

Candidat N° : 122599

Travail Pratique Individuel 2017

Amazon Reader



Nom du candidat : Onu Oktay Dogan

Classe du candidat : 304541

13 juin 2017

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 2 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

Table des matières

1	Introduction	5
2	Gestion du projet	5
3	Phase d'initialisation	6
3.1	Analyse de l'état actuel	6
3.1.1	Analyse de l'état désiré.....	6
3.2	Exigences de l'application.....	6
3.2.1	Source de données.....	6
3.2.2	Sauvegarde des articles dans une base de données.....	7
3.2.3	Serveur Web.....	7
3.2.4	La sécurité	7
3.2.5	Client Web	8
3.3	Architecture de l'application.....	8
3.4	Choix des technologies	9
3.4.1	Front-end	9
3.4.2	Back-end.....	9
3.4.3	Base de données	9
3.5	Maquettes de l'interface graphique	10
3.6	User stories.....	12
3.7	APIs du serveur web.....	14
3.8	Diagrammes	15
3.8.1	Login	15
3.8.2	Logout	15
3.8.3	Récupération de la liste des utilisateurs	16
3.8.4	Création d'un utilisateur	16
3.8.5	Suppression d'un utilisateur	17
3.8.6	Modification d'un utilisateur.....	17
3.8.7	Recherche d'articles	18
3.8.8	Import d'articles	18

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 3 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

3.8.9	Modification d'un article	19
3.8.10	Suppression d'un article	19
4	Phase d'exécution	20
4.1	Préparation de l'environnement.....	20
4.1.1	Angular-CLI.....	20
4.2	Import des données.....	21
4.2.1	Analyse du contenu du fichier json.....	21
4.2.2	Correction des erreurs.....	22
4.2.3	Mapping.....	24
4.2.4	Optimisations.....	25
4.2.5	Persistance.....	25
4.3	Utilitaire d'importation en ligne de commande	26
4.4	Serveur RESTful (API).....	27
4.4.1	Serveur https	27
4.4.2	Routes.....	27
4.4.3	Réponses	28
4.4.4	Middleware	28
4.5	Recherche des articles dans la base de données	29
4.5.1	Indexation des données	29
4.5.2	Ajout des articles	29
4.5.3	Filtres.....	29
4.5.4	Recherche des articles.....	30
4.5.5	Limite	31
4.6	Client Web	32
4.6.1	Architecture	32
4.6.2	Les composants.....	32
4.6.3	Les services.....	33
4.6.4	Affichage de messages.....	33
4.6.5	Stratégie d'édition et de suppression	34
4.6.6	Importation des données.....	34

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 4 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.7 Sécurité.....	35
4.7.1 Authentification.....	35
4.7.2 Autorisations.....	36
4.7.3 Protection des mots de passe.....	36
4.7.4 Protection de l'accès à la base de données	36
4.7.5 Attaques par injection SQL	37
4.7.6 Minification.....	37
4.8 Revue de code.....	38
4.9 Test.....	38
4.9.1 Tests manuels.....	38
4.9.2 Tests unitaires.....	38
4.9.3 Tests de bout-en-bout.....	40
4.9.4 Tests de montée en charge.....	40
5 Documentation.....	41
5.1 Documentation des librairies	41
6 Améliorations	42
7 Conclusion.....	42
8 Annexes	43
9 Bibliographie.....	43
10 Remerciements	43

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 5 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

1 Introduction

Lors de ce travail pratique individuel (TPI) final, j'aurai à cœur de montrer mes connaissances en développement de logiciels acquises lors de ces 4 années d'apprentissage. Je vais donc m'appliquer à utiliser les meilleures pratiques.

2 Gestion du projet

J'ai choisi de gérer ce projet en utilisant une variante simplifiée de la méthode *Agile Scrum* puisque je suis seul dans ce projet. Ma démarche sera la suivante :

- Phase d'initialisation
 - Analyse du cahier des charges
 - Choix technologiques
 - Architecture
 - Création des user stories pour l'établissement du *product backlog*
- Phase de planning
 - Préparation du planning du projet
 - Maquettes de l'application client
 - Diagrammes de séquences
 - Diagramme de cas d'utilisation
- Phase d'exécution
 - Préparation de 5 sprints pour la livraison du projet (le dernier sprint concernera la finition de la documentation et la programmation des tests end-to-end)
 - Import des données
 - API du serveur *RESTFul*
 - Client Web
 - Sécurité
 - Documentation et test end-to-end
 - Préparation du sprint *backlog* de chaque *sprint*
 - Utilisation du Kanban pour visualiser l'avancement de mon projet

Vu l'objectif et la durée du projet, je n'aurai pas de phase de "*monitoring*" ni de phase de "*closing*" durant ce projet.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 6 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

3 Phase d'initialisation

3.1 Analyse de l'état actuel

Il s'agit d'un nouveau développement. Il n'y a pas d'état actuel.

3.1.1 Analyse de l'état désiré

L'objectif de ce projet est de réaliser une application web permettant la recherche et la visualisation d'articles en se basant sur une source de données contenant plusieurs millions d'articles.

3.2 Exigences de l'application

3.2.1 Source de données

Pour réaliser ce projet j'ai besoin d'importer un fichier json (de 10 GB) qui contient quelques millions d'articles Amazon avec leur titre, leur prix, un lien vers une image, une liste de catégories.

L'obtention des données est conditionnelle à la citation des documents SIGIR et KDD ci-dessous.

Citation

Please cite one or both of the following if you use the data in any way:

Image-based recommendations on styles and substitutes

J. McAuley, C. Targett, J. Shi, A. van den Hengel
SIGIR, 2015

[pdf](#)

Inferring networks of substitutable and complementary products

J. McAuley, R. Pandey, J. Leskovec
Knowledge Discovery and Data Mining, 2015

[pdf](#)

L'importation des données devra se faire de 2 manières différents :

- Directement depuis le client web (package GO)
- Depuis une application desktop (en ligne de commande, en mentionnant le chemin du fichier à importer).

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 7 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

3.2.2 Sauvegarde des articles dans une base de données

Les données doivent être sauvegardée dans une base de données offrant :

- De très bonnes performances en écriture.
- D'excellentes performances en lecture.
- Le support de recherches complexes.

3.2.3 Serveur Web

- Le serveur web doit offrir des APIs qui permettront au client de récupérer les données.
- Le trafic de données qui transitent entre le client et le serveur doit être crypté.

3.2.4 La sécurité

- Les utilisateurs doivent s'authentifier
- L'application se compose de 3 niveaux d'autorisations prédéfinis (rôles).
 - Les utilisateurs peuvent effectuer des recherches et afficher les articles
 - Les rédacteurs disposent des droits des utilisateurs et peuvent gérer les articles (modification et suppression).
 - Les administrateurs disposent des droits des rédacteurs et peuvent gérer les comptes des utilisateurs (création, modification et suppression).
- Les mots de passe doivent être cryptés et salés.
- Le serveur génère un jeton jwt ne contenant que l'identifiant de l'utilisateur. Le jeton circule avec les requêtes, encapsulé dans un cookie qui ne devra pas pouvoir être visible par le client.
- Le cookie doit avoir une durée de vie limitée à :
 - 1 jour pour les administrateurs
 - 1 semaine pour les rédacteurs
 - 1 mois pour les utilisateurs
- Il faudra générer un certificat (auto-signé).
- La connexion entre le serveur et la base de données doit être également protégée.
- Une protection contre les attaques de type injection sql doit être implémentée.
- La minification du code javascript

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 8 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

3.2.5 Client Web

Une librairie permettant d'assurer une qualité à l'interface utilisateur.

- Si l'utilisateur n'a pas de compte, il peut en créer un lui-même. Par défaut, les nouveaux comptes ont un niveau d'autorisation « utilisateur ».
- Une fois authentifié, l'utilisateur peut effectuer des recherches pour afficher les articles :
 - Par mots-clés contenus dans le titre
 - Par catégorie
 - Par prix
 - Combinaison de plusieurs de ces choix
- S'il dispose du privilège « rédacteur » il peut également modifier ou supprimer des articles.
- L'administrateur doit pouvoir modifier les privilèges et supprimer des comptes utilisateurs.

3.3 Architecture de l'application

L'application se compose des 3 éléments principaux suivants :

- Une interface utilisateur qui s'exécutera dans le navigateur de l'utilisateur et qui récupérera les données sur le serveur
- Un serveur *RESTful* qui fournira les API permettant de transférer les données qu'il récupère dans la base de données. Il mettra également une adresse à disposition afin de charger l'application web dans le navigateur. Le serveur est également le garant de la sécurité de l'application.
- Une base de données qui traite les requêtes du serveur et lui fournit les données.



 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 9 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

3.4 Choix des technologies

3.4.1 Front-end

J'ai passé une année à utiliser *Angular.js* et 6 mois à utiliser *Angular 2*. Ce sont des frameworks que je maîtrise et que j'apprécie beaucoup. Le choix s'impose donc à moi naturellement.

La version actuelle de *Angular* est cependant la 4. Par chance, elle est identique à la version 2, je n'aurai donc pas à réapprendre de nouveaux concepts.

En ce qui concerne les éléments graphiques, je vais m'appuyer sur la bibliothèque *Angular Material 2* que j'ai déjà eu l'occasion d'utiliser. Je suis le seul chez Xpert à la connaître car elle est encore en version bêta mais je pense qu'elle est suffisamment stable pour ne pas rencontrer de problèmes insurmontables.

3.4.2 Back-end

Mon choix s'est tout de suite porté sur Go car c'est le langage que je maîtrise le mieux pour le back-end et également parce qu'il est très performant. Il possède d'autres qualités qui en font un choix idéal (voir annexe 1 → Pourquoi Xpert Technologies utilise Go).

De plus, je vais pouvoir très rapidement développer mon serveur *RESTful* sans devoir installer et configurer un serveur http.

3.4.3 Base de données

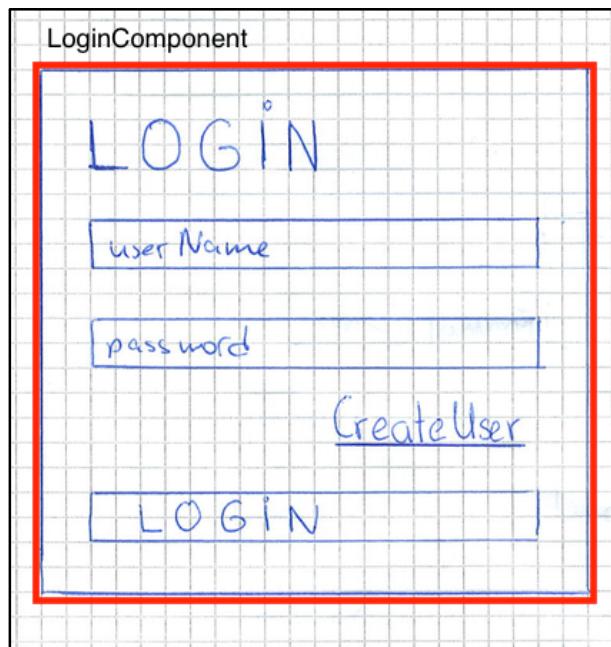
J'avais initialement opté pour *CouchBase* qui est une base de données *NoSQL* orientée document offrant d'excellentes performances en lecture et écriture.

Après discussion avec mon maître de stage, j'ai finalement choisi d'utiliser *ElasticSearch* qui est également une base de données *NoSQL* orientée document mais offre un moteur de recherche *full-text* qui sera, m'a-t-on dit, très utile pour effectuer des recherches performantes dans les millions d'articles disponibles.

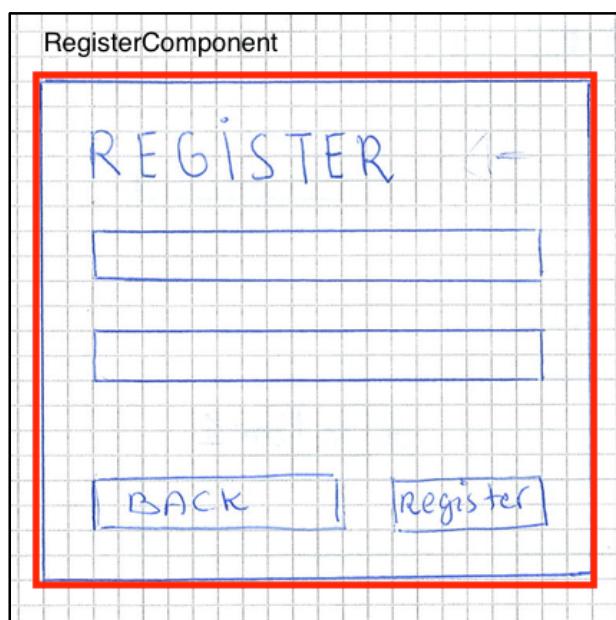
 ICT Berufsbildung <i>Formation professionnelle</i> Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 10 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

3.5 Maquettes de l'interface graphique

Mon application va débuter par l'identification de l'utilisateur. En cas de succès, le serveur retournera le nom de l'utilisateur ainsi que son rôle.



Si un utilisateur ne dispose pas encore de compte, il pourra en créer un lui-même. Initialement, il recevra le rôle d'utilisateur.



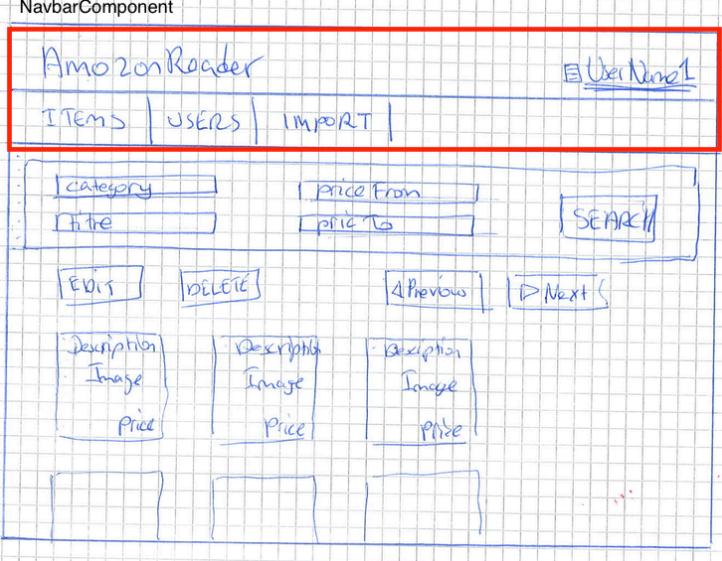
 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 11 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

Lorsque l'utilisateur a été identifié avec succès, il accèdera à l'écran principal de l'application où trois onglets seront disponibles en fonction de son privilège :

- **Articles**

permet de rechercher et d'afficher les articles par page de 15 éléments. En outre, les rédacteurs et les administrateurs peuvent modifier les articles.

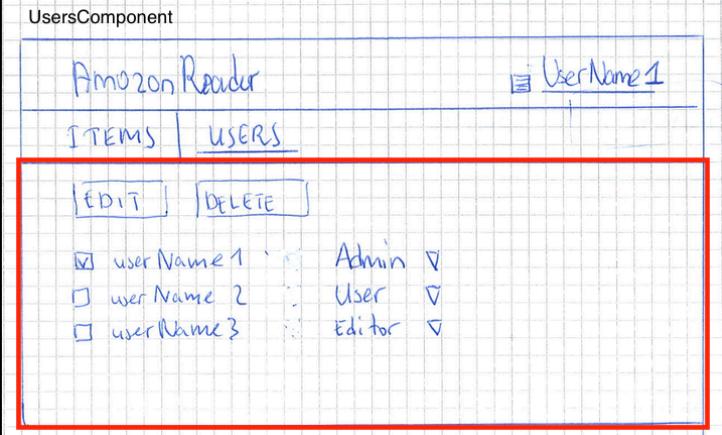
NavbarComponent



- **Utilisateurs** (uniquement pour les administrateurs)

permet d'accéder à la liste des utilisateurs et de modifier leur rôle.

UsersComponent



<input checked="" type="checkbox"/> user Name 1	:	Admin	▼
<input type="checkbox"/> user Name 2	:	User	▼
<input type="checkbox"/> user Name 3	:	Editor	▼

- **Import** (uniquement pour les administrateurs)

permet d'importer un nouveau fichier json pour ajouter ou mettre à jour des articles.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 12 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

3.6 User stories

Après lecture attentive du cahier des charges, j'ai identifié les fonctionnalités principales que j'ai saisie dans des *users stories*.

Les *user stories* sont composée de 2 parties :

- La description d'une fonctionnalité sous la forme "en tant que <personne>, je peux <quoi> afin de <pourquoi>", où le pourquoi est optionnel.
- La liste des critères d'acceptation qui, une fois validée durant la phase de test, me permettront de considérer la tâche comme effectuée.

Titre : Création d'un compte utilisateur	N° : 1
Description : En tant qu' utilisateur de l'application, je peux créer mon propre compte.	
Critères d'acceptation : <ul style="list-style-type: none"> • Le compte créé reçoit un rôle minimal, c'est-à-dire celui d'utilisateur. 	

Titre : Accès à l'application	N° : 2
Description : En tant qu' utilisateur, rédacteur ou administrateur je dois m'authentifier afin d'accéder à l'application.	
Critères d'acceptation : <ul style="list-style-type: none"> • L'application n'est disponible que pour les personnes authentifiées, sinon l'application redirige l'utilisateur vers la fenêtre du login. • Si l'authentification est incorrecte, l'application affiche un message d'erreur. • Une fois l'authentification effectuée avec succès, le nom de l'utilisateur est affiché de manière visible dans l'application. 	

Titre : Gestion des utilisateurs	N° : 3
Description : En tant qu' administrateur je peux accéder à la gestion des utilisateurs.	
Critères d'acceptation : <ul style="list-style-type: none"> • Le menu de gestion des utilisateurs ne doit être visible que pour les administrateurs. • Les administrateurs, et eux uniquement, peuvent modifier le rôle des utilisateurs. • Les administrateurs, et eux uniquement, peuvent supprimer un utilisateur. • La modification ou la suppression n'est possible que si au minimum un administrateur reste actif. 	

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 13 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

Titre : Import des articles

N° : 4

Description :

En tant qu'administrateur, je peux importer les articles d'un fichier json.

Critères d'acceptation :

- Le menu d'import ne doit être visible que pour les administrateurs.
- Le fichier à importer doit être du type json.

Titre : Recherche des articles

N° : 5

Description :

En tant qu'**utilisateur, rédacteur ou administrateur** de l'application, je peux effectuer des recherches sur les articles.

Critères d'acceptation :

- Par mots-clés sur le titre.
- Par catégorie.
- Par prix (plage de – à).
- Combinaison de plusieurs de ces choix.
- L'affichage contient 15 articles par page.
- Une pagination permet d'afficher les articles suivants ou précédents.

Titre : Gestion des articles

N° : 6

Description :

En tant que **rédacteur ou administrateur** je peux éditer et supprimer un article.

