

Rapport DAR

Victor ANTUNES DA SILVA, Ladislas HALIFA, Patrick TRAN

November 2016

1 Présentation

L'application CheapMeal permet à ses utilisateurs de partager des recettes personnelles. Là où notre application se démarque des autres applications similaires, c'est qu'elle introduit un côté économique. En effet, chaque fiche recette affiche le prix de la recette calculé à partir des prix minimums de chaque ingrédient. Cela se fait à partir d'appels effectués vers une API externe nommée Datagram (MasterCourses). Le côté social est aussi introduit par les profils utilisateur où les recettes ajoutées par l'utilisateur sont affichées, ainsi que par le système de notation de recette. L'application est disponible à l'adresse suivante: <http://vps197081.ovh.net:8080/CheapMeal/>

2 Analyse détaillée

2.1 Fonctionnalités

2.1.1 Connexion

Ce service permet à un utilisateur de créer une session valide pendant un temps donné sur le site et d'accéder aux autres fonctionnalités. L'utilisateur Joël (qui possède déjà un compte sur CheapMeal) est sur le formulaire de connexion. Il rentre son nom d'utilisateur : Joel43 et son mot de passe : steakFrites457! puis il appuie sur le bouton de connexion. S'il a rentré les bonnes informations, il est redirigé vers la page d'accueil, sinon la page lui indique un message d'erreur. De plus des cookies contenant la clé de session, son id utilisateur, ainsi que son nom d'utilisateur seront créés.

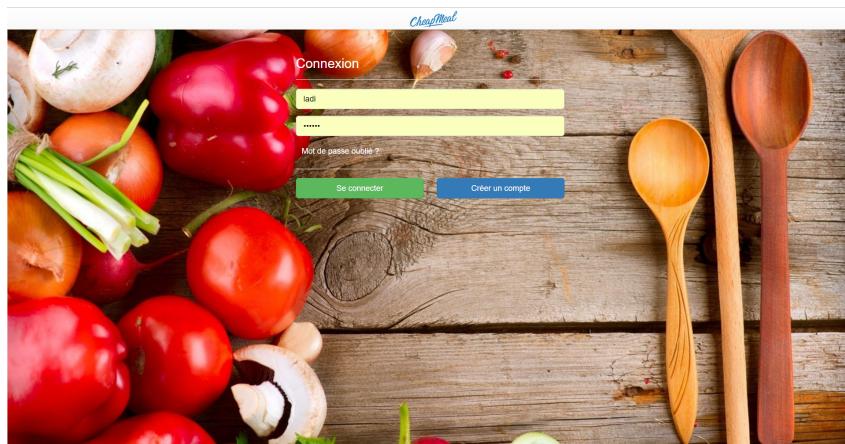


Figure 1: Formulaire de connexion

2.1.2 Inscription

Ce service permet à un utilisateur de s'inscrire sur CheapMeal lui permettant ensuite de pouvoir s'authentifier. Un utilisateur souhaite s'inscrire il rentre donc son nom et prénom, son email ainsi que son nom d'utilisateur et finalement son mot de passe. Si les champs sont correctement remplis, lorsqu'il valide son inscription il est redirigé vers la page de connexion et ses informations sont stockées coté serveur.

2.1.3 Récupérer mot de passe

Ce service permet à un utilisateur de pouvoir récupérer son mot de passe. L'utilisateur Joël a oublié son mot de passe, il appuie sur le bouton "Mot de passe oublié?", il renseigne ainsi son email et valide le formulaire. Il reçoit alors par mail un nouveau mot de passe provisoire attribué à son compte qu'il peut utiliser afin de se connecter.

2.1.4 Ajouter une photo de profil

Ce service permet à un utilisateur d'ajouter une photo de profil associée à son compte et visible par tous les utilisateurs. L'utilisateur Joël est connecté sur CheapMeal et souhaite ajouter une image de lui, il va dans l'onglet "Profil" et clique sur le bouton "Ajouter une photo" puis il choisit la photo qui lui convient. Ce changement sera reflété instantanément sur la page web.

