

Documents de stratégie de test



Projet allocation de lits d'hôpital pour les urgences

Table des matières

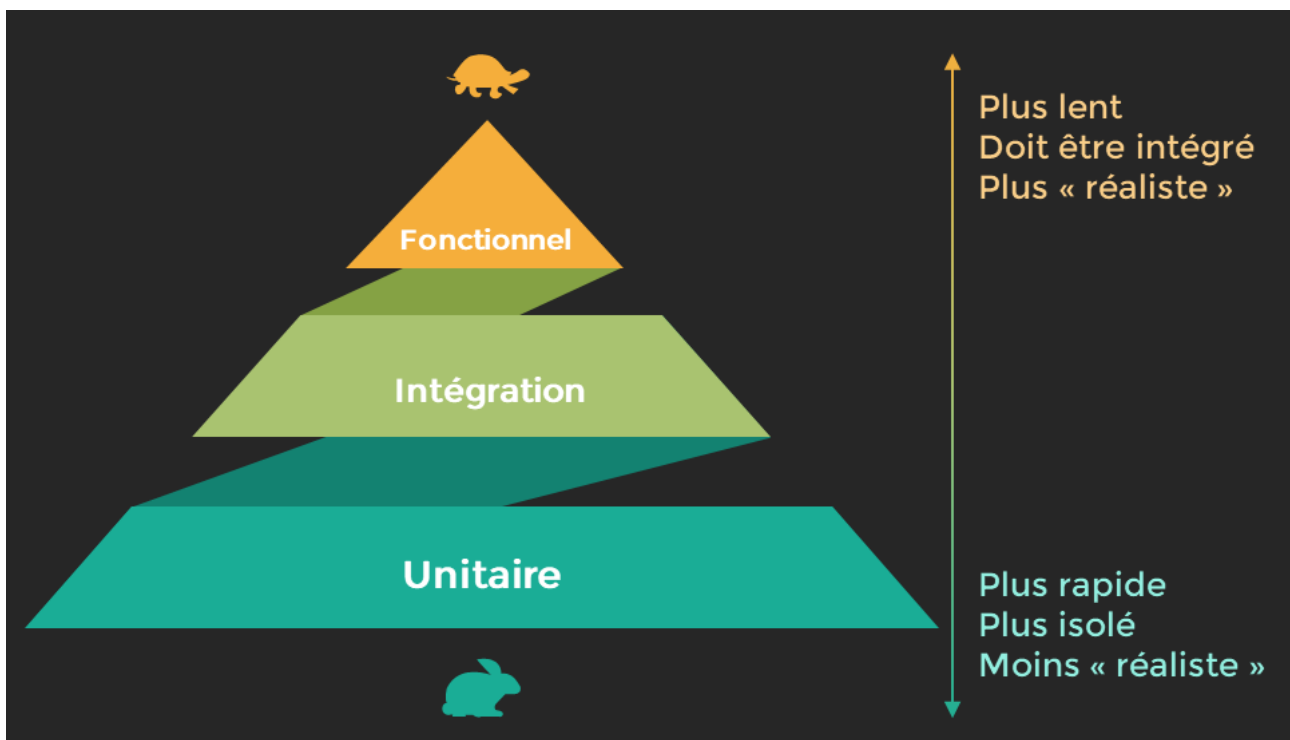
| | |
|---|----|
| Introduction..... | 3 |
| Environnement..... | 4 |
| Environnement de développement..... | 4 |
| Jeu de données de test..... | 4 |
| Bibliothèques et outils..... | 5 |
| JUnit 5..... | 5 |
| Apache JMeter..... | 6 |
| Code coverage..... | 6 |
| Mockito..... | 7 |
| Analyseur de code : Code Climate..... | 7 |
| Swagger 2 : Spécifications OpenAPI..... | 8 |
| Tests..... | 9 |
| Tests unitaires..... | 9 |
| Tests d'intégration..... | 10 |
| Tests fonctionnels..... | 11 |
| Stress tests..... | 12 |
| Rapport de tests..... | 13 |

Introduction

Ce document a pour objectif de donner une description la plus complète possible de la stratégie de test mis en place dans le cadre de la preuve de concept (PoC)

Il décrit les différents type de tests, leur portée et la méthodologie utilisé pour les mettre en place.