Critères d'acceptation :

- Seuls les rédacteurs et administrateurs ont accès à la gestion des articles.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 14 sur 43	
	AMAZON READER		Version 1 F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

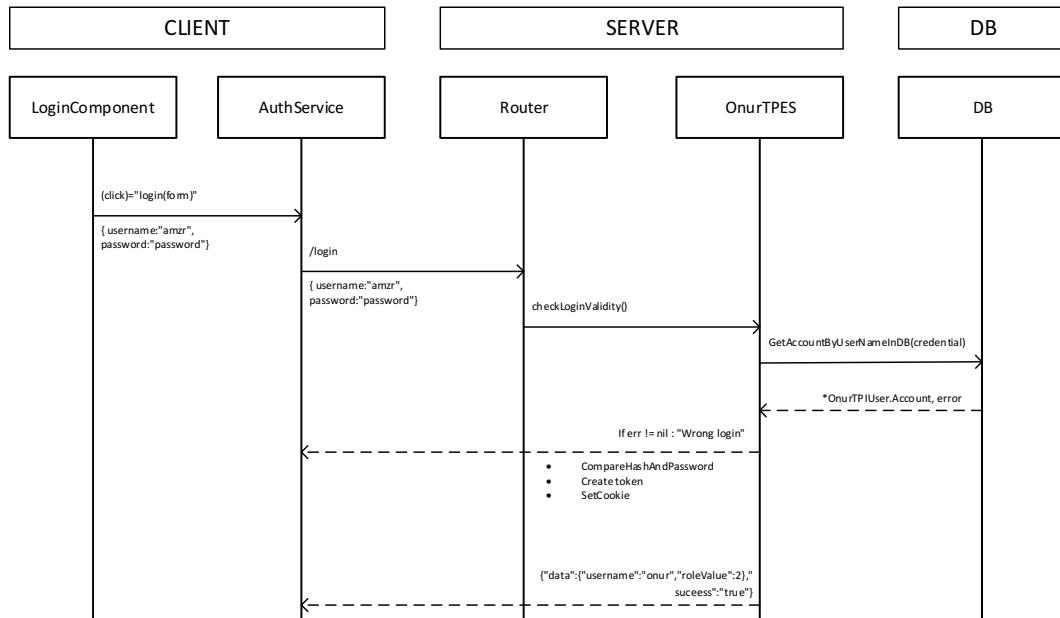
3.7 APIs du serveur web

Méthode	Accès	URI	Action
POST	Utilisateur	/user	Créer un utilisateur
GET	Administrateur	/user/id	Obtenir un utilisateur
PUT	Administrateur	/user/id/role	Mettre à jour un utilisateur
DELETE	Administrateur	/user/id	Supprimer un utilisateur
GET	Administrateur Rédacteur Utilisateur	/users	Récupérer tous les utilisateurs
GET	Administrateur Rédacteur Utilisateur	/item/id	Obtenir un item
PUT	Administrateur Rédacteur	/item/id	Mettre à jour un article
DELETE	Administrateur Rédacteur	/item/id	Supprimer un article
GET	Administrateur Rédacteur Utilisateur	/items	Récupérer les articles en fonction des critères de recherche données dans le body de la requête
GET	Administrateur	/import	Importation des articles
POST	Administrateur Rédacteur Utilisateur	/login	Authentification
GET	Administrateur Rédacteur Utilisateur	/logout	Déconnexion et suppression du cookie

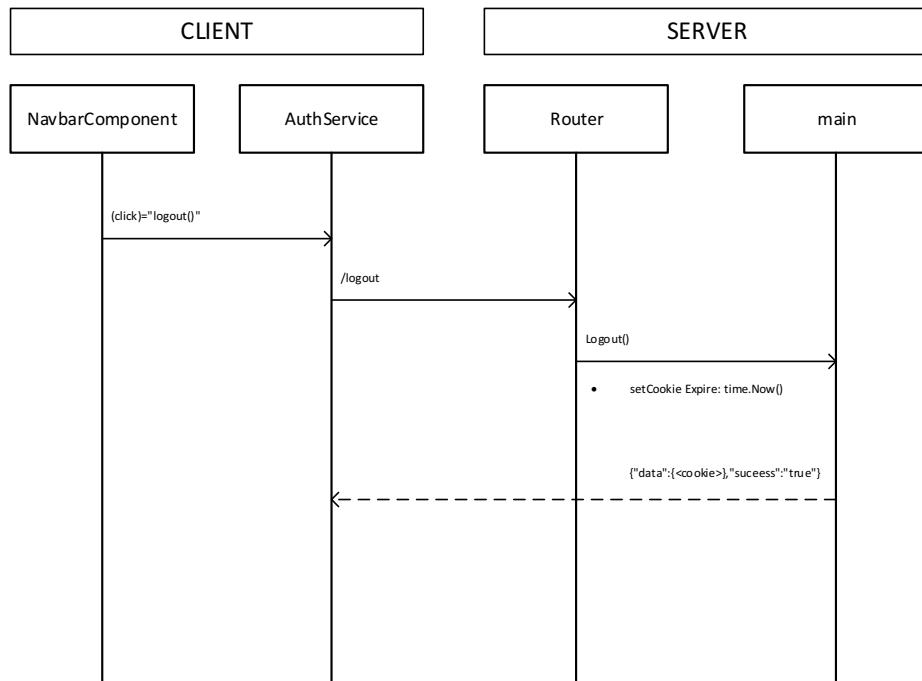
 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 15 sur 43
	AMAZON READER	Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

3.8 Diagrammes

3.8.1 Login

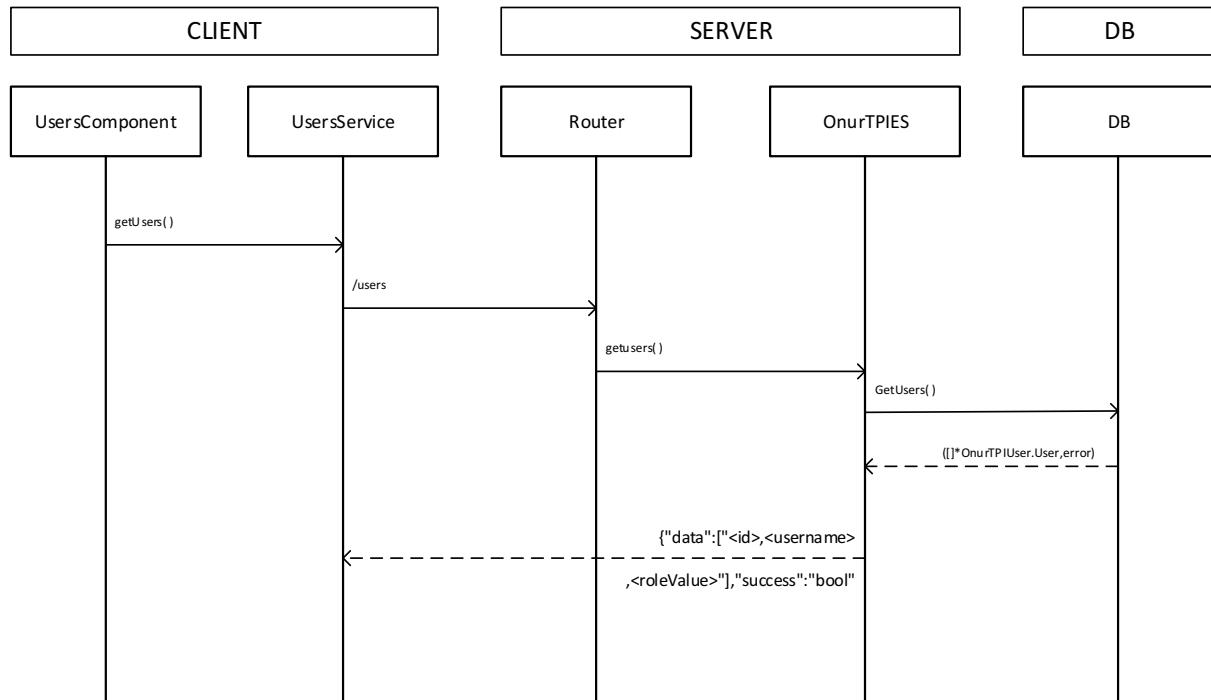


3.8.2 Logout

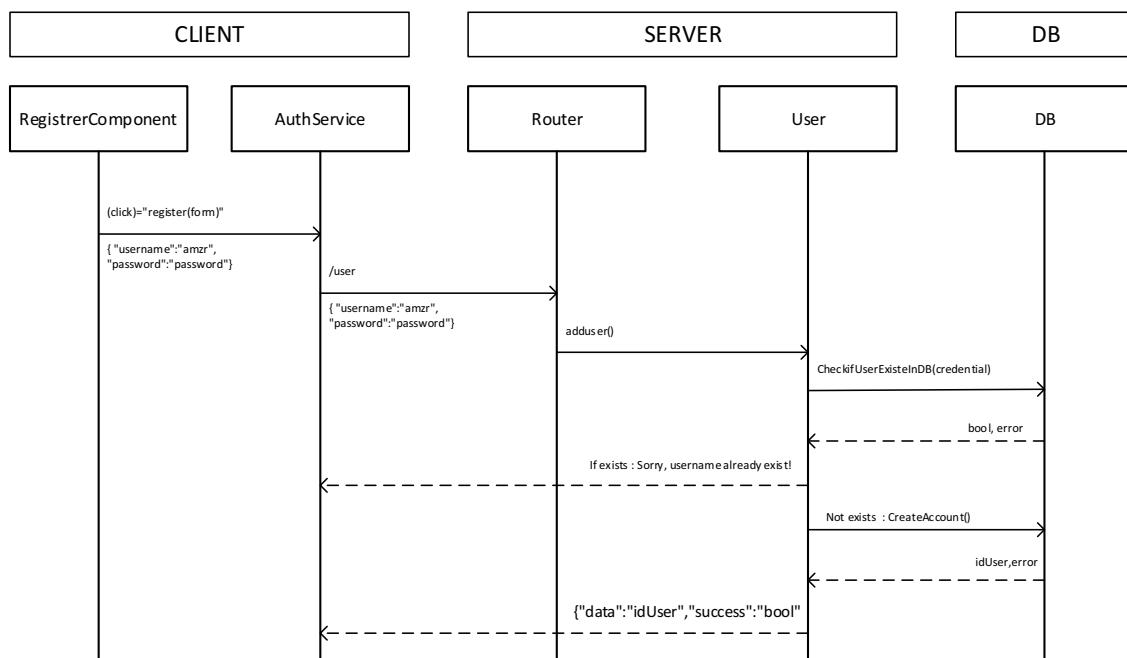


 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 16 sur 43
	AMAZON READER	Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

3.8.3 Récupération de la liste des utilisateurs

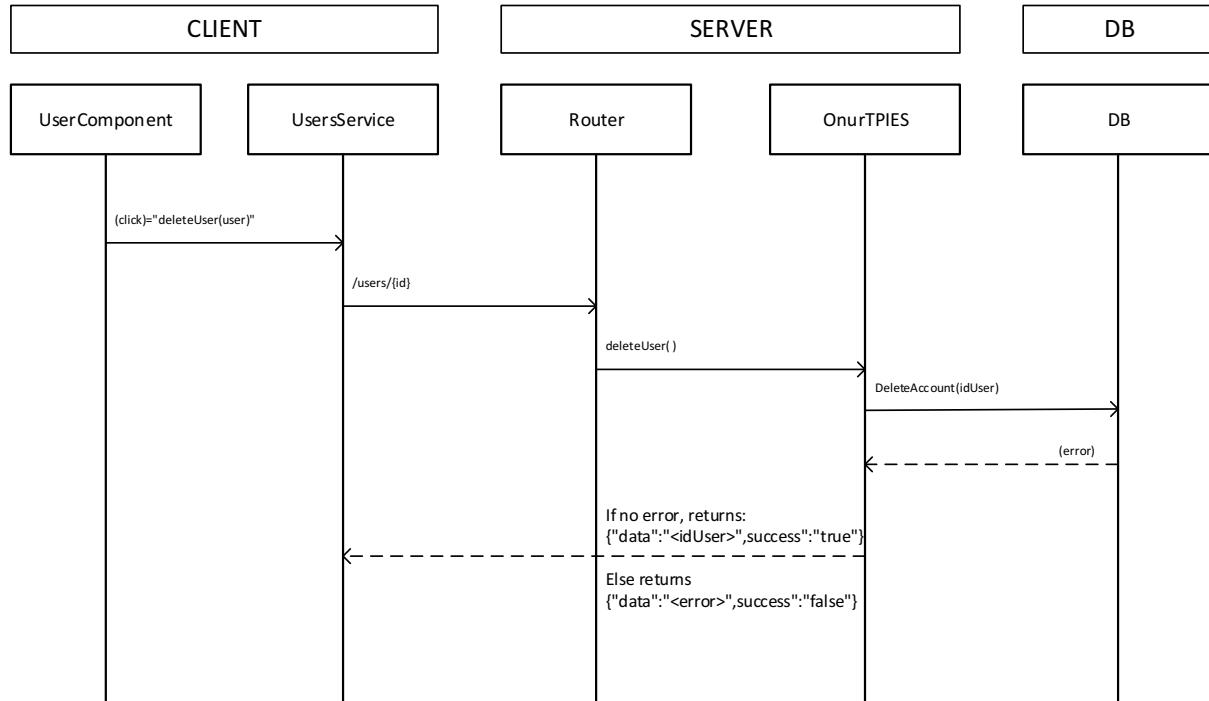


3.8.4 Crédation d'un utilisateur

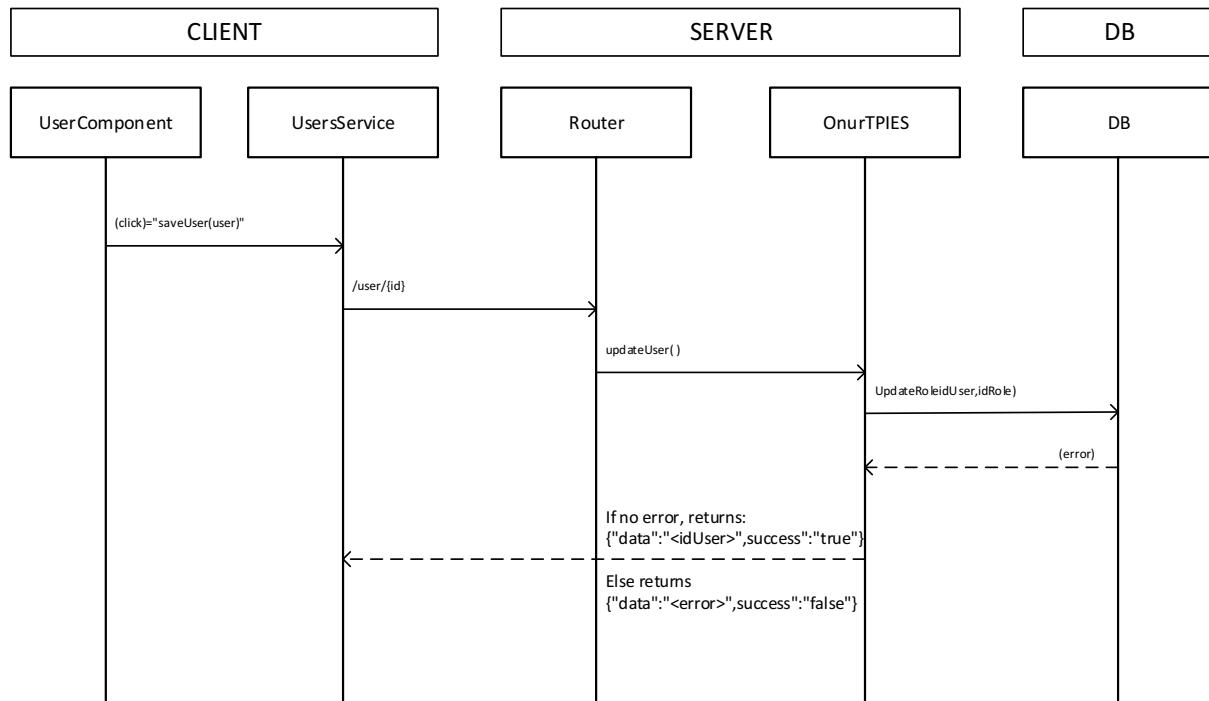


 ICT Berufsbildung <i>Formation professionnelle</i> Freiburg-Fribourg	Travail Pratique Individuel 2017 AMAZON READER	Page 17 sur 43 Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

3.8.5 Suppression d'un utilisateur

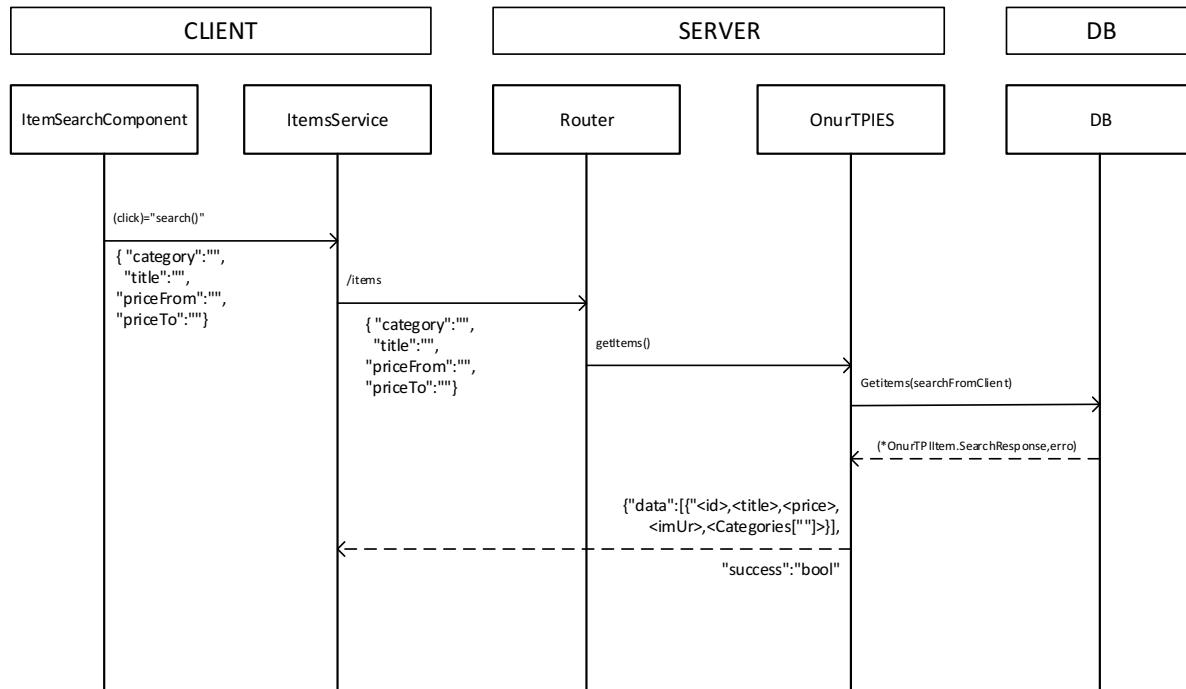


3.8.6 Modification d'un utilisateur

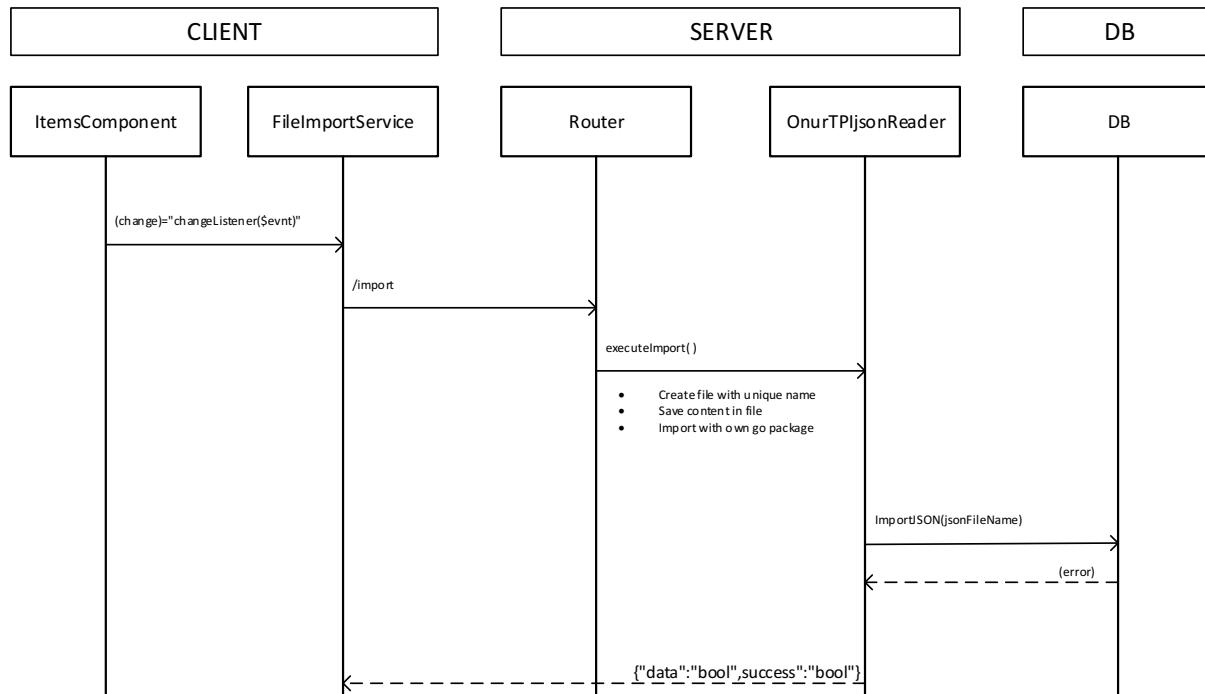


 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 18 sur 43
	AMAZON READER	Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