2.1.5 Ajouter une recette

Ce service permet à un utilisateur de publier une recette. L'utilisateur Joël souhaite ajouter une recette, il clique sur le bouton "ajouter une recette" depuis l'accueil ou son profil, il renseigne les informations de la recette (ingrédients, quantités, mesures, étapes de préparation) puis valide le formulaire. Sa recette est ajoutée à la base de donnée et il est redirigé vers l'accueil.

The screenshot shows the 'Ajouter une nouvelle recette' (Add new recipe) form on the CheapMeal website. At the top, there's a note: 'Utilisez ce formulaire pour sauvegarder et partager vos recettes avec votre famille, vos amis et les autres membres de CheapMeal. Essayez d'être précis et clair pour que tous nos utilisateurs puissent réaliser votre recette sans problème !'. Below this, there's a 'Titre de la recette' (Title of the recipe) input field with placeholder text 'Choisissez un titre pour votre recette'. Underneath it are two buttons: 'Ajouter une photo' (Add a photo) with a camera icon and 'Ajouter une image' (Add an image). The next section is 'Liste d'ingrédients' (List of ingredients), which has columns for 'Ingrédient' (Ingredient), 'Quantité' (Quantity), and 'Mesure' (Measure). There's also a '+' button to add more ingredients. The final section is 'Préparation' (Preparation), with a text area for instructions and a '+' button to add more steps.

Figure 2: Formulaire d'ajout d'une recette

2.1.6 Supprimer une recette

Ce service permet de supprimer une recette si l'utilisateur authentifié est le propriétaire de celle-ci. Dans l'onglet "Profil" Bénédicte peut voir ses recettes, elle trouve que son "Riz aux fruits de mer" ne lui plaît plus. Elle clique sur l'icône en forme de corbeille qui permet de la supprimer, puis elle valide la suppression. La recette disparaît de la base de données et elle n'est plus affichée sur le site.

2.1.7 Afficher une recette

Ce service est appelé lorsque l'utilisateur souhaite afficher la page d'une recette. Dans la page d'accueil ou un profil (son propre profil ou celui d'un autre utilisateur), Jacques clique sur une recette et est redirigé vers la page décrivant la recette. Cette page contient les ingrédients (avec la quantité et les mesures), le prix de la recette calculé à partir des ingrédients, les étapes de préparation ainsi que la note moyenne attribuée à cette recette. De plus, elle contient les informations relatives à l'auteur qui sont son nom d'utilisateur et sa photo de profil.

2.1.8 Noter une recette

L'utilisateur Bénédicte regarde la recette "Poulet braisé", elle trouve que la recette n'est pas assez détaillée elle décide donc de mettre une note de 2 étoiles sur 5 (équivalent à 2/5). Lorsqu'elle valide son vote, la note moyenne de la recette est modifiée et son vote est pris en compte. Elle ne peut plus changer son avis.

2.1.9 Recherche d'une recette

Ce service est appelé lorsque l'utilisateur souhaite trouver une recette, que ce soit avec un mot contenu dans le titre, le nom d'utilisateur de l'auteur, un ingrédient ou bien un mot clé dans les étapes de préparation. L'utilisateur René désire trouver une recette de lapin chasseur, il entre dans la barre de recherche "lapin" ou "chasseur" ou "lapin chasseur", les résultats de la recherche s'affichent là où se trouvaient les recettes récentes et le top des recettes. Il peut cliquer sur les recettes pour afficher les pages détaillées de celles-ci.

2.1.10 Afficher le profil d'un utilisateur

L'utilisateur Bénédicte désire en savoir plus sur Joël, auteur de la recette "Poulet braisé", elle clique donc sur le lien menant à son profil dans la page de la recette. Elle est redirigée vers une page contenant toutes les informations de cet utilisateur. C'est à dire sa photo de profil, sa biographie, ainsi que toutes les recettes de Joël.

