

DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ RISPAL
Nom d'usage ▶
Prénom ▶ Gwenaelle
Adresse ▶ 4 square Daumesnil
94300 Vincennes

Titre professionnel visé

Développeur web et web mobile – Niveau III

MODALITE D'ACCES :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.
Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.



<http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	5
▶ Interface utilisateur liée à une API	p.	5
▶ WordPress e-commerce	p.	7
Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité	p.	9
▶ API JDBC	p.	9
Titres, diplômes, CQP, attestations de formation <i>(facultatif)</i>	p.	11
Déclaration sur l'honneur	p.	12
Documents illustrant la pratique professionnelle <i>(facultatif)</i>	p.	13
Annexes <i>(Si le RC le prévoit)</i>	p.	23

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°1 ► Interface utilisateur liée à une API

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Ce projet a été réalisé pendant la préformation à mon alternance, nous devons créer une application utilisant les données fournies par une API de notre choix, trouvable sur internet. J'ai choisi d'utiliser l'API Harry Potter mettant à disposition des informations sur les personnages de la saga. J'ai donc décidé de développer une application recensant les personnages d'Harry Potter, et donnant la possibilité de mettre en favori nos personnages préférés.

J'ai commencé par dessiner les maquettes, puis j'ai créé mon projet à l'aide d'**Angular CLI** et enfin j'ai installé **Angular Material**. J'ai ensuite commencé à créer les composants qui formeront la structure de mon projet et le service qui va permettre à mon application et à l'API de communiquer.

Quand l'application se lance, le composant en charge d'afficher les personnages va demander au service de récupérer les données de l'API. Cette opération étant asynchrone, le composant doit s'abonner au résultat que va renvoyer le service. Le composant stocke ensuite les données renvoyées par ce service dans un tableau.

Concernant le système de favoris, lorsque l'on clique sur l'icône de favori d'un personnage, le service va stocker une copie du personnage dans un tableau. Ainsi le composant s'occupant de l'affichage des favoris pourra y accéder facilement sans avoir à communiquer directement avec le composant des personnages.

Pour finir, je me suis occupée du design de mon application à l'aide d'Angular Material et de pages CSS.

2. Précisez les moyens utilisés :

Pour développer cette application, j'ai utilisé l'éditeur de texte **Sublime Text**.

J'ai dessiné les maquettes à la main sur une feuille blanche.

J'ai utilisé **Angular 4** ainsi que le framework **Angular Material** pour le design de l'application, et pour l'API, j'ai utilisé celle trouvable à l'adresse <https://hp-api.herokuapp.com/>.

Enfin j'ai utilisé **Git** et **GitHub** pour le versioning du projet.

3. Avec qui avez-vous travaillé ?

Ce projet étant un exercice personnel demandé par ma formatrice, j'ai travaillé seule dessus.

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ► *Simplon.co*

Chantier, atelier, service ► Projet exercice pendant la formation POEC JAVA

Période d'exercice ► Du : 25/10/2017 au : 27/10/2017

5. Informations complémentaires (facultatif)

Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 2 ► **WordPress e-commerce**

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai réalisé ce projet pendant mes semaines de formation en alternance. Le but étant de créer un site WordPress, nous avons le choix entre faire un e-commerce ou un blog. Personnellement, j'ai choisi de faire un e-commerce car nous avons déjà réalisé un blog quelques semaines auparavant à l'occasion d'un cours sur Angular.

J'ai donc commencé par installer **VirtualBox**, **Vagrant** et **Varying Vagrant Vagrants** (VVV) sur mon pc afin de pouvoir installer **WordPress** en local. J'ai aussi utilisé le plugin **vagrant-hostupdater** qui va modifier le fichier des hosts afin de pouvoir accéder aux instances de WordPress via des URLs simples comme <http://vvv.test> par exemple. Après avoir provisionné mon serveur avec tous les composants nécessaires, j'ai créé une nouvelle instance de site WordPress sur **VVV**. Après avoir vérifié que mon site marchait bien en local, j'ai installé le thème **Shop Isle** et les extensions nécessaires à mon e-commerce : j'ai utilisé **Elementor** pour construire la page d'accueil de façon plus personnelle, **WPForms Lite** afin de créer un formulaire de contact et **WooCommerce** couplé à **WooCommercePaypal Checkout Gateway** pour faire de mon site WordPress un véritable e-commerce.

