

Documentation

API de comparaison des risques

version prototype

Table des matières

1 Préambule.....	3
2 Documentation et client test.....	3
3 Opérations possibles.....	3
3.1 Lister les fiches patient.....	3
3.2 Récupérer une fiche patient en particulier.....	4
3.3 Créer une fiche patient.....	4
3.4 Mettre à jour une fiche patient.....	4
3.5 Supprimer une fiche patient.....	5
3.6 Récupérer toutes les valeurs des paramètres.....	5
3.7 Lister les noms des paramètres disponibles.....	6
3.8 Récupérer les données pour graphique de type ligne.....	6
3.9 Récupérer les données pour graphique de type nuage de point.....	7
4 Types rencontrés.....	7
4.1 Objet « Patient ».....	7
4.1.1 Description objet « patient » et valeurs possibles.....	7
4.1.2 Exemple pour objet « patient ».....	7
4.2 Objet « paramètre ».....	8
4.2.1 Description objet « paramètre ».....	8
4.2.2 Exemple pour objet « paramètre ».....	8
5 Exemple de retour.....	9
5.1 Exemple de retour « liste d'erreurs ».....	9
5.2 Exemple de retour « liste de paramètres disponibles ».....	9
5.3 Exemple de retour « pour graphique ».....	9
6 Exemple de code.....	10
6.1 Exemple de code pour « Lister les fiches patient ».....	10
6.2 Exemple de code pour « Récupérer une fiche patient en particulier».....	10
6.3 Exemple de code pour «Créer une fiche patient».....	10
6.4 Exemple de code pour « Mettre à jour une fiche patient».....	11
6.5 Exemple de code pour «Supprimer une fiche patient ».....	11
6.6 Exemple de code pour « Récupérer toutes les valeurs des paramètres».....	11
6.7 Exemple de code pour « Lister les noms des paramètres disponibles ».....	12
6.8 Exemple de code pour « Récupérer les données pour graphique de type ligne».....	12
6.9 Exemple de code pour « Récupérer les données pour graphique de type nuage de point »....	12

1 Préambule

Outil de comparaison des risques encourus par des patients sur base des paramètres médicaux, biométriques et d'assuétudes.

Le premier objectif de cette API est de gérer un ensemble de « fiche patient », cette fiche étant composée d'un certain nombre de paramètres. Il s'agira donc de pouvoir créer, lire, écrire ou supprimer un objet « patient » représentant cette fiche d'informations. (Gestion des données de type CRUD)

Le second objectif est de mettre à disposition un ensemble de paramètres afin de pouvoir établir des corrélations entre eux. L'ensemble de paramètres est constitué d'une part des paramètres renseignés dans la fiche patient et d'autre part de paramètres calculés à partir de ceux-ci.

L'API se charge de la synchronisation entre les fiches patients et des données mises à disposition ainsi que du calcul des paramètres manquant. (Un patient souhaitant être supprimé de la base de données verra ses données également supprimées des statistiques)

En plus de fournir les données brutes, l'API possède des fonction permettant de facilement comparer deux paramètres choisis sous forme de graphique. Les données sont optimisées pour être utilisées avec Google Chart mais peuvent facilement s'adapter à n'importe quel outil générant des graphiques.

2 Documentation et client test

Une page de documentation, permettant également de tester les fonctions est disponible.
{url_de_base}/swagger

3 Opérations possibles

3.1 Lister les fiches patient

Description	Liste les fiches patients
Méthode	GET
URL	{url_de_base}/api/patient[?search={search}&limit={limit}]
Paramètres URL	<i>search</i> [optionnel, string] (e.g. search=fred) - Filtre la liste par le début du nom de famille correspondant. <i>limit</i> [optionnel, integer] (e.g. limit=10) - Limite le nombre de Patient dans la liste .
Paramètres DATA	(aucun)
Succès	Code 200 Contenu : Tableau d'objet « patient », en format JSON Exemple : voir 4.1.2

Erreur	(aucun)
Exemple	Voir 6.1
Notes	(aucune)

