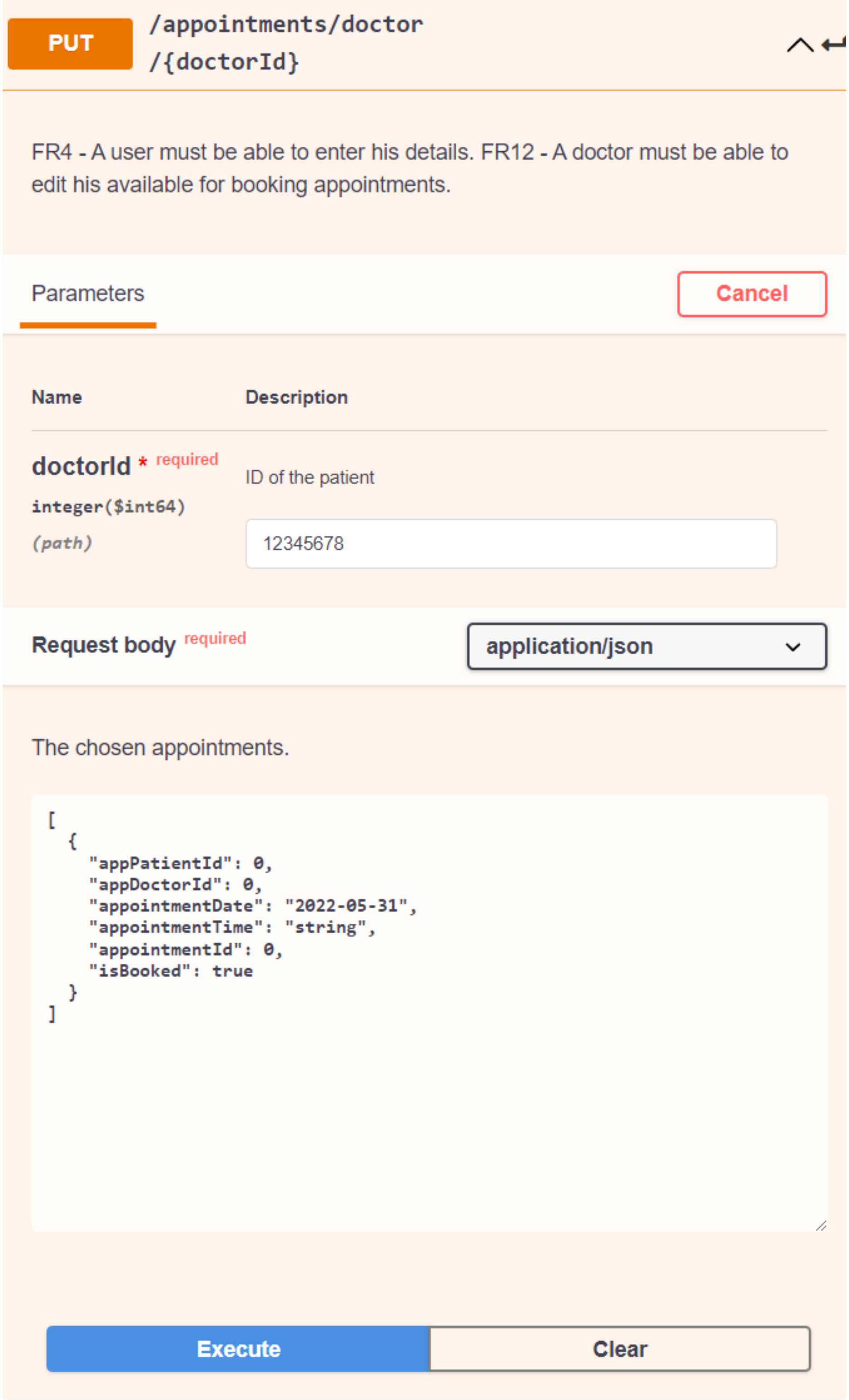
Responses

Code	Description	Links
200	Appointment successfully cancelled.	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error Media type application/json Controls Accept header. Example Value | Schema	*No links*

```
{
  "code": 0,
  "message": "string"
}
```

Figure 32: Endpoint testing and system responses

2.2.3.6 Endpoint PUT of resource appointments, for doctor and specific doctorId.



/appointments/doctor/{doctorId}

FR4 - A user must be able to enter his details. FR12 - A doctor must be able to edit his available for booking appointments.

Name	Description
doctorId * required integer($int64) (path)	ID of the patient 12345678

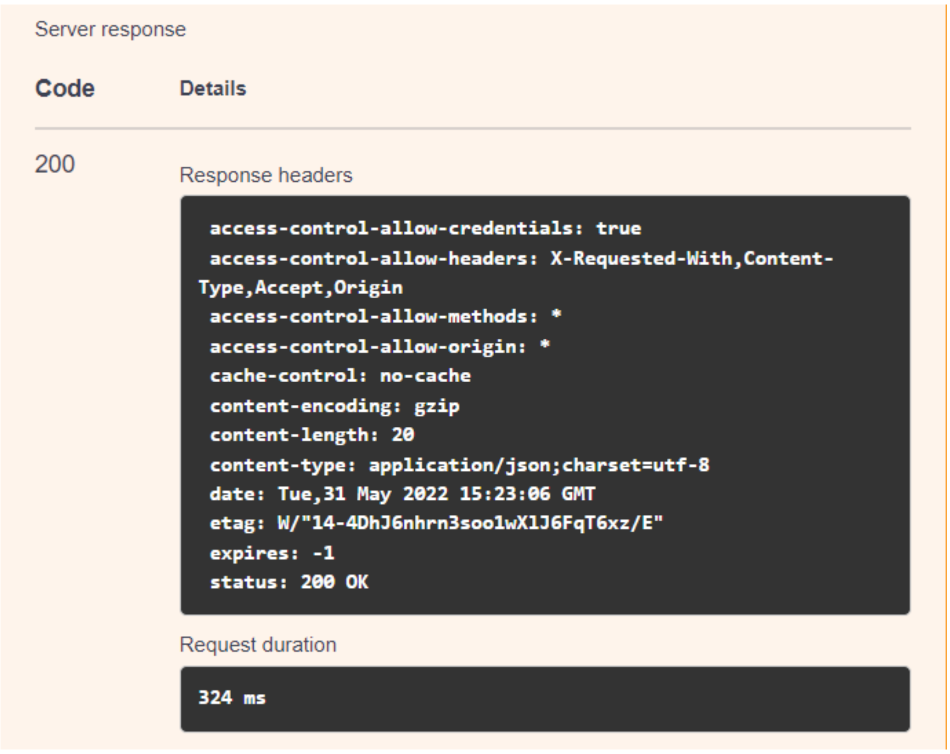
Request body required

application/json

The chosen appointments.

```
[
  {
    "appPatientId": 0,
    "appDoctorId": 0,
    "appointmentDate": "2022-05-31",
    "appointmentTime": "string",
    "appointmentId": 0,
    "isBooked": true
  }
]
```

Figure 33: Explanation and presentation of the resource parameters



Server response

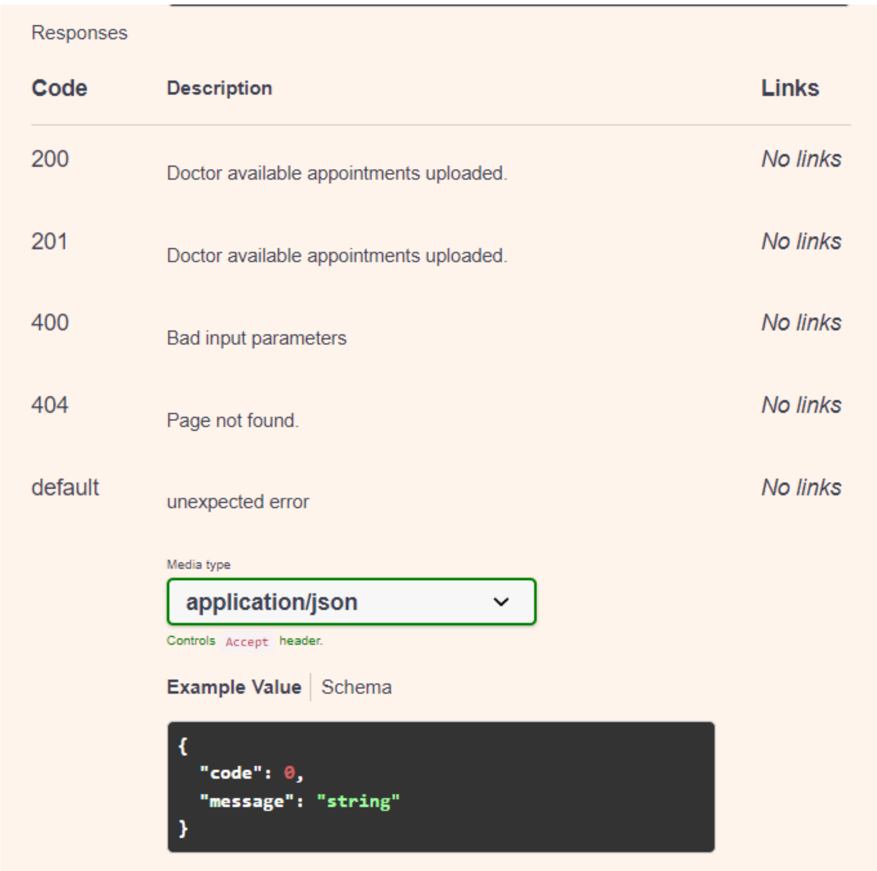
Code	Details
200	Response headers

```
 access-control-allow-credentials: true
 access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin
 access-control-allow-methods: *
 access-control-allow-origin: *
 cache-control: no-cache
 content-encoding: gzip
 content-length: 20
 content-type: application/json;charset=utf-8
 date: Tue,31 May 2022 15:23:06 GMT
 etag: W/"14-4DhJ6nhrn3soo1wX1J6FqT6xz/E"
 expires: -1
 status: 200 OK
```

Request duration

```
324 ms
```

Figure 34: endpoint testing



Responses

Code	Description	Links
200	Doctor available appointments uploaded.	*No links*
201	Doctor available appointments uploaded.	*No links*
400	Bad input parameters	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

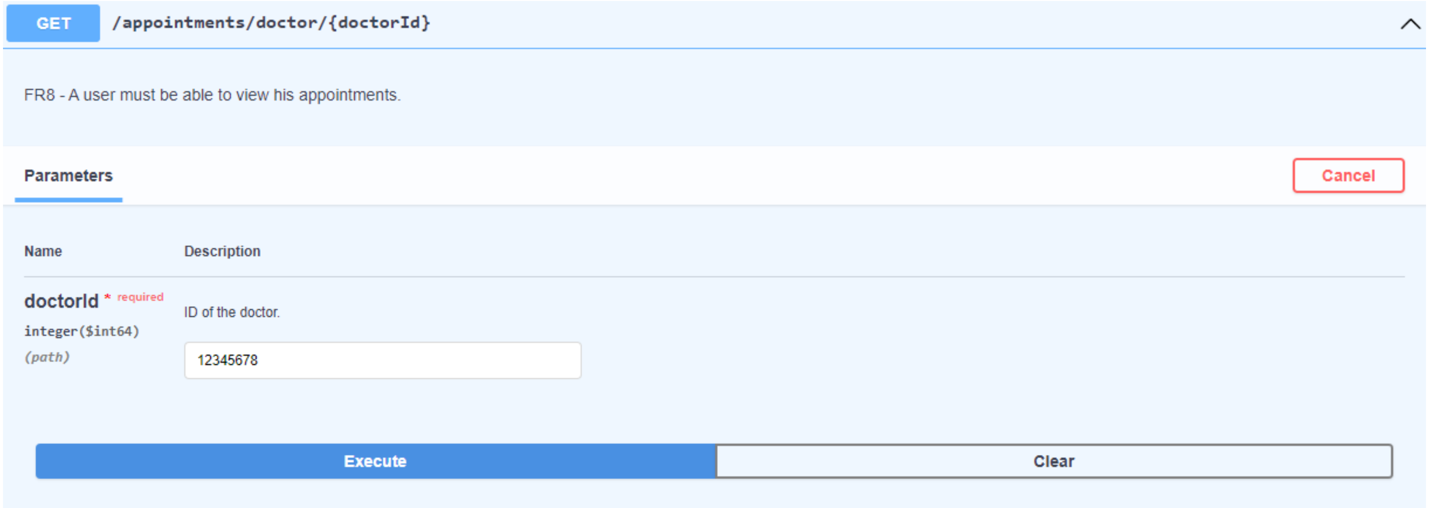
Media type
application/json
Controls `Accept` header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 35: System responses

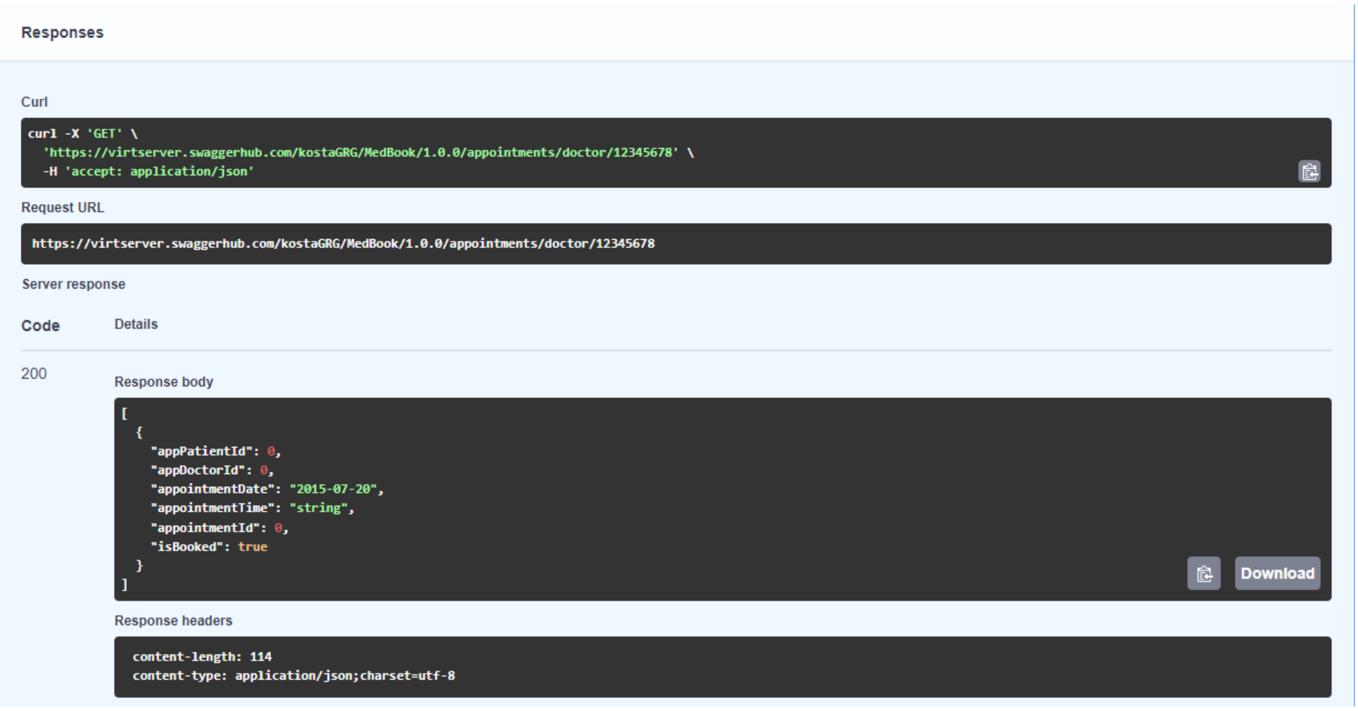
2.2.3.7 Endpoint GET of resource appointments, for doctor and specific doctorId.