J'ai créé mes articles que j'ai importé dans mon site via un fichier .csv. Puis j'ai modifié l'apparence du site pour qu'il corresponde plus à mes attentes.

J'ai ensuite créé deux nouveaux utilisateurs : un shop manager et un customer. N'en ayant pas d'utilité dans le cadre d'un e-commerce, je n'ai pas créé de compte éditeur, auteur, contributeur ou abonné.

Pour le déploiement du site, j'ai utilisé l'option gratuite de **Byethost.com** et leur installateur **Softaculous** pour installer WordPress, puis j'ai utilisé l'extension **All-in-One WP Migration** pour exporter mon site local dans un fichier .wordpress, puis l'importer sur mon nouveau serveur.

Mon e-commerce WordPress est donc visible à l'adresse <http://doggy-treats.byethost6.com/wp/>

2. Précisez les moyens utilisés :

VirtualBox : logiciel de virtualisation.

Vagrant : logiciel libre et open-source pour la création et la configuration des environnements de développement virtuel.

Varying Vagrant Vagrants : configuration open source de vagrant spécialement conçue pour WordPress.

Extensions WordPress : **Elementor**, **WP Forms Lite**, **WooCommerce**, **WooCommercePaypal Checkout Gateway**, **All-in-One WP Migration**.

DOSSIER PROFESSIONNEL (DP)

Pixabay : site web de partage de photographies et de vidéos libres de droit.

3. Avec qui avez-vous travaillé ?

Ce projet étant un exercice personnel demandé par ma formatrice, j'ai travaillé seule dessus.

4. Contexte

Nom de l'entreprise, organisme ou association ► *Simplon.co*

Chantier, atelier, service ► Projet exercice pendant la formation en alternance

Période d'exercice ► Du : *03/09/2018* au : *07/09/2018*

5. Informations complémentaires (facultatif)

Des captures d'écran du site sont disponibles dans la partie Documents illustrant la pratique professionnelle.

Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n° 1 ► API JDBC

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'ai effectué ce projet pendant ma formation en alternance dans l'optique de bien assimiler la façon de développer une API REST en JDBC. J'ai donc choisi de développer le backend d'une application de collection de livres.

J'ai commencé par faire un diagramme de classe et un diagramme Entité-Relation grâce au logiciel StarUML.

J'ai ensuite initialisé un projet SpringBoot via SpringCLI, j'ai créé les fichiers schema.sql et data.sql dans le dossier resources pour créer les tables de ma base de données et y insérer des entrées. Ensuite j'ai créé les entités de mes deux classes Book et Author ainsi que leurs mappers respectifs. Les mappers permettent de transformer directement le résultat des requêtes SQL en objet java. J'ai ensuite créé les repository des deux classes Book et Author qui sont chargés d'exécuter les requêtes SQL. Puis j'ai créé un service pour chaque classe, c'est lui qui se chargera de faire communiquer les repository avec les controllers.

Et enfin, j'ai créé les controllers de Book et Author, puis j'ai testé que les requêtes marchaient bien grâce à Insomnia.

2. Précisez les moyens utilisés :

J'ai utilisé StarUML pour faire le diagramme Entité-Relation et le diagramme de classe.

Pour développer ce projet j'ai utilisé l'environnement de développement IntelliJ IDEA et j'ai utilisé git et GitHub pour le versioning. Il s'agit d'un projet SpringBoot avec les dépendances Web, JDBC et PostgreSQL.

Pour la base de données, j'ai utilisé PostgreSQL avec PgAdmin et pour tester les requêtes, j'ai utilisé Insomnia.

3. Avec qui avez-vous travaillé ?

Ce projet étant un exercice que j'ai fait à titre personnel pour comprendre le fonctionnement d'un CRUD en JDBC, j'ai travaillé seule dessus.