Enfin, il précisera les modalités d'automatisation des différents tests.



La stratégie de test fera intervenir les 3 types de tests présents dans la pyramide de test :

- Des tests fonctionnels
- Des tests d'intégrations
- Des tests fonctionnels

Environnement

Environnement de développement

Système d'exploitation : Ubuntu 21,10

Java : 11 'Temurin'

Interface de développement : IntelliJ IDEA Community Edition v2022.1.3

Maven : 4.0.0

SpringBoot : 2.7.0

Git : 2.32.0

Jeu de données de test

Les données de test utilisés sont basés sur les données du NHS (National Health Service) : <https://assets.nhs.uk/data/foi/Hospital.csv>

Ces données ont été modifiées pour ajouter une liste de spécialité aléatoirement tirée de la liste des spécialités, toujours basée sur des données du NHS, fusionner les différentes colonnes d'adresse, et en supprimer d'autres.

Enfin, une valeur entre 0 et 150 à été aléatoirement mise dans un paramètre pour spécifier le nombre de lits disponibles.

| OrganisationName | Longitude | Latitude | specialities | address | freedbed |
|--|------------------------|--------------------|---|---|----------|
| Walton Community Hospital - Virgin Care Services Ltd | -0.40604206919670105 | 51.379997253417969 | oral surgery,oral surgery,general practice (gp) 6 month training,endodontic | Rodney Road Walton-on-Thames Surrey KT12 3LD | 118 |
| Woking Community Hospital (Virgin Care) | -0.55628949403762817 | 51.315132141113281 | oral and maxillo-facial surgery,genito-urinary medicine,dermatology,intens | Heathside Road Woking Surrey GU22 7HS | 89 |
| North Somerset Community Hospital | -2.8471927642822266 | 51.43719482421875 | restorative dentistry,restorative dentistry,paediatric dentistry,audio vestib | North Somerset Community Hospital Old Street Clevedon Avo | 83 |
| Bridgewater Hospital | -2.2454688549041748 | 53.459743499755859 | clinical genetics,restorative dentistry,clinical pharmacology and therapeut | 120 Princess Road Manchester Greater Manchester M15 5AT | 0 |
| Kneeworth House | -3.0604055151343346E-2 | 52.078121185302734 | audio vestibular medicine,dermatology,oral medicine,intensive care medic | Old North Road Bassingbourn Royston SG8 5JP | 0 |
| Stockton Hall Hospital | -1.0025526285171509 | 53.995403289794922 | dermatology,dental and maxillofacial radiology,general med practitioner, re | Stockton Hall The Village Stockton On The Forest York North | 0 |
| Lakeside Hospital | -0.31203153729438782 | 52.198459625244141 | oral and maxillofacial pathology,periodontics,clinical genetics,prosthodont | The Lane Wyboston Bedford Bedfordshire MK44 3AS | 140 |
| Orthopaedics and Spine Specialist Hospital | -0.28302180767059326 | 52.604942321777344 | special care dentistry,periodontics,clinical oncology,clinical genetics,oral | 1 Stirling Way Bretton Peterborough Cambridgeshire PE3 8YA | 0 |
| Lee Mill Hospital | -3.9596600532531738 | 50.389484405517578 | oral medicine,additional dental specialties,special care dentistry,dermatol | Beech Road Ivybridge Plymouth Devon PL21 9HL | 0 |
| Mount Gould Hospital | -4.1127405166625977 | 50.378402709960938 | general practice (gp) 6 month training,emergency medicine,acute internal | Mount Gould Road Plymouth PL4 7QD | 137 |
| Nuffield Health | -1.8697808980941772 | 50.729755401611328 | prosthodontics,emergency medicine,audio vestibular medicine,clinical ne | Bournemouth Hospital 67 Landsdowne Road Bournemouth Do | 76 |
| Nuffield Health | 0.31283530592918396 | 51.623332977294922 | intensive care medicine,oral and maxillofacial pathology,endodontics,card | Brentwood Hospital Shenfield Road Brentwood Essex CM15 8 | 31 |