3.2 Récupérer une fiche patient en particulier

Description	Récupère un patient particulier (par son id).
Méthode	GET
URL	{url_de_base}/api/patient/{id}
Paramètres URL	<i>id</i> [requis, integer] - identifiant de l'objet
Paramètres DATA	(aucun)
Succès	Code : 200 Contenu : « objet patient », en format JSON Exemple : voir 4.1.2
Erreur	Code 404 Contenu : description de l'erreur en format string
Exemple	Voir 6.2
Notes	(aucune)

3.3 Créer une fiche patient

Description	Crée une fiche patient
Méthode	POST
URL	{url_de_base}/api/patient
Paramètres URL	(aucun)
Paramètres DATA	« objet patient », en format JSON (id=0)
Succès	Code 200 Contenu : « objet patient » nouvellement créé (avec son id), en format JSON Exemple : voir 4.1.2
Erreur	Code 400 Contenu : description de l'erreur en format string OU Code 400 Contenu : Liste d'erreur, tableau de string en format JSON Exemple : voir 5.1
Exemple	Voir 6.3
Notes	L'id doit être égal à 0.

3.4 Mettre à jour une fiche patient

Description	Met à jour une fiche patient
-------------	------------------------------

Méthode	PUT
URL	{url_de_base}/api/patient/{id}
Paramètres URL	<i>id</i> [requis, integer] - identifiant de l'objet
Paramètres DATA	« objet patient », en format JSON Exemple : voir 4.1.2
Succès	Code 204 Contenu : aucun (No Content)
Erreur	Code 400 Contenu : description de l'erreur, en format string OU Code 400 Contenu : Liste d'erreur, tableau de string, en format JSON Exemple : Voir créer un patient OU Code 404 Contenu : description de l'erreur, en format string
Exemple	Voir 6.4
Notes	(aucune)

3.5 Supprimer une fiche patient

Description	Supprime une fiche patient
Méthode	DELETE
URL	{url_de_base}/api/patient/{id}
Paramètres URL	<i>id</i> [requis, integer] - identifiant de l'objet
Paramètres DATA	(aucun)
Succès	Code 204 Contenu : aucun (No Content)
Erreur	(aucun)
Exemple	Voir 6.5
Notes	(aucune)

3.6 Récupérer toutes les valeurs des paramètres

Description	Liste toutes les valeurs des paramètres. Utile dans le cas où le client souhaite traiter les données par lui-même.
Méthode	GET
URL	{url_de_base}/api/parametre
Paramètres URL	(aucun)
Paramètres DATA	(aucun)
Succès	Code 200

	Contenu : tableau de « objet paramètre », en format JSON Exemple : voir 4.2.2
Erreur	(aucun)
Exemple	Voir 6.6
Notes	(aucune)

3.7 Lister les noms des paramètres disponibles

Description	Donne la liste des paramètres disponibles.
Méthode	GET
URL	{url_de_base}/api/parametre/info
Paramètres URL	(aucun)
Paramètres DATA	(aucun)
Succès	Code 200 Contenu : liste des noms des paramètres disponibles, tableau de string, en format JSON Exemple : voir 5.2
Erreur	(aucun)
Exemple	Voir 6.7
Notes	(aucune)

3.8 Récupérer les données pour graphique de type ligne

Description	Retourne un ensemble de point destinés à alimenter un graphique de type "ligne" L'abscisse et l'ordonnée sont choisies parmi les paramètres.
Méthode	GET
URL	{url_de_base}/api/parametre/line_chart/{abscisse}/{ordonnee}
Paramètres URL	<i>abscisse</i> [requis, string] - Abscisse du graphique, doit être un des nom des paramètres disponibles <i>ordonnee</i> [requis, string] - Ordonnée du graphique, doit être un des nom des paramètres disponibles
Paramètres DATA	(aucun)
Succès	Code 200 Contenu : tableau (nombre d'éléments indéterminé) de tableau (deux éléments) de double. Exemple : voir 5.3
Erreur	Code 400 Contenu : description de l'erreur, en format string
Exemple	Voir 6.8
Notes	(aucune)