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ► *Simplon.co*

Chantier, atelier, service ► Projet exercice pendant la formation en alternance

Période d'exercice ► Du : *09/11/2018* au : *11/11/2018*

5. Informations complémentaires (*facultatif*)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Visual Basic for Applications	Next Formation – Formation dans le cadre du DIF Entreprise	2013
Apprenez à créer votre site web avec HTML5 et CSS3	OpenClassrooms	17/11/2014
EU Scholarship – Android Development for Beginners	Udacity – Formation dans le cadre d’une bourse offerte par Google	01/04/2017
Modélisez, implémentez et requêtez une base de données relationnelle avec UML et SQL	OpenClassrooms	26/01/2018

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Gwenaëlle RISPAL ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur(e) des réalisations jointes.

Fait à Vincennes le 08/12/2018

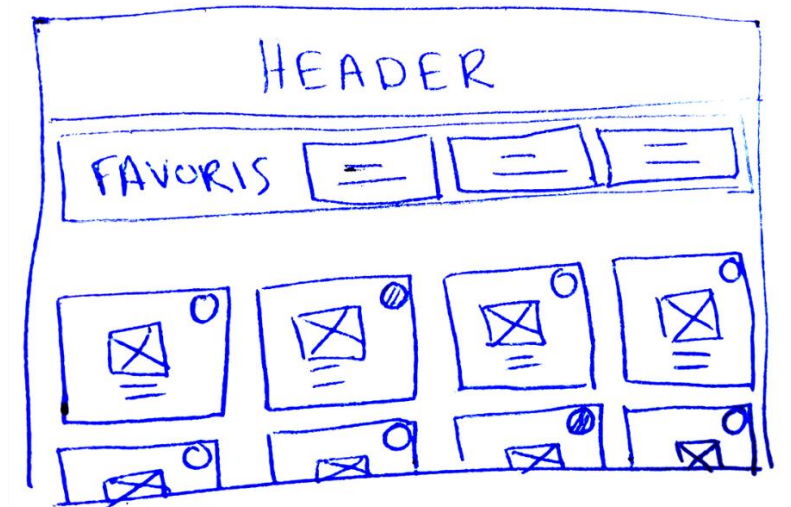
pour faire valoir ce que de droit.

Signature :

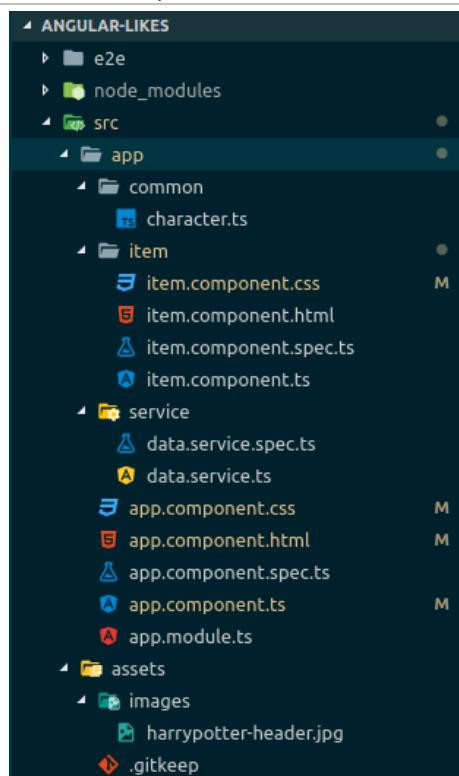
Documents illustrant la pratique professionnelle

(facultatif)