A partir de ces données, un fichier SQL a été généré, et sera utilisé à chaque démarrage de l'application, puis stocké dans une base de donnée H2 (Hibernate)

```
DROP TABLE IF EXISTS hospital;
CREATE TABLE hospital (
  id INT AUTO_INCREMENT NOT NULL,
  hospital_name VARCHAR(255),
  longitude FLOAT,
  latitude FLOAT,
  spec VARCHAR(255),
  address VARCHAR(255),
  freebed INTEGER,
  CONSTRAINT pk_hospital PRIMARY KEY (id)
);
INSERT INTO HOSPITAL(hospital_name,Longitude,Latitude,spec,address,freebed) VALUES ('Walton Community Hospital - Virgin Care Services Ltd','-0.40604206919670105','51.379997253417969','oral surgery,oral surgery,general practice (gp) 6 month training,endodontics,oral and maxillofacial pathology','Rodney Road Walton-on-Thames Surrey KT12 3LD',118);
...
```

Bibliothèques et outils

JUnit 5



JUnit 5 est un framework de test pour le langage Java. Il est intégré par défaut sur les environnements de développement (IDE)

Il intervient dans le cadre de la mise en place de tests unitaires

Apache JMeter



Apache JMeter est une application libre permettant d'effectuer des tests de performance d'application ou de serveurs.

Il fonctionne en simulant le comportement de plusieurs utilisateurs agissant de manière simultanée sur une application.

Dans notre cas, il sera utilisé pour effectuer les stress tests de l'application.

L'utilisation de JMeter dans le projet sera détaillé dans ce document, dans la partie « Stress tests »

Code coverage

| Coverage: coverage x | | | |
|------------------------|------------|-------------|--------------|
| Element | Class, % | Method, % | Line, % |
| com | 70% (7/10) | 65% (27/41) | 63% (76/119) |
| medhead | 70% (7/10) | 65% (27/41) | 63% (76/119) |
| api | 70% (7/10) | 65% (27/41) | 63% (76/119) |
| RunApi | 100% (1/1) | 0% (0/1) | 50% (1/2) |
| service | 100% (2/2) | 80% (12/15) | 75% (48/64) |
| HospitalService | 100% (1/1) | 83% (10/12) | 88% (22/25) |
| GisOperations | 100% (1/1) | 66% (2/3) | 66% (26/39) |
| repository | 100% (1/1) | 100% (1/1) | 100% (2/2) |
| HospitalRepositoryCust | 100% (0/0) | 100% (0/0) | 100% (0/0) |
| HospitalRepository | 100% (0/0) | 100% (0/0) | 100% (0/0) |
| customProperty | 100% (1/1) | 100% (1/1) | 100% (2/2) |
| model | 50% (1/2) | 80% (8/10) | 61% (8/13) |
| Itineraire | 0% (0/1) | 0% (0/2) | 0% (0/5) |
| Hospital | 100% (1/1) | 100% (8/8) | 100% (8/8) |
| exceptions | 33% (1/3) | 33% (1/3) | 33% (1/3) |
| NotYetImplemented | 0% (0/1) | 0% (0/1) | 0% (0/1) |
| NoParametersEntered | 0% (0/1) | 0% (0/1) | 0% (0/1) |
| NoHospitalFound | 100% (1/1) | 100% (1/1) | 100% (1/1) |
| controller | 100% (1/1) | 45% (5/11) | 45% (16/35) |