3.9 Récupérer les données pour graphique de type nuage de point

Description	Retourne un ensemble de point destinés à alimenter un graphique de type "nuage de point" L'abscisse et l'ordonnée sont choisies parmi les paramètres.
Méthode	GET
URL	{url_de_base}/api/parametre/scatter_chart/{abscisse}/{ordonnee}
Paramètres URL	<i>abscisse</i> [requis, string] - Abscisse du graphique, doit être un des nom des paramètres disponibles <i>ordonnee</i> [requis, string] - Ordonnée du graphique, doit être un des nom des paramètres disponibles
Paramètres DATA	(aucun)
Succès	Code 200 Contenu : tableau (nombre d'éléments indéterminé) de tableau (deux éléments) de double. Exemple : voir 5.3
Erreur	Code 400 Contenu : description de l'erreur, en format string
Exemple	Voir 6.9
Notes	(aucune)

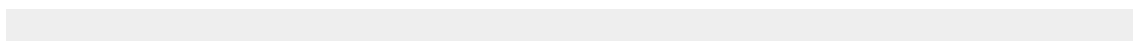
4 Types rencontrés

4.1 Objet « Patient »

4.1.1 Description objet « patient » et valeurs possibles

Nom	Description	Type	Limitations
id	Identifiant	integer	(aucune)
admin.prenom	Prénom	string	(aucune)
admin.nom	Nom	string	(aucune)
admin.date_de_naissance	Date de naissance	string	entre 1/1/1930 et 31/12/2017
admin.genre	Genre	string	"Male" ou "Female"
biometrie.poids	Poids	integer	entre 45 et 140
biometrie.taille	Taills	integer	entre 155 et 195
const_biologique.hbA1c	Hémoglobine glyquée	number	entre 0.05 et 0.12
const_biologique.cholesterol_total	Cholesterol total	integer	entre 130 et 320
const_biologique.cholesterol_HDL	Cholesterol HDL	integer	entre 20 et 100
parametres.pss	Pression sanguine systolique	integer	90 et 200
assuetudes.consommation_tabagique	Consommation tabagique	integer	entre 0 et 80

4.1.2 Exemple pour objet « patient »



```
{
  "id": 163,
  "admin": {
    "prenom": "Frédéric",
    "nom": "Liénard",
    "date_de_naissance": "1983-07-21T07:45:13Z",
    "genre": "Male"
  },
  "biometrie": {
    "poids": 78,
    "taille": 175
  },
  "const_biologique": {
    "hbA1c": 0.09,
    "cholesterol_total": 303,
    "cholesterol_HDL": 78
  },
  "parametres": {
    "pss": 101
  },
  "assuetudes": {
    "consommation_tabagique": 0
  }
}
```

4.2 Objet « paramètre »

4.2.1 Description objet « paramètre »

Nom	Description	Type
id	Identifiant	integer
age	Age	integer
bmi	Indice de masse corporelle (IMC)	integer
hbA1c	Hémoglobine glyquée	number
cholesterol_total	Cholesterol total	integer
cholesterol_HDC	Cholesterol HDL	integer
pss	Pression sanguine systolique	integer

4.2.2 Exemple pour objet « paramètre »

```
{
  "id": 1,
```



```
    "age": 55,  
    "bmi": 17,  
    "hbA1c": 0.1,  
    "cholesterol_total": 171,  
    "cholesterol_HDC": 103,  
    "pss": 106,  
    "consommation_tabagique": 60  
}
```

5 Exemple de retour

5.1 Exemple de retour « liste d'erreurs »

```
[  
  "Le genre doit être Male ou Female.",  
  "La date de naissance doit être comprise entre 1/1/1930 et 31/12/2017.",  
  "Le poids doit être compris entre 45 et 140 kg.",  
  "La taille doit être comprise entre 155 et 195 cm.",  
  "La PSS doit être comprise entre 90 et 200 mmHg.",  
  "La HbA1c doit être compris entre 0.05 et 0.12 ).",  
  "Le cholesterol HDL doit être compris entre 20 et 100 mg/dl).",  
  "Le cholesterol total doit être compris entre 130 et 320 mg/dl)."  
]
```