FR8 - A user must be able to view his appointments.

Name	Description
doctorId * required integer($int64) (path)	ID of the doctor. 12345678

Figure 36: Explanation and presentation of the resource parameters



Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/doctor/12345678' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/doctor/12345678
```

Server response

Code Details

200

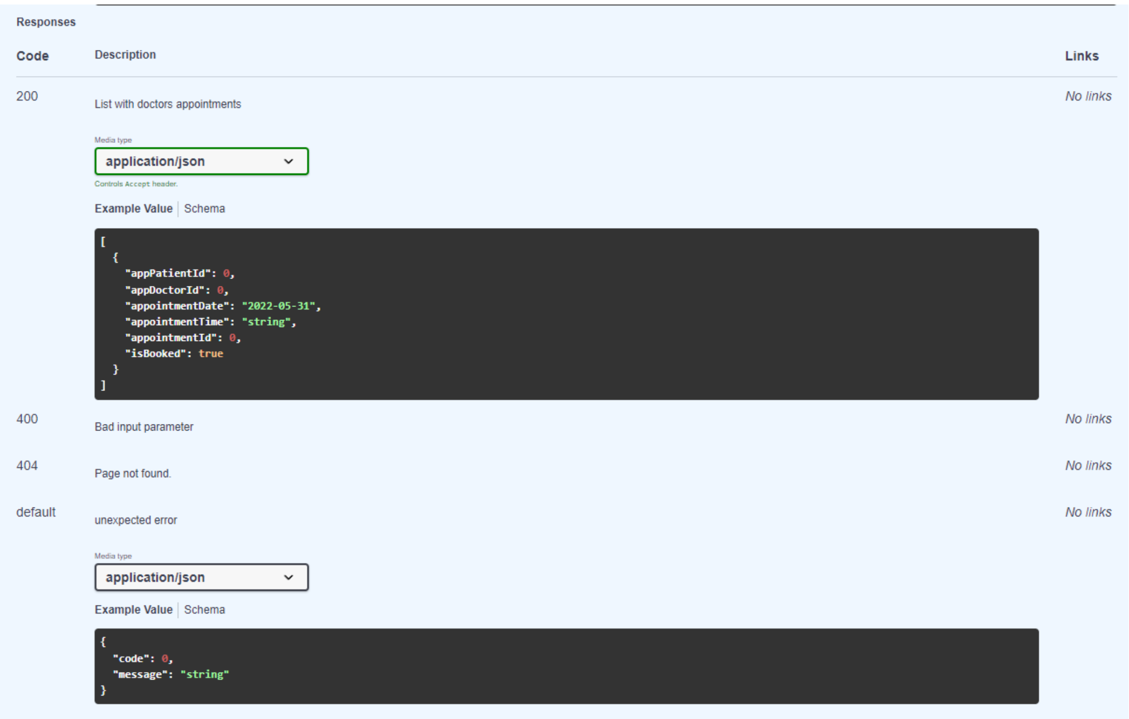
Response body

```
[
  {
    "appPatientId": 0,
    "appDoctorId": 0,
    "appointmentDate": "2015-07-20",
    "appointmentTime": "string",
    "appointmentId": 0,
    "isBooked": true
  }
]
```

Response headers

```
content-length: 114
content-type: application/json;charset=utf-8
```

Figure 37: endpoint testing



Responses

Code	Description	Links
200	List with doctors appointments	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

200 — Media type: application/json. Controls Accept header. Example Value | Schema

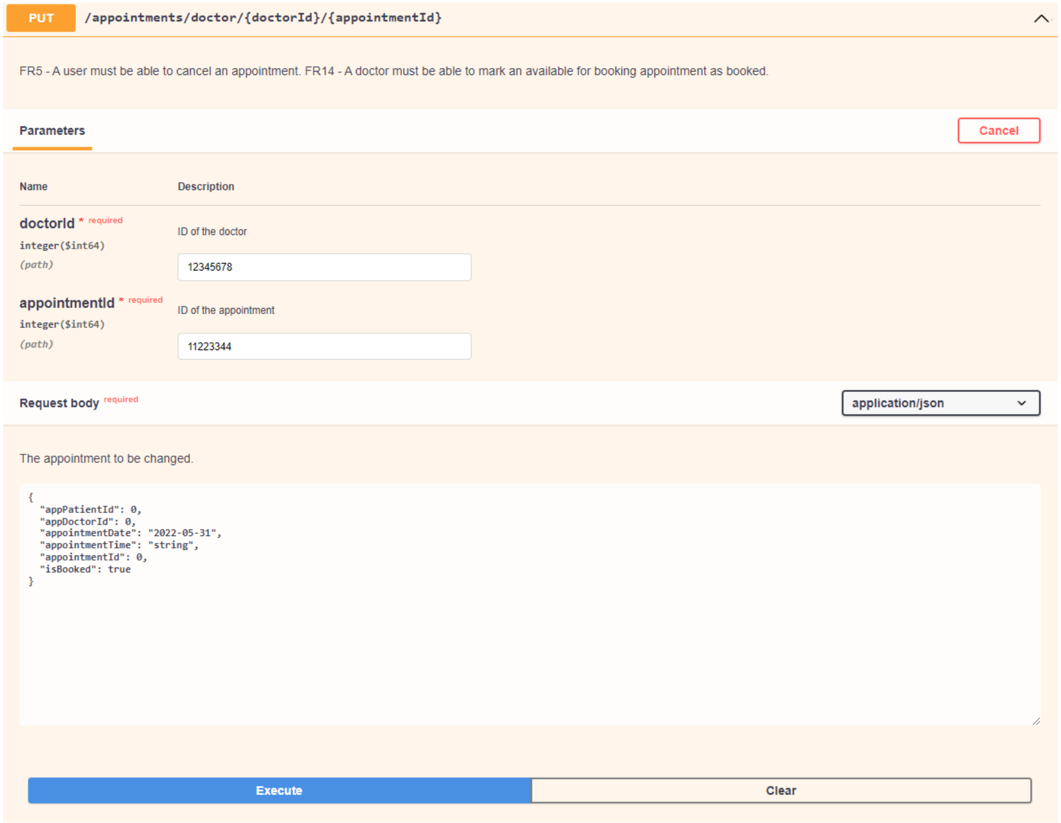
```
[
  {
    "appPatientId": 0,
    "appDoctorId": 0,
    "appointmentDate": "2022-05-31",
    "appointmentTime": "string",
    "appointmentId": 0,
    "isBooked": true
  }
]
```

default — Media type: application/json. Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 38: System responses

2.2.3.9 Endpoint PUT of resource appointments, for doctor and specific doctorId, appointmentId.



/appointments/doctor/{doctorId}/{appointmentId}

FR5 - A user must be able to cancel an appointment. FR14 - A doctor must be able to mark an available for booking appointment as booked.

Parameters

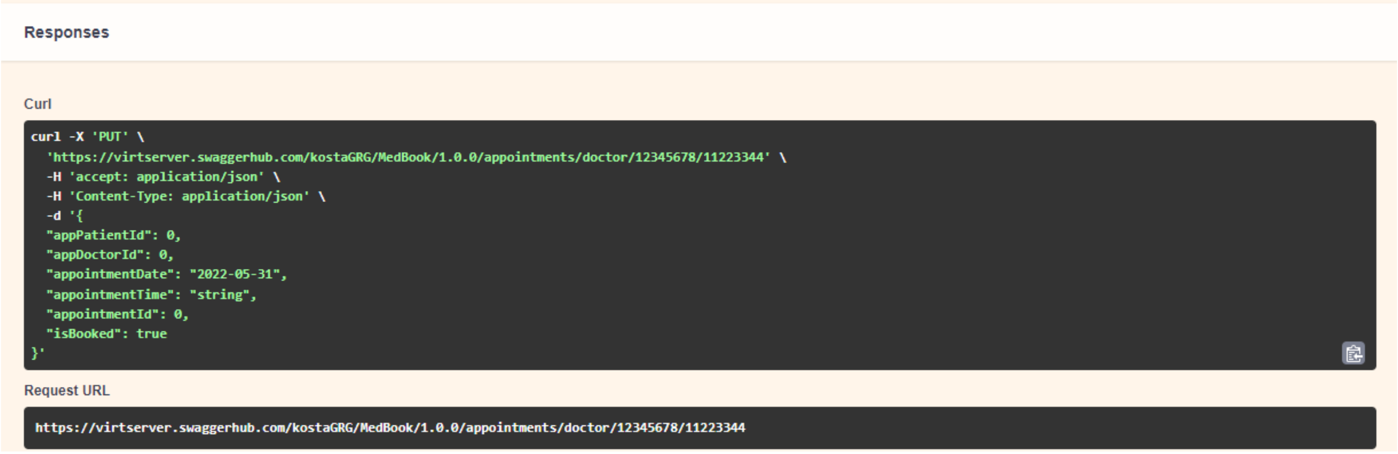
Name	Description
doctorId * required integer($int64) *(path)*	ID of the doctor 12345678
appointmentId * required integer($int64) *(path)*	ID of the appointment 11223344

Request body required — application/json

The appointment to be changed.

```
{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2022-05-31",
  "appointmentTime": "string",
  "appointmentId": 0,
  "isBooked": true
}
```

Figure 39: Explanation and presentation of the resource parameters



Responses

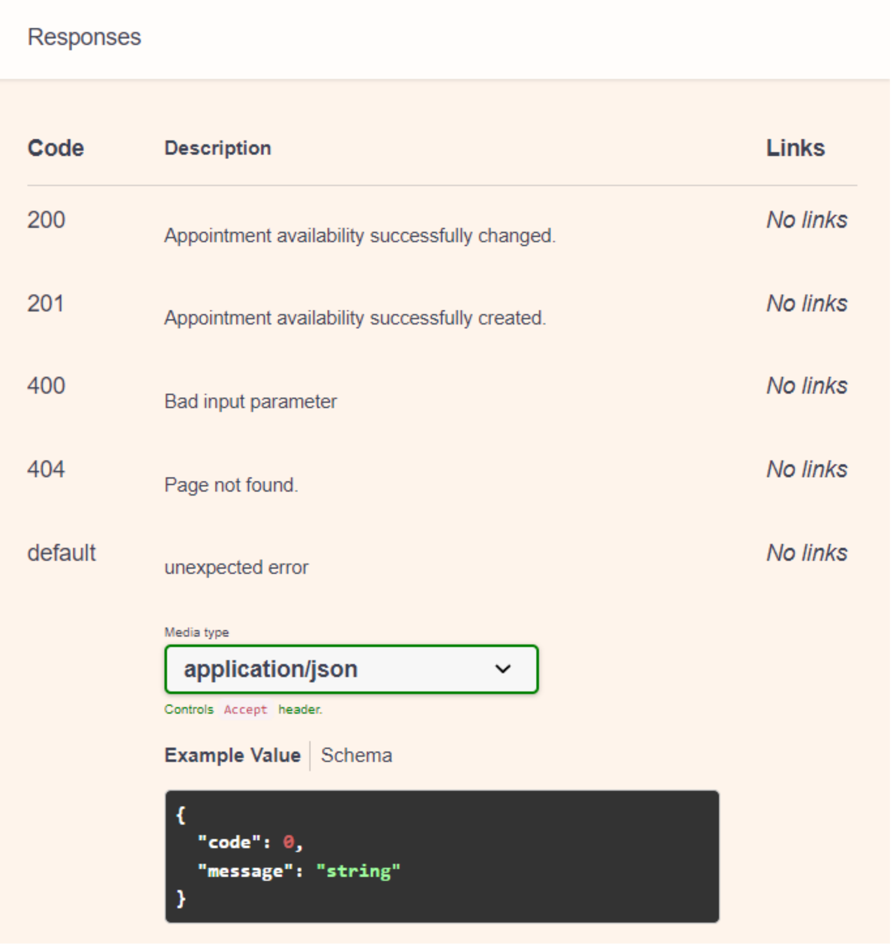
Curl

```
curl -X 'PUT' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/doctor/12345678/11223344' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "appPatientId": 0,
  "appDoctorId": 0,
  "appointmentDate": "2022-05-31",
  "appointmentTime": "string",
  "appointmentId": 0,
  "isBooked": true
}'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/appointments/doctor/12345678/11223344
```

Figure 40: endpoint testing



Responses

Code	Description	Links
200	Appointment availability successfully changed.	*No links*
201	Appointment availability successfully created.	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

Media type: application/json

Controls `Accept` header.

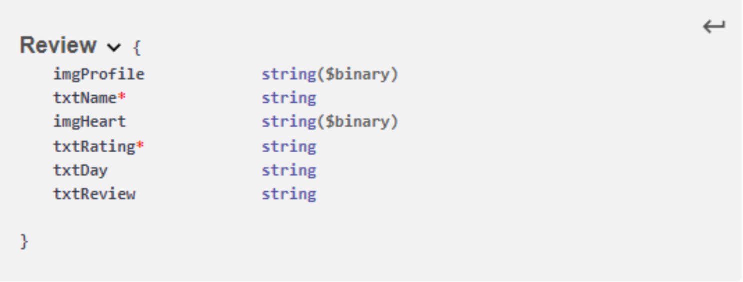
Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 41: System responses

2.2.4 Resource Review

2.2.4.1 Review data model

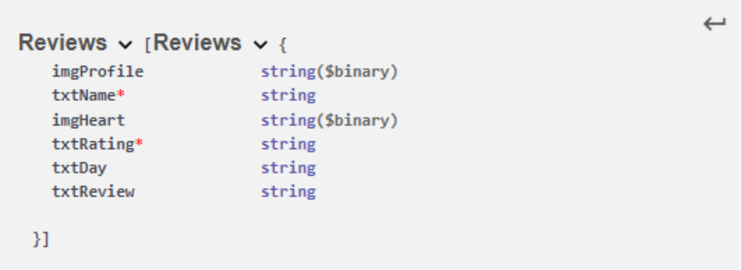