2.2 Modèle de données

2.2.1 SQL

Nous avons utilisé une base de donnée SQL pour stocker toutes les informations liées aux utilisateurs: les informations de comptes, les informations du profil, les informations de session. Nous avons donc 3 tables dont la structure est décrite dans la figure 3. Les tables Sessions et Profils sont liées à la table Utilisateurs grâce aux clés étrangères qui pointent vers l'id contenu dans la table Utilisateurs. Cela permet de retrouver facilement la session et le profil associé à un utilisateur. Nous avons choisi le système MySQL, libre et open-source, qui nous permet de visualiser facilement nos données avec son application web phpMyAdmin.

UTILISATEURS

Colonne	Type	Null	Défaut	Commentaires
id	int(11)	Non		
login	varchar(255)	Oui	<i>NULL</i>	
mail	varchar(255)	Oui	<i>NULL</i>	
mdp	varchar(255)	Oui	<i>NULL</i>	
nom	varchar(255)	Oui	<i>NULL</i>	
prenom	varchar(255)	Oui	<i>NULL</i>	

SESSIONS

Colonne	Type	Null	Défaut	Relié à
idSession	int(11)	Non		UTILISATEURS -> id
cleSession	varchar(255)	Oui	<i>NULL</i>	
dateExpiration	datetime	Oui	<i>NULL</i>	

PROFILS

Colonne	Type	Null	Défaut	Relié à
idProfil	int(11)	Non		UTILISATEURS -> id
bio	varchar(255)	Oui	<i>NULL</i>	

Figure 3: Structures des tables SQL utilisées

2.2.2 NoSQL avec MongoDB

Pour ce projet, nous avons utilisé une base de données MongoDB pour stocker les recettes et les ingrédients. La flexibilité d'une base de données NoSQL ainsi que la possibilité d'y coupler un outil de recherche ont orienté notre choix vers MongoDB. Des exemples de représentation des données (documents) dans la base MongoDB se trouvent sur la figure 4.

Ingrédients	Recettes
<pre>{ "_id": "5826295d96aa584d0800b69d", "nomIngredient": "chocolat noir", "ean": "7613033113697", "quantite": 200 } { "_id": "5826295e96aa584d0800b6ce", "nomIngredient": "fraise", "prix": 0.134939759, "quantite": 1 }</pre>	<pre>{ "_id": "582610d396aa583c1fb845b7", "titre": "Cr\u00e9pes", "auteur": { "idAuteur": 25, "loginAuteur": "max77" }, "ingredients": [{ "nomIngredient": "farine", "quantite": 150, "mesure": "g" }, { "nomIngredient": "oeufs frais", "quantite": 1, "mesure": "unit\u00e9(s)" }, { "nomIngredient": "lait", "quantite": 20, "mesure": "cl" }], "preparation": ["\u00e9mou\u00eblanger la farine et l\u00e9g\u00e9rer le lait", "Ajouter progressivement le lait"], "note": { "moyenne": 3.5, "nbNotes": 2, "usersNotes": [{ "idUser": 24, "userNote": 4 }, { "idUser": 10, "userNote": 3 }] }, "date": "1478889683809", "photo": "582610d396aa583c1fb845b7" }</pre>

Figure 4: Formats de document utilisés dans la base Mongo

Pour les ingrédients, deux formats de document existent dans la collection ingrédients. Le premier concerne les fruits et légumes (cf. figure 4, exemple du document de l'ingrédient "fraise") pour lesquels il n'existe pas de code barre. Les appels vers l'API s'effectuent avec le code barre, les prix des fruits et légumes ont donc été stocké statiquement dans la base de données. Le second format concerne tous les autres produits et contient un champ "ean" correspondant au

code barre à la place du prix pour le premier format. Ce code barre permettra d'obtenir le prix du produit dynamiquement à travers des appels à l'API. Il faut savoir que tous les codes barres sont identiques pour un même produit en France. Dans les deux formats de document pour un ingrédient, la quantité correspond à la quantité du produit associé au code barre. Pour les fruits et légumes la quantité est exprimée en unité (dans l'exemple, 1 fraise vaut environ 0.13 euros). Les ingrédients stockés dans cette collection ingrédients sont les ingrédients que l'on peut sélectionner et mettre dans une recette. Tout autre ingrédient ne pourra pas y être inséré.