3.8.7 Recherche d'articles

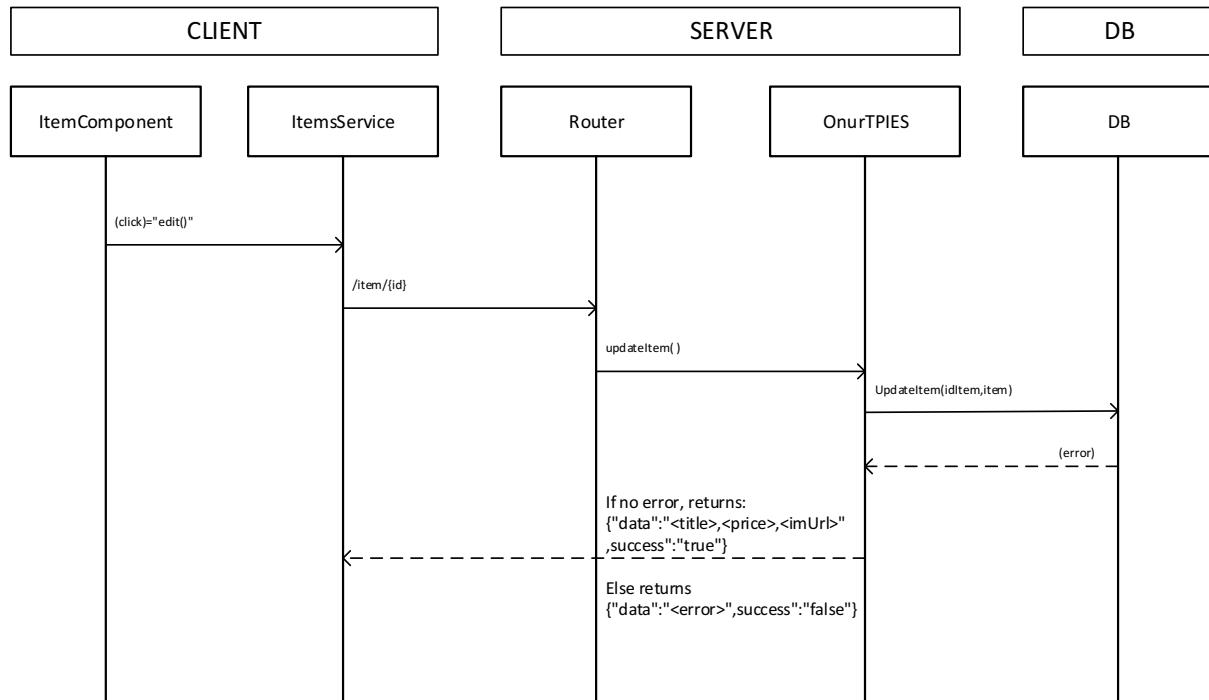


3.8.8 Import d'articles

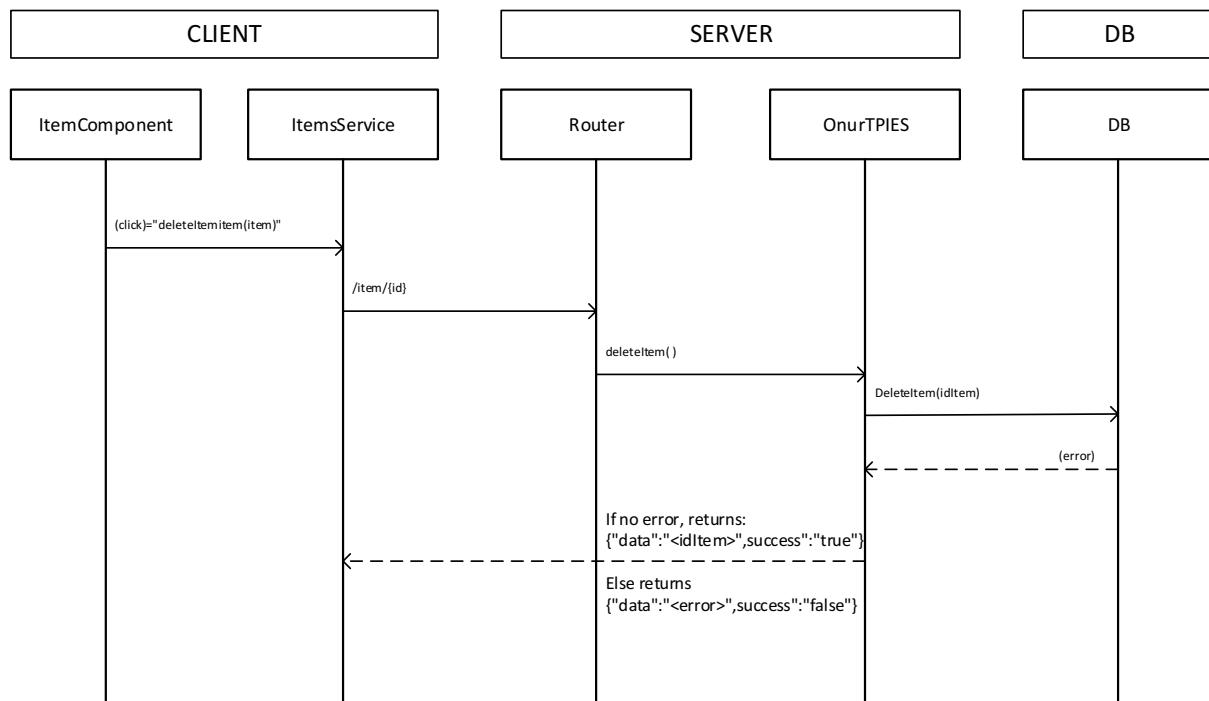


 ICT Berufsbildung <i>Formation professionnelle</i> Freiburg-Fribourg	Travail Pratique Individuel 2017 AMAZON READER	Page 19 sur 43 Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

3.8.9 Modification d'un article



3.8.10 Suppression d'un article



 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 20 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4 Phase d'exécution

4.1 Préparation de l'environnement

Il n'est pas prévu que je déploie l'application sur un vrai serveur. La présentation s'effectuera sur mon ordinateur.

La préparation consiste donc à mettre à jour Angular-CLI (V4.1.3) et Go (V1.8.3) qui sont déjà installés sur ma machine.

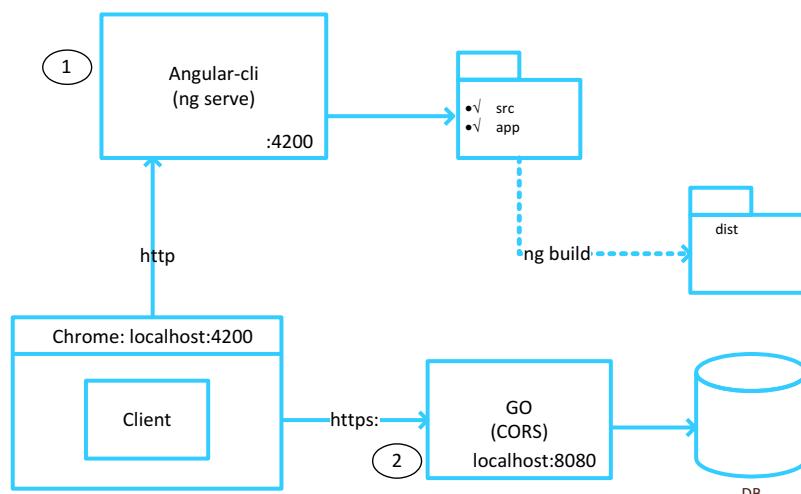
La base de données est le seul élément nouveau de mon environnement. J'ai été impressionné par la facilité d'installation d'*ElasticSearch* puisqu'il m'a fallu 10 minutes, téléchargement compris, pour être en mesure de l'utiliser.

4.1.1 Angular-CLI

Angular-CLI est un outil très pratique qui simplifie le développement de l'application client en transpilant le code TypeScript en javascript, en rechargeant les pages dans le navigateur lorsqu'une modification a été enregistrée, et bien d'autres choses encore. Il me sera également très utile à la fin du projet pour générer le code minifié.

Toutefois, pour pouvoir profiter de ces avantages, je vais travailler avec 2 serveurs durant la phase de développement : l'un pour télécharger l'application web au démarrage de l'application et l'autre pour récupérer les données.

Les navigateurs modernes considèrent ce fonctionnement comme dangereux. Ils y voient une faille potentielle de sécurité connue sous le nom de CORS (Cross-Origin Resource Sharing) et bloquent les requêtes effectuées sur le second serveur. Pour le résoudre, il faut indiquer dans les entêtes des réponses de mon serveur Go que ce cas de figure est permis. **Cette permission devra cependant être retirée dans la version productive.**



 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 21 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.2 Import des données

4.2.1 Analyse du contenu du fichier json

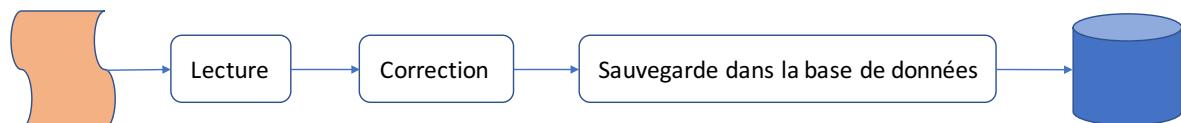
L'importation des données a été un défi coriace.

Afin d'analyser le contenu du fichier de 10,54 GB, j'ai écrit quelques lignes de code en Go permettant de limiter la lecture à quelques lignes.

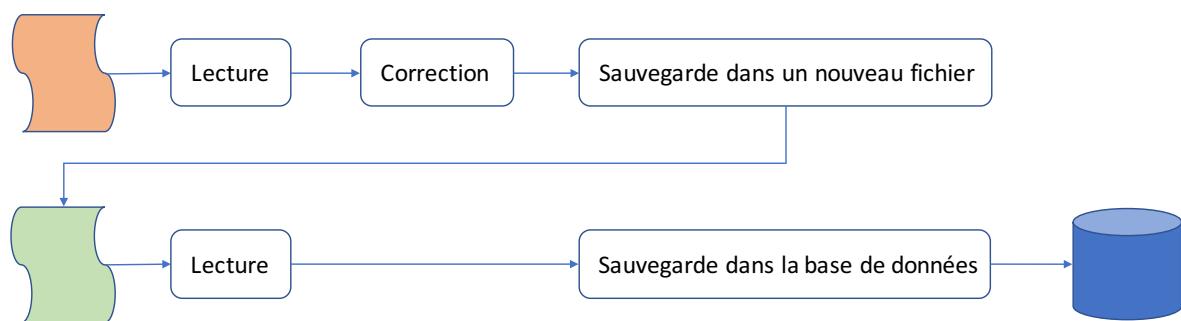
J'ai ensuite pu constater que le fichier était mal formaté. A la place des guillemets, les délimiteurs utilisés étaient des apostrophes ... sauf lorsque l'information contenait des apostrophes.

```
{'asin': '0000401048', 'title': "The rogue of publishers' row;: Confessions of a publisher (A Banner Book)", 'imUrl': 'http://ecx.images-amazon.com/images/I/41bchvIfgaL.jpg', 'related': {'also_viewed': ['068240103X']}, 'salesRank': {'Books': 6448843}, 'categories': [['Books']]}
```

Une correction des données s'est avéré nécessaire avant de pouvoir procéder à l'importation.



Imaginant qu'il faudra plusieurs imports, notamment pour retrouver des données correctes après mes tests, j'ai cependant opté pour une autre approche qui consiste à générer un fichier corrigé qui sera réutilisé pour les imports suivants afin de perdre moins de temps.



Le fichier contient 9'430'088 lignes mais je ne sais pas encore s'il y a des doublons ou non.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 22 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.2.2 Correction des erreurs

Sur la base des constatations faites durant l'analyse, j'ai établi mon concept de correction des données :

- Parcours du fichier, ligne par ligne.
- Parcours de la ligne, caractère par caractère.
- Recherche du délimiteur de départ (soit un guillemet, soit une apostrophe).
- Recherche du délimiteur de fin (le même que celui de départ non précédé d'un back slash).
- Remplacement des guillemets trouvés entre les délimiteurs par des apostrophes.
- Ignore les back slashes (\) trouvés entre les délimiteurs.
- Remplacement des délimiteurs par des guillemets dans la nouvelle chaîne.
- Copie des informations dans la nouvelle chaîne.
- Sauvegarde de la nouvelle chaîne corrigée.

Le code Go correspondant au concept ci-dessus est présenté à la page suivante.

 <p>ICT Berufsbildung Formation professionnelle Freiburg-Fribourg</p>	Travail Pratique Individuel 2017	Page 23 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

```

func CorrectQuotes(s1 string) (s2 string) {
    inside := false
    marker := ""
    c := ""

    for pos, char := range s1 {
        c = string(char)

        // Detect Start Marker
        if !inside && (c == singleQuote || c == doubleQuote) {
            inside = true
            s2 += doubleQuote

            if c == singleQuote {
                marker = singleQuote
            } else {
                marker = doubleQuote
            }

            continue
        }

        // Detect End Marker
        if inside {
            if c == marker && s1[pos-1:pos] != backslash {
                inside = false
                s2 += doubleQuote

                continue
            }
        }

        // Replace " by '
        if inside && (c == doubleQuote) {
            s2 += singleQuote
            continue
        }

        // Ignore \
        if inside && (c == backslash) {
            continue
        }

        // Transfer char in new string
        s2 += c
    }

    return
}

```

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 24 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.2.3 Mapping

Toutes les informations du fichier json ne sont pas intéressantes. De plus, je vais utiliser mes propres noms : *Asin* est spécifique à Amazon, je préfère utiliser *id*.

Les informations *related* sont trop détaillées. J'ai décidé de ne conserver que les liens avec les articles achetés (*also_bought*). Il n'est plus donc plus nécessaire que *related* soit un objet, un tableau de *string* convient tout à fait.

En ce qui concerne les catégories, je ne sais pas à quoi sert un tableau de tableau chez Amazon mais, après discussion avec mon maître d'apprentissage, nous avons convenu de garder uniquement la première chaîne de catégories :

```
[
  [
    [
      Clothing, Shoes & Jewelry Girls
    ],
    [
      Clothing, Shoes & Jewelry Novelty,
      Costumes & More Costumes & Accessories More Accessories Kids & Baby
    ]
]
```

Ainsi le champ *categories*, qui est de type `[][]string` chez Amazon, devient un simple tableau de *string* dans mon application.

Structure Amazon	Structure de mon application
<code>asin string</code>	<code>Id string</code>
<code>title string</code>	<code>Title string</code>
<code>price float64</code>	<code>Price float64</code>
<code>imURL string</code>	<code>ImURL string</code>
<code>related {</code> <code> alsoBought []string</code> <code> alsoViewed []string</code> <code> boughtTogether []string</code> <code>}</code>	<code>Related []string</code>
<code>brand string</code>	<code>Brand string</code>
<code>categories [][]string</code>	<code>Categories []string</code>

Les autres informations sont ignorées.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 25 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.2.4 Optimisations

Il est spécifié dans le cahier des charges que l'optimisation du code est importante. Une fois mon programme d'importation fonctionnel, j'ai donc pris soin de l'améliorer de la façon suivante :

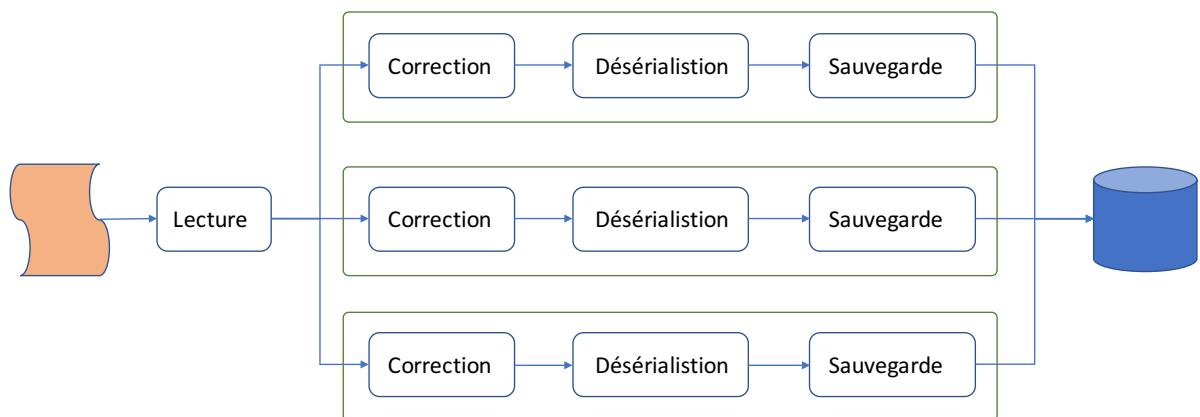
- **Optimisation de la désérialisation**

Utilisation d'une librairie externe plus performante que celle de la librairie standard

- **Optimisation par parallélisme**

La lecture des données dans le fichier json est très rapide et ne peut de toute manière pas être optimisée. Le processus de correction et désérialisation peut, par contre, être accéléré en le traitant en parallèle.

Ces optimisations m'ont permis de réduire le temps de lecture des articles qui était initialement de 128 secondes à 41 secondes, soit près de 70% !



4.2.5 Persistance

La phase finale consiste à sauvegarder les articles dans la base de données. Chaque article est enregistré sous la forme clé-valeur dans *ElasticSearch* :

- La clé est son n° d'article provenant du fichier json.
- La valeur est un document json contenant toutes les données de l'article.

Grâce au document fourni par mon maître de stage, j'ai pu rapidement me connecter à *ElasticSearch* et connaître les requêtes de base. Pour la sauvegarde des articles, j'utilise une librairie qui me simplifie la tâche.

L'import a été effectué avec succès en 2h15'. Au final, 9'430'088 articles ont été ajoutés, ce qui correspond au nombre de lignes du fichier json.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 26 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.3 Utilitaire d'importation en ligne de commande

Mon utilitaire en ligne de commande s'appuie sur le code que j'ai développé pour importer les données et que j'ai transformé en librairie afin qu'il soit réutilisable.

Du coup, il ne faut que quelques lignes pour le rendre opérationnel :

```
package main

import (
    "flag"
    "fmt"
    "github.com/onuroktay/Amazon-Reader/AmzR-pkg-import"
)

func main() {
    // Read filename given by the user
    fileName := flag.String("file", "amazon.json", "name of the json file to read")
    flag.Parse()

    fmt.Println("Import data from ", *fileName)

    // Import data
    err := OnurTPIjsonImporter.ImportJSON(*fileName)
    if err != nil {
        fmt.Println(err.Error())
    }
}
```

Utilisation de l'application en ligne de commande

Afficher l'aide en utilisant les options -h ou --help ou ---h ou --help :

```
AmzR-import-cli -h
```

Résultat :

```
Usage of ./AmzR-import-cli:
  -file string
    name of the json file to read (default "amazon.json")
```

Sans indication de l'option *-file*, le fichier *amazon.json* se trouvant dans le même répertoire que l'application est importé (s'il existe) :

```
./AmzR-import-cli
```

Il est possible d'indiquer le nom du fichier json à importer en utilisant l'option *-file* ou *--file* suivi d'un espace ou de =

```
./AmzR-import-cli -file=../json/amazon.json
```

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 27 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.4 Serveur RESTful (API)

4.4.1 Serveur https

En Go, il n'est pas nécessaire d'installer et de configurer un serveur http, tels que *Apache Tomcat* ou *Microsoft IIS*. Le serveur est intégré dans la librairie *net/http* :

```
err = http.ListenAndServe(":8080", nil)
if err != nil {
    fmt.Println(err.Error())
}
```

Le serveur étant le principal garant de la sécurité, il est nécessaire de crypter les communications entre le client et le serveur. A cette fin, j'ai généré un certificat auto-signé avec *openssl*. Comme l'adresse *ip* de mon ordinateur n'est pas fixe, j'ai choisi de le faire pour *localhost*.

Le démarrage du serveur n'est guère plus compliqué :

```
err = http.ListenAndServeTLS(":8080", certPath + "server.pem", certPath + "server.key", nil)
if err != nil {
    fmt.Println(err.Error())
}
```

Cerise sur le gâteau, mon serveur prend également en charge le protocole HTTP/2 lorsque le navigateur le supporte.