```
Review ⌄ {
    imgProfile              string($binary)
    txtName*                string
    imgHeart                string($binary)
    txtRating*              string
    txtDay                  string
    txtReview               string

}
```

Figure 42: Review data model

2.2.4.2 Data model Reviews (List of objects Review)

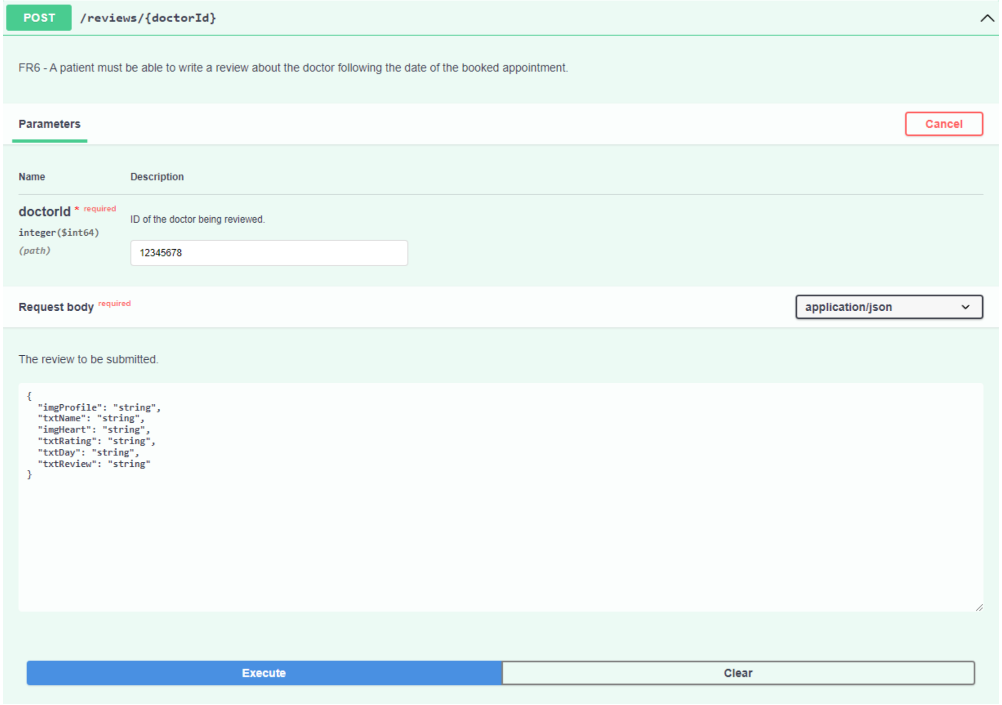


```
Reviews ⌄ [Reviews ⌄ {
    imgProfile              string($binary)
    txtName*                string
    imgHeart                string($binary)
    txtRating*              string
    txtDay                  string
    txtReview               string

  }]
```

Figure 43: Reviews data model

2.2.4.3 Endpoint POST of resource reviews, for specific doctorId.



FR6 - A patient must be able to write a review about the doctor following the date of the booked appointment.

doctorId * required
integer($int64)
(path)

ID of the doctor being reviewed.

12345678

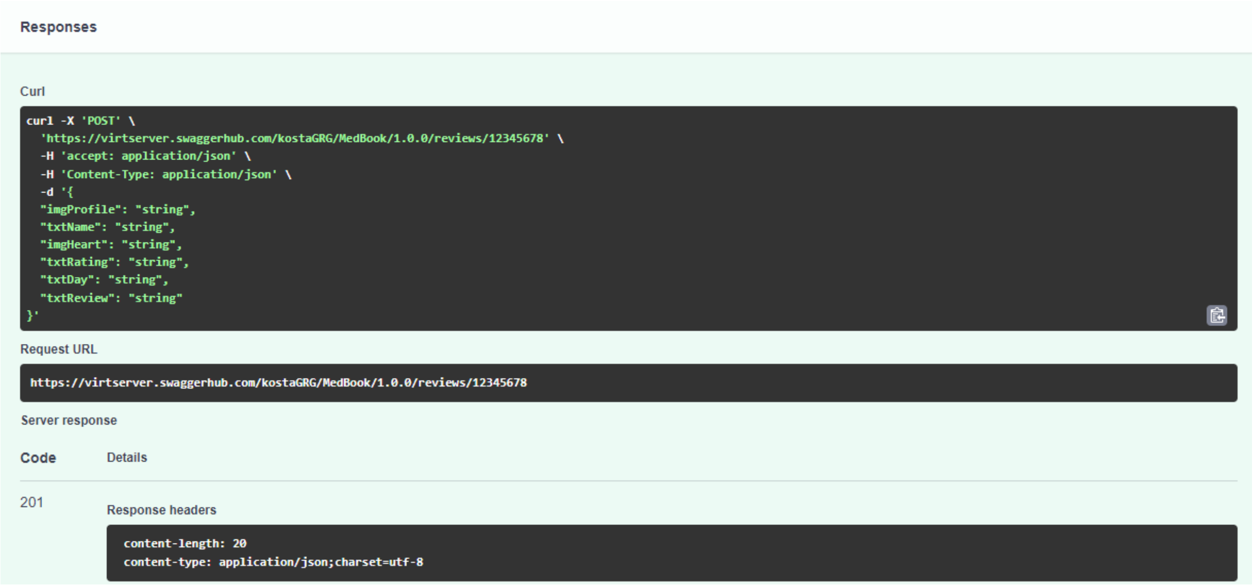
Request body required

application/json

The review to be submitted.

```
{
  "imgProfile": "string",
  "txtName": "string",
  "imgHeart": "string",
  "txtRating": "string",
  "txtDay": "string",
  "txtReview": "string"
}
```

Figure 44: Explanation and presentation of the resource parameters



Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reviews/12345678' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "imgProfile": "string",
  "txtName": "string",
  "imgHeart": "string",
  "txtRating": "string",
  "txtDay": "string",
  "txtReview": "string"
}'
```

Request URL

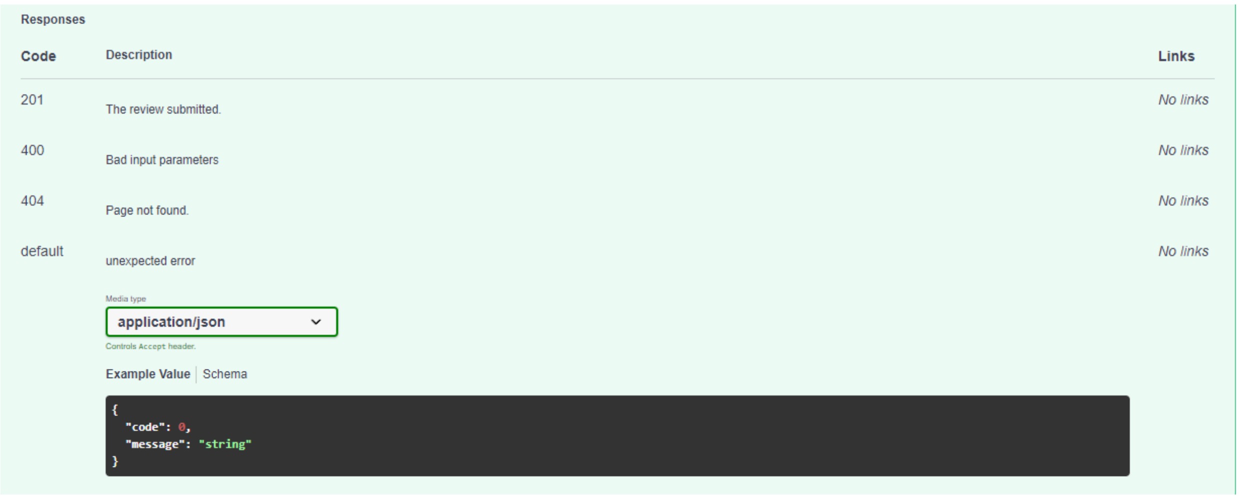
```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reviews/12345678
```

Server response

Code	Details
201	Response headers

```
content-length: 20
content-type: application/json;charset=utf-8
```

Figure 45: endpoint testing



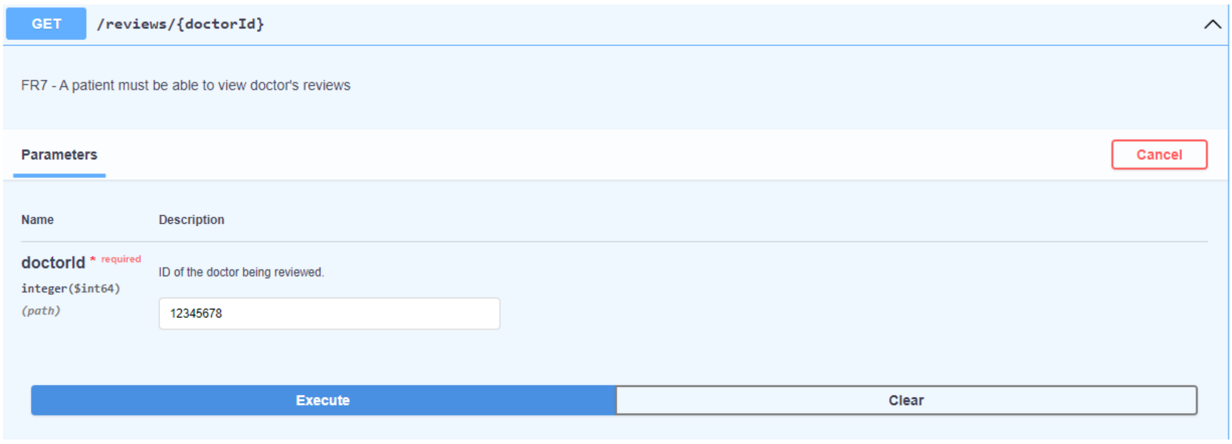
Responses

Code	Description	Links
201	The review submitted.	*No links*
400	Bad input parameters	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

```
{
  "code": 0,
  "message": "string"
}
```

Figure 46: System responses

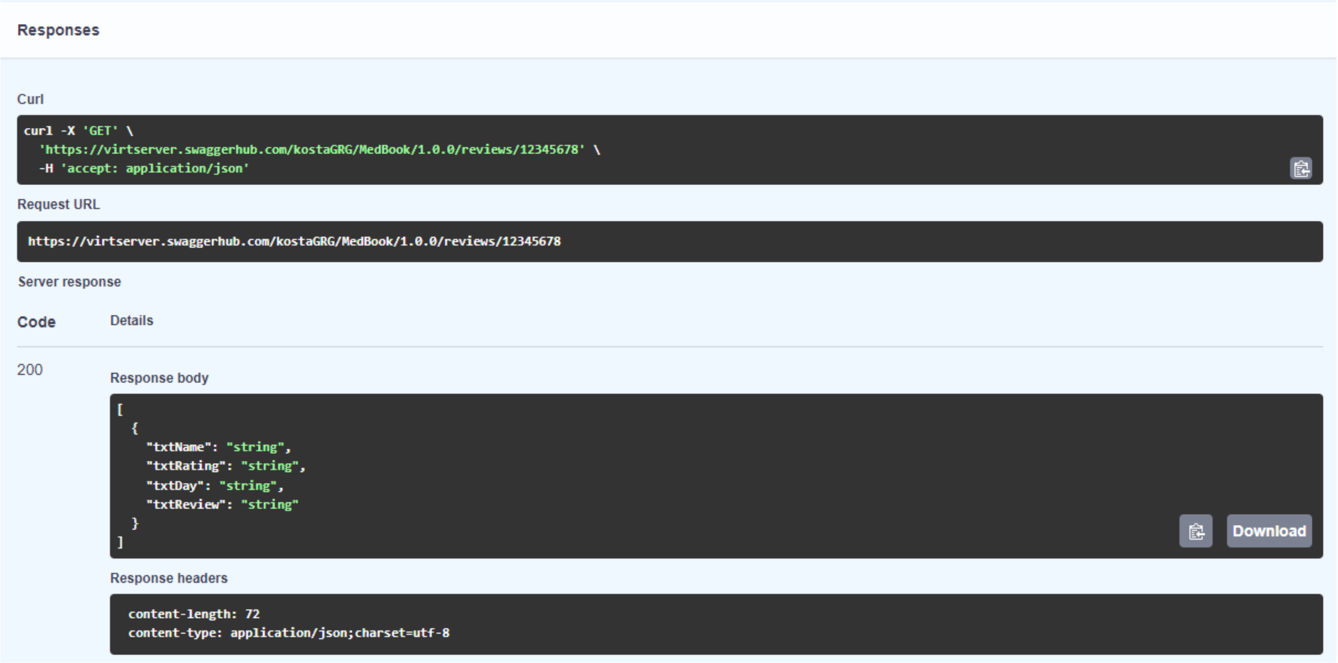
2.2.4.4 Endpoint GET of resource reviews, for specific doctorId.