En ce qui concerne les recettes on y retrouve le titre, l'auteur (id, nom d'utilisateur), les ingrédients sous forme de liste (nom, quantité, mesures utilisée pour exprimer la quantité), les étapes de préparation sous forme de liste de chaîne de caractères, les informations relatives à la note de la recette (moyenne, nombre de notes données, les utilisateurs ayant noté la recette), la date (Long), la photo (dont la valeur est l'id de la recette). La flexibilité de MongoDB nous permet, ici, d'ajouter facilement des valeurs aux listes contenues dans le document.

3 Architectures et Technologies

3.1 Architecture web

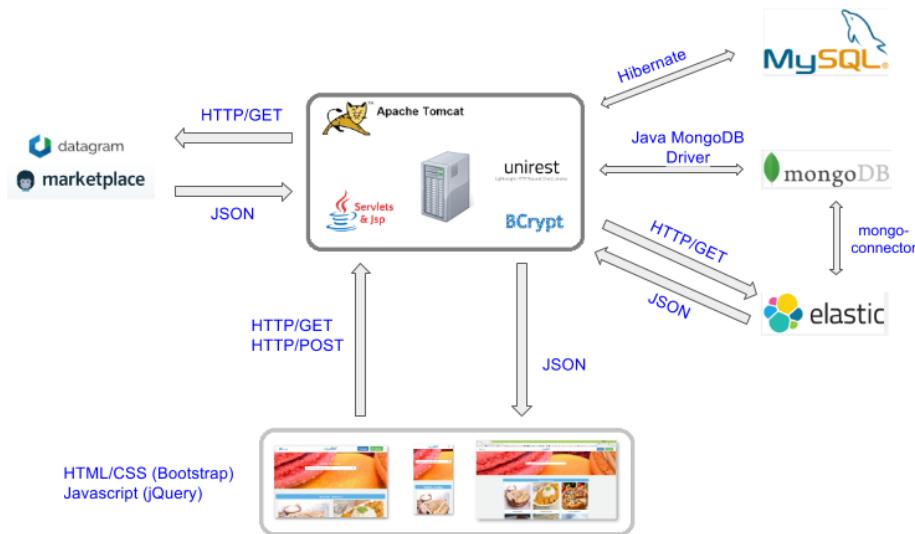


Figure 5: Schéma de l'application

La figure ci-dessus schématise l'architecture de notre application. Le client fait appel à un service du serveur à travers notre API Rest. Il communique avec le serveur au moyen de requêtes HTTP/GET et HTTP/POST. En réponse à

une requête HTTP, la servlet appelée effectue le traitement correspondant et répond au client avec du contenu JSON. Notre serveur accède à notre base de donnée MySQL avec Hibernate, et il accède à notre base MongoDB à travers le driver JAVA officiel. Concernant les appels à l'API Datagram pour récupérer les prix des ingrédients, nous faisons de simples requêtes HTTP et le résultat renvoyé est un document JSON. Enfin nous utilisons ElasticSearch pour notre moteur de recherche, on y accède également avec de simples requêtes HTTP et le résultat renvoyé est un aussi document JSON. Nous avons synchronisé MongoDb et ElasticSearch avec Mongo-connector, qui permet de reporter chaque modification dans la base MongoDB vers les documents indexés de ElasticSearch.