| Element | Missed Instructions | Cov. | Missed Branches | Cov. | Missed Cxty | Missed Lines | Missed Methods | Missed Classes |
|----------------------------|------------------------|-------|------------------------|------|-------------|--------------|----------------|----------------|
| com.medhead.api.service | <div><div></div></div> | 29 % | <div><div></div></div> | 22 % | 20 32 | 45 66 | 11 21 | 0 2 |
| com.medhead.api.model | <div><div></div></div> | 13 % | <div><div></div></div> | 0 % | 36 47 | 10 14 | 11 22 | 1 2 |
| com.medhead.api.controller | <div><div></div></div> | 1 % | <div><div></div></div> | 0 % | 19 20 | 34 35 | 11 12 | 0 1 |
| com.medhead.api.exceptions | <div><div></div></div> | 0 % | <div><div></div></div> | n/a | 3 3 | 6 6 | 3 3 | 3 3 |
| com.medhead.api | <div><div></div></div> | 37 % | <div><div></div></div> | n/a | 1 2 | 2 3 | 1 2 | 0 1 |
| com.medhead.api.repository | <div><div></div></div> | 100 % | <div><div></div></div> | n/a | 0 2 | 0 2 | 0 2 | 0 1 |
| Total | 749 of 934 | 19 % | 83 of 88 | 5 % | 79 106 | 97 126 | 37 62 | 4 10 |

Le code coverage, que ce soit via IntelliJ ou Jacoco, est un programme qui permet de mesurer le nombre de ligne de code qui sont couvertes par les tests. Le rapport Jacoco est disponible dans le dossier `./target/site` après build du projet dans l'environnement de développement.

Mockito



Tout comme Junit, Mockito est un framework de test pour les applications en Java.

Il permet de créer des « Mock », des objets fictifs qui vont intervenir dans le cadre des tests unitaires.

Analyseur de code : Code Climate

Showing 3 of 3 total issues

Method `getItineraireOSRM` has 31 lines of code (exceeds 25 allowed).

Consider refactoring.

OPEN

```
49     public static Iterable<Itineraire> getItineraireOSRM(float longCentre, float latCentre, It
50
51     String apiUrl = "https://api.openrouteservice.org/v2/directions/driving-car?api_key=";
52     String apiKey = "5b3ce3597851110001cf6248dc9db6fb880d46b7b63a907d42d826e5" ;
53     ArrayList<Itineraire> myItineraireList = new ArrayList<>();
```

Found in `src/main/java/com/medhead/api/service/GisOperations.java` - About 1 hr to fix

Similar blocks of code found in 2 locations. Consider refactoring.

OPEN

```
127     @GetMapping("/hospital/free/speciality/{spec}")
128     public Iterable<Hospital> getAllFreeBySpec(@PathVariable("spec") final String speciality)
129     ArrayList<Hospital> myHospitalList = (ArrayList<Hospital>) hospitalService.findByFreeb
130     if (!myHospitalList.isEmpty()){
131         return hospitalService.findByFreebedAndBySpecialities(speciality);
```

Found in `src/main/java/com/medhead/api/controller/HospitalController.java` and 1 other location - About 35 mins to fix

Similar blocks of code found in 2 locations. Consider refactoring.

OPEN

```
111     @GetMapping("/hospital/speciality/{spec}")
112     public Iterable<Hospital> getAllBySpec(@PathVariable("spec") final String speciality) thro
113     ArrayList<Hospital> myHospitalList = (ArrayList<Hospital>) hospitalService.findBySpeci
114     if (!myHospitalList.isEmpty()){
115         return hospitalService.findBySpecialities(speciality);
```

Found in `src/main/java/com/medhead/api/controller/HospitalController.java` and 1 other location - About 35 mins to fix

Swagger 2 : Spécifications OpenAPI



Swagger est un outil de documentation d'API. Il permet d'obtenir rapidement une documentation ordonnée et compréhensible.