FR7 - A patient must be able to view doctor's reviews

Name	Description
doctorId * required integer($int64) *(path)*	ID of the doctor being reviewed. 12345678

Figure 47: Explanation and presentation of the resource parameters



Responses

Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reviews/12345678' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reviews/12345678
```

Server response

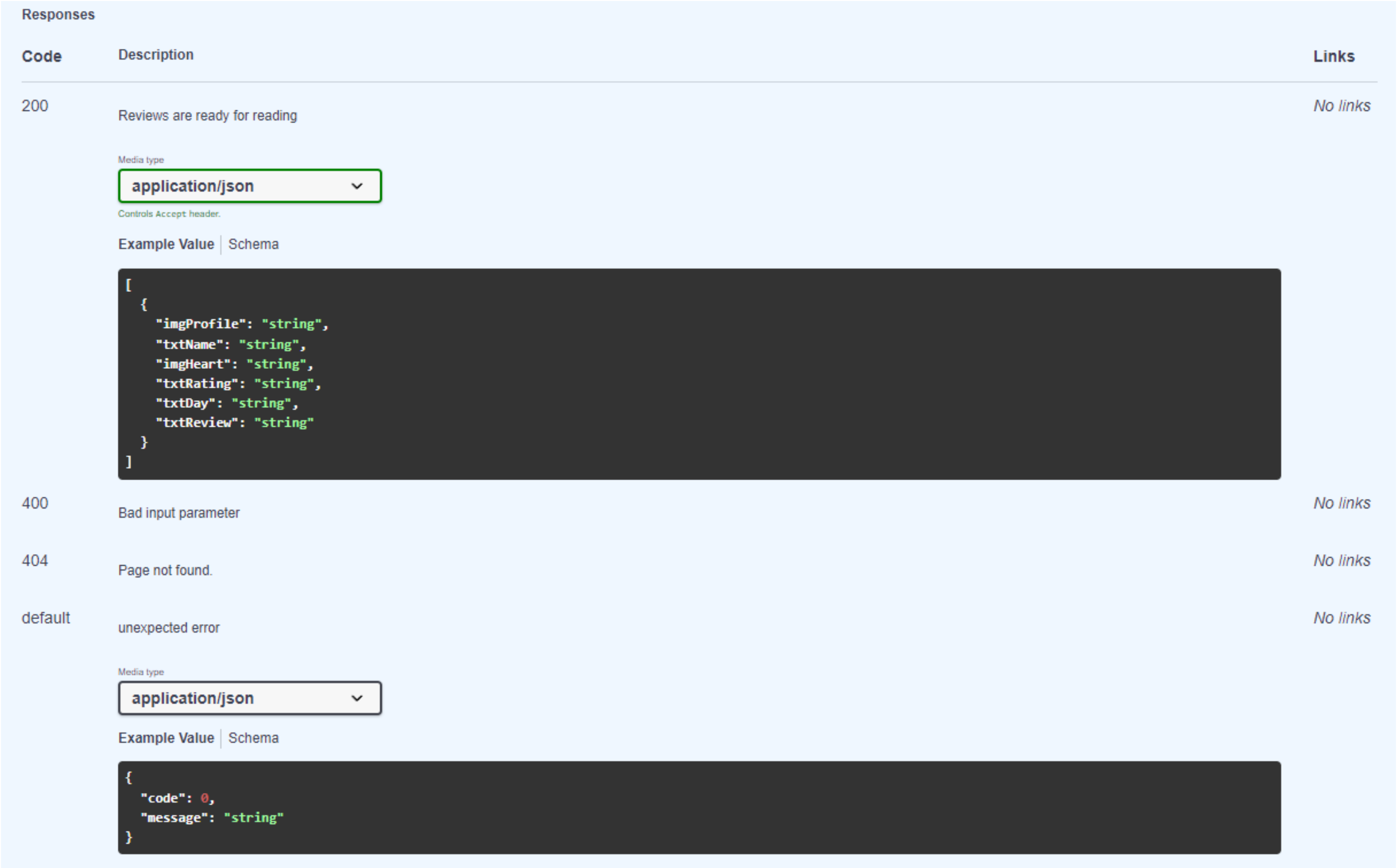
Code	Details
200	Response body

```
[
  {
    "txtName": "string",
    "txtRating": "string",
    "txtDay": "string",
    "txtReview": "string"
  }
]
```

Response headers

```
content-length: 72
content-type: application/json;charset=utf-8
```

Figure 48: endpoint testing



Responses

Code	Description	Links
200	Reviews are ready for reading	*No links*

```
[
  {
    "imgProfile": "string",
    "txtName": "string",
    "imgHeart": "string",
    "txtRating": "string",
    "txtDay": "string",
    "txtReview": "string"
  }
]
```

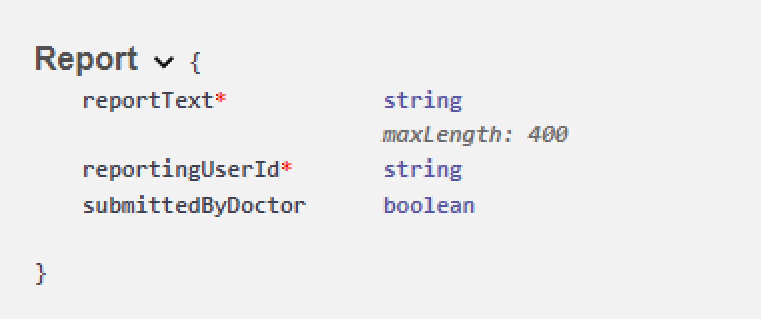
Code	Description	Links
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

```
{
  "code": 0,
  "message": "string"
}
```

Figure 49: System responses

2.2.5 Resource Report

2.2.5.1 Report data model



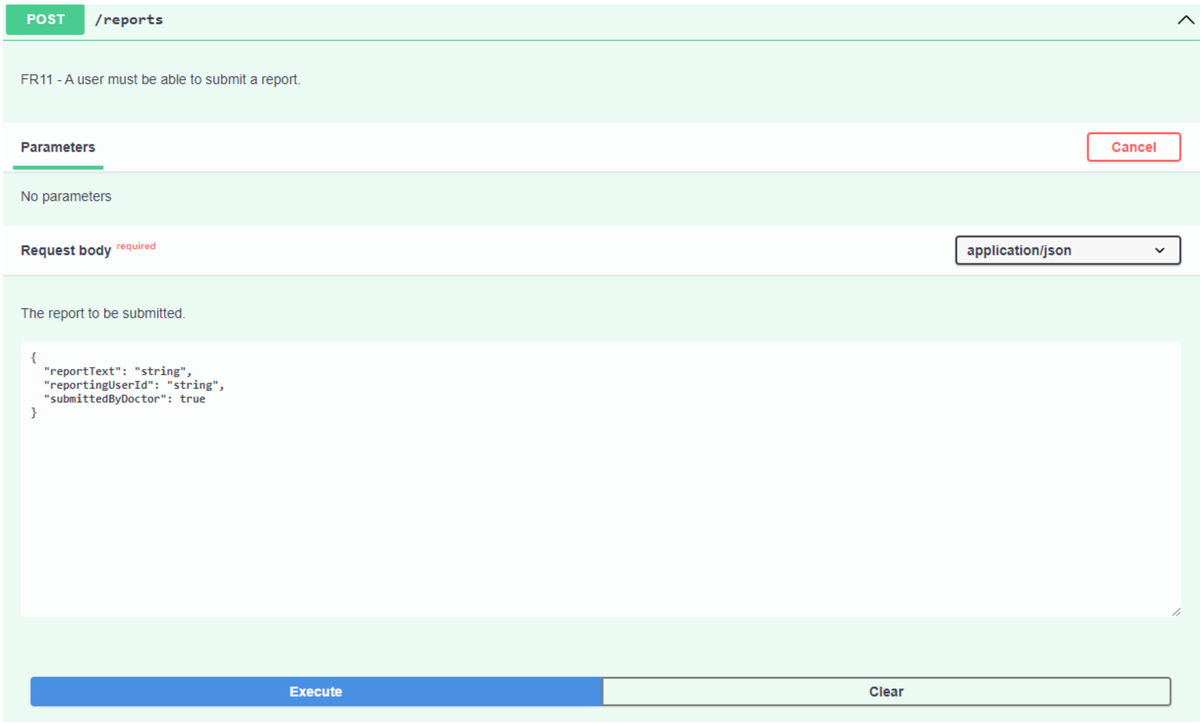
```
Report ⌄ {
    reportText*           string
                          maxLength: 400

    reportingUserId*      string
    submittedByDoctor     boolean

}
```

Figure 50: Report data model

2.2.5.2 Endpoint POST of resource reports



FR11 - A user must be able to submit a report.

No parameters

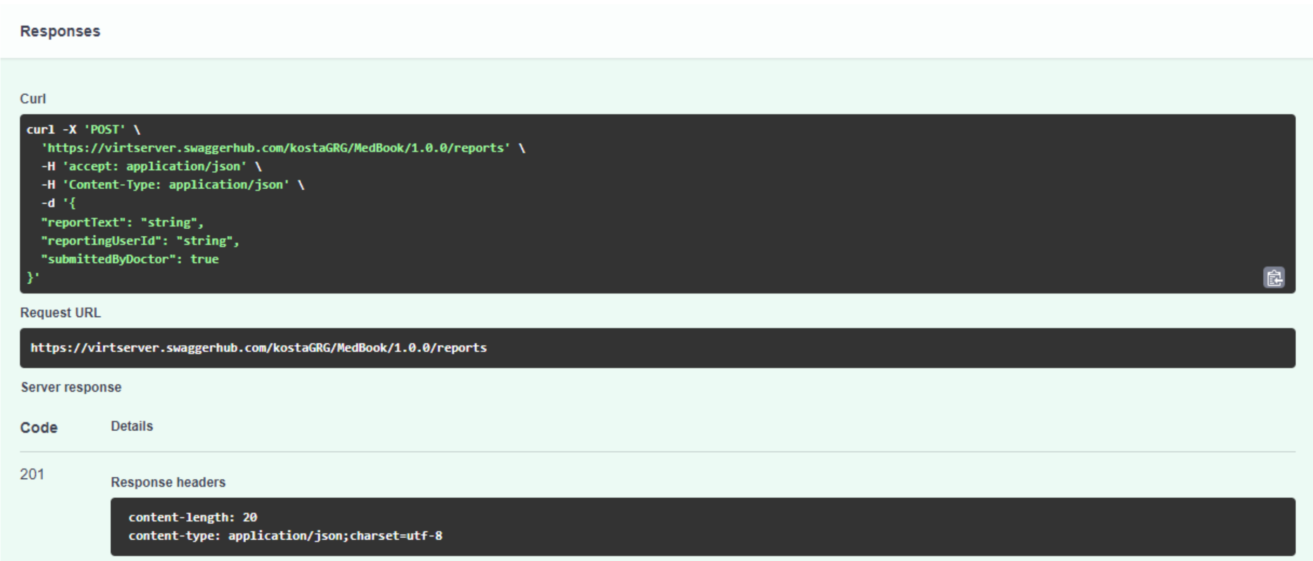
Request body required

application/json

The report to be submitted.

```
{
  "reportText": "string",
  "reportingUserId": "string",
  "submittedByDoctor": true
}
```

Figure 51: Explanation and presentation of the resource parameters



Responses

Curl

```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reports' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "reportText": "string",
  "reportingUserId": "string",
  "submittedByDoctor": true
}'
```

Request URL

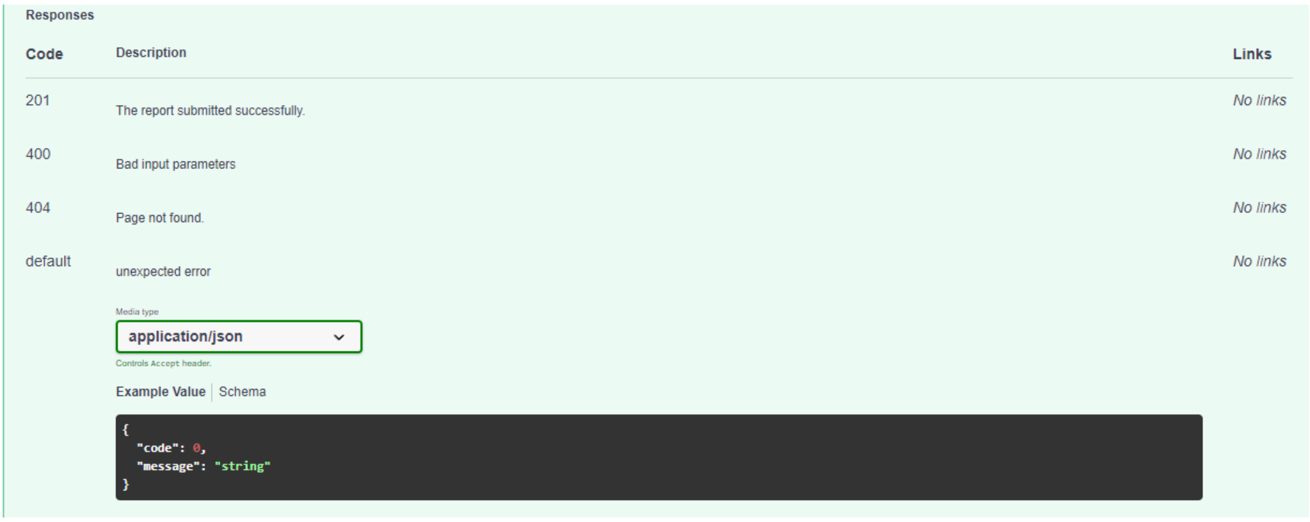
```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/reports
```

Server response

Code	Details
201	Response headers

```
content-length: 20
content-type: application/json;charset=utf-8
```

Figure 52: endpoint testing



Responses

Code	Description	Links
201	The report submitted successfully.	*No links*
400	Bad input parameters	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

Media type
application/json
Controls Accept header.

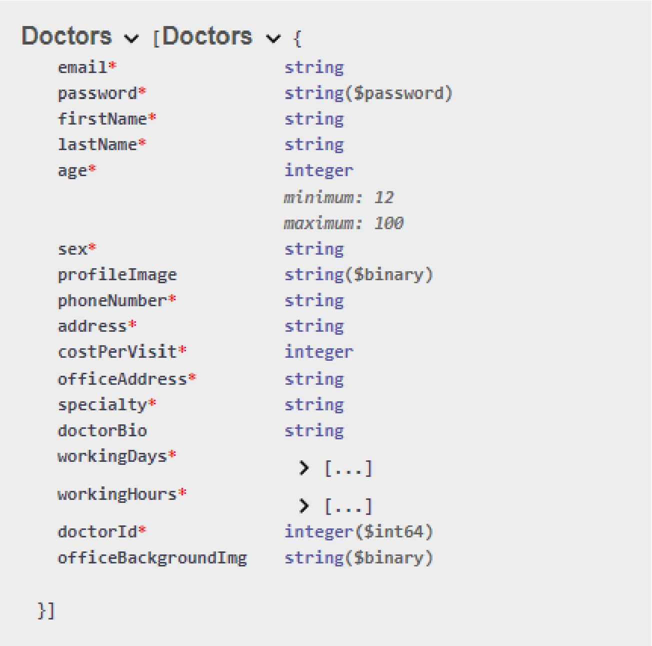
Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 53: System responses

2.2.6 Resource Search

2.2.6.1 Doctors data model



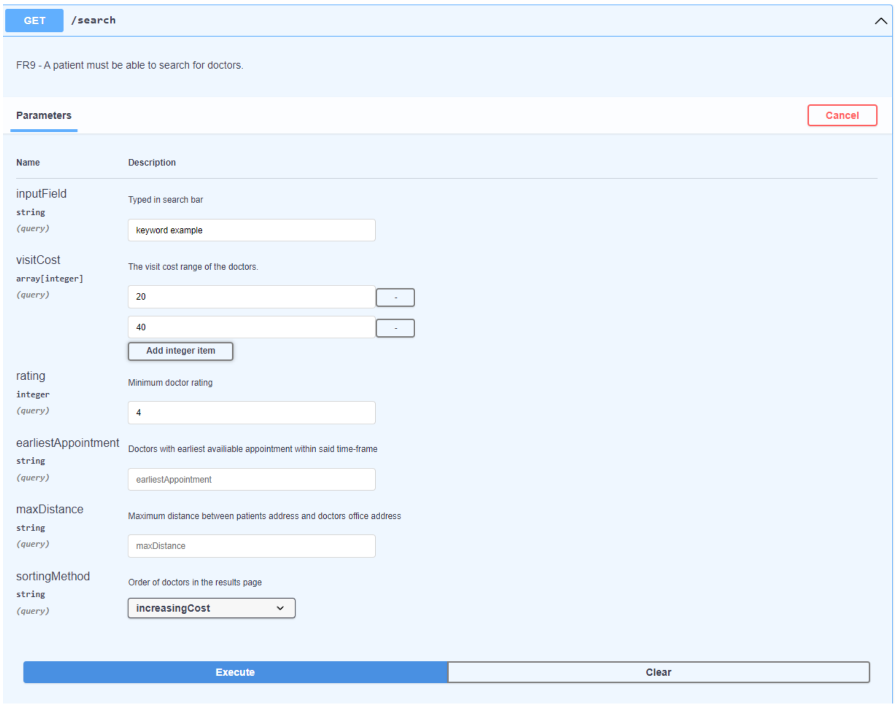
```
Doctors ⌄ [Doctors ⌄ {
    email*                string
    password*             string($password)
    firstName*            string
    lastName*             string
    age*                  integer
                          minimum: 12
                          maximum: 100

    sex*                  string
    profileImage          string($binary)
    phoneNumber*          string
    address*              string
    costPerVisit*         integer
    officeAddress*        string
    specialty*            string
    doctorBio             string
    workingDays*
                            > [...]
    workingHours*
                            > [...]
    doctorId*             integer($int64)
    officeBackgroundImg   string($binary)