3.2 Technologies utilisées

3.2.1 Bibliothèque Unirest

Nous utilisons Unirest pour effectuer nos appels vers l'API Datagram. C'est une bibliothèque qui permet d'effectuer des requêtes HTTP dans différents langages. Nous avons fait le choix de l'utiliser car elle permet de faire des requêtes HTTP très facilement. Voici un exemple de requête :

```
Unirest.post("http://httpbin.org/post")
    .queryString("name", "Mark")
    .field("last", "Polo")
    .asJson()
```

Figure 6: Exemple de requête HTTP avec Unirest

3.2.2 ElasticSearch

ElasticSearch est un moteur de recherche open-source (Apache 2). Il utilise la bibliothèque Apache Lucene et indexe les données sous forme de documents. Il possède des avantages indéniables dont :

- la recherche en quasi temps réel
- la synchronisation automatique avec notre base MongoDB
- l'indexation automatique des documents de MongoDB
- API REST

ElasticSearch offre une API REST permettant d'effectuer tout type d'opération.
Il supporte les méthodes HTTP (GET, PUT, POST, DELETE)

```
curl -XGET 'http://localhost:9200/[index]/[type]/[id]/[action]'
```

Figure 7: Exemple de requête HTTP/GET avec curl

- index : nom de la database MongoDB
- type : nom d'une collection MongoDB
- id : id du document
- action : action à effectuer

Dans notre application, la seule action effectuée est celle de la recherche. Par exemple lorsqu'un utilisateur souhaite taper le mot "chocolat" dans la barre de recherche, nous interrogeons ElasticSearch sur l'adresse suivante :
http://localhost:9200/cheapmeal/Recettes/_search?q=chocolat~

```
Settings settings = Settings.builder()
    .put("client.transport.sniff", true).build();
TransportClient client = new PreBuiltTransportClient(settings);
client.addTransportAddress(new InetSocketTransportAddress(
    InetAddress.getByName("localhost"), 9300));

SearchResponse res = client.prepareSearch()
    .setIndices("cheapmeal")
    .setTypes("Recettes")
    .setQuery(QueryBuilders.queryStringQuery("chocolat" + "~"))
    .execute()
    .actionGet();
```

Figure 8: Exemple de d'utilisation de l'API Java d'ElasticSearch

3.3 Hibernate et JPA

Hibernate est l'un des ORM les plus utilisés dans le domaine du web. Nous avons décidé d'utiliser ce cadriel (framework) car, bien que difficile à configurer, il était intéressant de nous y familiariser.

3.4 Bootstrap et JQuery

Finalement, coté client, nous avons décidé d'utiliser Bootstrap qui permet un développement rapide et responsive de nos pages web et facilite la maintenance.

De même JQuery a été utilisé afin de simplifier le développement JavaScript notamment grâce à la facilité d'usage d'Ajax dans celui-ci et des différents plugins pouvant être utilisés (autocomplete et validator).

3.5 Hébergement

L'hébergement a été fait sur Tomcat 8, un conteneur web libre de servlets JAVA. Il est installé sur un serveur virtuel privé OVH (merci à Issa Mahamat qui nous a partagé l'accès à son serveur). Cela nous a permis de choisir et de configurer nous même les différentes technologies utilisées : MongoDB et ElasticSearch notamment.

3.6 Outils

Parmis les outils utilisés nous trouvons : Eclipse JEE, GitHub pour le travail parallèle et collaboratif, Maven pour la gestion des dépendances, Discord pour la communication entre les différents membres, et pour finir les outils de développement du navigateur Google Chrome.