4.4.2 Routes

Le rôle du routeur est, en se basant sur l'URL reçu, de diriger une requête vers une fonction à exécuter afin de retourner la réponse attendue.

La ligne suivante, par exemple, exécute la fonction *startPage* lorsque l'URL racine est reçue, qui retourne la page *index.html* au navigateur :

```
r.HandleFunc("/", startPage).Methods("GET")
```

Dans le cas des API *RESTful*, la réponse contiendra les données désirées au format json. J'ai prévu les API suivantes :

```
// USERS
r.HandleFunc("/users", getUsers).Methods("GET")
r.HandleFunc("/user", addUser).Methods("POST")
r.HandleFunc("/{id}/{role}", updateUser).Methods("PUT")
r.HandleFunc("/{id}", deleteUser).Methods("DELETE")

// ITEMS
r.HandleFunc("/item/{id}", getItem)).Methods("GET")
r.HandleFunc("/items", getItems)).Methods("POST")
r.HandleFunc("/{id}", updateItem)).Methods("PUT")
r.HandleFunc("/{id}", deleteItem)).Methods("DELETE")
```

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 28 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.4.3 Réponses

Si l'URL demandée n'existe pas, la réponse aura un statut **404**, ressource non trouvée.

Je gère également les cas d'erreurs suivants :

- **401**, non autorisée, lorsque l'information d'authentification est manquante, erronée ou échue.
- **406**, pas acceptable, lorsque la taille des informations dans le corps de la requête dépasse la limite acceptée.

Dans les autres cas, mon serveur renvoie une réponse au format json avec un statut **200** contenant *data* et *success* :

- **success** est faux en cas de problème, par exemple si les données reçues dans la requête n'ont pas le format désiré ou que la base de données a retourné un message d'erreur, et vrai si la réponse est utilisable.
- **data** contient le résultat de la requête si *success* est à vrai ou un message d'erreur dans le cas contraire.

4.4.4 Middleware

Le middleware permet d'emballer (wrap) les fonctions afin d'effectuer des actions ou contrôles avant et/ou après leur exécution. J'ai prévu de les utiliser pour :

- **Journaliser des requêtes** : cette tâche n'était pas demandée dans le cahier des charges et n'a pas été faite, faute de temps.
- **Authentifier les utilisateurs** : Vérification de l'existence de l'utilisateur.
- **Contrôler les requêtes demandées** : Autoriser ou bloquer les requêtes en fonction du rôle de l'utilisateur.

Exemple d'ajout de middleware aux routes :

```
// Delete user
r.Handle("/user/{id}", checkAuth(ADMIN,
    http.HandlerFunc(deleteUser))).Methods("DELETE")

// Update item
r.Handle("/item/{id}", checkAuth(EDITOR,
    http.HandlerFunc(updateItem))).Methods("PUT") // update item
```

Dans mon implémentation, je passe au middleware le droit minimum dont il faut disposer pour être autorisé à utiliser l'API.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 29 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.5 Recherche des articles dans la base de données

4.5.1 Indexation des données

ElasticSearch offre un moteur de recherche dit *full-text* qui permet de rechercher un ou une suite de mots. Chaque mot présent dans les enregistrements est indexé de manière à rendre sa recherche très rapide.

Il est cependant important de noter que seuls les mots complets seront trouvés.

4.5.2 Ajout des articles

J'ai utilisé la librairie "github.com/olivere/elastic" pour sauvegarder les données dans la base.

Comme on peut le voir, son utilisation rend l'enregistrement d'un item est très simple :

```
func (es *ELASTICSEARCH) SaveItem(item *ITEM) (err error) {
    _, err = es.client.Index().
        Index(es._indexName).
        Type(es._type).
        Id(item.ID).
        BodyJson(item).
        Refresh("false").
        Do(context.TODO())

    return
}
```

4.5.3 Filtres

Parmi toutes ces fonctions, *ElasticSearch* permet d'effectuer des recherches en spécifiant les critères suivants :

- **match**
permet de rechercher les éléments contenant au moins l'un des mots de la liste (OR)
- **match_phrase**
permet de rechercher les éléments contenant tous les mots de la liste (AND)
- **size**
permet de récupérer *n* éléments
- **from**
permet d'indiquer un décalage à partir duquel on veut récupérer les éléments
- **sort**
permet de trier les éléments

Il est bien sûr possible de combiner ces critères, y compris les "match" et "match_phrase".

J'ai choisi de ne pas utiliser match car je veux que le résultat de la requête ne contienne que des éléments répondant à tous les mots-clés que l'utilisateur a indiqués.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 30 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

En me basant sur ces fonctionnalités, j'ai décidé d'offrir 4 critères de recherche dans mon application web :

- **Titre**
Liste de mots-clés à rechercher dans le titre de l'article
- **Catégorie**
Liste de mots-clés à rechercher dans la catégorie de l'article
- **Prix, à partir de**
Prix minimum de l'article à rechercher
- **Prix, jusqu'à**
Prix maximum de l'article à rechercher

4.5.4 Recherche des articles

Malheureusement, j'ai rencontré quelques soucis lors de l'utilisation de cette librairie pour effectuer des requêtes : aucun résultat n'était retourné dès que la requête devenait plus complexe.

Heureusement, *ElasticSearch* dispose d'une interface RESTFul, ce qui m'a permis d'envoyer les requêtes sous une autre forme et d'obtenir le résultat souhaité. Il a cependant fallu pour cela que je construise les requêtes moi-même, ce que j'ai fait en me référant à la requête suivante :

```
GET /amazonreader/item/_search
{
  "query": {
    "bool": {
      "must": [
        {
          "match_phrase": {
            "title": {
              "query": "lazy dog"
            }
          }
        }
      ],
      "filter": {
        "range": {
          "price": {
            "gte": 10,
            "lte": 20
          }
        }
      }
    },
    "sort": [
      {"price": {"order": "asc"}}
    ],
    "size": 2,
    "from": 10
  }
}
```

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 31 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.5.5 Limite

J'ai constaté durant mes tests que je ne pouvais pas dépasser le 10'000^{ème} enregistrement de ma requête. Si, par exemple, je veux afficher 15 articles par page, je ne pourrai afficher que les 666 premières pages. La page suivante me retournera un résultat vide.

Après recherche sur Internet, j'ai pu constater qu'il s'agit d'une limite imposée par *ElasticSearch*¹.

J'imagine que, pour une application comme la mienne, ça n'est pas très important car il faut avoir de la patience et du temps pour consulter autant de produits en passant en revue des centaines de pages.

J'ai implémenté un contrôle pour vérifier le n° de page dans l'URL : si celle-ci dépasse la limite, je renvoie à la dernière page possible.

¹ <https://www.elastic.co/guide/en/elasticsearch/reference/5.2/index-modules.html#dynamic-index-settings>

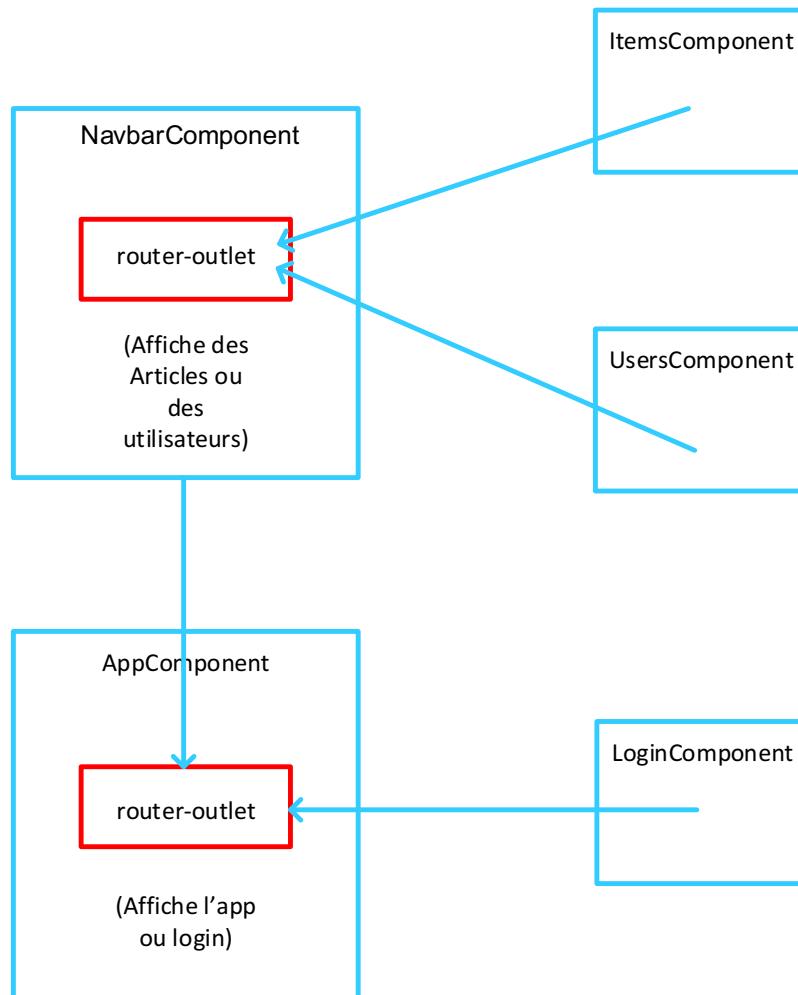
 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 32 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.6 Client Web

4.6.1 Architecture

Le client web a une architecture Modèle – Vue – Contrôleur (MVC).

L'utilisation du router Angular simplifie le développement et permet le contrôle d'accès aux URL internes (*CanActivate*) et permet d'éviter de quitter un composant dans lequel des données doivent être validées (*CanDeactivate*).



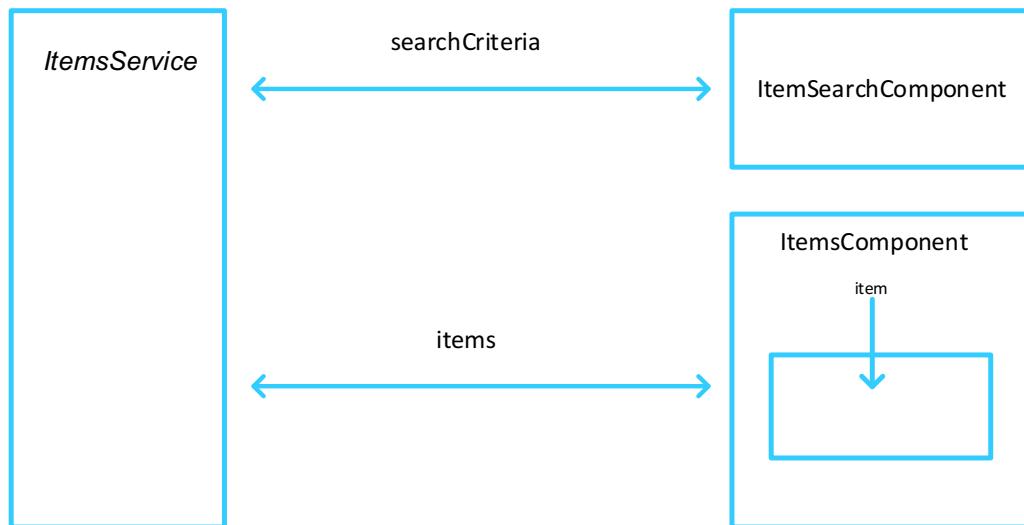
4.6.2 Les composants

Les composants *Angular* contiennent un template (la vue) et une classe (le contrôleur) dans lequel le code métier est implémenté.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 33 sur 43
	AMAZON READER	Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

4.6.3 Les services

L'accès aux données s'effectue par l'intermédiaire des services (modèle).



J'ai choisi de sauvegarder les données des articles (maximum 15) et des utilisateurs (tous) dans les services. Lors de leur utilisation dans les composants, le tableau d'objet local pointera vers celui se trouvant dans le service. En cas de suppression d'un élément dans le service, le composant supprime donc automatiquement l'élément dans la vue.

4.6.4 Affichage de messages

J'ai prévu l'affichage de messages afin que l'utilisateur sache si son action a été effectuée ou non. Ceci ne concerne bien sûr que les actions "invisibles" : par exemple, lors de la modification d'un article dans la base de données. L'affichage de messages n'auraient pas de sens pour valider une authentification correcte, par exemple, car l'utilisateur peut voir le résultat correct lorsque l'application affiche la page suivante.

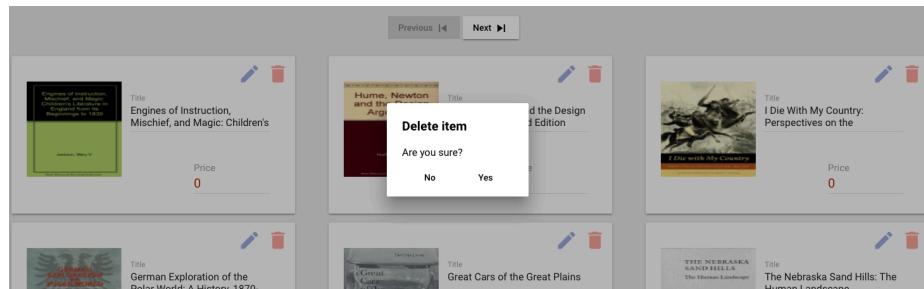
 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 34 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.6.5 Stratégie d'édition et de suppression

Pour ce projet, j'ai décidé de permettre l'édition et la suppression directement dans la liste affichée. Si la personne est autorisée, les boutons correspondants à ces actions s'afficheront sur la ligne (utilisateur) ou dans la carte (article) de l'élément.

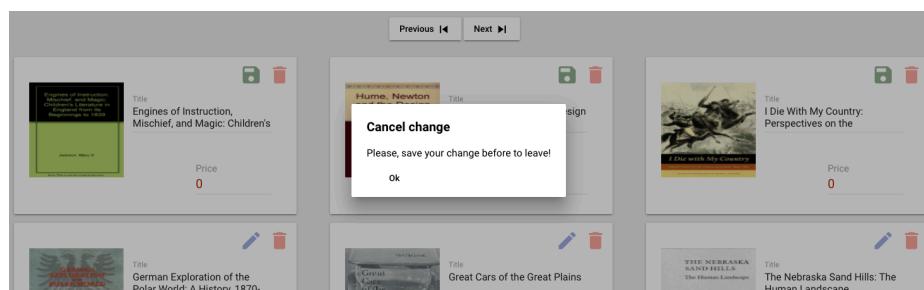
a) Suppression

Lors de la suppression d'un élément (article ou utilisateur), un message de confirmation apparaît. Une fois validé, l'élément supprimé disparaît de la liste. Aucun nouvel élément vient le remplacer dans le cas des articles.



b) Modification

Il est possible de modifier tous les éléments présents à l'écran. Cependant, si des éléments sont toujours en édition, l'application affichera un message indiquant que tous les changements d'avoir avoir été validés avant de quitter la page.



4.6.6 Importation des données

En raison du manque de temps, j'ai dû simplifier l'importation des données. J'avais prévu un onglet permettant de sélectionner un fichier et affichant l'avancement dans une barre de progression. J'aurais également interdit de quitter la page tant que la tâche était en cours.

En lieu et place, j'ai juste ajouté un bouton de sélection de fichier dans l'onglet des articles : aucun onglet dédié, aucune barre de progression et aucun contrôle de fin de tâche.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 35 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.7 Sécurité

4.7.1 Authentification

La sécurité est principalement assurée par le serveur. Lors de la requête d'authentification, le serveur génère un token jwt contenant l'identifiant de l'utilisateur en cours. J'utilise jwt pour m'assurer que le token n'a pas été modifié, bien que ça ne soit guère probable.

Le serveur génère également un cookie afin de pouvoir retrouver ce token à chaque nouvelle requête. De cette manière, je n'ai rien à développer au niveau du client.

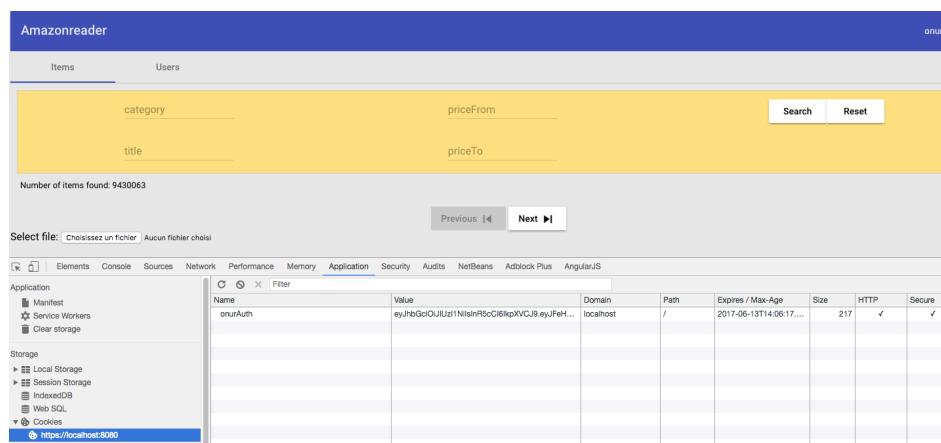
Pour que la sécurité par cookie fasse sens, il est impératif que la communication soit sécurisé car il faut éviter son vol lors du transport. Ce point a déjà été traité dans le chapitre 4.4.1 (*Serveur https*). Cela ne suffit cependant pas. J'ai également configuré le cookie en activant les 2 paramètres suivants :

- Secure = true**

Le cookie n'est transféré que si la connexion est sécurisée (https) de bout en bout.

- HttpOnly = true**

Afin d'éviter le vol du cookie dans le navigateur, ce flag spécifie qu'il n'est pas lisible par Javascript. C'est le navigateur qui se charge de le renvoyer à chaque requête effectuée sur l'URL lié.



Name	Value	Domain	Path	Expires / Max-Age	Size	HTTP	Secure
onurAuth	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJFeH... [REDACTED]	localhost	/	2017-06-13T14:06:17....	217	✓	✓

La durée de vie du token n'est pas importante. Je l'ai toutefois limitée à 32 jours (1 jour de plus que la durée de vie maximum du cookie). La durée de vie du cookie a été configurée comme spécifiée dans le cahier des charges :

- 1 jour pour les administrateurs
- 1 semaine pour les rédacteurs
- 1 mois pour les utilisateurs

Au-delà de cette période, l'utilisateur devra à nouveau s'authentifier.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 36 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.7.2 Autorisations

Grâce à l'identifiant de l'utilisateur qui se trouve dans le token qui, lui, se trouve dans le cookie, le serveur sait qui effectue la requête. Avant d'y accéder, il vérifie dans la base de données le rôle de cet utilisateur et le compare au privilège minimum transmis au middleware.

Si l'utilisateur est autorisé, le serveur exécute la fonction et retourne le résultat. Autrement, il renvoie un statut 401, signifiant que la requête n'est pas autorisée.

4.7.3 Protection des mots de passe

Afin de vérifier l'authentification des utilisateurs, il est nécessaire de disposer de leur mot de passe. Il faut donc les sauvegarder dans la base de données, ce qui est dangereux et fortement déconseillé.

La meilleure pratique consiste à ne pas sauvegarder le mot de passe directement mais une version hachée de ce mot de passe. C'est ce que j'ai fait en utilisant une librairie implémentant *bcrypt*².

En raison des performances des ordinateurs, le hachage n'est aujourd'hui plus suffisant pour protéger les mots de passe. C'est pourquoi, *bcrypt* effectue également un salage. Les avantages de *bcrypt* sont :

- Les mots de passe identiques apparaissent différents dans la base de données.
- Les attaques par force brute prennent plus de temps et coûtent plus cher en capacité de calcul.

4.7.4 Protection de l'accès à la base de données

J'ai trouvé les informations nécessaires afin de protéger l'accès à la base de données. La sécurité s'effectue en installant un utilitaire supplémentaire qui s'appelle X-Pack et qui permet de protéger l'accès par mot de passe et de crypter le transfert de données en utilisant un certificat à la norme X.509³.

L'usage de cet utilitaire est payant mais il est possible de l'installer pour des tests. Toutefois, le temps m'a manqué dans le projet et j'ai choisi de ne pas appliquer cet outil comme le cahier des charges m'y autorisait.