 }]
```

Figure 54: Doctors data model

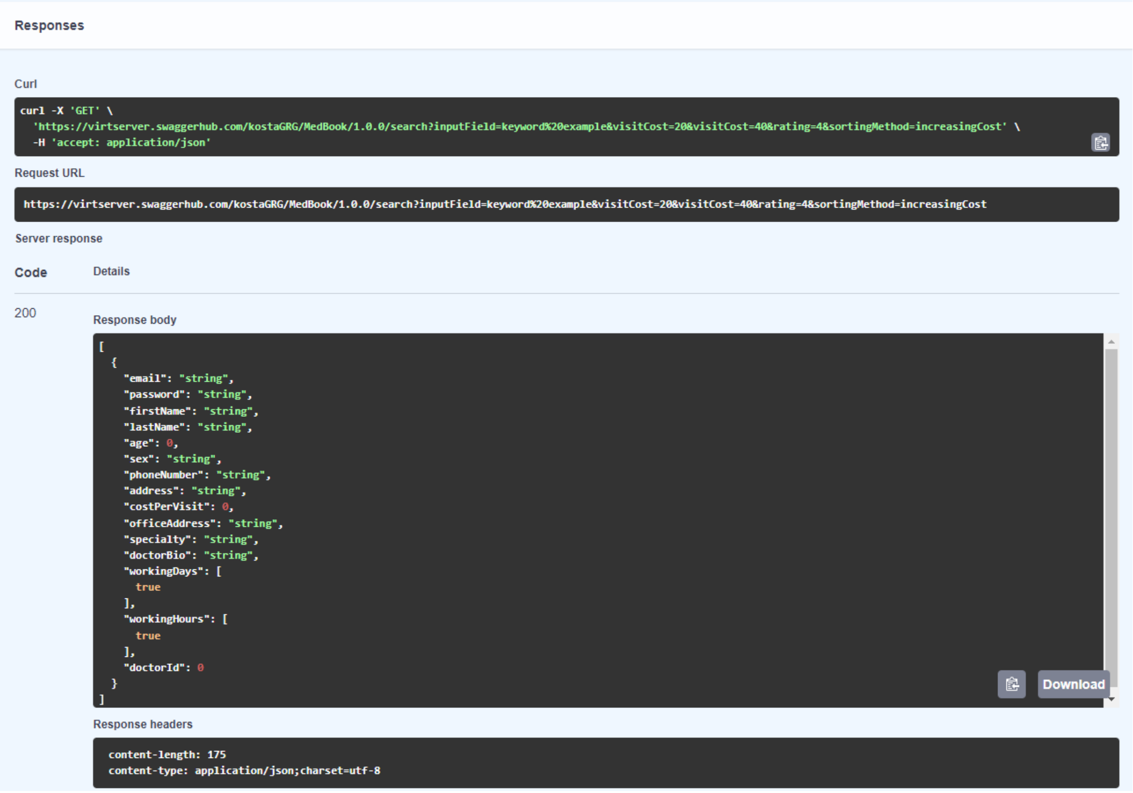
2.2.6.2 Endpoint GET of resource search



FR9 - A patient must be able to search for doctors.

Name	Description
inputField string *(query)*	Typed in search bar keyword example
visitCost array[integer] *(query)*	The visit cost range of the doctors. 20 - 40 - Add integer item
rating integer *(query)*	Minimum doctor rating 4
earliestAppointment string *(query)*	Doctors with earliest availiable appointment within said time-frame earliestAppointment
maxDistance string *(query)*	Maximum distance between patients address and doctors office address maxDistance
sortingMethod string *(query)*	Order of doctors in the results page increasingCost ⌄

Figure 55: Explanation and presentation of the resource parameters



Responses

Curl

```
curl -X 'GET' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/search?inputField=keyword%20example&visitCost=20&visitCost=40&rating=4&sortingMethod=increasingCost' \
  -H 'accept: application/json'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/search?inputField=keyword%20example&visitCost=20&visitCost=40&rating=4&sortingMethod=increasingCost
```

Server response

Code Details

200

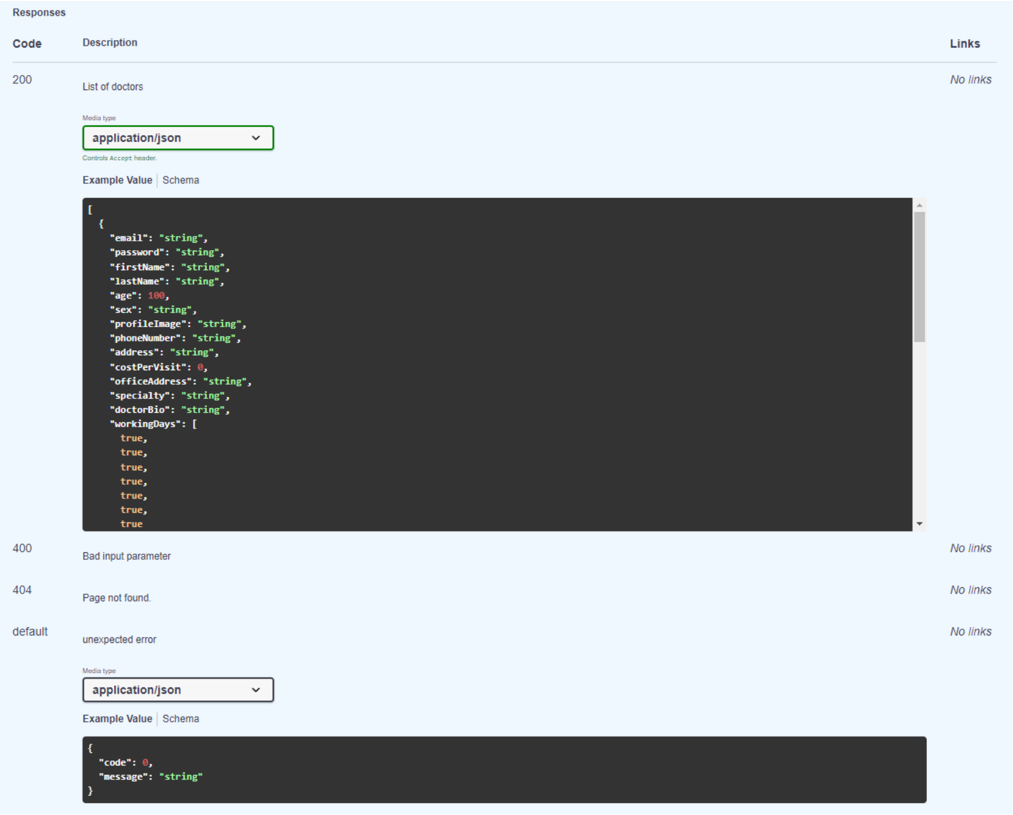
Response body

```
[
  {
    "email": "string",
    "password": "string",
    "firstName": "string",
    "lastName": "string",
    "age": 0,
    "sex": "string",
    "phoneNumber": "string",
    "address": "string",
    "costPerVisit": 0,
    "officeAddress": "string",
    "specialty": "string",
    "doctorBio": "string",
    "workingDays": [
      true
    ],
    "workingHours": [
      true
    ],
    "doctorId": 0
  }
]
```

Response headers

```
 content-length: 175
 content-type: application/json;charset=utf-8
```

Figure 56: endpoint testing



Responses

Code | Description | Links

200 List of doctors — No links

Media type

Controls Accept header.

```
[
  {
    "email": "string",
    "password": "string",
    "firstName": "string",
    "lastName": "string",
    "age": 100,
    "sex": "string",
    "profileImage": "string",
    "phoneNumber": "string",
    "address": "string",
    "costPerVisit": 0,
    "officeAddress": "string",
    "specialty": "string",
    "doctorBio": "string",
    "workingDays": [
      true,
      true,
      true,
      true,
      true,
      true,
      true
```

400 Bad input parameter — No links

404 Page not found. — No links

default unexpected error — No links

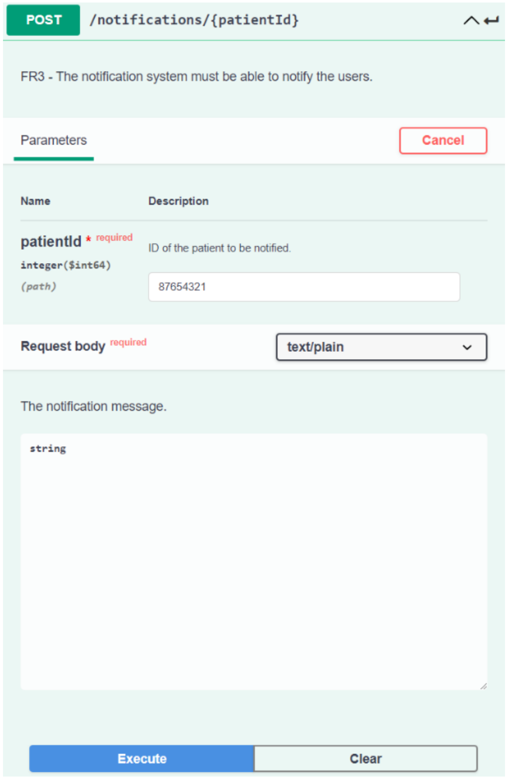
Media type

```
{
  "code": 0,
  "message": "string"
}
```

Figure 57: System responses

2.2.7 Resource Notifications

2.2.7.1 Endpoint POST of the resource notifications, for specific patientId.



FR3 - The notification system must be able to notify the users.

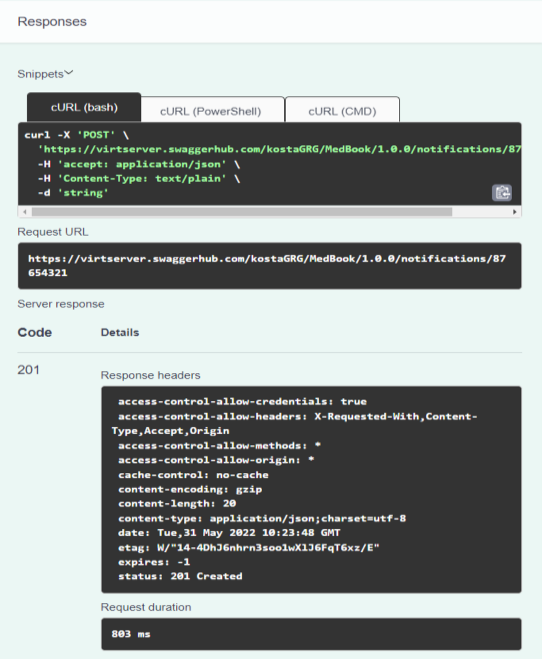
Name	Description
patientId * required integer($int64) (path)	ID of the patient to be notified. 87654321

Request body required: text/plain

The notification message.

```
string
```

Figure 58: Explanation and presentation of the resource parameters



```
curl -X 'POST' \
  'https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/notifications/87
  -H 'accept: application/json' \
  -H 'Content-Type: text/plain' \
  -d 'string'
```

Request URL

```
https://virtserver.swaggerhub.com/kostaGRG/MedBook/1.0.0/notifications/87
654321
```

Server response

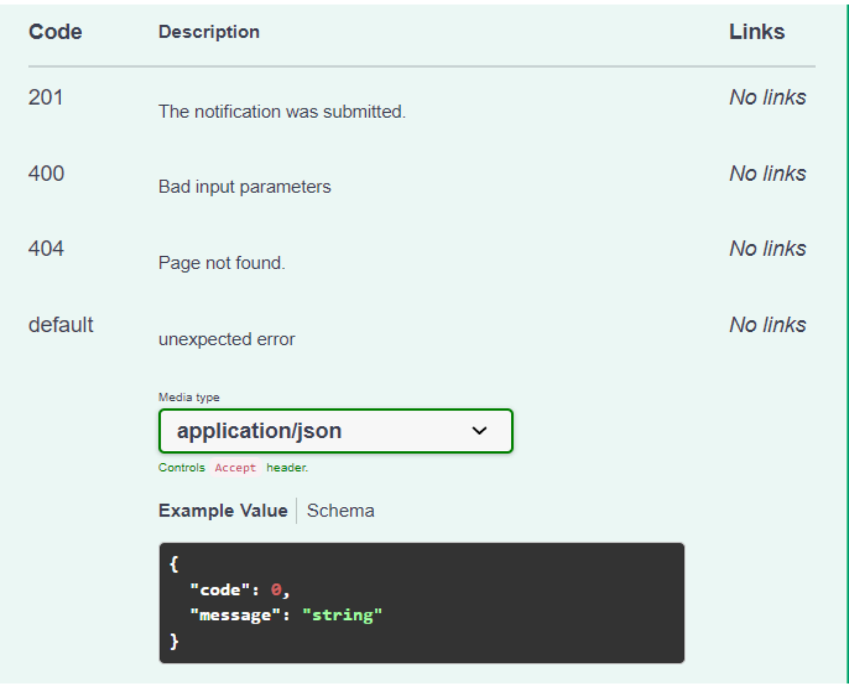
Code	Details
201	Response headers

```
 access-control-allow-credentials: true
 access-control-allow-headers: X-Requested-With,Content-
Type,Accept,Origin
 access-control-allow-methods: *
 access-control-allow-origin: *
 cache-control: no-cache
 content-encoding: gzip
 content-length: 20
 content-type: application/json;charset=utf-8
 date: Tue,31 May 2022 10:23:48 GMT
 etag: W/"14-4DhJ6nhrn3soolwX1J6FqT6xz/E"
 expires: -1
 status: 201 Created
```

Request duration