Swagger est capable de représenter presque tous les services Web et les informations qui on trait à l'interface. La documentation évolue en parallèle que le système.

Une interface, swagger_ui est disponible via l'url : <http://localhost:9000/swagger-ui/>

The screenshot shows the Swagger UI interface. At the top, there's a dark header with the Swagger logo and "Supported by SMARTBEAR" on the left, and "Select a definition" with a dropdown menu showing "default" on the right. Below the header, the main content area has a title "PoC API HeadMed 0.1" with a version badge. Underneath, it says "[Base URL: localhost:9000/]" and "http://localhost:9000/v2/api-docs". A subtitle reads "Api pour le PoC HeadMed". There are two expandable sections: "basic-error-controller" (Basic Error Controller) and "hospital-controller" (Hospital Controller). The "hospital-controller" is expanded, showing a list of API endpoints with their HTTP methods and paths: GET /handleRequest, HEAD /handleRequest, POST /handleRequest, PUT /handleRequest, DELETE /handleRequest, OPTIONS /handleRequest, PATCH /handleRequest, and GET /app-info handleAppInfoRequest.

Tests

Tests unitaires

Les tests unitaires consistent en des tests d'unités plus restreintes du code.

Chaque fonctionnalité est testé de manière isolée pour vérifier que le comportement se déroule comme convenu.

Comme le montre la pyramide des tests, ils sont très rapides et simples, pour le programme de test, à effectuer.

Les tests unitaires ont été mis en place pour les méthodes déclarées manuellement dans la classe *HospitalRepositoryCustom* :

| Test | Donnée | Attendu |
|----------------------------------|---|-----------------------|
| findByFreebed() | Data.sql | Liste de 859 réponses |
| findBySpecialities() | Data.sql Spécialité imposée : <i>pharmacology</i> | Liste de 200 réponses |
| findByFreebedAndBySpecialities() | Data.sql Spécialité imposée : <i>pharmacology</i> | Liste de 132 réponses |

La suite des tests unitaires sera ceux qui vont impacter la classe de service *HospitalService*

Pour ces tests, il ne sera pas utilisé les données du fichier *data.sql*, mais 3 objets de type *Hospital* :

```
Hopital1 : {
id = 1,
name = Cromwell Hospital,
longitude = -1.21137082993984222
latitude = 51.766518859863281,
spécialités : audio vestibular medicine,prosthodontics,endocrinology and diabetes mellitus,restorative dentistry
Address = 164-178 Cromwell Road London SW5 0TU
Freebed = 145
}
Hopital2 : {
id = 1,
```

```

name = Princess Grace Hospital,
longitude = -0.15291328728199005
latitude = 51.495105743408203,
spécialités : pharmacology, acute internal medicine,oral and maxillo-facial surgery,paediatric dentistry,clinical
neurophysiology
Address = 42-52 Nottingham Place London W1U 5NY
Freebed = 0
}
Hopital3 : {
id = 1,
name = Spire Tunbridge Wells Hospital,
longitude = 0.1881597638130188
latitude = 51.134513854980469,
spécialités : gastroenterology,oral medicine,restorative dentistry,additional dental specialties
Address = Tunbridge Wells Hospital Fordcombe Road Fordcombe Tunbridge Wells TN3 0RD
Freebed = 107
}

```

| Test | Donnée | Attendu |
|-------------------------|------------------|---------------------------------|
| getAllHospital() | Liste d'hopitaux | Liste de 3 réponses |
| getHospital() | Liste d'hopitaux | Le nom de l'hôpital avec l'ID 1 |
| findByFreebed() | Liste d'hopitaux | Liste de 2 réponses |
| getAllHospitalInRange() | Liste d'hopitaux | Liste de 1 réponse |

Tests d'intégration

Les tests d'intégration sont là pour vérifier que les différentes unités du code (classes, méthodes) fonctionnent correctement ensemble, comme prévu. Ils viennent après les tests unitaires, et sont plus fiables sur le bon fonctionnement de l'application final, du fait de la vérification des interactions entre les unités.