² <https://fr.wikipedia.org/wiki/Bcrypt>

³ <https://fr.wikipedia.org/wiki/X.509>

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 37 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.7.5 Attaques par injection SQL

Les attaques par injection SQL ne représentent pas de danger avec *ElasticSearch* qui utilise un autre langage pour les requêtes. J'ai cependant contrôlé ce qui se passe lors de l'utilisation des apostrophes, guillemets et back slashes.

Lors de l'écriture, la librairie que j'utilise tient compte de ces caractères et leur enregistrement est effectué correctement dans la base de données.

Lors de la lecture, la requête que je créé peut effectivement être affectée et je dois les gérer. Il ne peut pas y avoir de conséquences autres qu'une réponse vide due à une mauvaise requête mais cela peut être gênant pour l'utilisateur. C'est pourquoi j'ai décidé de filtrer ces caractères. Comme ils ne sont pas indexés, la recherche s'effectue correctement et le résultat est celui attendu.

4.7.6 Minification

Angular-CLI prend en charge la minification⁴. Cela rend le code plus difficile à lire côté client pour d'éventuels hackers car tous les espaces inutiles et les retours à la ligne sont supprimés (c'est pourquoi il ne faut pas oublier les ; à la fin des lignes – heureusement Tint y veille). De plus, les noms des variables sont remplacés par un nom court.

Lorsque le développement est terminé, la commande suivante permet de créer un répertoire *dist* (pour distribution) qui contiendra tous les fichiers nécessaires après concaténation et minification.

```
ng build
```

⁴ <https://fr.wikipedia.org/wiki/Minification>

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 38 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4.8 Revue de code

Le contrôle de la qualité du code est une bonne pratique. Il existe des outils (Lint) pour Go comme pour *TypeScript*.

Pour *TypeScript*, il est intégré à *Angular-CLI*. En utilisant un IDE⁵ moderne, les erreurs ou propositions d'amélioration sont ajoutées lors du codage.

Il doit être démarré manuellement pour Go et indique également les commentaires manquants lorsqu'une fonction est publique :

```
golint
```

4.9 Test

4.9.1 Tests manuels

Les tests manuels sont effectués par une personne sur la base de protocoles décrivant précisément les tâches à effectuer.

- Les protocoles de tests des APIs du serveur se trouvent dans l'annexe 2 du rapport.
- Les protocoles de tests des fonctionnalités du client se trouvent dans l'annexe 3.

4.9.2 Tests unitaires

Écrits pendant la phase de développement (ou alors commencé avant dans le cas de la TDD), ils permettent de s'assurer que notre code fonctionne après une phase de refactorisation ou suite à l'ajout de fonctionnalités.

c) En Go

En Go, les tests unitaires font partie du langage et sont très simples à écrire. Il suffit de créer un fichier dont le nom se termine par `_test.go` et d'importer la librairie `testing`. Ces fichiers ne seront pas pris en compte par le compilateur lors de la création de l'exécutable.

Ensuite, chaque test consiste à écrire une fonction dont le nom commence par `"Test"` et qui reçoit un paramètre de type `*testing.T`:

```
func TestUserFunctions(t *testing.T) {
    // Create account
    account := &OnurTPIUser.Account{
        account.UserName = "test1"
        account.RoleValue = 2
        account.Password = "test"

        database.accesser.CreateAccount(account)
}
```

⁵ EDI en français pour Environnement de Développement Intégré

 <p>ICT Berufsbildung Formation professionnelle Freiburg-Fribourg</p>	Travail Pratique Individuel 2017	Page 39 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

```
// Read account in DB
cred := &OnurTPIUser.CredentialsClient{}
cred.UserName = "test1"
cred.Password = "test"

readAccount, err := database.accesser.GetAccountByUserNameInDB(cred)
if err != nil {
    t.Error()
}

// Check account
if readAccount == nil {
    t.Error()
}

if readAccount.UserName != account.UserName ||
    readAccount.Password != account.Password {
    t.Error()
}

// SaveUser sets rolevalue to 0
if readAccount.RoleValue != 0 {
    t.Error()
}

// Delete Account
err = database.accesser.DeleteAccount(readAccount.ID)
if err != nil {
    t.Error()
}
```

La commande `t.Error()` permet de faire échouer le test lorsque la valeur contrôlée n'est pas celle attendue.

Tous les tests sont exécutés avec la commande suivante :

Sur le même principe, il est également possible de créer des exemples qui seront utilisés pour la documentation et pour les tests, ainsi que des benchmarks qui permettront de vérifier si l'on a dégradé ou amélioré les performances de notre fonction en la modifiant ou en installant une nouvelle version de Go.

Les exemples et benchmarks sont de bonnes pratiques que je n'ai cependant pas pu implémenter car le temps m'a manqué

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 40 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

d) **En Angular**

Angular-CLI prend également nativement les tests unitaires en charge. Lors de la création d'un composant ou d'un service, un fichier avec une extension ".spec.ts" est créé.

J'ai malheureusement très peu d'expérience avec les tests unitaires en Angular. En raison de la grande quantité de travail à fournir et en accord avec mon maître d'apprentissage, il a été décidé de les retirer du cahier des charges afin de libérer ce temps au profit d'autres tâches.

4.9.3 **Tests de bout-en-bout**

Les tests de bout-en-bout (ou end-to-end), permettent d'automatiser des tâches comme si un utilisateur les exécutait, et s'assurer que le résultat est conforme aux besoins du client.

J'ai malheureusement très peu d'expérience dans ce domaine car ils ne font pas partie de notre programme scolaire. En raison de la grande quantité de travail à fournir et en accord avec mon maître d'apprentissage, il a été décidé de les retirer du cahier des charges afin de libérer ce temps au profit d'autres tâches.

4.9.4 **Tests de montée en charge**

Les tests de charge permettent de contrôler les performances de notre serveur en lui envoyant un grand nombre de requêtes en parallèle pendant quelques minutes et en analysant le temps de réponse.

L'outil le plus connu pour les réaliser est *ab*⁶, très utilisé dans le monde Java. J'avais prévu d'utiliser *Vegeta*⁷ qui est un outil écrit en Go, malheureusement, le temps m'a manqué. Comme cette tâche ne figure pas dans le cahier des charges,

⁶ <https://httpd.apache.org/docs/2.4/fr/programs/ab.html>

⁷ <https://github.com/tsenart/vegeta>

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 41 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

5 Documentation

5.1 Documentation des librairies

Le langage Go offre un outil permettant de générer automatiquement une documentation de qualité pour les librairies. Pour cela, il faut respecter les prescriptions du langage et utiliser GoLint pour s'en assurer.

Les documents générés se trouvent dans l'annexe 4 du rapport.

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 42 sur 43	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

6 Améliorations

Voici la liste des points que je n'ai pas eu le temps de développer :

- Trier les articles selon l'ordre alphabétique du titre.
- Proposer aux utilisateurs des catégories lors de la saisie.
- Offrir à l'utilisateur de choisir le nombre d'articles à afficher par page.
- Importation des données comme expliqué au chapitre 4.6.6.
- Ajouter un choix permettant de sauvegarder tous les éléments édités (valable pour les articles et les utilisateurs) ou d'annuler toutes les modifications.
- Ajouter lors de l'édition un bouton pour annuler une modification (valable pour les articles et les utilisateurs).
- Apporter plus de soins à l'interface graphique.

7 Conclusion

Ce projet a été passionnant car il couvre la plupart des connaissances que j'ai acquises dans le développement d'application web durant mon apprentissage. Il m'a également permis de me familiariser avec une nouvelle technologie très intéressante que je ne connaissais pas, *ElasticSearch*.

Dans l'ensemble, le projet s'est assez bien déroulé. J'ai été confronté à quelques soucis que j'ai pu régler seul la plupart du temps ou avec l'aide de collègues.

J'éprouve cependant une petite frustration car je n'ai pas pu apporter autant de soin à mon application que je l'aurais voulu en raison de la grande quantité de travail à fournir. Au final, j'aurais aimé disposer de 2 jours supplémentaires, notamment pour les tests dans Angular. J'ai très peu d'expérience dans ce domaine, c'est pourquoi nous avons décidé avec mon maître d'apprentissage, de soulager la charge de travail en supprimant les tests unitaires et end-to-end dans Angular du cahier des charges.

C'est la première fois que j'avais à gérer autant de données dans une application. En utilisant les technologies adéquates, j'ai pu constater que la quantité n'était pas un problème : en utilisant la pagination, l'affichage des articles s'effectue à la même vitesse si ma base de données contient 1'000 ou 10'000'000 d'éléments.

Je remercie mon maître d'apprentissage pour ce projet et me réjouis d'avoir l'occasion

 ICT Berufsbildung Formation professionnelle Freiburg-Fribourg	Travail Pratique Individuel 2017	Page 43 sur 43	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

8 Annexes

Annexe 1 : Pourquoi Xpert Technologies utilise Go.

Annexe 2 : Tests manuels des APIs du serveur.

Annexe 3 : Tests manuels des fonctionnalités du client.

Annexe 4 : Documentation des librairies que j'ai créées.

Annexe 5 : Guide d'utilisation.

9 Bibliographie

https://blog.meshup.net/chrome-58-corriger-lerreur-missing_subjectaltname-avec-openssl/

<https://material.angular.io/>

<https://github.com/elastic/elasticsearch>

10 Remerciements

Je tiens à remercier M. Laurent Brülhart, maître d'apprentissage, ainsi que M. Cédric Ackermann pour leur soutien technique durant le projet.

Un énorme merci également à M. Ibrahim Nimaga et M. Cédric Ackermann pour la correction de mes documents. Ils ont vécu avec moi les derniers moments de stress du projet et ont été d'un soutien sans faille.

Annexe 1

Pourquoi Xpert Technologies utilise Go

Pourquoi Go ?

Il existe beaucoup de langages pour écrire des applications web. Parmi les plus connus, voici ceux que nous avons évalués mais pas retenus :

- Java : usine à gaz + déploiement chez les clients + stratégie peu claire d'Oracle
- C / C++ : trop compliqués
- PHP / Node.js : trop lents + déploiement chez les clients

Voici les raisons qui nous ont fait opter pour Go :

Compilation

Go est un langage compilé. Cette compilation s'effectue extrêmement rapidement (un grand nombre de type de plateformes cibles peuvent être indiquées). Ainsi, beaucoup de soucis peuvent être évités lors du déploiement (pas de machines virtuelles ou autres dépendances à préinstaller), ce qui est particulièrement appréciable lorsque le déploiement s'effectue chez le client.

Taille

Une fois compilées, les applications ont une taille de quelques mégas, souvent entre 5 et 15 MB (serveur http inclus !).

Performances

Go est un langage jeune (2009) mais a déjà des performances similaires à celles de Java. Son avenir paraît très prometteur.

Stabilité

Au cours de nos différents tests (de charge notamment), l'application ne s'est jamais arrêtée.

Footprint

En plus d'être de petite taille, les applications écrites en Go sont très économies en ressources CPU et en mémoire RAM.

Communauté

Le langage dispose d'une grande communauté. Il est ainsi possible de trouver très facilement des ressources (tutoriaux, livres, vidéos, aide sur Stackoverflow ou autres plateformes) pour la formation ou en cas de problèmes.

Librairies

Go est très populaire et dispose d'un nombre considérable de librairies et driver.

Simplicité

Le langage est très facile à prendre en main. Il ne dispose d'ailleurs que de 25 mots-clés. Il suffit ainsi de quelques heures à un développeur pour être capable de l'utiliser et de quelques jours (~2 semaines) pour le maîtriser.

Multithread

La programmation parallèle est très simple en Go. De plus, le coût d'utilisation des *goroutines* (2kB) est très limité par rapport à l'utilisation de threads (2MB) dans la plupart des autres langages.

Concurrence

Les solutions pour gérer la concurrence est également remarquable avec Go. En plus des *channels* (prônés par l'équipe de Go), il est possible d'utiliser des *mutex* ainsi que des opérations atomiques pour les nombres entiers.

Gestion des erreurs

La gestion des erreurs est plus verbeuse que dans d'autres langages car Go pousse à traiter tous les cas d'erreur (notamment grâce au fait que les procédures peuvent retourner plusieurs résultats). C'est une critique que l'on peut lire quelques fois sur les sites.

En ce qui nous concerne, c'est non seulement une bonne pratique mais ce que tout développeur professionnel devrait faire.

Lisibilité du code

Le code écrit en Go est en règle générale très simple à lire. C'est un point capital pour la maintenance des applications.

Qualité du code

Quelques bonnes pratiques proposées sur le site de Go (qui peuvent pour la plupart être contrôlées à l'aide de l'outil *golint*) ainsi que l'outil *gofmt* pour le formatage permettent d'assurer une excellente qualité du code.

Soutien

Go est soutenu par Google qui est une entreprise sachant ce qu'innover veut dire et propose de fréquentes nouvelles versions. C'est d'ailleurs cette stratégie qui a permis à Chrome de devenir le navigateur le plus utilisé en l'espace de 3 ans.

Une nouvelle version de Go paraît tous les 6 mois et, régulièrement, des correctifs sont proposées (principalement pour des problèmes liés à la sécurité).

Compatibilité des nouvelles versions

Depuis le release 1.0, les nouvelles versions n'ont pas amené d'incompatibilités. Il suffit d'installer la nouvelle version et de compiler son programme pour qu'il marche.

Annexe 2

Tests manuels des APIs du serveur

Amazon Reader

Protocole de test des APIs du serveur

Dogan Onur Oktay

Classe : 304541

13 Juin 2017

Entreprise formatrice : Xpert Technologies

 ICT Berufsbildung Formation professionnelle Formazione professionale	PROTOCOLE DE TEST DES APIS	Page 2 sur 8	
	AMAZON READER	Version 1	F
	Etabli par : ONUR OKTAY DOGAN	13 juin 2017	

Table des matières

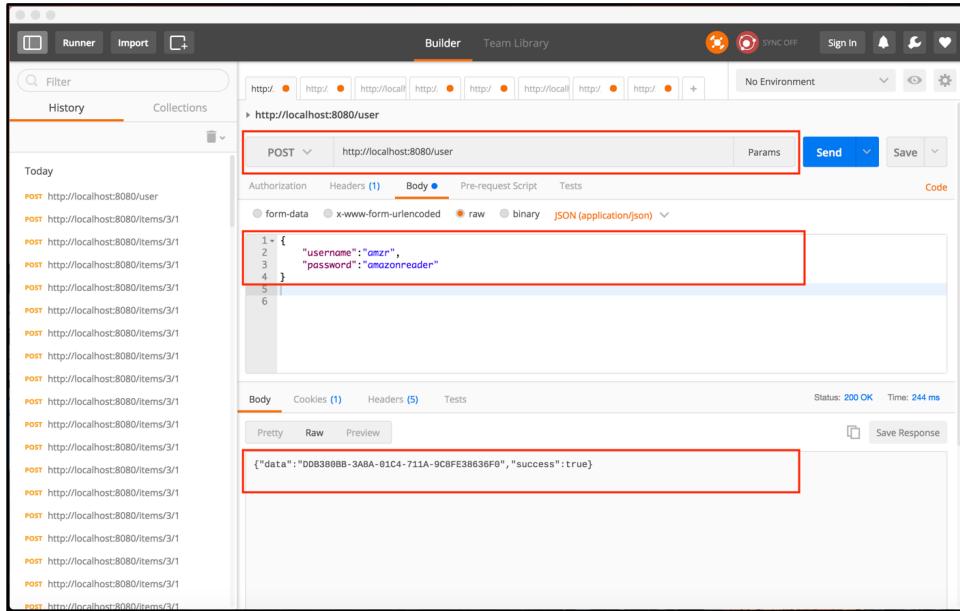
1	Introduction	3
2	Ajout d'un utilisateur :.....	3
3	Récupération des données d'un utilisateur :	3
4	Modification du rôle d'un utilisateur	4
5	Suppression d'un utilisateur	4
6	Récupération des données de tous les utilisateurs :	5
7	Login	5
8	Rechercher des articles	7
9	Modification d'un article.....	7
10	Suppression d'un article	8
11	Rechercher d'articles	8

 <p>ICT Berufsbildung Formation professionnelle Formazione professionale</p>	PROTOCOLE DE TEST DES APIS	
	Page 3 sur 8	
	AMAZON READER	Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

1 *Introduction*

J'ai utilisé Postman pour les tests manuels des APIs. Chaque test a sa propre configuration dans Postman.

2 *Ajout d'un utilisateur :*



The screenshot shows the Postman interface with a POST request to `http://localhost:8080/user`. The request body is set to `JSON (application/json)` and contains the following JSON:

```

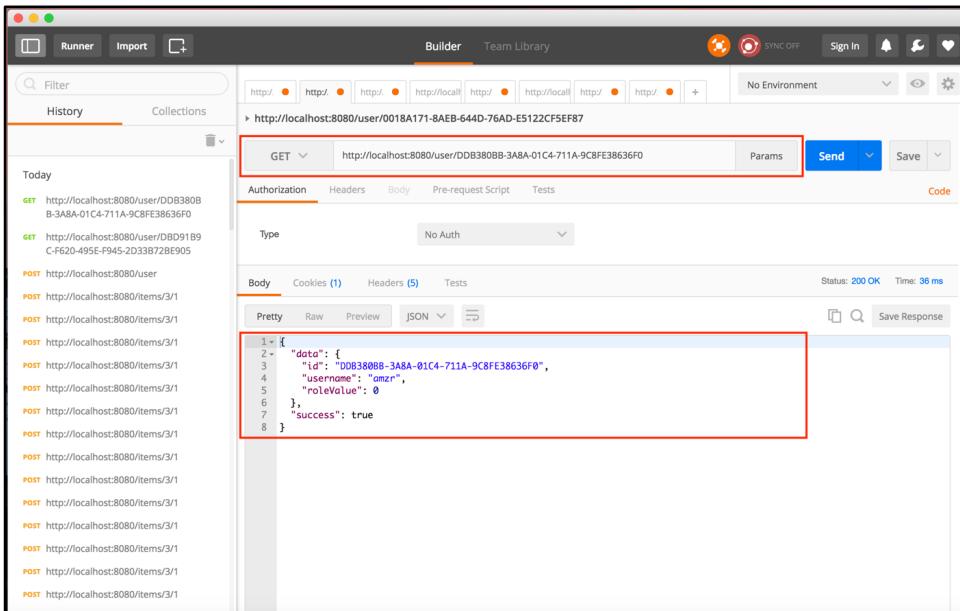
1+ {
2-   "username": "omzr",
3-   "password": "amazonreader"
4}

```

The response status is `200 OK` with a time of `244 ms`. The response body is:

```
{"data":"DDB380BB-3A8A-01C4-711A-9C8FE38636F0", "success":true}
```

3 *Récupération des données d'un utilisateur :*



The screenshot shows the Postman interface with a GET request to `http://localhost:8080/user/0018A171-8AEB-644D-76AD-E5122CF5EF87`. The response body is:

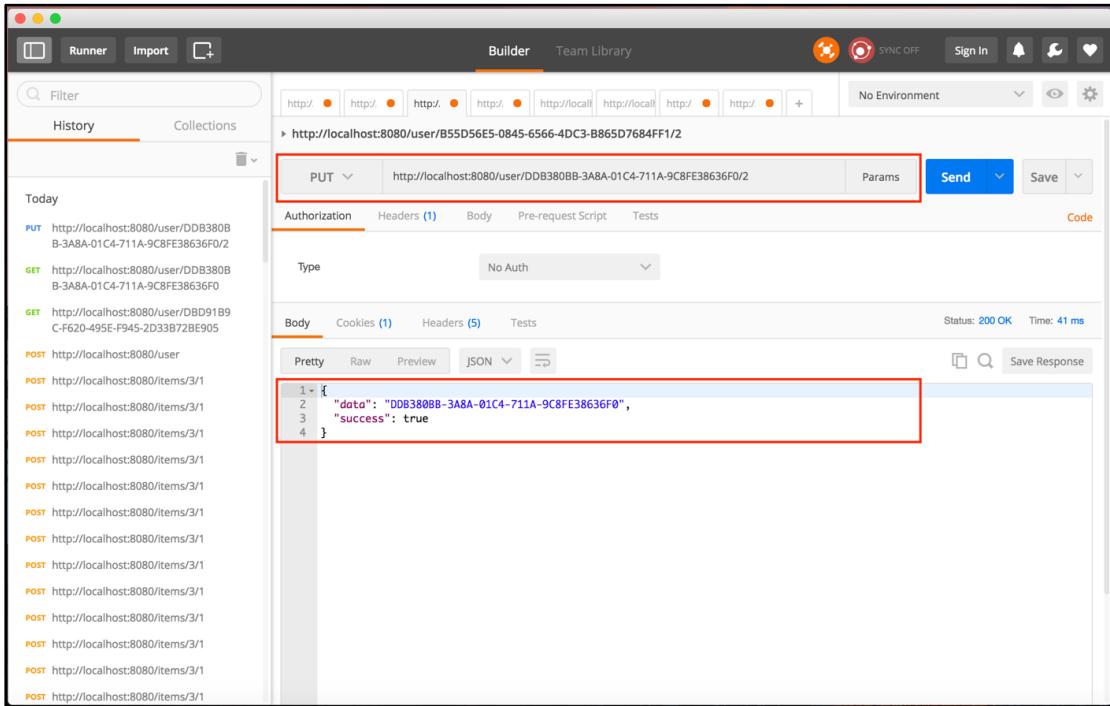
```

1+ {
2-   "data": {
3-     "id": "DDB380BB-3A8A-01C4-711A-9C8FE38636F0",
4-     "username": "omzr",
5-     "roleValue": 0
6   },
7   "success": true
8 }