```
803 ms
```

Figure 59: endpoint testing



Code	Description	Links
201	The notification was submitted.	*No links*
400	Bad input parameters	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

Media type

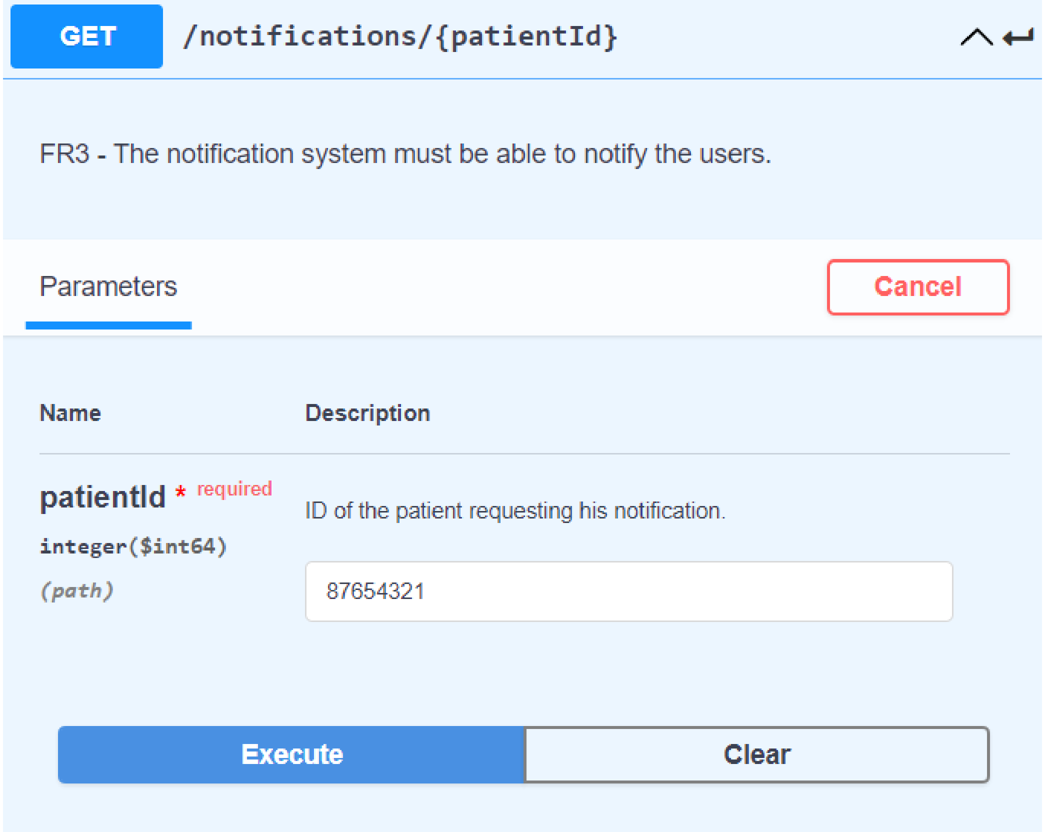
application/json

Controls Accept header.

```
{
  "code": 0,
  "message": "string"
}
```

Figure 60: System responses

2.2.7.2 Endpoint GET of the resource notifications, for specific patientId.

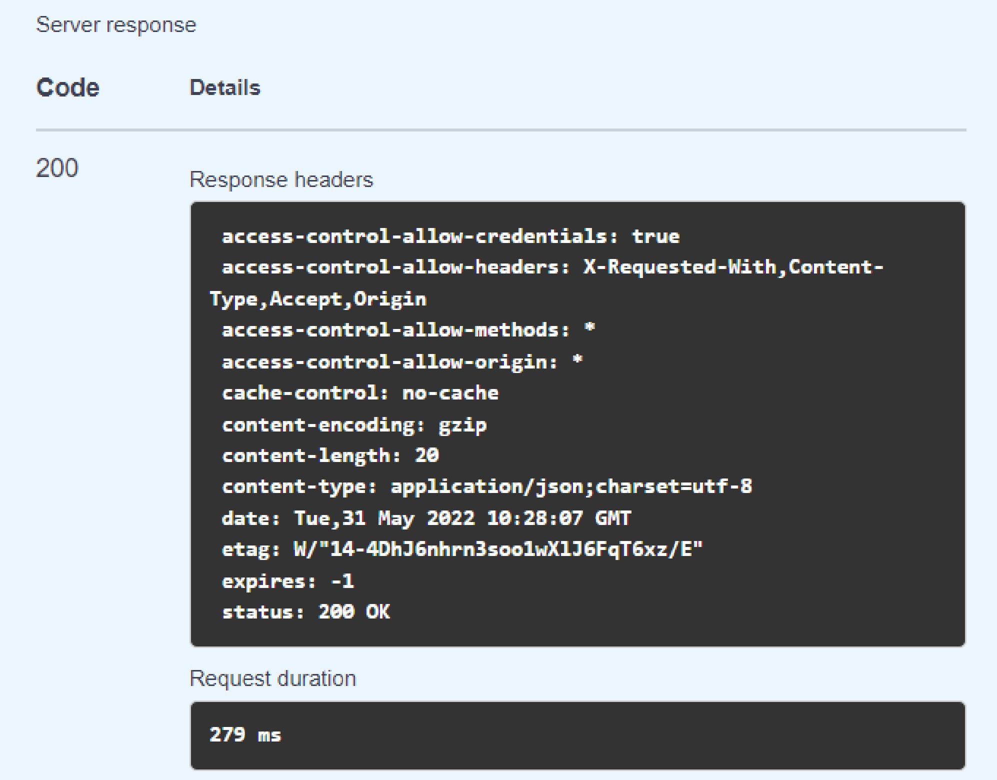


GET `/notifications/{patientId}`

FR3 - The notification system must be able to notify the users.

Name	Description
patientId * required integer($int64) *(path)*	ID of the patient requesting his notification. 87654321

Figure 61: Explanation and presentation of the resource parameters



Server response

Code Details

200

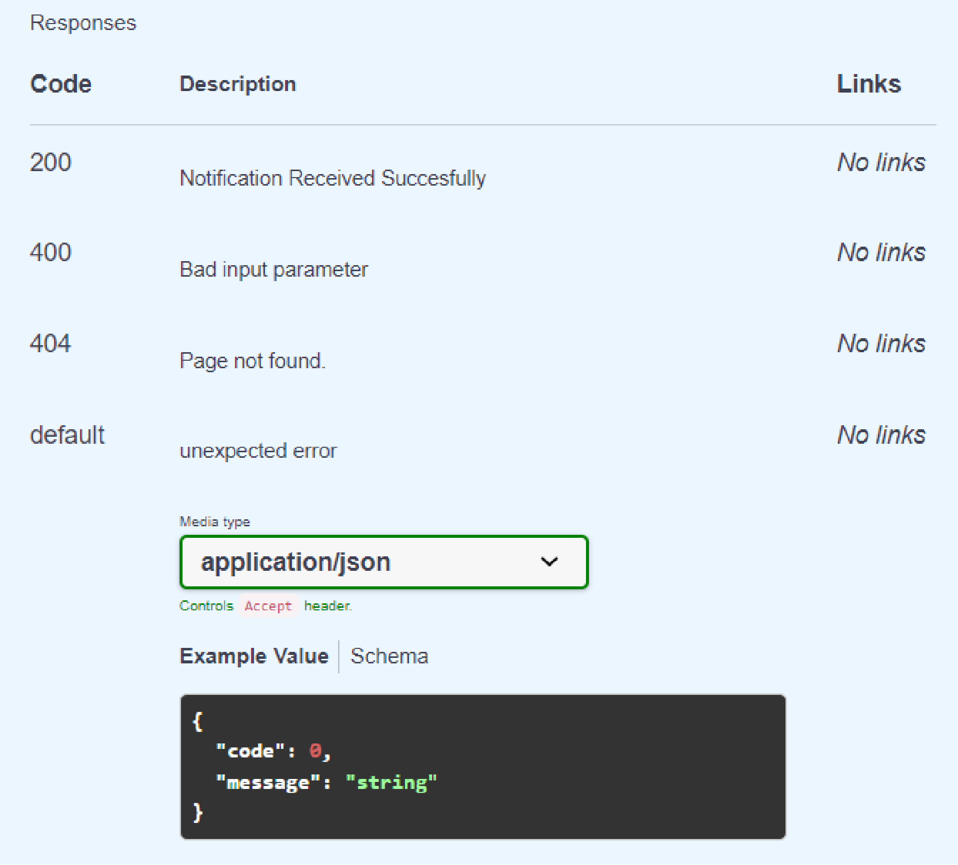
Response headers

```
 access-control-allow-credentials: true
 access-control-allow-headers: X-Requested-With,Content-Type,Accept,Origin
 access-control-allow-methods: *
 access-control-allow-origin: *
 cache-control: no-cache
 content-encoding: gzip
 content-length: 20
 content-type: application/json;charset=utf-8
 date: Tue,31 May 2022 10:28:07 GMT
 etag: W/"14-4DhJ6nhrn3soo1wX1J6FqT6xz/E"
 expires: -1
 status: 200 OK
```

Request duration

```
279 ms
```

Figure 62: endpoint testing



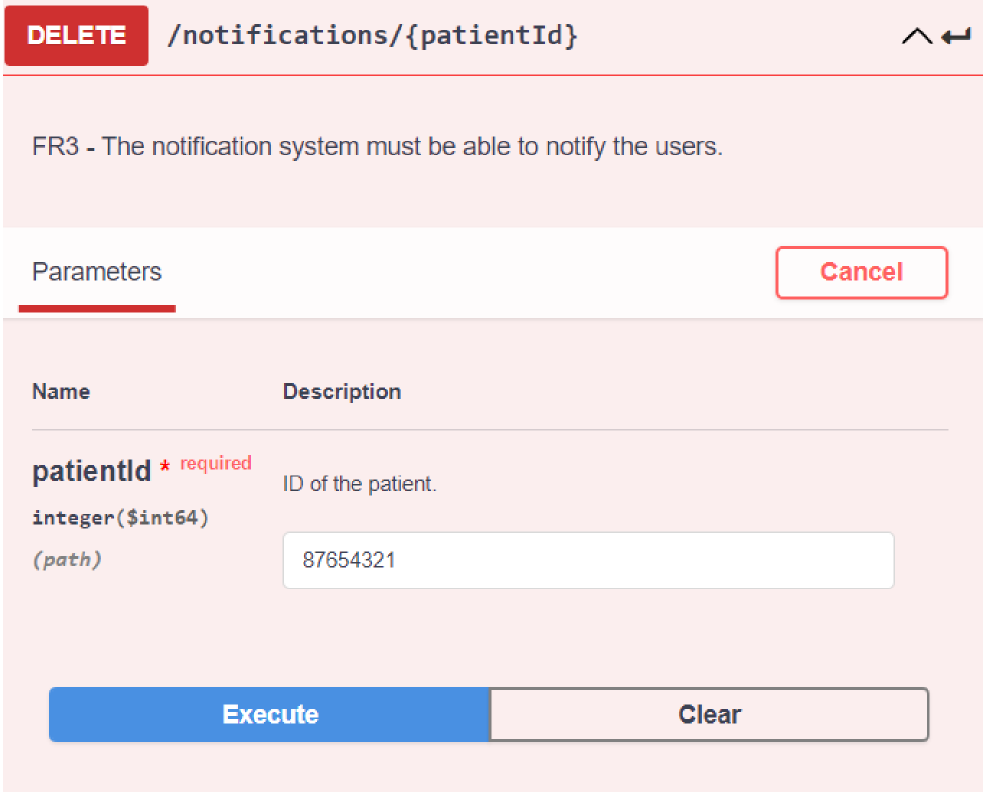
Responses

Code	Description	Links
200	Notification Received Succesfully	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

```
{
  "code": 0,
  "message": "string"
}
```

Figure 63: System responses

2.2.7.3 Endpoint DELETE of the resource notifications, for specific patientId.

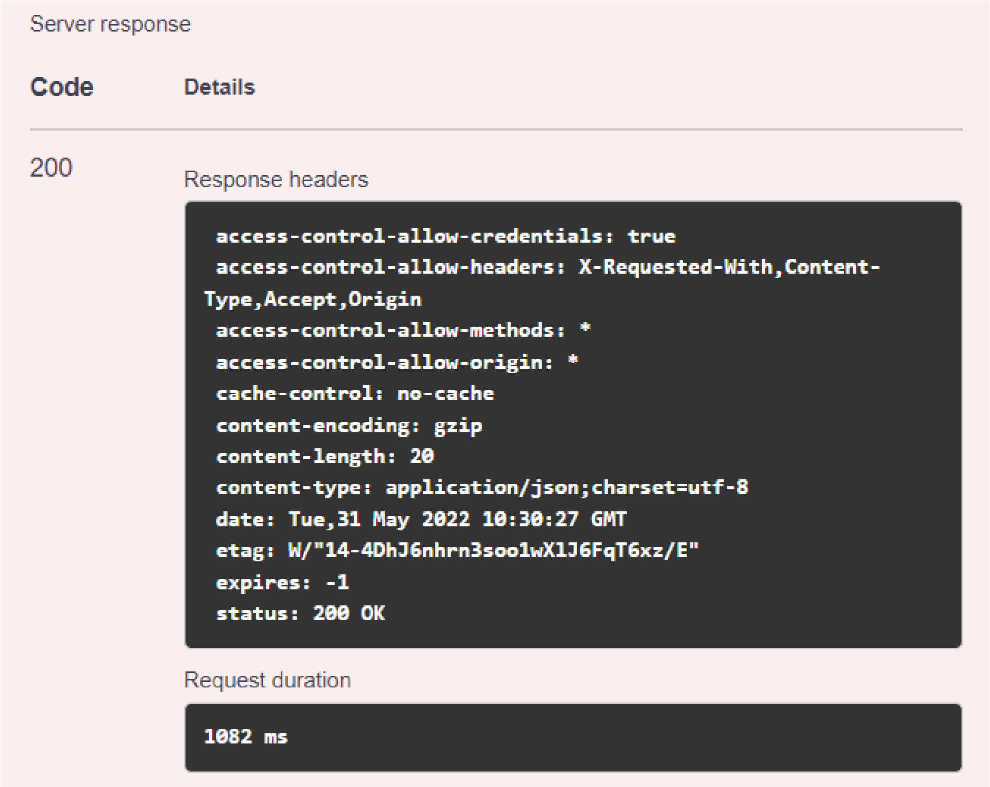


/notifications/{patientId}

FR3 - The notification system must be able to notify the users.

Name	Description
patientId * required integer($int64) *(path)*	ID of the patient. 87654321

Figure 64: Explanation and presentation of the resource parameters


Server response

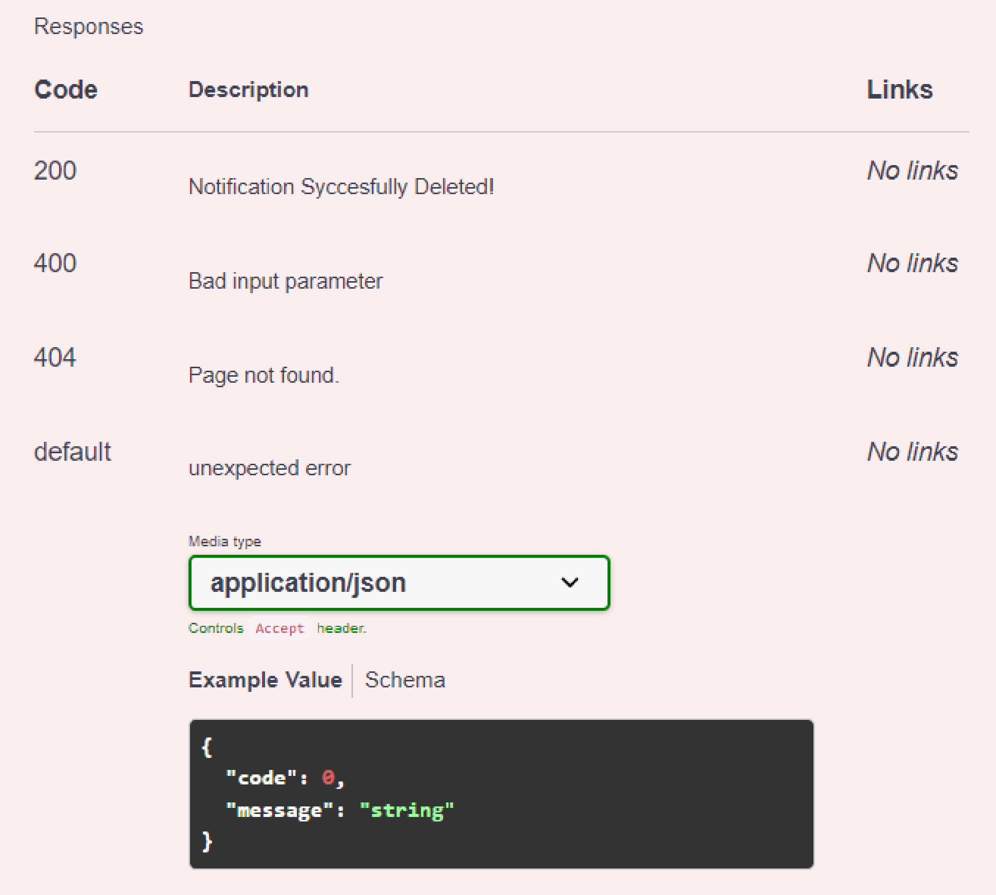
Code	Details
200	Response headers

