

Mediscreen



Modèle de documentation de l'API REST

Liste des microservices :

PatientInformation : <http://localhost:8081> (swagger documentation)

PatientUI : <http://localhost:8083> (swagger documentation)

MedicalNotes : <http://localhost:8082> (swagger documentation)

PatientRisk : <http://localhost:8080> (swagger documentation)

Microservice : PatientInformation

<http://localhost:8081>
(swagger documentation)

POST /patient/add addNewPatient

DELETE /patient/delete/{patientId} deletePatient

GET /patient/getAll getAllPatients

GET /patient/getPatient getPatientsByName

GET /patient/getPatient/{patientId} getPatientById

PUT /patient/update/{patientId} updatePatient

Microservice : PatientInformation

Nom : getPatientById

URL : /patient/getPatient/{patientId}

Méthode : GET

Paramètres d'URL :

Obligatoire : patientId = Integer (path)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu :

```
{
  "address": "string",
  "age": 0,
  "dateOfBirth": "string",
  "familyName": "string",
  "givenName": "string",
  "patientId": 0,
  "phone": "string",
  "sex": "string"
}
```

Le patient recherché

Réponse (échec) :

Code : 404

Contenu : "The patient with id 100 was not found."

Exemple d'appel :

curl -X GET "http://localhost:8081/patient/getPatient/1" -H "accept: */*"

Remarques :

Microservice : PatientInformation

Nom : getPatientsByName

URL : /patient/getPatient

Méthode : GET

Paramètres d'URL :

Facultatifs : familyName = String (query)
givenName = String (query)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu :

```
[ {  
  "address": "string",  
  "age": 0,  
  "dateOfBirth": "string",  
  "familyName": "string",  
  "givenName": "string",  
  "patientId": 0,  
  "phone": "string",  
  "sex": "string"  
}]
```

Une liste des patients trouvés

Exemple d'appel :

```
curl -X GET "http://localhost:8081/patient/getPatient?familyName=family" -H "accept:  
*/*"
```

Remarques :

La recherche s'effectue par nom de famille et/ou prénom :

- si seul le nom de famille est fourni, la recherche s'effectue sur le nom de famille
- si seul le prénom est fourni, la recherche s'effectue sur le prénom
- si le nom de famille et le prénom sont fournis, la recherche s'effectue sur les 2 critères
- si aucun paramètre n'est fourni, une liste vide est renvoyée
- si aucun patient n'est trouvé, une liste vide est renvoyée

Microservice : PatientInformation

Nom : getAllPatients

URL : /patient/getAll

Méthode : GET

Paramètres d'URL :

aucun

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : [

```
{  
  "address": "string",  
  "age": 0,  
  "dateOfBirth": "string",  
  "familyName": "string",  
  "givenName": "string",  
  "patientId": 0,  
  "phone": "string",  
  "sex": "string"  
}
```

] Une liste des tous les patients enregistrés

Exemple d'appel :

```
curl -X GET "http://localhost:8081/patient/getAll" -H "accept: */*"
```

Remarques :

S'il n'y a aucun patient enregistré une liste vide sera renvoyée

Microservice : PatientInformation

Nom : addNewPatient

URL : /patient/add

Méthode : POST

Paramètres d'URL :

Obligatoire : patientDTO = PatientDTO (body)

Paramètres de données :

```
PatientDTO {  
    address      : String ( minLength : 0, maxLength : 256)  
    age          : Integer  
    dateOfBirth* : String (format yyyy-mm-dd)  
    familyName*  : String ( minLength : 0, maxLength : 50)  
    givenName*   : String ( minLength : 0, maxLength : 50)  
    patientId    : Integer  
    phone        : String (format ***-***-****)  
    sex*         : String ("M" or "F")  
}
```

Réponse (réussite) :

Code : 201

Contenu : "The patient named family given has been created."

Réponse (échec) :

Code : 400

Contenu : "The request is not correct : please verify the request's body."