4 Analyse de la sécurité

Nous mettons un point d'honneur à protéger les informations confidentielles de nos clients. C'est pourquoi nous avons décidé d'utiliser BCrypt, une bibliothèque qui implémente un algorithme de chiffrement Blowfish, et qui nous permet de protéger les mots de passe de nos utilisateurs dans nos bases de données. Le cryptage se fait du côté serveur et non du côté client. Si le cryptage avait été fait côté client, il aurait été facile de savoir quel type de cryptage a été utilisé et en interceptant les requêtes transportant des mots de passes, il aurait été facile d'obtenir des mots de passe utilisateur. Pour éviter ce problème, il aurait fallu utiliser le protocole HTTPS pour les requêtes vers le serveur. Malheureusement, ce protocole n'est pas utilisable gratuitement, c'est pourquoi nous n'avons pas pu nous en servir.

Nous avons aussi prévenus notre site de failles couramment utilisées dans le domaine du web. La première est l'injection SQL, pour cela nous avons utilisé l'API de Hibernate qui permet de les éviter en nommant nos paramètres lors des requêtes vers la base de données. La deuxième est la gestion du cross site scripting sur notre site en utilisant des méthodes fournies par Apache Commons, qui permettent d'échapper les chaînes de caractères HTML.

5 Rentabilité

5.1 Coût du projet

Nous avons principalement utilisé des outils et technologies libres et open-sources pour réaliser notre application Web. Concernant les supports infor-

matiques, nous avons utilisé nos machines personnelles ainsi que les ressources matérielles disponibles à l'université.

Les seules dépenses concernent la location du serveur qui nous aurait coûté 3,59 euros/mois minimum, ainsi que l'utilisation de la version payante à 99 euros/mois de l'API Datagram qui permettrait de localiser un produit à proximité en temps réel.

5.2 Étude du marché et monétisation

Cependant, il serait préférable de récupérer automatiquement le contenu des sites de courses en ligne des groupes de la grande distribution et de remplir notre propre base de données contenant le prix de plusieurs produits alimentaires. Il faudrait en revanche se renseigner pour savoir si cette pratique est légale. Cela nous permettrait aussi de monétiser notre application de deux manières :

- En affichant des publicités vers les différents sites de courses en ligne.
- En plaçant un lien dans chaque recette permettant de pré-remplir un panier d'achat sur le site <http://courses.carrefour.fr/drive/accueil> contenant les ingrédients de la recette par exemple et de recevoir une commission à chaque commande validée.
- Permettre à des entreprises de proposer, moyennant finance, leurs contenus à des utilisateurs sous forme de produits sponsorisés.

Enfin de plus en plus de personnes ne font plus confiance aux industries agro-alimentaires et préfèrent donc préparer elles-mêmes leurs plats et leur desserts. Notre application s'adresse essentiellement à ces personnes. De plus il y a un côté économique, car un produit en magasin coûte plus cher qu'un produit fait-maison et notre application permet d'estimer en un clin d'œil le prix d'une recette.

6 Améliorations

De nombreuses améliorations sont envisageables. La liste ci-dessous présente une liste non-exhaustive des extensions que nous envisageons.

1. Ajout de commentaires dans les recettes permettant de donner des avis supplémentaires ou de demander des conseils par rapport à la préparation.
2. Catégoriser les recettes afin de permettre une recherche plus facile. Par exemple : dessert, entrée, plat ou bien même spécialités Portugaises.
3. Algorithme de proposition de contenu qui étudierait les habitudes d'un utilisateur et lui proposerait des recettes qu'il pourrait apprécier.
4. Permettre à un utilisateur d'ajouter une recette en favoris et d'afficher les favoris d'un utilisateur sur son profil.

5. Soumission d'ingrédients supplémentaires sur le site. Ainsi un utilisateur pourrait ajouter à sa guise un ingrédient qui n'est pas référencé dans notre base de données. Cela implique aussi un système de validation des ingrédients "disponibles" par des administrateurs.
6. Affichage de la valeur énergétique d'un plat, en fonction de ses ingrédients.