```
 access-control-allow-credentials: true
 access-control-allow-headers: X-Requested-With,Content-
Type,Accept,Origin
 access-control-allow-methods: *
 access-control-allow-origin: *
 cache-control: no-cache
 content-encoding: gzip
 content-length: 20
 content-type: application/json;charset=utf-8
 date: Tue,31 May 2022 10:30:27 GMT
 etag: W/"14-4DhJ6nhrn3soo1wX1J6FqT6xz/E"
 expires: -1
 status: 200 OK
```

Request duration

```
1082 ms
```

Figure 65: endpoint testing



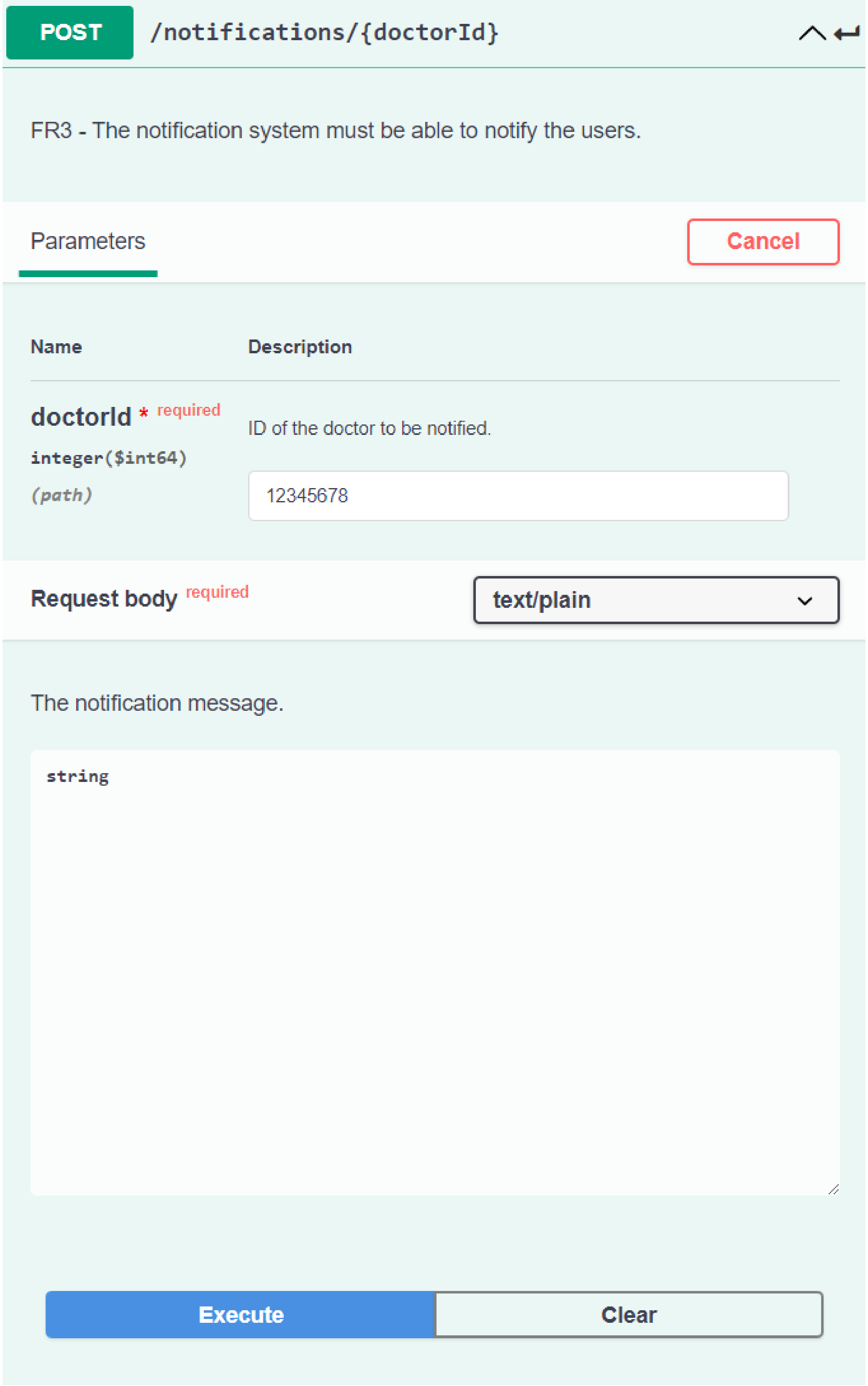
Responses

Code	Description	Links
200	Notification Syccesfully Deleted!	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

```
{
  "code": 0,
  "message": "string"
}
```

Figure 66: System responses

2.2.7.4 Endpoint POST of the resource notifications, for specific doctorId.



POST /notifications/{doctorId}

FR3 - The notification system must be able to notify the users.

Parameters

Cancel

Name	Description
doctorId * required integer($int64) (path)	ID of the doctor to be notified. 12345678

Request body required

text/plain

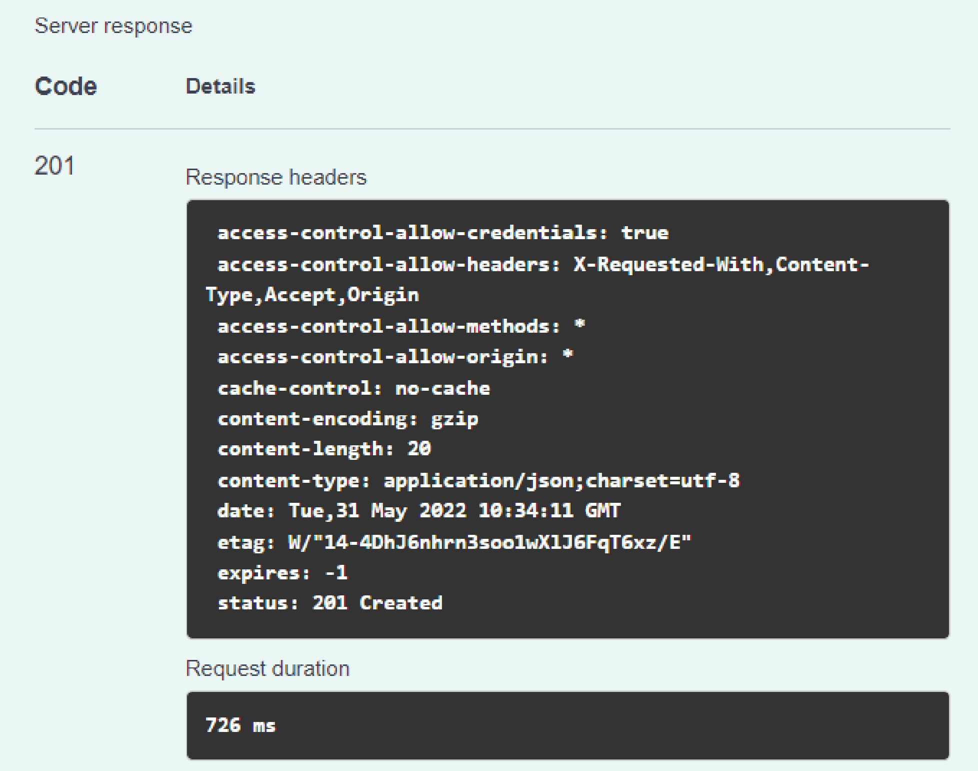
The notification message.

```
string
```

Execute

Clear

Figure 67: Explanation and presentation of the resource parameters



Server response

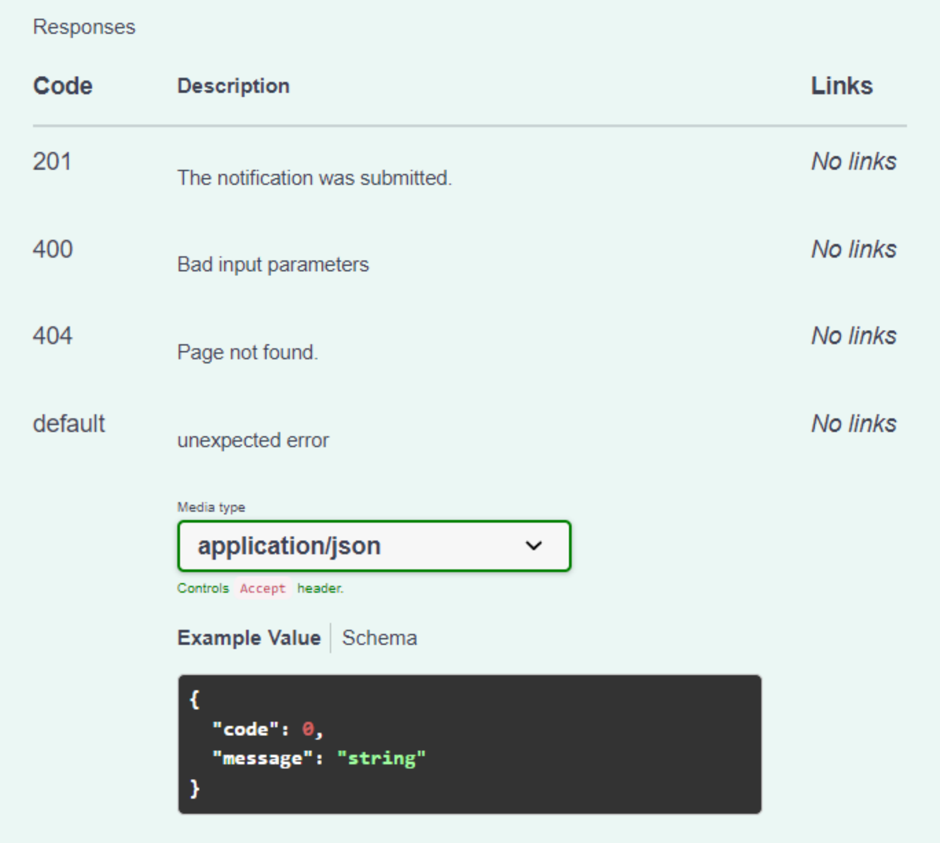
Code	Details
201	Response headers

```
 access-control-allow-credentials: true
 access-control-allow-headers: X-Requested-With,Content-
Type,Accept,Origin
 access-control-allow-methods: *
 access-control-allow-origin: *
 cache-control: no-cache
 content-encoding: gzip
 content-length: 20
 content-type: application/json;charset=utf-8
 date: Tue,31 May 2022 10:34:11 GMT
 etag: W/"14-4DhJ6nhrn3soo1wX1J6FqT6xz/E"
 expires: -1
 status: 201 Created
```

Request duration