Code : 400

Contenu : "There is a problem with the following given data :

- Please enter "M" or "F" for sex.
- The date of birth is not correct :
The format must be yyyy/mm/dd.
- Please enter a phone number with the following format: "***-***-****"

(Exemple de contenu, le contenu est adapté en fonction des erreurs de validation des données)

Exemple d'appel :

```
curl -X POST "http://localhost:8081/patient/add" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"address\": \"string\", \"dateOfBirth\": \"2000-02-02\", \"familyName\": \"family\", \"givenName\": \"given\", \"phone\": \"111-111-1111\", \"sex\": \"M\"}"
```

Remarques : Concernant les paramètres de données :

- Les champs « dateOfBirth », « familyName », « givenName » et « sex » sont obligatoires pour valider la requête.
- Les champs « address » et « phone » sont facultatifs
- Les champs « age » et « patientId » ne sont pas à fournir, ils sont calculés automatiquement. S'ils sont fournis ils seront ignorés et remplacés par les valeurs calculées.
- Des critères de validation conditionnent l'acceptation paramètres :
 - o address : ne doit pas dépasser 256 caractères
 - o familyName : ne doit pas dépasser 50 caractères
 - o givenName : ne doit pas dépasser 50 caractères
 - o sex : correspond à une énumération, les valeurs acceptées sont « M » ou « F »
 - o phone : doit être au format « ***-***-**** »
 - o dateOfBirth : -doit être au format « yyyy-mm-dd »
 - l'année doit être comprise entre 1900 et 3000
 - le mois doit être valide (<=12)
 - le jour du mois doit être valide (en fonction du mois concerné)

Microservice : PatientInformation

Nom : updatePatient

URL : /patient/update/{patientId}

Méthode : PUT

Paramètres d'URL :

Obligatoires : infoPatientToUpdateDTO = InfoPatientToUpdateDTO (body)
 patientId : Integer (path)

Paramètres de données :

InfoPatientToUpdateDTO {
 address : String (minLength : 0, maxLength : 256)
 dateOfBirth : String (format yyyy-mm-dd)
 familyName : String (minLength : 0, maxLength : 50)
 givenName : String (minLength : 0, maxLength : 50)
 phone : String (format ***-***-*****)
 sex : String ("M" or "F")}

Réponse (réussite) :

Code : 200

Contenu : "The patient number 5 has been update."

Réponse (échec) :

Code : 404

Contenu : "The patient with id 9 was not found, so it couldn't have been updated."

Code : 400

Contenu : "There is a problem with the following given data :

- Please enter "M" or "F" for sex.
- The date of birth is not correct :
The format must be yyyy/mm/dd.
- Please enter a phone number with the following format: "***-***-****"

(Exemple de contenu, le contenu est adapté en fonction des erreurs de validation des données)

Exemple d'appel :

```
curl -X PUT "http://localhost:8081/patient/update/5" -H "accept: */*" -H "Content-Type: application/json" -d "{ \"address\": \"string\", \"dateOfBirth\": \"2000-01-01\", \"familyName\": \"string\", \"givenName\": \"string\", \"phone\": \"111-111-1111\", \"sex\": \"M\"}"
```

Remarques : Concernant les paramètres de données :

- Les champs ne sont pas obligatoires
- Des critères de validation conditionnent l'acceptation paramètres :
 - address : ne doit pas dépasser 256 caractères
 - familyName : ne doit pas dépasser 50 caractères
 - givenName : ne doit pas dépasser 50 caractères
 - sex : correspond à une énumération, les valeurs acceptées sont « M » ou « F »
 - phone : doit être au format « ***-***-**** »
 - dateOfBirth : -doit être au format « yyyy-mm-dd »
 - l'année doit être comprise entre 1900 et 3000
 - le mois doit être valide (<=12)
 - le jour du mois doit être valide (en fonction du mois concerné)

Microservice : PatientInformation

Nom : deletePatient

URL : /patient/delete/{patientId}

Méthode : DELETE

Paramètres d'URL :

Obligatoire : patientId = Integer (path)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : "The patient 5 has been deleted"

Réponse (échec) :

Code : 404

Contenu : "The patient with id 100 was not found, so it couldn't have been deleted."