Les tests d'intégrations ont été mis en place pour les différentes méthodes du contrôleur via les appels de l'API:

| Test | Donnée | Attendu |
|-----------|----------|---|
| /hospital | Data.sql | <ul style="list-style-type: none"> Liste de 1290 réponses Champ « name » de la première réponse : « Walton Community Hospital - Virgin Care » |

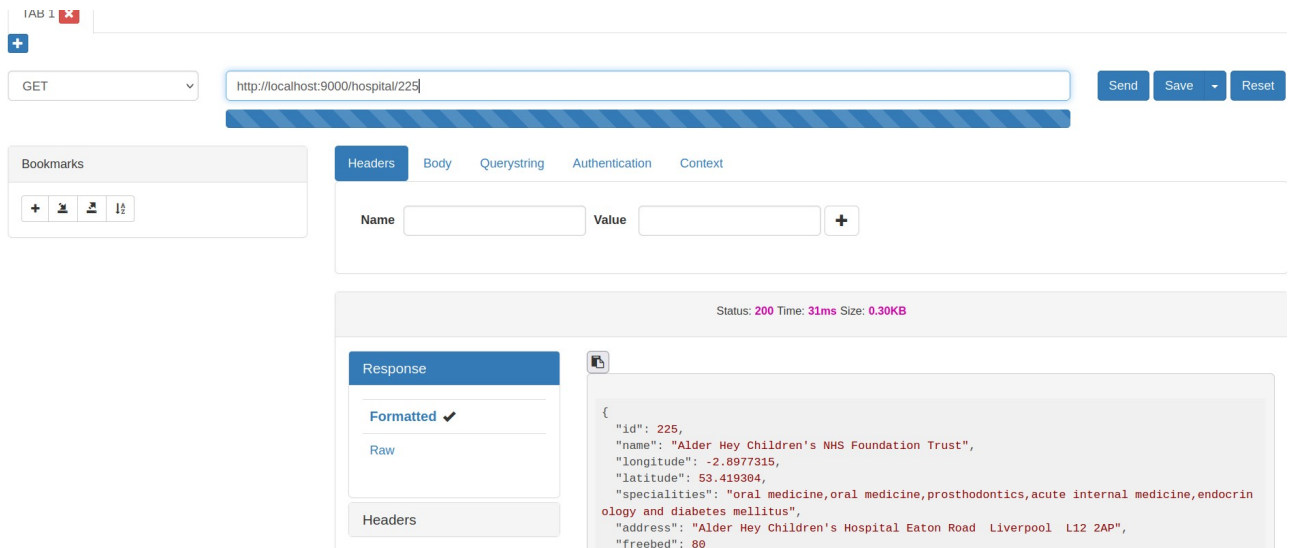
| | | |
|--|--|--|
| | | <i>Services Ltd »</i> <ul style="list-style-type: none"> Champ « <i>address</i> » de la seconde réponse : « <i>Heathside Road Woking Surrey GU22 7HS</i> » Champ « <i>freebed</i> » de la troisième réponse : 83 |
| /hospital/4 | Data.sql Id imposé : 4 | <ul style="list-style-type: none"> Champ « <i>Id</i> » : 4 Champ « <i>name</i> » : « <i>Bridgewater Hospital</i> » Champ « <i>address</i> » : « <i>120 Princess Road Manchester Greater Manchester M15 5AT</i> » Champ « <i>freebed</i> » : 0 |
| /hospital/speciality/ pharmacology | Data.sql Spécialité imposée : pharmacology | <ul style="list-style-type: none"> Liste de 200 réponses Champ « <i>name</i> » de la première réponse : « <i>Bridgewater Hospital</i> » Champ « <i>address</i> » de la 7^e réponse : « <i>The Alexandra Hospital Mill Lane Cheadle Cheshire SK8 2PX</i> » Champ « <i>freebed</i> » de la 4^e réponse : 0 |
| /hospital/free/ speciality/ pharmacology | | <ul style="list-style-type: none"> Liste de 132 réponses Champ « <i>name</i> » de la première réponse : « <i>Nuffield Health</i> » Champ « <i>address</i> » de la 7^e réponse : « <i>Benslow Lane Hitchin Hertfordshire SG4 9QZ</i> » Champ « <i>freebed</i> » de la 4^e réponse : 29 |
| /hospital/400258 | Data.sql | Status : Not found |
| /nearest/-1.8/51.25/2/ pharmacology | Data.sql | Status : Not found |