```

 ICT Berufsbildung Formation professionnelle Formazione professionale	PROTOCOLE DE TEST DES APIS	Page 4 sur 8	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4 Modification du rôle d'un utilisateur



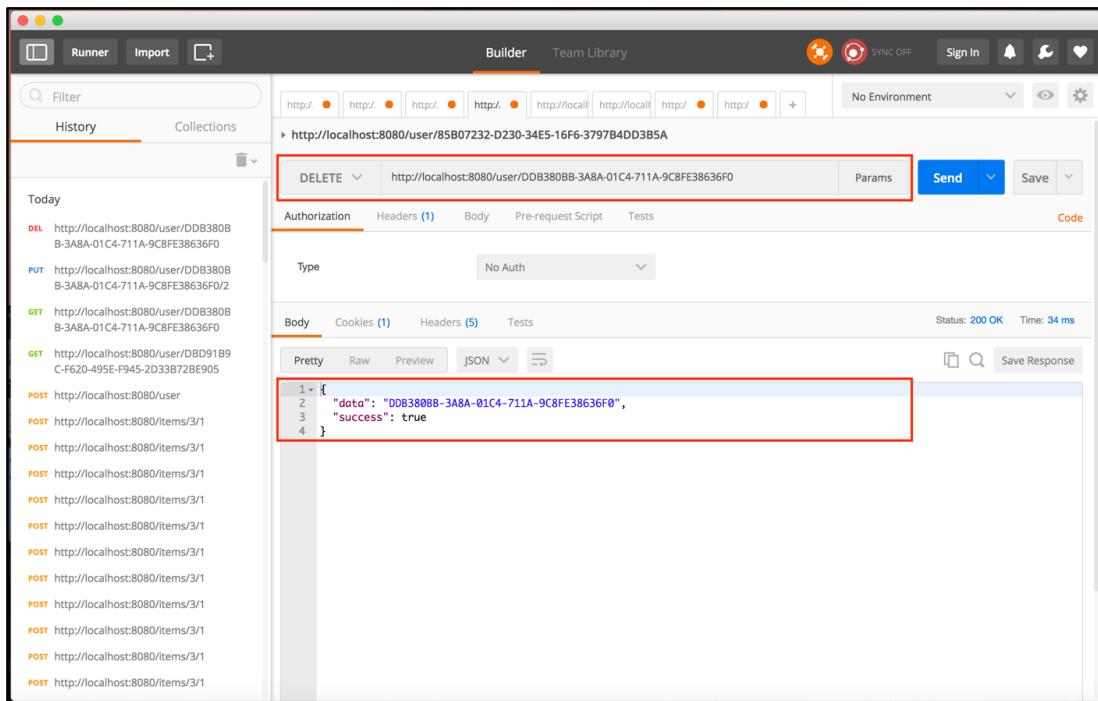
The screenshot shows the Postman interface with a successful PUT request to update a user's role. The URL is `http://localhost:8080/user/DDB380BB-3ABA-01C4-711A-9C8FE38636F0/2`. The response body is:

```

1 + {
2   "data": "DDB380BB-3ABA-01C4-711A-9C8FE38636F0",
3   "success": true
4 }

```

5 Suppression d'un utilisateur



The screenshot shows the Postman interface with a successful DELETE request to delete a user. The URL is `http://localhost:8080/user/DDB380BB-3ABA-01C4-711A-9C8FE38636F0`. The response body is:

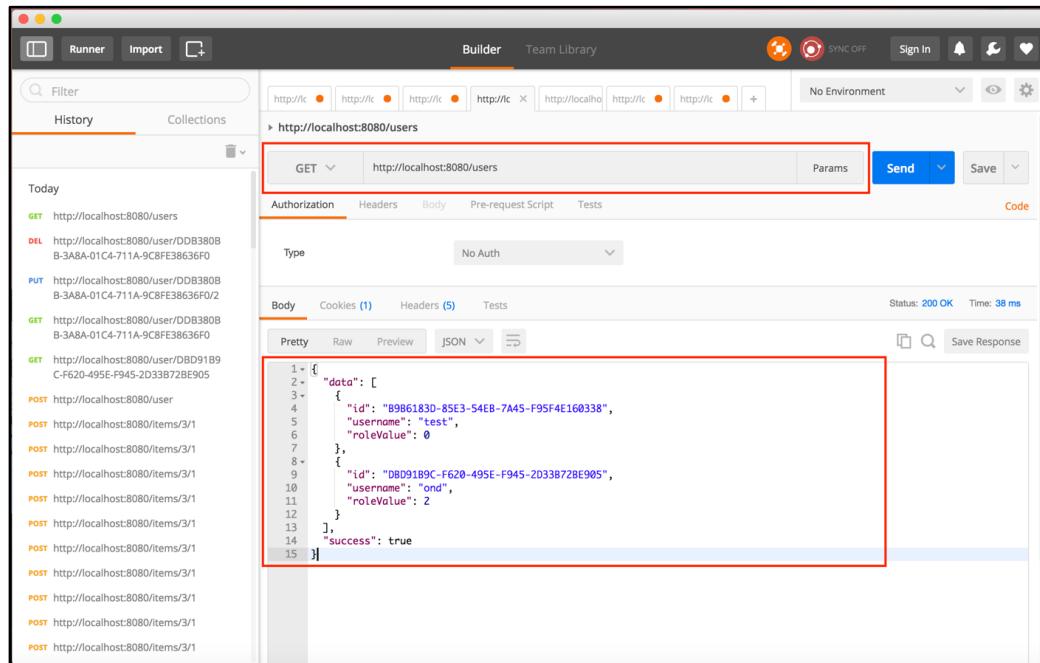
```

1 + {
2   "data": "DDB380BB-3ABA-01C4-711A-9C8FE38636F0",
3   "success": true
4 }

```

 ICT Berufsbildung Formation professionnelle Formazione professionale	PROTOCOLE DE TEST DES APIS	Page 5 sur 8	
AMAZON READER		Version 1	F
Etabli par : ONUR OKTAY DOGAN	13 juin 2017		

6 Récupération des données de tous les utilisateurs :



The screenshot shows the Postman application interface. A GET request is made to `http://localhost:8080/users`. The response body contains the following JSON data:

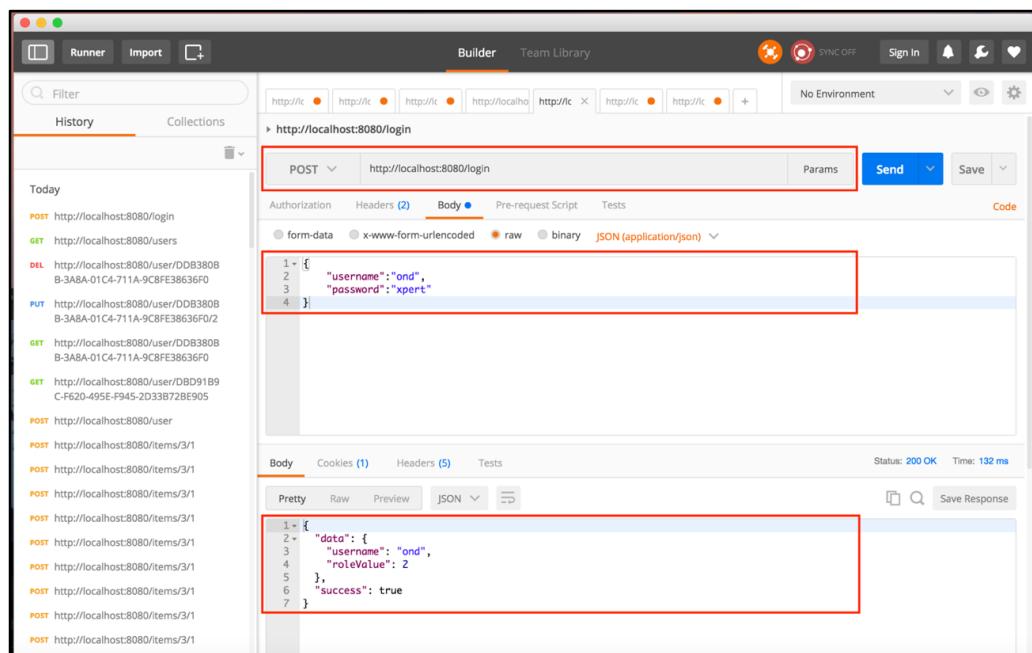
```

1+ [
2+   {
3+     "id": "B986183D-85E3-54EB-7A45-F95F4E160338",
4+     "username": "test",
5+     "roleValue": 0
6+   },
7+   {
8+     "id": "DBD91B9C-F620-495E-F945-2033B72BE905",
9+     "username": "ond",
10+    "roleValue": 2
11+  }
12+ ],
13+   "success": true
14+
15+

```

7 Login

Si les données sont correctes, le serveur retourne le nom de l'utilisateur ainsi que son rôle :



The screenshot shows the Postman application interface. A POST request is made to `http://localhost:8080/login`. The request body contains the following JSON data:

```

1+ {
2+   "username": "ond",
3+   "password": "xpert"
4+

```

The response body is highlighted with a red box and shows the user information returned by the server:

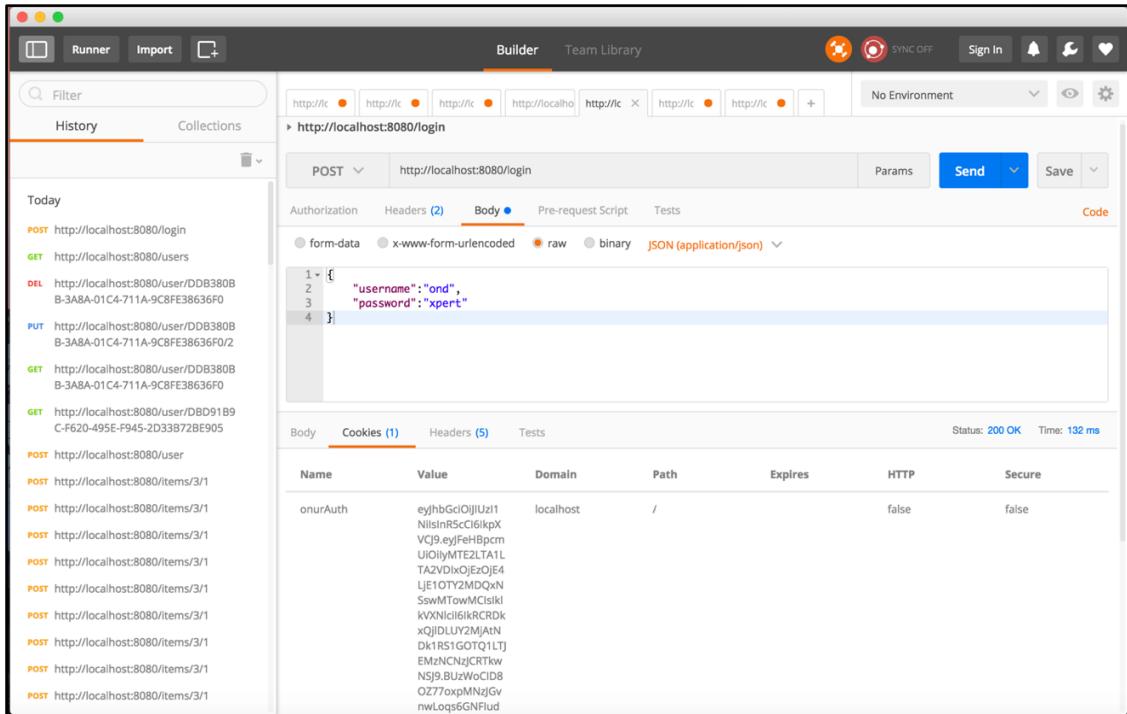
```

1+ {
2+   "data": [
3+     {
4+       "id": "DBD91B9C-F620-495E-F945-2033B72BE905",
5+       "username": "ond",
6+       "roleValue": 2
7+     }
8+   ],
9+   "success": true
10+

```

 <p>ICT Berufsbildung Formation professionnelle Formazione professionale</p>	PROTOCOLE DE TEST DES APIS		Page 6 sur 8	
	AMAZON READER		Version 1	F
	Etabli par : ONUR OKTAY DOGAN		13 juin 2017	

Le serveur retourne également un cookie qui sera utilisé plus tard pour la sécurité. Comme je n'ai pas encore crypté la communication avec le serveur, j'ai provisoirement désactivé les options http et Secure.



The screenshot shows the Postman application interface. In the top navigation bar, 'Builder' is selected. Below it, a search bar shows 'http://localhost:8080/login'. The main area displays a POST request to 'http://localhost:8080/login'. The 'Body' tab is active, showing a JSON payload:

```

1 - {
2   "username": "ond",
3   "password": "xpert"
4 }

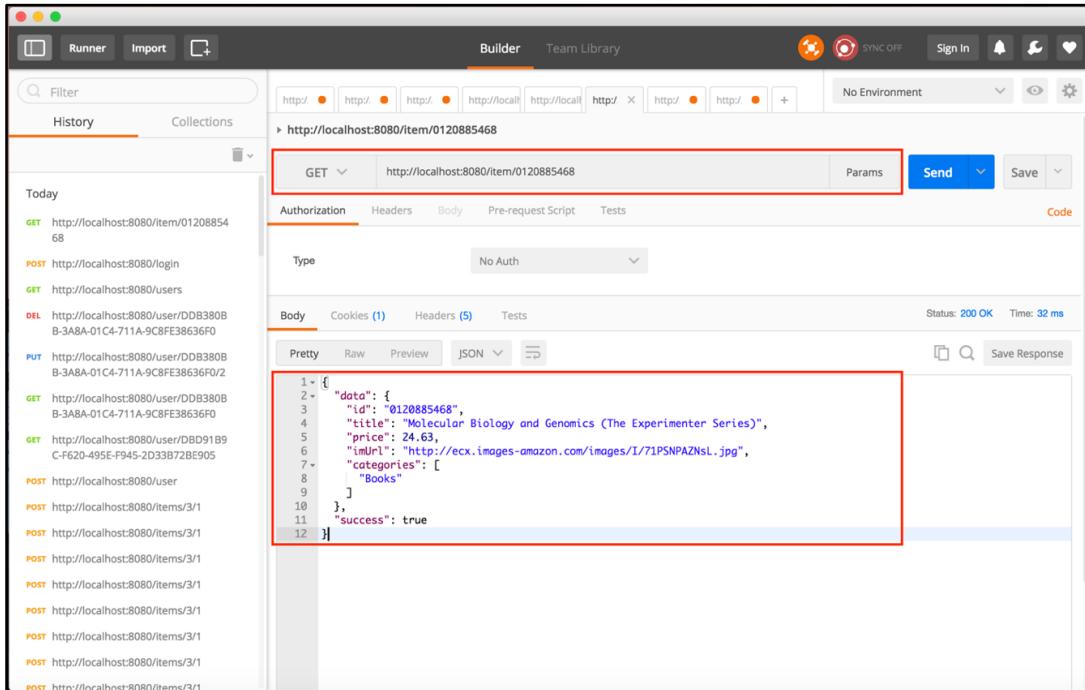
```

Below the body, the 'Cookies (1)' tab is selected, showing a single cookie named 'onurAuth' with the value: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9eyJFeHlBpcmUlOilyMTE2LTAlTAZ2VDixOjEzJE4LjE1OTY2MDQxNNSwMTowMCisIkJVNlciI6IkRCRDkxQjIDLUY2MjAtNDk1RS1GOTQ1LTJEM2NCNzjCRTkwNSj9.BUzWcID8OZ77oxpMhNzGvnwLogq6GNFlud

The status bar at the bottom indicates 'Status: 200 OK' and 'Time: 132 ms'.

 ICT Berufsbildung Formation professionnelle Formazione professionale	PROTOCOLE DE TEST DES APIS	Page 7 sur 8	
	AMAZON READER		Version 1 F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

8 Rechercher des articles



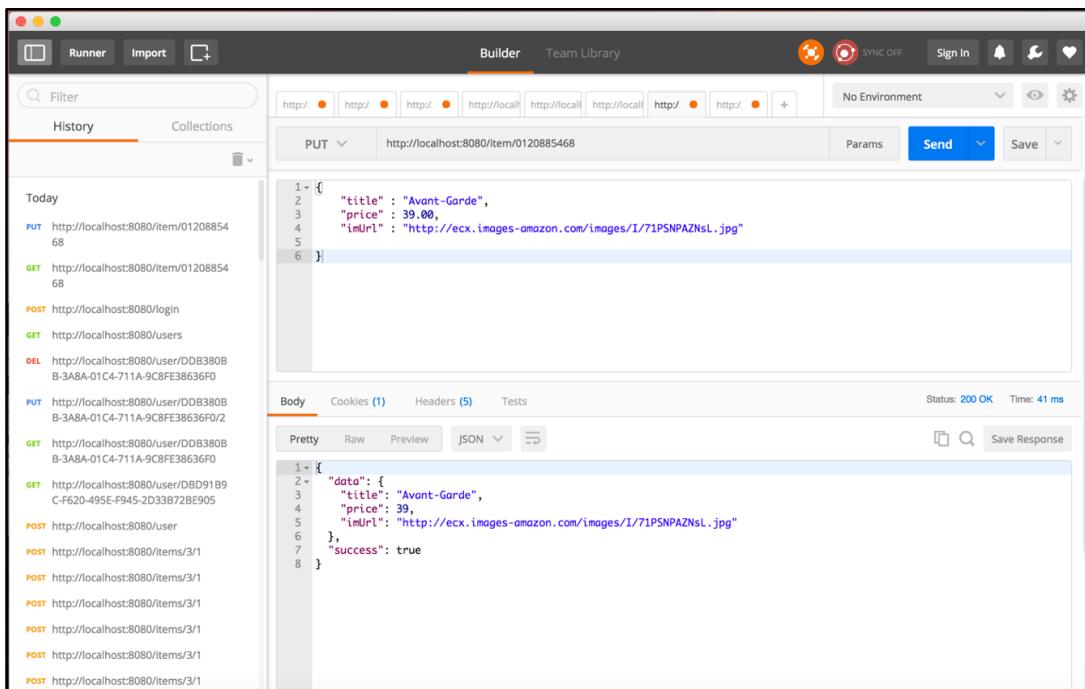
The screenshot shows the Postman interface with a successful API call. The URL in the header is `http://localhost:8080/item/0120885468`. The response body is:

```

1 - {
2 -   "data": {
3 -     "id": "0120885468",
4 -     "title": "Molecular Biology and Genomics (The Experimenter Series)",
5 -     "price": 24.63,
6 -     "imUrl": "http://ecx.images-amazon.com/images/I/71PSNPAZNsL.jpg",
7 -     "categories": [
8 -       "Books"
9 -     ]
10 -   },
11 -   "success": true
12 -

```

9 Modification d'un article



The screenshot shows the Postman interface with a successful API call. The URL in the header is `http://localhost:8080/item/0120885468`. The request body is:

```

1 - {
2 -   "title" : "Avant-Garde",
3 -   "price" : 39.00,
4 -   "imUrl" : "http://ecx.images-amazon.com/images/I/71PSNPAZNsL.jpg"
5 - }
6 -

```

The response body is:

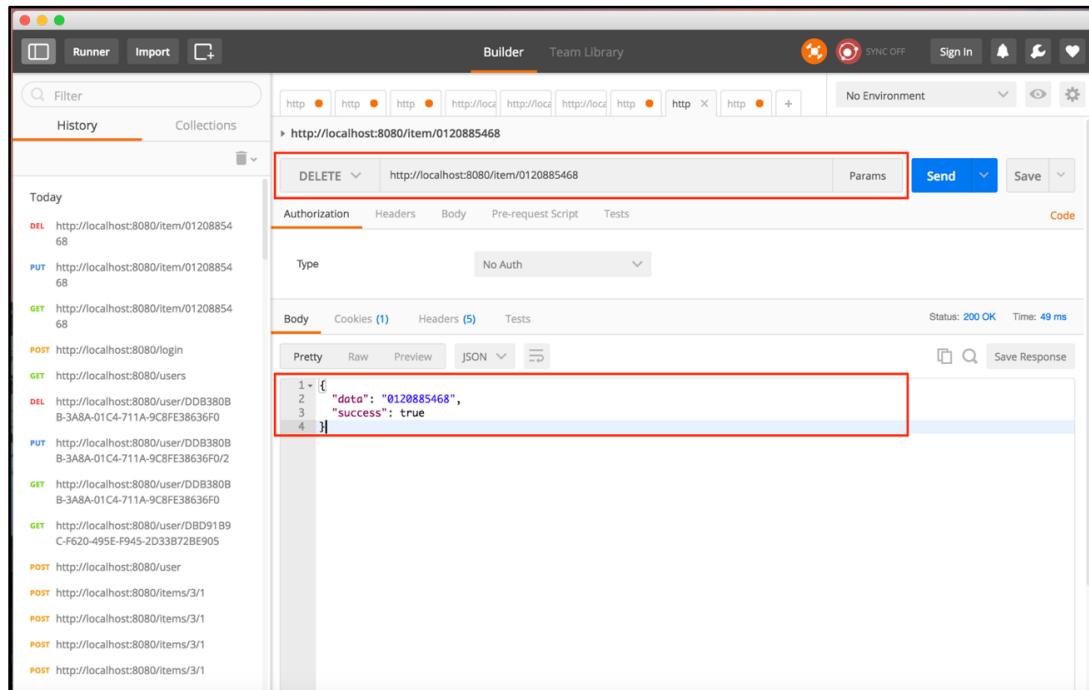
```

1 - {
2 -   "data": {
3 -     "title": "Avant-Garde",
4 -     "price": 39,
5 -     "imUrl": "http://ecx.images-amazon.com/images/I/71PSNPAZNsL.jpg"
6 -   },
7 -   "success": true
8 -

```

 ICT Berufsbildung Formation professionnelle Formazione professionale	PROTOCOLE DE TEST DES APIS	Page 8 sur 8	
	AMAZON READER		Version 1 F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

10 Suppression d'un article



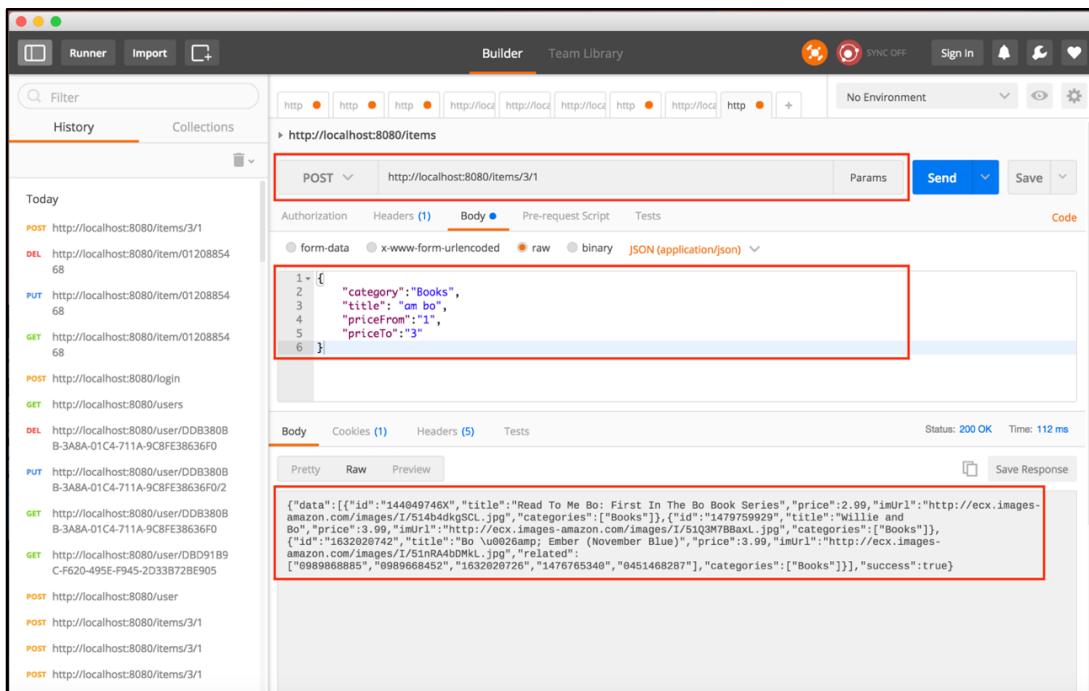
The screenshot shows the Postman interface. In the 'History' section, there is a list of various API requests. In the main panel, a DELETE request is selected with the URL `http://localhost:8080/item/0120885468`. The response body is displayed as JSON:

```

1 - [
2 -   "data": "0120885468",
3 -   "success": true
4 - ]

```

11 Rechercher d'articles



The screenshot shows the Postman interface. In the 'History' section, there is a list of various API requests. In the main panel, a POST request is selected with the URL `http://localhost:8080/items/3/1`. The request type is set to 'Body' and the content type is 'raw' with 'JSON (application/json)' selected. The request body is a JSON object:

```

1 - [
2 -   "category": "Books",
3 -   "title": "am bo",
4 -   "priceFrom": "1",
5 -   "priceTo": "3"
6 - ]

```

The response body is also a JSON object, indicating a successful search for items:

```

{"data": [{"id": "14049746X", "title": "Read To Me Bo: First In The Bo Book Series", "price": 2.99, "imUrl": "http://ecx.images-amazon.com/images/I/514bdkgSCL.jpg", "categories": ["Books"]}, {"id": "1479759929", "title": "Willie and Bo", "price": 3.99, "imUrl": "http://ecx.images-amazon.com/images/I/51Q3M7BaxL.jpg", "categories": ["Books"]}, {"id": "1632026742", "title": "Bo (Boober (November Blue)", "price": 3.99, "imUrl": "http://ecx.images-amazon.com/images/I/51nR4bDMKL.jpg", "related": [{"id": "0989868885", "imUrl": "http://ecx.images-amazon.com/images/I/51o026amp; Ember (November Blue)", "price": 3.99}], "categories": ["Books"]}], "success": true}

```

Annexe 3

Tests manuels des fonctionnalités
du client

Amazon Reader

Protocole de test des fonctionnalités du client

Dogan Onur Oktay

Classe : 304541

13 Juin 2017

Entreprise formatrice : Xpert Technologies

 ICT Berufsbildung Formation professionnelle Formazione professionale	PROTOCOLE DE TEST DES FONCTIONALITES DU CLIENT	Page 2 sur 5	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

Table des matières

1	Introduction	3
2	Création d'un compte d'utilisateur et connexion	3
2.1	Création d'un nouvel utilisateur.....	3
2.2	Connexion	3
3	Gestion des articles	3
3.1	Le message en cas de succès	4
3.2	Le message en cas d'erreur	4
4	Gestion des utilisateurs	5
4.1	Le message en cas de succès	5
4.2	Le message en cas d'erreur	5

 ICT Berufsbildung Formation professionnelle Formazione professionale	PROTOCOLE DE TEST DES FONCTIONALITES DU CLIENT	Page 3 sur 5	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

1 **Introduction**

Le protocole de tests s'adresse aux testeurs de l'application Amazon Reader et a pour but de valider manuellement toutes les fonctionnalités de l'application.

2 **Création d'un compte d'utilisateur et connexion**

2.1 **Création d'un nouvel utilisateur**

- Aller sur la route « /login ».
- Cliquer sur le lien « Create a User » et vérifier qu'il se dirige vers la route « /register ».
- Saisir un « username » et un « password » et cliquer sur le bouton « Register ».
- On doit se retrouver dans la route « /login » et un message « A new user has been successfully created » en bas de l'écran doit s'afficher.

2.2 **Connexion**

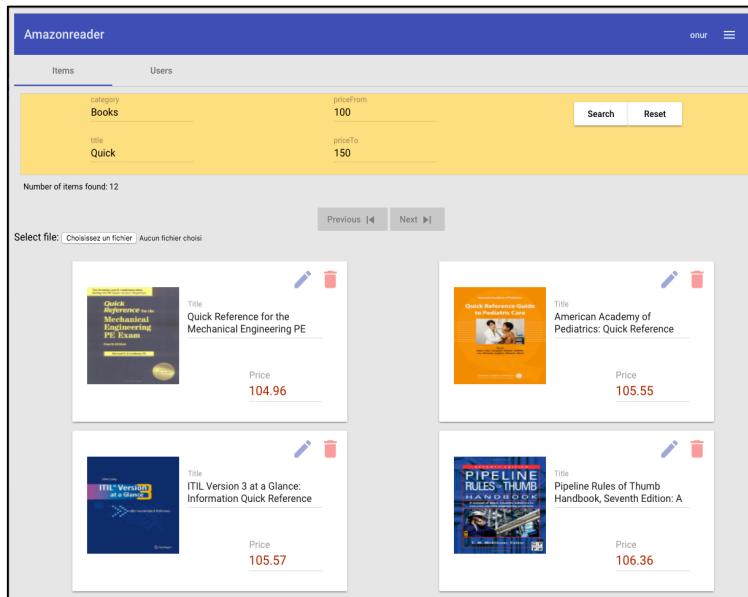
- Sur la route « /login ».
- Saisir le nom d'utilisateur (username) qui vient d'être créé et le mot de passe (password) correspondant et cliquer sur « Login ».
- Une fois l'opération effectuée, le nom d'utilisateur est affiché en droite de l'écran.
- En cas de non succès, un message « Wrong login » est affiché en bas de l'écran.

3 **Gestion des articles**

- Une fois connectée à l'application, vérifier la route « /app/items/page/1 ».
- Contrôler que tous les utilisateurs puissent faire des recherches.
- Effectuer des recherches par « category », « title », « priceFrom », « priceTo ».
- Vérifier lors de clique sur le bouton « Reset », les critères de recherches soit mis à zéro.

	PROTOCOLE DE TEST DES FONCTIONALITES DU CLIENT	Page 4 sur 5
	AMAZON READER	Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

- Vérifier si le résultat de recherche plus petit ou égal à 15 articles que l'on ne puisse pas cliquer sur le bouton « Next ».



- Contrôler le fonctionnement de bouton « Next » et « Previous ».
- Connectez-vous à l'application avec un compte « User » et vérifier que l'on ne puisse pas éditer ou supprimer un article.
- Connectez-vous à l'application avec un compte « Redactor » et « Admin » et vérifier que les boutons modifier et supprimer apparaissent. Contrôler leur fonctionnement.
- Contrôler que les utilisateurs « Admin » puissent importer des articles d'un fichier JSON.

3.1 *Le message en cas de succès*

- Modifier un article : « The item has been modified successfully »
- Supprimer un article : « The item has been modified successfully »

3.2 *Le message en cas d'erreur*

- Modifier un article: « The item can't be modified! »
- Supprimer un article : « The item can't be deleted! »

 ICT Berufsbildung Formation professionnelle Formazione professionale	PROTOCOLE DE TEST DES FONCTIONALITES DU CLIENT	Page 5 sur 5	
	AMAZON READER	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4 **Gestion des utilisateurs**

- Vérifier que cet onglet soit disponible uniquement pour les utilisateurs « Admin ».
- Vérifier que l'on ne puisse pas supprimer le dernier utilisateur « Admin ».
- Vérifier que l'on puisse modifier les rôles des utilisateurs sauf celui d'Admin.
- Vérifier que l'on puisse supprimer un utilisateur « User » et « Redactor ».

4.1 **Le message en cas de succès**

- Modifier un utilisateur : « The user role has been modified successfully »
- Supprimer un utilisateur : « The user role has been deleted successfully »

4.2 **Le message en cas d'erreur**

- Modifier un article : « The user role has not been modified! »
- Supprimer un article : « Sorry, the user can't be deleted! »

Annexe 4

Documentation des librairies
que j'ai créées

The Go Programming Language



Package util

```
import "github.com/onuroktay/amazon-reader/AmzR-Server/util"
```

[Overview](#)[Index](#)[Overview ▾](#)[Index ▾](#)

```
func CleanQuote(t string) string
func GetEncryptionKey() []byte
func GetUUID() string
```

Package files

[util.go](#)

func CleanQuote

```
func CleanQuote(t string) string
```

func GetEncryptionKey

```
func GetEncryptionKey() []byte
```

GetEncryptionKey encrypt key

func GetUUID

```
func GetUUID() string
```

GetUUID unique uuid

Build version go1.8.3.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)



The Go Programming Language

Package OnurTPIUser

```
import "github.com/onuroktay/amazon-reader/AmzR-Server/user-data"
```

[Overview](#)[Index](#)[Overview ▾](#)[Index ▾](#)

```
func CompareHashAndPassword(hashedPassword, password string) bool
type Account
type CredentialsClient
    func (c *CredentialsClient) CreateAccount() *Account
type ESResponse
type MainHits
type SubHits
type User
```

Package files

[data.go](#) [user.go](#)

func CompareHashAndPassword

```
func CompareHashAndPassword(hashedPassword, password string) bool
```

CompareHashAndPassword With bcryptHash I compare the password

type Account

Account contains the users and password data

```
type Account struct {
    User
    Password string `json:"password, omitempty"`
}
```

type CredentialsClient

CredentialsClient contains login data from client

```
type CredentialsClient struct {
    UserName string `json:"username"`
    Password string `json:"password,omitempty"`
}
```

func (*CredentialsClient) CreateAccount

```
func (c *CredentialsClient) CreateAccount() *Account
```

NewAccount creates an account with credential data

type ESResponse

```
type ESResponse struct {
    Took    int      `json:"took"`
    TimeOut bool     `json:"time_out"`
    Hits    MainHits `json:"hits"`
}
```

type MainHits

```
type MainHits struct {
    Hits  []SubHits `json:"hits"`
    Total int       `json:"total"`
}
```

type SubHits

SubHits contains a sub-data structure returns by elasticsearch

```
type SubHits struct {
    Source Account `json:"_source"`
    ID     string   `json:"_id"`
}
```

type User

User contains user data

```
type User struct {
    ID        string `json:"id,omitempty"`
    UserName string `json:"username"`
    RoleValue int    `json:"roleValue"`
}
```



The Go Programming Language

Package OnurTPIES

```
import "github.com/onuroktay/amazon-reader/AmzR-Server/elasticsearch"
```

[Overview](#)[Index](#)

Overview ▾

Index ▾

Constants

```
func SetTestMode(status bool)
type ElasticSearch
    func NewElasticSearch(indexName string) (es *ElasticSearch, err error)
    func (es *ElasticSearch) CheckIfUserExistsInDB(cred *OnurTPIUser.CredentialsClient) (bool, error)
    func (es *ElasticSearch) CreateAccount(account *OnurTPIUser.Account) (string, error)
    func (es *ElasticSearch) DeleteAccount(id string) error
    func (es *ElasticSearch) DeleteItem(id string) error
    func (es *ElasticSearch) GetAccountByIDInDB(id string) (*OnurTPIUser.Account, error)
    func (es *ElasticSearch) GetAccountByUsernameInDB(cred *OnurTPIUser.CredentialsClient)
        (*OnurTPIUser.Account, error)
    func (es *ElasticSearch) GetItemByIDInDB(id string) (*OnurTPIItem.Item, error)
    func (es *ElasticSearch) GetItems(criteria *Search) (*OnurTPIItem.SearchResponse, error)
    func (es *ElasticSearch) GetUsers() ([]*OnurTPIUser.User, error)
    func (es *ElasticSearch) SaveItem(item *OnurTPIItem.Item) (err error)
    func (es *ElasticSearch) UpdateItem(id string, item *OnurTPIItem.Item) error
    func (es *ElasticSearch) UpdateRole(id string, roleValue int) error
type Search
```

Package files

[elasticsearch.go](#) [item.go](#) [search.go](#) [user.go](#)

Constants

```
const (
    ITEM = "item"
    USER = "user"
)
```

func SetTestMode

```
func SetTestMode(status bool)
```

type ElasticSearch

```
type ElasticSearch struct {
    // contains filtered or unexported fields
}
```

func NewElasticSearch

```
func NewElasticSearch(indexName string) (es *ElasticSearch, err error)
```

NewElasticSearch open a connection to the ElasticSearch database

func (*ElasticSearch) CheckIfUserExistsInDB

```
func (es *ElasticSearch) CheckIfUserExistsInDB(cred *OnurTPIUser.CredentialsClient)
(bool, error)
```

func (*ElasticSearch) CreateAccount

```
func (es *ElasticSearch) CreateAccount(account *OnurTPIUser.Account) (string,
error)
```

SaveAccount save an account in ElasticSearch

func (*ElasticSearch) DeleteAccount

```
func (es *ElasticSearch) DeleteAccount(id string) error
```

func (*ElasticSearch) DeleteItem

```
func (es *ElasticSearch) DeleteItem(id string) error
```

func (*ElasticSearch) GetAccountByIDInDB

```
func (es *ElasticSearch) GetAccountByIDInDB(id string) (*OnurTPIUser.Account,
error)
```

func (*ElasticSearch) GetAccountByUserNameInDB

```
func (es *ElasticSearch) GetAccountByUserNameInDB(cred
*OnurTPIUser.CredentialsClient) (*OnurTPIUser.Account, error)
```

func (*ElasticSearch) GetItemByIDInDB

```
func (es *ElasticSearch) GetItemByIDInDB(id string) (*OnurTPIItem.Item, error)
```

func (*ElasticSearch) GetItems

```
func (es *ElasticSearch) GetItems(criteria *Search) (*OnurTPIItem.SearchResponse, error)
```

func (*ElasticSearch) GetUsers

```
func (es *ElasticSearch) GetUsers() ([]*OnurTPIUser.User, error)
```

func (*ElasticSearch) SaveItem

```
func (es *ElasticSearch) SaveItem(item *OnurTPIItem.Item) (err error)
```

SaveItem save an item in ElasticSearch

func (*ElasticSearch) UpdateItem

```
func (es *ElasticSearch) UpdateItem(id string, item *OnurTPIItem.Item) error
```

func (*ElasticSearch) UpdateRole

```
func (es *ElasticSearch) UpdateRole(id string, roleValue int) error
```

type Search

```
type Search struct {
    Category  string `json:"category"`
    Title     string `json:"title"`
    PriceFrom string `json:"priceFrom"`
    PriceTo   string `json:"priceTo"`
    Size      int    `json:"size"`
    From      int    `json:"from"`
}
```

Build version go1.8.3.

Except as noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)



The Go Programming Language

Package OnurTPIjsonImporter

```
import "github.com/onuroktay/amazon-reader/AmzR-pkg-import"
```

[Overview](#)
[Index](#)

[Overview ▾](#)

[Index ▾](#)

```
func ImportJSON(jsonFileName string) error
type DATABASE
type ELASTICSEARCH
    func NewElasticSearch(indexName, typeName string) (es *ELASTICSEARCH, err error)
    func (es *ELASTICSEARCH) SaveItem(item *ITEM) (err error)
type ITEM
```

Package files

database.go elasticsearch.go parser.go util.go

func ImportJSON

```
func ImportJSON(jsonFileName string) error
```

ImportJSON allows to import data coming from the given json file into the database

type DATABASE

DATABASE contains an interface with the method to implement in order to use de database

```
type DATABASE struct {
    // contains filtered or unexported fields
}
```

type ELASTICSEARCH

ELASTICSEARCH contains the information to connect with ElasticSearch

```
type ELASTICSEARCH struct {
    // contains filtered or unexported fields
}
```

}

func NewElasticSearch

```
func NewElasticSearch(indexName, typeName string) (es *ELASTICSEARCH, err error)
```

NewElasticSearch open a connection to the ElasticSearch database

func (*ELASTICSEARCH) SaveItem

```
func (es *ELASTICSEARCH) SaveItem(item *ITEM) (err error)
```

SaveItem save an item in ElasticSearch

type ITEM

ITEM contains all information about 1 item (book, ...)

```
type ITEM struct {
    ID      string `json:"id"`
    Title   string `json:"title"`
    Price   float64 `json:"price"`
    ImURL  string `json:"imUrl"`
    Related []string `json:"related"`
    Brand   string `json:"brand"`
    Categories []string `json:"categories"`
}
```

Build version go1.8.3.