Exemple d'appel :

curl -X DELETE "http://localhost:8081/patient/delete/9" -H "accept: */*"

Remarques :

Microservice : MedicalNotes

<http://localhost:8082>
(swagger documentation)

POST /pathHistory/add/{patientId} addMedicalNote

DELETE /pathHistory/delete/{noteId} deleteMedicalNote

GET /pathHistory/getAllInformation/{patientId} getAllInformation

GET /pathHistory/getAllMedicalNotes/{patientId} getMedicalNotesByPatient

GET /pathHistory/getMedicalNote/{id} getMedicalNote

PUT /pathHistory/update/{noteId} updateMedicalNote

Microservice : MedicalNotes

Nom : addMedicalNote

URL : /patHistory/add/{patientId}

Méthode : POST

Paramètres d'URL :

Obligatoires : patientId = String (path)
 content = String[JSON] (body)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 201

Contenu : "A medical note have been added to the patient number 9."

Réponse (échec) :

Code : 400

Contenu : "The request is not correct : please verify the request's body."

Exemple d'appel :

```
curl -X POST "http://localhost:8082/patHistory/add/9" -H "accept: */*" -H "Content-Type: application/json" -d "content of the note"
```

Remarques :

Microservice : MedicalNotes

Nom : deleteMedicalNote

URL : /patHistory/delete/{noteId}

Méthode : DELETE

Paramètres d'URL :

Obligatoire : noteId = String (path)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : "The medical note number 62fdc162ac51da52514c77c9 have been deleted."

Réponse (échec) :

Code : 404

Contenu : "The medical note number 55 was not found."

Exemple d'appel :

curl -X DELETE "http://localhost:8082/patHistory/delete/55" -H "accept: */*"

Remarques :

Microservice : MedicalNotes

Nom : getAllInformation

URL : /patHistory/getAllInformation/{patientId}

Méthode : GET

Paramètres d'URL :

Obligatoire : patientId = String (path)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : String

(Une chaîne de caractères comportant le contenu de tous les notes concernant le patient recherché concaténées)

Réponse (échec) :

Code : 400

Contenu : "Bad request."

Exemple d'appel :

curl -X GET "http://localhost:8082/patHistory/getAllInformation/1" -H "accept: */*"

Remarques :

Le contenu de la réponse est une chaîne de caractères comportant la concaténation des tous les contenus de toutes les notes du patient. Cette chaîne de caractère est utilisée pour faciliter la recherche de termes déclencheurs dans le microservice PatientRisk.

S'il n'y a aucune note pour l'id de patient recherché une chaîne vide est retournée

Microservice : MedicalNotes

Nom : getMedicalNotesByPatient

URL : /patHistory/getAllMedicalNotes/{patientId}

Méthode : GET

Paramètres d'URL :

Obligatoire : patientId = String (path)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : [

```
{
  "id": "string",
  "noteContent": "string",
  "noteDate": "string",
  "patientId": "string"
},
{
  "id": "string",
  "noteContent": "string",
  "noteDate": "string",
  "patientId": "string"
}
]
```

(Une liste de toutes les notes trouvées pour le patient recherché.)

Réponse (échec) :

Code : 400

Contenu : "Bad request."

Exemple d'appel :

curl -X GET "http://localhost:8082/patHistory/getAllMedicalNotes/2" -H "accept: */*"

Remarques : Si aucune note n'est trouvée pour le patient recherché, une liste vide est renvoyée.

Microservice : MedicalNotes

Nom : getMedicalNote

URL : /patHistory/getAMedicalNotes/{id}

Méthode : GET

Paramètres d'URL :

Obligatoire : id = String (path)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu :

```
{
  "id": "string",
  "noteContent": "string",
  "noteDate": "string",
  "patientId": "string"
}
```

(La note recherchée.)