Tests fonctionnels

Les tests fonctionnels ont été réalisés sur navigateur avec le plugin pour le navigateur *Mozilla Firefox « Resting »*, version 1.4.0

Les différents tests ont été passés avec le démarrage de l'API dans l'environnement de développement (Intellij IDEA) ainsi que qu'après démarrage de l'image docker, ce qui a permis une validation du bon fonctionnement de celle-ci.

Ces tests ont couvert toutes les adresses présentes dans la classe *HospitalController* de l'API.



GET Send Save Reset

Bookmarks

Headers Body Querystring Authentication Context

Name Value +

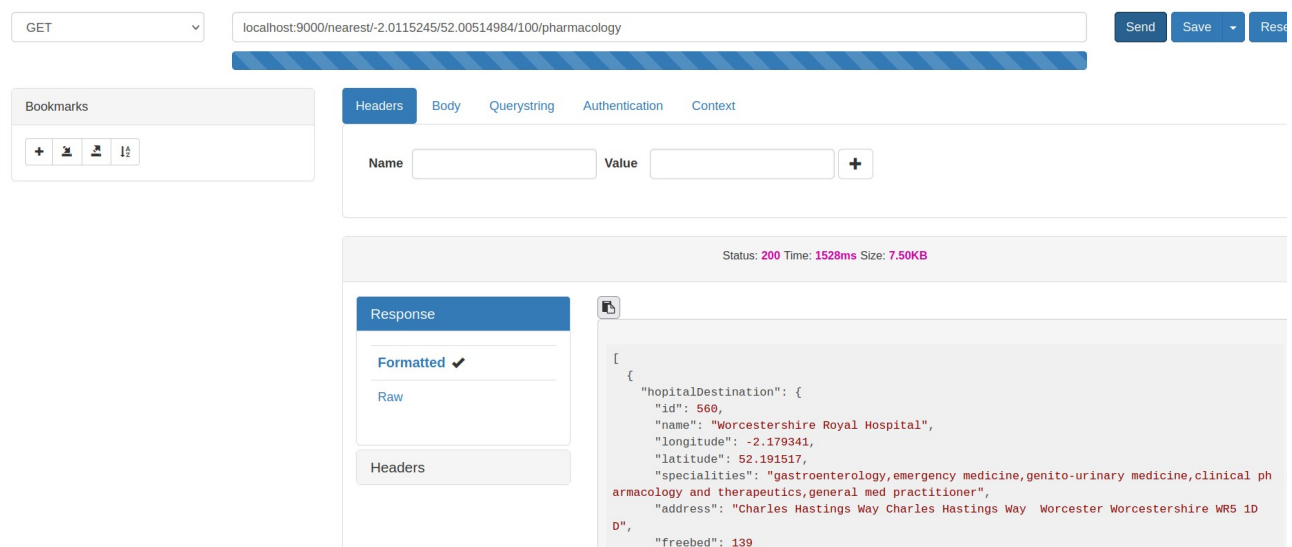
Status: 200 Time: 31ms Size: 0.30KB

Response

Formatted ✓ Raw

Headers

```
{
  "id": 225,
  "name": "Alder Hey Children's NHS Foundation Trust",
  "longitude": -2.8977315,
  "latitude": 53.419384,
  "specialities": "oral medicine,oral medicine,prosthodontics,acute internal medicine,endocrinology and diabetes mellitus",
  "address": "Alder Hey Children's Hospital Eaton Road Liverpool L12 2AP",
  "freedbed": 80
}
```