```
726 ms
```

Figure 68: endpoint testing



Responses

Code	Description	Links
201	The notification was submitted.	*No links*
400	Bad input parameters	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

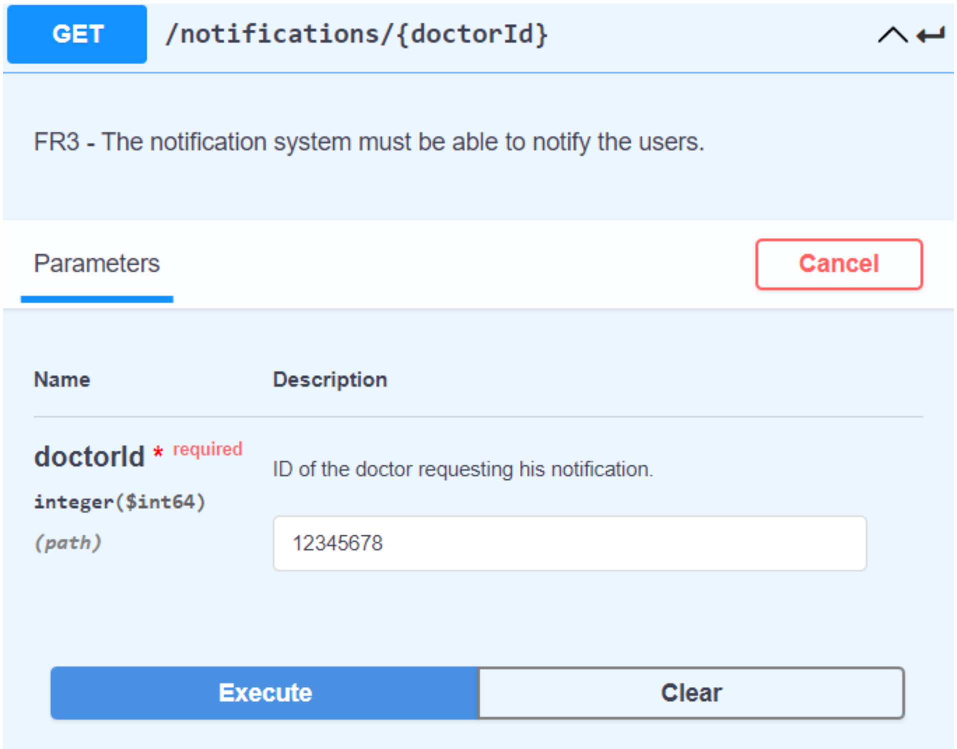
Media type
application/json
Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 69: System responses

2.2.7.5 Endpoint GET of the resource notifications, for specific doctorId.



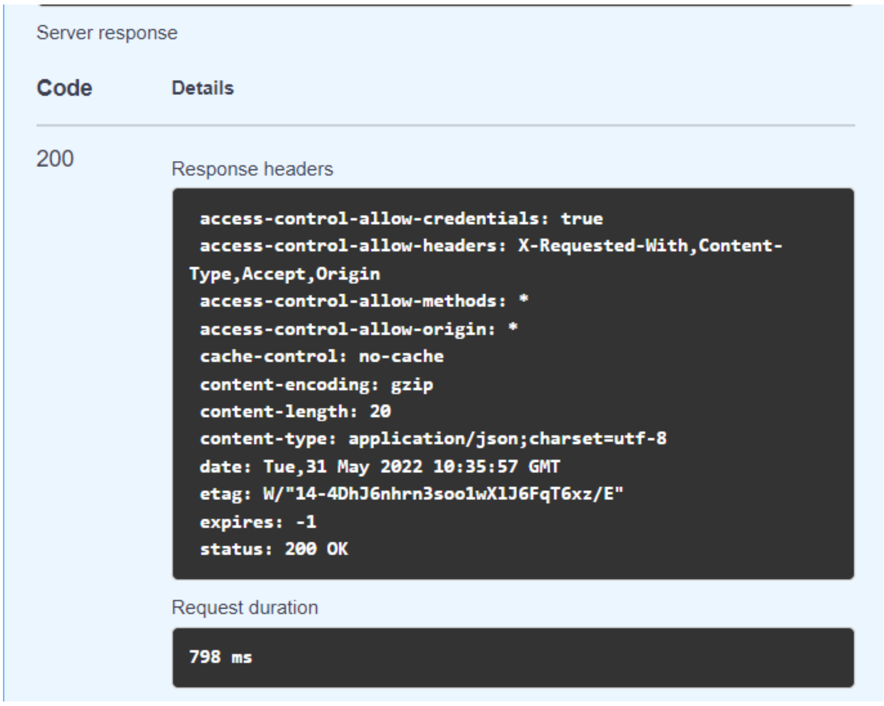
/notifications/{doctorId}

FR3 - The notification system must be able to notify the users.

Parameters

Name	Description
doctorId * required integer($int64) (path)	ID of the doctor requesting his notification. 12345678

Figure 70: Explanation and presentation of the resource parameters



Server response

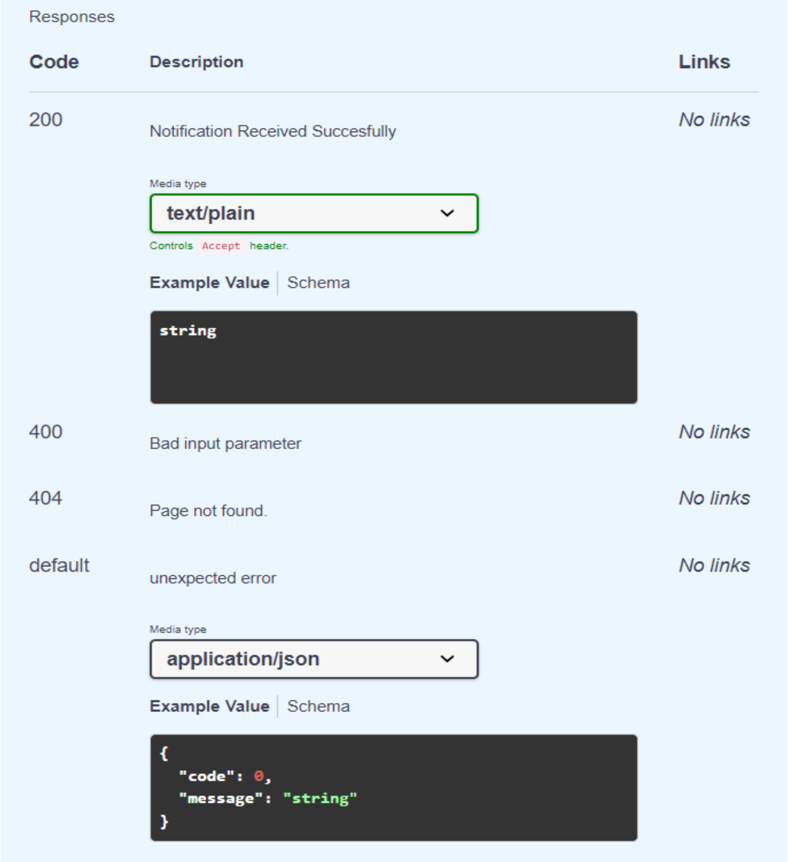
Code	Details
200	Response headers

```
 access-control-allow-credentials: true
 access-control-allow-headers: X-Requested-With,Content-
Type,Accept,Origin
 access-control-allow-methods: *
 access-control-allow-origin: *
 cache-control: no-cache
 content-encoding: gzip
 content-length: 20
 content-type: application/json;charset=utf-8
 date: Tue,31 May 2022 10:35:57 GMT
 etag: W/"14-4DhJ6nhrn3soolwX1J6FqT6xz/E"
 expires: -1
 status: 200 OK
```

Request duration

```
798 ms
```

Figure 71: endpoint testing



Responses

Code	Description	Links
200	Notification Received Succesfully	*No links*
400	Bad input parameter	*No links*
404	Page not found.	*No links*
default	unexpected error	*No links*

200 — Media type: text/plain. Controls `Accept` header.

Example Value | Schema

```
string
```

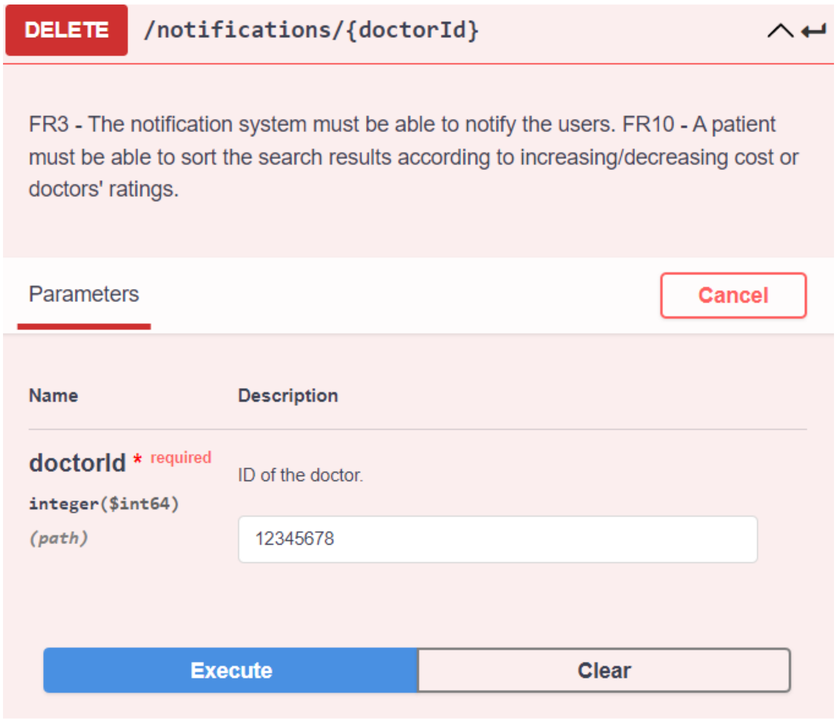
default — Media type: application/json

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 72: System responses

2.2.7.6 Endpoint DELETE of the resource notifications, for specific doctorId.

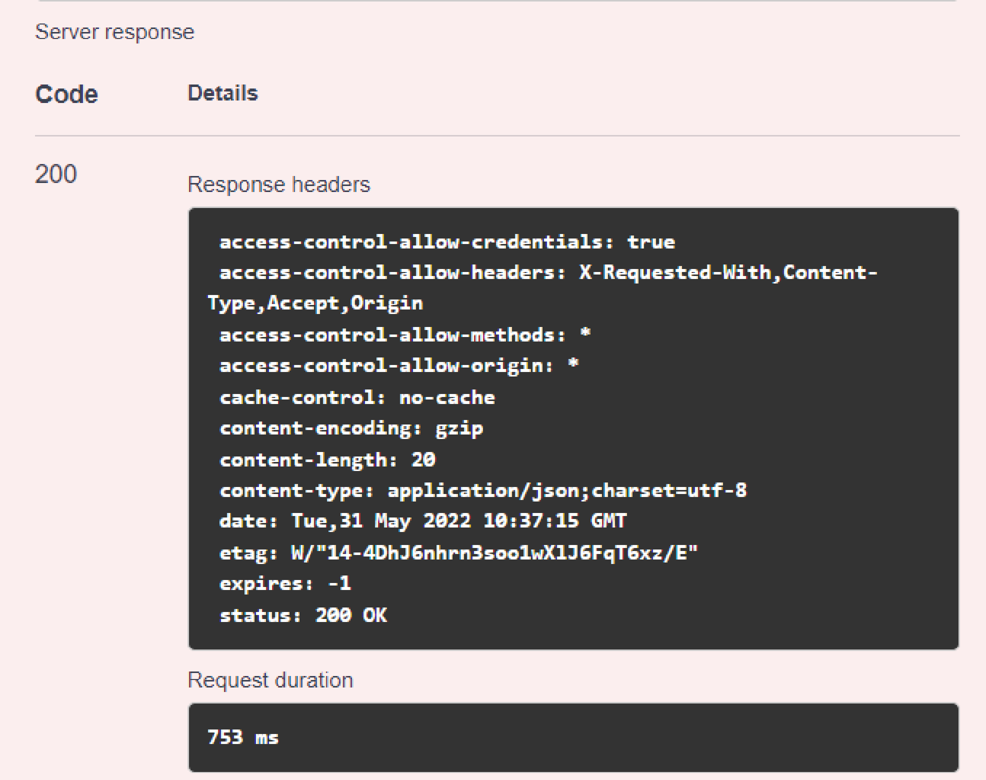


DELETE `/notifications/{doctorId}`

FR3 - The notification system must be able to notify the users. FR10 - A patient must be able to sort the search results according to increasing/decreasing cost or doctors' ratings.

Name	Description
doctorId * required `integer($int64)` *(path)*	ID of the doctor. 12345678

Figure 73: Explanation and presentation of the resource parameters



Server response

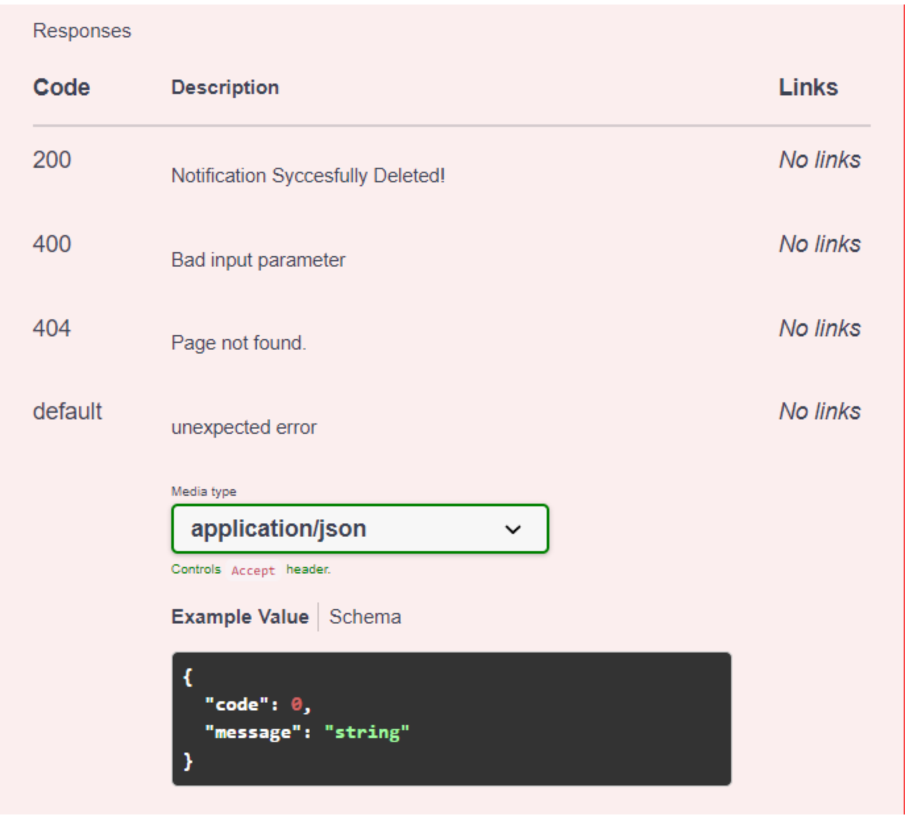
Code	Details
200	Response headers

```
 access-control-allow-credentials: true
 access-control-allow-headers: X-Requested-With,Content-
Type,Accept,Origin
 access-control-allow-methods: *
 access-control-allow-origin: *
 cache-control: no-cache
 content-encoding: gzip
 content-length: 20
 content-type: application/json;charset=utf-8
 date: Tue,31 May 2022 10:37:15 GMT
 etag: W/"14-4DhJ6nhrn3soo1wX1J6FqT6xz/E"
 expires: -1
 status: 200 OK
```

Request duration

```
753 ms
```

Figure 74: endpoint testing



Responses

Code	Description	Links
200	Notification Syccesfully Deleted!	No links
400	Bad input parameter	No links
404	Page not found.	No links
default	unexpected error	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```
{
  "code": 0,
  "message": "string"
}
```

Figure 75: System responses

Appendix I – Open Issues

- There is room for the implementation of additional design patterns for other non-functional and/or functional requirements. In this specific deliverable, it was requested to implement indicatively 3 design patterns.