Except as noted, the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)

Annexe 5

Guide d'utilisation

AmazonReader

Guide d'utilisateur

The screenshot shows the 'Amazonreader' application interface. At the top, there is a blue header bar with the title 'Amazonreader' and a user icon 'onur'. Below the header, there are two tabs: 'Items' (which is selected) and 'Users'. The main area has a yellow background. It contains search filters for 'category' (set to 'Books'), 'priceFrom', 'title', and 'priceTo'. Below these filters, a message states 'Number of items found: 2376270'. There are 'Previous' and 'Next' navigation buttons. A note says 'Select file: Choisissez un fichier Aucun fichier choisi'. Below this, there are four product cards:

- LeapFrog Leapster® Talking WORDS Factory**: An educational video game for children. It shows the product box and a small preview image.
- Pelle the Conqueror / My Life as a Dog**: A book by Axel Scheffler. It shows the book cover and a small preview image.
- TEAMWORK Your Way To Success**: A book by Michael Hyatt. It shows the book cover and a small preview image.
- The Top Ten Fears Of... Your Professional Guide To Overcoming Perceived Obstacles And Overcoming Fears**: A book by Ray H. Johnson. It shows the book cover and a small preview image.

Dogan Onur Oktay

Classe : 304541

13 Juin 2017

Entreprise formatrice : Xpert - Technologies

	Amazon Reader	Page 2 sur 9	
ICT Berufsbildung Formation professionnelle Formazione professionale	Guide Utilisateur		Version 1 F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

Table des matières

1	Introduction	3
2	Définition des différents rôles d'utilisateurs (User Role)	3
2.1	User (utilisateur).....	3
2.2	Redactor (Rédacteur).....	3
2.3	Admin (Administrateur).....	3
3	Création d'un compte d'utilisateur et connexion	4
3.1	Création d'un nouvel utilisateur.....	4
3.2	Connexion	5
3.2.1	Erreur de connexion	5
4	Opérations	6
4.1	Recherche d'articles	6
4.2	Gestion des articles.....	7
4.2.1	Édition	7
4.2.2	Suppression	8
4.3	Gestion des comptes d'utilisateurs	8
4.3.1	Édition	8
4.3.2	Suppression	9

 ICT Berufsbildung Formation professionnelle Formazione professionale	Amazon Reader	Page 3 sur 9	
	Guide Utilisateur	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

1 **Introduction**

Ce document, conçu comme un guide d'utilisateur, s'adresse aux nouveaux utilisateurs d'Amazon Reader et a pour but de décrire les différentes fonctionnalités disponibles ainsi que leurs utilisations au sein de l'API.

2 **Définition des différents rôles d'utilisateurs (User Role)**

2.1 **User (utilisateur)**

Le rôle « User » (utilisateur) est le rôle attribué par défaut lors de la création d'un nouveau compte. La seule possibilité qui lui est offerte est d'effectuer des recherches d'articles dans la base de données.

Ceux-ci reçoivent un cookie d'authentification qui expire après un mois, l'utilisateur n'a donc pas besoin de se logger à nouveau pendant cette période.

2.2 **Redactor (Rédacteur)**

Le Redactor (rédacteur) possède des droits plus étendus que le simple « User ». Il a la possibilité, en plus d'effectuer des recherches, d'édition et de supprimer des articles de la base de données.

Le cookie d'authentification du redactor à une durée de validité de sept jours.

2.3 **Admin (Administrateur)**

L'Admin (administrateur) peut également effectuer des recherches, modifier ou supprimer des articles, mais il a aussi la possibilité de gérer les différents comptes d'utilisateurs ainsi que d'édition leur rôle.

La possibilité d'édition permet à un administrateur de modifier le rôle d'un compte, comme par exemple, de le passer d'un rôle « User » à « Redactor ».

La gestion des comptes permet simplement de supprimer un compte (hormis un compte administrateur).

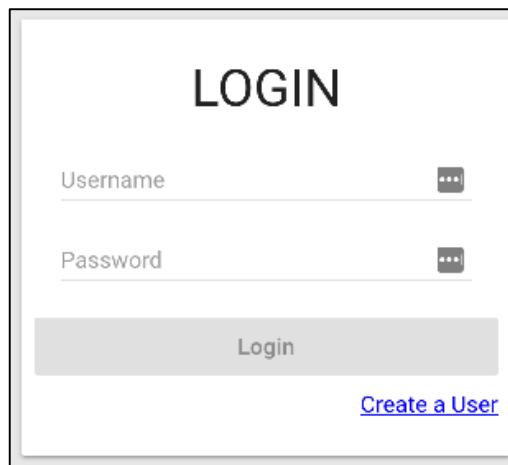
Le cookie d'authentification d'un administrateur n'est valable qu'un jour.

	Amazon Reader	Page 4 sur 9	
Guide Utilisateur		Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

3 *Création d'un compte d'utilisateur et connexion*

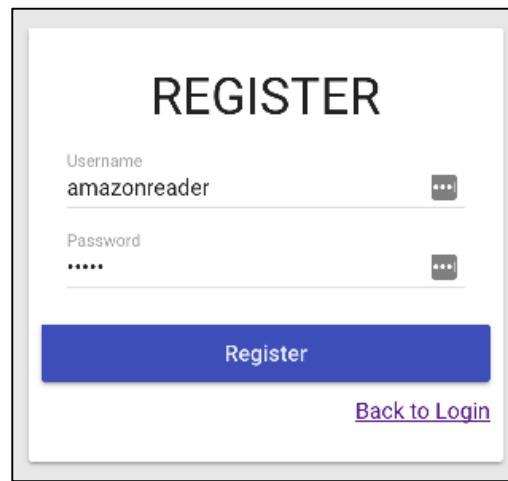
3.1 *Création d'un nouvel utilisateur*

Afin de créer un nouveau compte, il faut cliquer sur le bouton « Create a User », puis insérer le nom d'utilisateur (username) et le mot de passe (password) désirés.



The screenshot shows a 'LOGIN' page with two input fields: 'Username' and 'Password', each with a character count indicator. Below the fields is a 'Login' button. At the bottom right of the form area is a blue link labeled 'Create a User'.

En cliquant sur « s'enregistrer » (register), la saisie est validée et une redirection est effectuée sur la page de login.

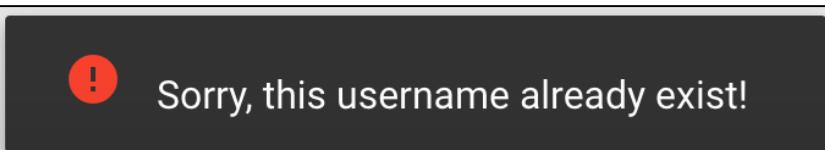


The screenshot shows a 'REGISTER' page with two input fields: 'Username' containing 'amazonreader' and 'Password' containing '.....'. Below the fields is a large blue 'Register' button. At the bottom right of the form area is a blue link labeled 'Back to Login'.

Information : Par défaut, tous les nouveaux comptes utilisent le rôle utilisateur (User).

	Amazon Reader	Page 5 sur 9	
	Guide Utilisateur		Version 1 F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

Si le nom d'utilisateur existe déjà dans la base de données, un message « Sorry, this username already exist ! » est affiché en bas de l'écran.



3.2 Connexion

Saisir le nom d'utilisateur (username) et le mot de passe (password) correspondant au compte et cliquer sur « Login ».

Une fois l'opération effectuée, le nom d'utilisateur connecté est affiché en haut à droite de l'écran, et un menu déroulant lui offre la possibilité de se déconnecter.

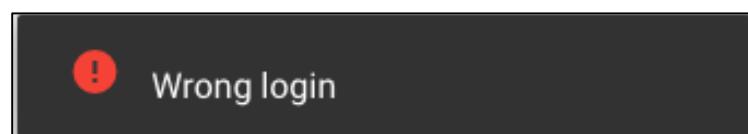
The screenshot shows the 'Amazonreader' application running in a web browser. At the top, there's a header bar with the title 'Amazonreader' and a user profile icon. Below the header, a search form has fields for 'category' and 'title', and buttons for 'Search' and 'Reset'. A message indicates 'Number of items found: 9430070'. Below the search form, there's a section titled 'Select file' with a 'Choisir le fichier...' button. The main area displays four product items in a grid:

- Top-left: Suntegrity SUN BLOCK - SPF 25+ 4 oz. (Price: 0)
- Top-right: Natural Factors Echinamide Echinacea/Goldenseal Extract (Price: 0)
- Bottom-left: Supreme Vital-Hair Hi Potency Hair & Scalp Formula w (Price: 0)
- Bottom-right: New Chapter Digestion Tonic, Ginger Honey, 4-Ounce (Price: 0)

At the bottom right of the screen, a user profile menu is visible with options like 'Déconnexion' (Logout).

3.2.1 Erreur de connexion

Si le nom d'utilisateur ou le mot de passe saisis ne correspondent pas à un compte existant, un message « Mauvaise identification » (Wrong Login) est affiché en bas de l'écran.



 ICT Berufsbildung Formation professionnelle Formazione professionale	Amazon Reader	Page 6 sur 9	
	Guide Utilisateur	Version 1	F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

4 Opérations

4.1 Recherche d'articles

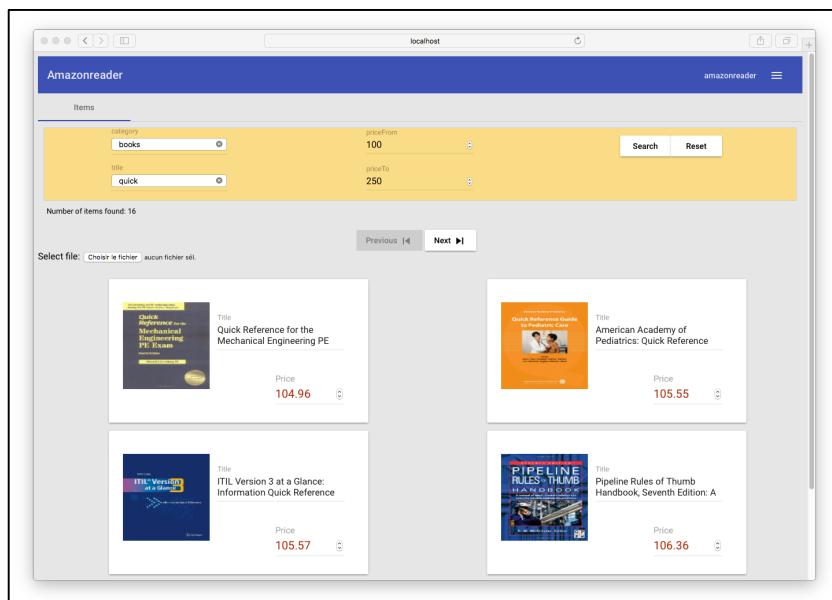
L'utilisateur a la possibilité de faire défiler les articles pages par pages ou de filtrer les résultats selon une requête. Le système affiche au maximum 15 articles par page, un bouton « précédent » (Previous) et « suivant » (Next) nous permettant de naviguer parmi les résultats.

Afin de filtrer les résultats, certains champs peuvent être renseignés suivant ce que l'utilisateur désire effectuer comme recherche. Les champs suivants sont disponibles :

- Catégorie (exemple : books)
- Titre (exemple : quick)
- Prix minimum
- Prix maximum

Une fois les critères sélectionnés, le bouton « search » permet de filtrer les résultats, alors que le bouton « reset » permet de réinitialiser tous les champs de recherche.

Après chaque chargement, une indication sur le nombre de résultats obtenus est retournée juste en dessous des filtres (Number of items found : XX).



 ICT Berufsbildung Formation professionnelle Formazione professionale	Amazon Reader	Page 7 sur 9
	Guide Utilisateur	Version 1 F
Etabli par : ONUR OKTAY DOGAN		13 juin 2017

4.2 Gestion des articles

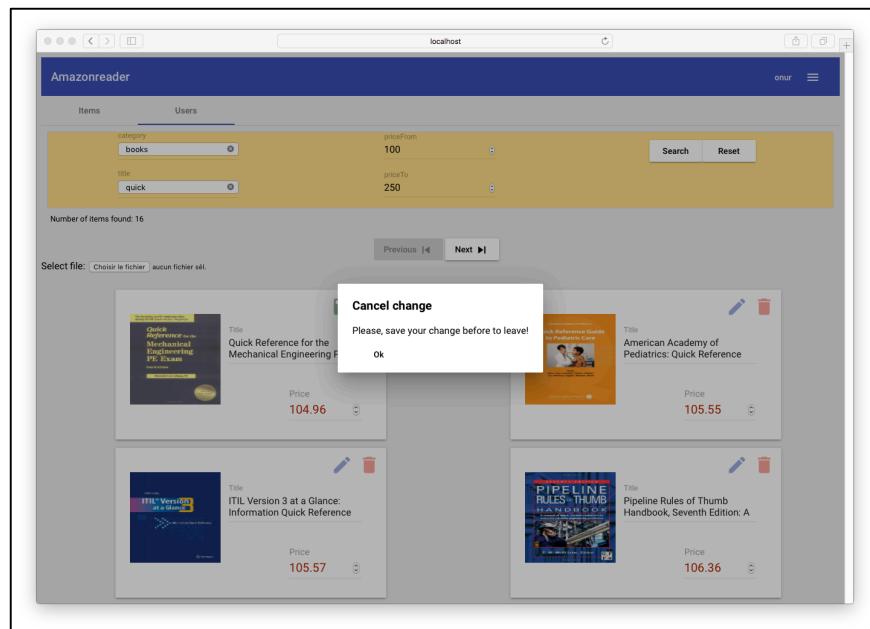
La modification (édition ou suppression) d'articles en base de données requiert un rôle particulier (Redactor ou Admin).

4.2.1 Edition

Afin d'éditer un article, le rédacteur doit cliquer sur le symbole crayon , ce qui activera la modification des champs. Une fois la modification effectuée, il est

nécessaire de cliquer sur la disquette  afin d'enregistrer la sauvegarde. Un message est affiché en bas de l'écran afin de confirmer que la sauvegarde a bien été prise en compte.

Information : Lorsque l'on se situe dans l'édition d'un élément (et donc après avoir cliqué sur le crayon), la navigation sur l'application est désactivée et un message de confirmation sera affiché avant que l'élément édité soit sauvegardé.

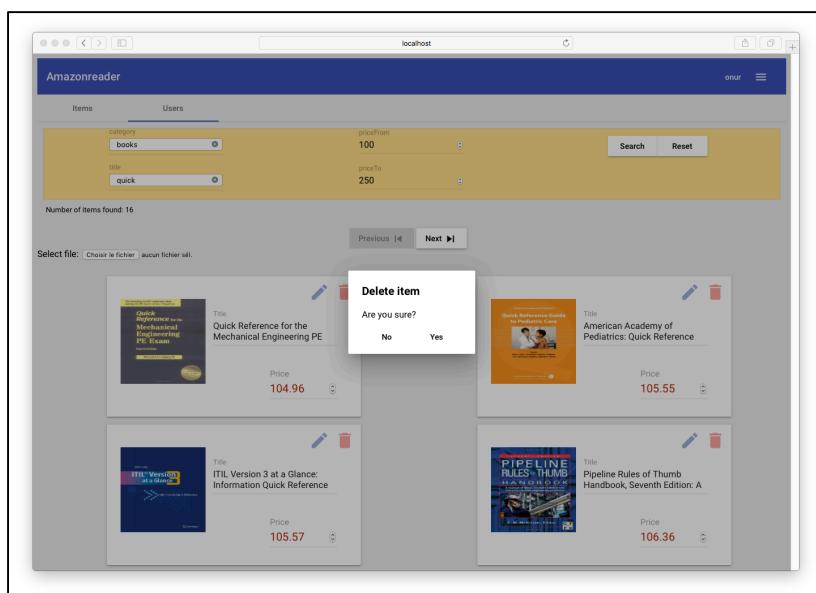


 ICT Berufsbildung Formation professionnelle Formazione professionale	Amazon Reader	Page 8 sur 9	
Etabli par : ONUR OKTAY DOGAN	Guide Utilisateur	Version 1	F
		13 juin 2017	

4.2.2 Suppression



Pour supprimer un élément, il faut cliquer sur l'icône corbeille. Une fenêtre de validation s'ouvrira afin de vérifier qu'il s'agit bien de l'opération désirée. Si le choix est validé, l'élément est supprimé de la base de données et un message de confirmation est affiché au bas de l'écran.



4.3 Gestion des comptes d'utilisateurs

L'administrateur est le seul rôle à posséder un onglet supplémentaire nommé « Utilisateurs » (Users), affiché à côté de l'onglet « articles » (Items). Celui-ci lui permet la gestion des comptes (à l'exception des comptes Administrateurs).

Lorsque l'on clique sur celui-ci, une liste des utilisateurs est visible.

4.3.1 Édition

Afin d'éditer un utilisateur, il est nécessaire de cliquer sur l'icône « crayon » afin d'activer la modification du champ permettant de sélectionner le rôle de l'utilisateur. Après avoir modifié le rôle comme désiré, il faut cliquer sur la disquette pour sauvegarder la modification effectuée. Un message est affiché en bas de l'écran pour confirmer le changement.

	Amazon Reader	Page 9 sur 9	
ICT Berufsbildung Formation professionnelle Formazione professionale	Guide Utilisateur		Version 1 F
Etabli par : ONUR OKTAY DOGAN			13 juin 2017

Information : Lorsque l'on se situe dans l'édition d'un élément (et donc après avoir cliqué sur le crayon), la navigation sur l'application est désactivée et un message d'erreur sera affiché avant que l'élément édité soit sauvegardé.

4.3.2 Suppression

Pour supprimer un compte d'utilisateur, il faut cliquer sur l'icône représentant une corbeille. Une fenêtre de validation s'ouvrira afin de vérifier qu'il s'agit bien de l'opération désirée. Si le choix est validé, l'élément est supprimé de la base de données et un message de confirmation est affiché au bas de l'écran.

Information : Si le système rencontre une erreur lors de la suppression d'un compte, un message informera l'utilisateur qu'une erreur est survenue et que la suppression n'est donc pas effective.



The Go Programming Language

Package OnurTPIItem

```
import "github.com/onuroktay/amazon-reader/AmzR-Server/item-data"
```

[Overview](#)[Index](#)

Overview ▾

Index ▾

```
type ESResponse
type Item
type MainHits
type SearchResponse
type SubHits
```

Package files

[data.go](#) [item.go](#)

type ESResponse

```
type ESResponse struct {
    Took      int      `json:"took"`
    TimeOut  bool     `json:"time_out"`
    Hits     MainHits `json:"hits"`
}
```

type Item

Item structure

```
type Item struct {
    ID          string   `json:"id,omitempty"`
    Title       string   `json:"title,omitempty"`
    Price       float64  `json:"price"`
    ImgURL     string   `json:"imUrl,omitempty"`
    Related    []string  `json:"related,omitempty"`
    SalesRank  string   `json:"salesRank,omitempty"`
    Brand      string   `json:"brand,omitempty"`
    Categories []string  `json:"categories,omitempty"`
}
```

type MainHits

```
type MainHits struct {
    Hits  []SubHits `json:"hits"`
    Total int       `json:"total"`
}
```

type SearchResponse

```
type SearchResponse struct {
    Items []*Item `json:"items"`
    Total int     `json:"total"`
}
```

type SubHits

SubHits contains a sub-data structure returns by elasticsearch

```
type SubHits struct {
    Source Item   `json:"_source"`
    ID     string `json:"_id"`
}
```

Build version go1.8.3.

Except as [noted](#), the content of this page is licensed under the Creative Commons Attribution 3.0 License, and code is licensed under a [BSD license](#).

[Terms of Service](#) | [Privacy Policy](#)