GET Send Save Reset

Bookmarks

Headers Body Querystring Authentication Context

Name Value +

Status: 200 Time: 1528ms Size: 7.50KB

Response

Formatted ✓ Raw

Headers

```
[
  {
    "hospitalDestination": {
      "id": 560,
      "name": "Worcestershire Royal Hospital",
      "longitude": -2.179341,
      "latitude": 52.191517,
      "specialities": "gastroenterology,emergency medicine,genito-urinary medicine,clinical pharmacology and therapeutics,general med practitioner",
      "address": "Charles Hastings Way Charles Hastings Way Worcester Worcestershire WR5 1D",
      "freedbed": 139
    }
  }
]
```

Stress tests

Des tests de stress ont été mis en place en local avec l'outil JMeter.

Plusieurs scénarios ont été configurés, pour permettre de vérifier les contraintes de temps d'exécution, et ce pour vérifier entre autres le principe suivant : « *que nous obtenons un temps de*

réponse de moins de 200 millisecondes avec une charge de travail allant jusqu'à 800 requêtes par seconde, par instance de service »

Exemple pour le paramétrage suivant :

| Propriétés du groupe d'unités | |
|---|-----------------------------------|
| Nombre d'unités (utilisateurs) : | 800 |
| Durée de montée en charge (en secondes) : | 1 |
| Nombre d'itérations : | <input type="checkbox"/> Infini 1 |

| Libellé | # Echantillons | Moyenne | Médiane | 90% centile | 95% centile | 99% centile | Min | Max | % Erreur | Débit | Ko/sec |
|------------------|----------------|---------|---------|-------------|-------------|-------------|-----|------|----------|-----------|--------|
| GET Freebad R... | 800 | 71 | 18 | 154 | 200 | 1032 | 1 | 1041 | 0,00% | 503,5/sec | 719,8 |
| TOTAL | 800 | 71 | 18 | 154 | 200 | 1032 | 1 | 1041 | 0,00% | 503,5/sec | 719,8 |

Les résultats nous montrent une tenue de la charge, en local, avec un respect de la réponse de 200 millisecondes ou moins pour une charge de 800 requêtes par seconde.

Cependant, l'appel effectué ne fait pas intervenir l'API externe. En effet, le service proposé dans la PoC faisant appel a un prestataire externe pour la sélection de l'hôpital le plus proche, un appel vers cet API peut provoquer un dépassement du temps de réponse. Celui-ci pouvant être amélioré en fonction de l'offre choisit.

Rapport de tests

Les résultats des différents tests, aussi bien unitaires que d'intégration peuvent être retrouvés dans le détail de la pipeline CI/CD, dans le job de Build.

Dans le cas du développement du projet, un export des rapports sous forme d'artefact est possible Une modification du fichier de Github Actions sera à effectuer pour assurer cette prise en charge.

Dans le cadre d'un build sur un environnement de développement, les résultats des tests unitaires peuvent être retrouvés dans le dossier `./target/surefire_report` et `./target/sfailsafe_report`, où un fichier texte est généré par classe, avec le nombre de tests effectués, le nombre d'échecs, d'erreurs, ou encore de tests non exécutés, et le temps passé à l'exécution des tests de la classe.

Test set: `com.medhead.api.repository.HospitalRepositoryCustomTest`

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 9.515 s - in `com.medhead.api.repository.HospitalRepositoryCustomTest`