5.2 Exemple de retour « liste de paramètres disponibles »

Retour pour opération 3.7

Tableau de string.

```
[  
  "Age",  
  "BMI",  
  "HbA1c",  
  "Cholesterol_total",  
  "Cholesterol_HDC",  
  "PSS",  
  "Consommation_tabagique"  
]
```

5.3 Exemple de retour « pour graphique »

Retour pour opération 3.8 et 3.9

Tableau (nombre d'éléments indéterminé) de tableau (deux éléments) de double.

```
[  
  [ 0, 31.67 ],  
  [ 1, 30.38 ],  
  [ 2, 27.25 ]  
]
```

6 Exemple de code

Exemples d'utilisation avec curl.

6.1 Exemple de code pour « Lister les fiches patient »

```
curl -X GET --header 'Accept: application/json' 'http://localhost:59824/api/patient'
```

6.2 Exemple de code pour « Récupérer une fiche patient en particulier »

```
curl -X GET --header 'Accept: application/json' 'http://localhost:59824/api/patient/13'
```

6.3 Exemple de code pour « Créer une fiche patient »

```
curl -X POST --header 'Content-Type: application/json-patch+json' --header 'Accept: application/json' -d '{ \n  "id": 0, \n  "admin": { \n    "prenom": "Ruggiero", \n    "nom": "Semerad", \n    "date_de_naissance": "1957-06-14T19:47:34Z", \n    "genre": "Male" \n  }, \n  "biometrie": { \n    "poids": 97, \n    "taille": 166 \n  }, \n  "const_biologique": { \n    "hbA1c": 0.1, \n    "cholesterol_total": 280, \n    "cholesterol_HDL": 100 \n  }, \n  "parametres": { \n    "pss": 195 \n  } \n}'
```

```
}, \
"assuetudes": { \
  "consommation_tabagique": 0 \
} \
}' http://localhost:59824/api/patient'
```

6.4 Exemple de code pour « Mettre à jour une fiche patient »

```
curl -X PUT --header 'Content-Type: application/json-patch+json' -d '{ \
  "id": 10, \
  "admin": { \
    "prenom": "Puff", \
    "nom": "Di Bartolommeo", \
    "date_de_naissance": "1969-03-23T05:19:51Z", \
    "genre": "Male" \
  }, \
  "biometrie": { \
    "poids": 82, \
    "taille": 171 \
  }, \
  "const_biologique": { \
    "hbA1c": 0.05, \
    "cholesterol_total": 207, \
    "cholesterol_HDL": 30 \
  }, \
  "parametres": { \
    "pss": 148 \
  }, \
  "assuetudes": { \
    "consommation_tabagique": 60 \
  } \
}' http://localhost:59824/api/patient/10'
```

6.5 Exemple de code pour « Supprimer une fiche patient »

```
curl -X DELETE 'http://localhost:59824/api/patient/163'
```

6.6 Exemple de code pour « Récupérer toutes les valeurs des paramètres »

```
curl -X GET --header 'Accept: application/json' 'http://localhost:59824/api/parametre'
```

6.7 Exemple de code pour « Lister les noms des paramètres disponibles »

```
curl -X GET --header 'Accept: application/json'  
'http://localhost:59824/api/parametre/info'
```

6.8 Exemple de code pour « Récupérer les données pour graphique de type ligne »

```
curl -X GET --header 'Accept: application/json'  
'http://localhost:59824/api/parametre/line_chart/Age/BMI'
```

6.9 Exemple de code pour « Récupérer les données pour graphique de type nuage de point »

```
curl -X GET --header 'Accept: application/json'  
'http://localhost:59824/api/parametre/scatter_chart/Age/BMI'
```