7 Point forts

7.1 Bande Passante

Pour augmenter la réactivité lors de la navigation dans notre application web, nous avons décidé de minimiser le code Javascript et CSS ainsi que de compresser les différentes images et photos affichées. L'impact qu'a cette modification est mesurée et comparée (grâce à un outil que propose le navigateur Chrome) sur la figure 9 où en haut se trouve le chargement des ressources non-minimisées de la page de connexion et en bas avec les ressources minimisées. En gris est représenté le temps passé avant l'envoie de la requête pour le fichier, en orange le temps passé à établir la connexion, en vert le temps passé à attendre la réponse de la requête et en bleu le temps passé à télécharger le fichier. Nous remarquons que le temps de téléchargement des fichiers diminue légèrement mais c'est au niveau du temps d'attente avant l'envoi de la requête que le temps d'attente est significativement diminué. Après minimisation, la navigation est généralement plus fluide sur CheapMeal.

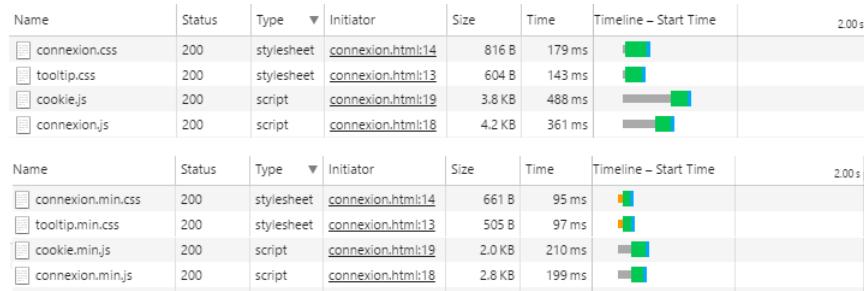


Figure 9: Comparaison du temps de chargement des ressources pour la page de connexion

7.2 API Rest CheapMeal

Au fil du développement de notre application, nous avons enrichi notre API qui consiste en plusieurs "endpoint", un pour chaque service. Des développeurs peuvent s'appuyer sur cette API Rest pour développer facilement un client différent de celui que nous proposons dans le cadre du projet. Elle offre aussi la possibilité de développer une version mobile de l'application CheapMeal. Nous avons utilisé MaShape, une plateforme open-source qui offre des services cloud facilitant la mise en place d'une API. Les services y sont documentés. Le lien d'accès à l'API CheapMeal est le suivant :

<https://market.mashape.com/patla12/cheapmeal>.

8 Points faibles

8.1 API externe

Nous avons rencontré quelques problèmes lors de l'utilisation de l'API Datagram. En effet, nous avons souscrit à leur offre "Freemium" qui limite les données reçues. Il n'est pas rare lors de la recherche du prix d'un produit, pourtant courant, de ne voir aucun magasin le proposant à Paris. Le problème qui en découle est qu'il est relativement difficile d'évaluer le prix d'une recette si l'API ne nous fournit pas des résultats fiables. Pour palier à ce problème nous avons du établir une liste d'ingrédients "autorisés" qui nous permettent de récupérer un prix à coup sûr. De plus il s'avère que les requêtes HTTP échouent sur une base constante, comme en témoigne le graphe de la figure 10 qui montre qu'à certains moments, nous avons plus de 75% des requêtes qui échouent.



Figure 10: Taux d'erreur des requêtes vers Datagram

9 Bilan

Ce projet nous a beaucoup appris sur le monde du développement web notamment sur certains outils (Hibernate, ElasticSearch, Bootstrap) très utilisés en entreprise. Il nous a permis de nous rendre compte des soucis de sécurité et de performance qui peuvent survenir au cours du développement et qui ne doivent pas être négligés pour que l'expérience utilisateur soit optimale. Au niveau de la gestion de projet, nous avons dû nous séparer les tâches par rapport aux fonctionnalités pour que chacun ait au moins une expérience côté serveur, client et base de données. L'utilisation de Github pour les sources et le travail collaboratif ainsi que de moyens de communication comme Discord nous ont permis de nous synchroniser, d'optimiser la répartition des tâches et de faciliter les prises de décisions en équipe.