Réponse (échec) :

Code : 404

Contenu : "The medical note number 1 was not found."

Code : 400

Contenu : "Bad request."

Exemple d'appel :

curl -X GET

"http://localhost:8082/patHistory/getMedicalNote/62fcbbaaac51da52514c77c6" -H

"accept: */*"

Remarques :

Microservice : MedicalNotes

Nom : updateMedicalNote

URL : /patHistory/update/{noteId}

Méthode : GET

Paramètres d'URL :

Obligatoires : content = String (body)
 noteId = String[JSON] (path)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : "The medical note number 62fdc048ac51da52514c77c7 have been updated."

Réponse (échec) :

Code : 404

Contenu : "The medical note number 1 was not found."

Code : 400

Contenu : "The request is not correct : please verify the request's body."

Exemple d'appel :

```
curl -X PUT "http://localhost:8082/patHistory/update/62fdc048ac51da52514c77c7" -H  
"accept: */*" -H "Content-Type: application/json" -d "new content for the note"
```

Remarques :

Microservice : PatientRisk

<http://localhost:8080>

(swagger documentation)

GET

/assess/familyName getAssessByName

GET

/assess/id getAssessById

GET

/diabetesRisk/{id} getDiabetesRisk

Microservice : PatientRisk

Nom : getAssessByName

URL : /assess/familyName

Méthode : GET

Paramètres d'URL :

Obligatoire : familyName = String (query)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : ["string"]

(Liste de chaînes de caractères indiquant le nom, l'âge et le niveau de risque des patients.)

Réponse (échec) :

Code : 404

Contenu : "There is not any patient named 'familyName' "

Code : 400

Contenu : "The request is not correct : a request parameter is missing or wrong."

Exemple d'appel :

```
curl -X GET "http://localhost:8080/assess/familyName?familyName=familyName" -H  
"accept: */*"
```

Remarques : Le résultat de la requête est une liste comportant les informations pour tous les patients portant le même nom de famille.

Microservice : PatientRisk

Nom : getAssessById

URL : /assess/id

Méthode : GET

Paramètres d'URL :

Obligatoire : id = int (query)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : string

(Chaine de caractère indiquant le nom, l'âge et le niveau de risque du patient.)

Réponse (échec) :

Code : 404

Contenu : "The patient with id 56 was not found."

Code : 400

Contenu : "The request is not correct : a request parameter is missing or wrong."

Exemple d'appel :

curl -X GET "http://localhost:8080/assess/id?patId=1" -H "accept: */*"

Remarques :

Microservice : PatientRisk

Nom : getDiabetesRisk

URL : /diabetesRisk/{id}

Méthode : GET

Paramètres d'URL :

Obligatoire : id = int (path)

Paramètres de données :

aucun

Réponse (réussite) :

Code : 200

Contenu : string

(stringEnum:[BORDERLINE, EARLY_ONSET, IN_DANGER, NONE])

Réponse (échec) :

Code : 404

Contenu : "The patient with id 56 was not found."

Code : 400

Contenu : "The request is not correct : a request parameter is missing or wrong."

Exemple d'appel :

curl -X GET "http://localhost:8080/diabetesRisk/1" -H "accept: */*"

Remarques :

Microservice : PatientUI

<http://localhost:8083>
(swagger documentation)

GET	/ home
POST	/patient/addNote submitNewNote
GET	/patient/addPatient addNewPatient
POST	/patient/addPatient createNewPatient
GET	/patient/delete/{patientId} deletePatient
GET	/patient/deleteNote/{patientId}/{noteId} deleteNote
GET	/patient/getAllPatients allPatient
GET	/patient/getPatient getPatientsByName
GET	/patient/getPatient/{patientId} getPatient
POST	/patient/searchPatient searchPatientsByName
POST	/patient/updateNote submitUpdateNote
POST	/patient/updatePatient/{patientId} updatePatient