Documents relatifs à Interface utilisateur liée à une API

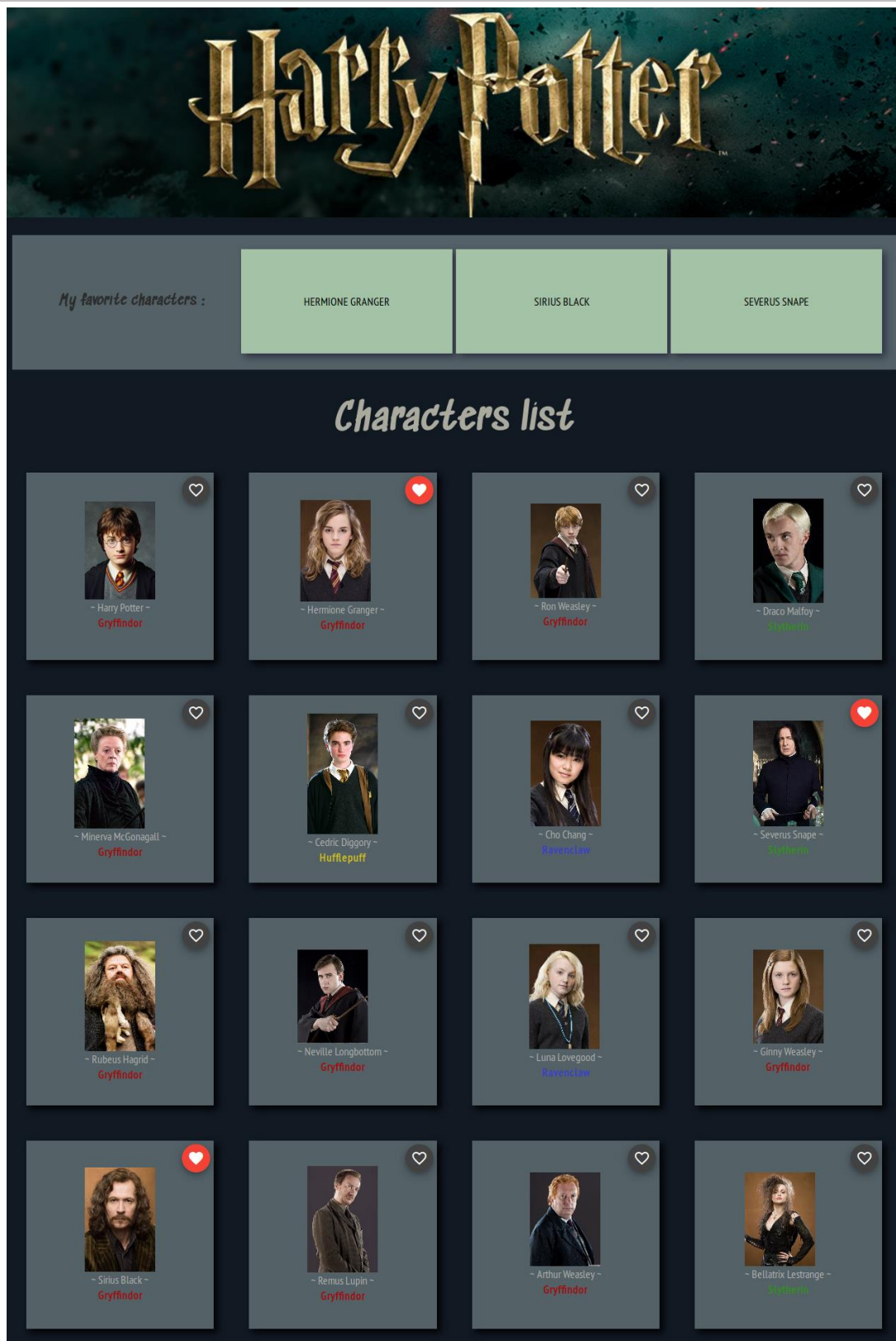


Maquette de l'interface



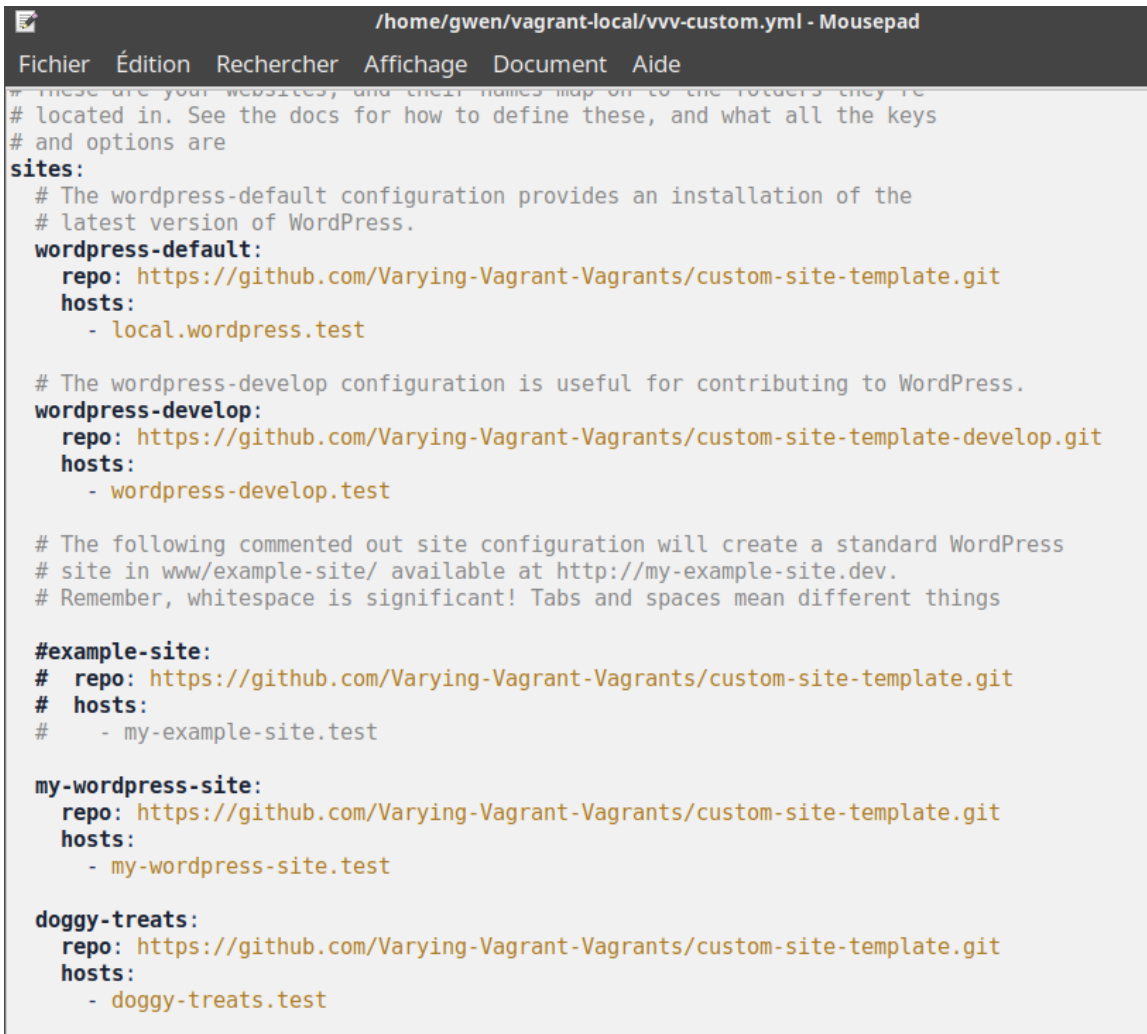
Structure de mon application Angular

DOSSIER PROFESSIONNEL (DP)



Capture d'écran de mon application finie

Documents relatifs au WordPress e-commerce



The screenshot shows a text editor window titled "/home/gwen/vagrant-local/vvv-custom.yml - Mousepad". The menu bar includes "Fichier", "Édition", "Rechercher", "Affichage", "Document", and "Aide". The content of the file is a YAML configuration for VVV sites. It starts with a comment about mapping folders to websites. Then it defines a "sites:" section with several configurations: "wordpress-default" (latest version), "wordpress-develop" (for contributing), and a commented-out "example-site". Below these, it defines "my-wordpress-site" and "doggy-treats", each with a repository URL and a host. The repository URL for all is "https://github.com/Varying-Vagrant-Vagrants/custom-site-template.git".

```
# These are your websites, and their names map on to the folders they're
# located in. See the docs for how to define these, and what all the keys
# and options are
sites:
  # The wordpress-default configuration provides an installation of the
  # latest version of WordPress.
  wordpress-default:
    repo: https://github.com/Varying-Vagrant-Vagrants/custom-site-template.git
    hosts:
      - local.wordpress.test

  # The wordpress-develop configuration is useful for contributing to WordPress.
  wordpress-develop:
    repo: https://github.com/Varying-Vagrant-Vagrants/custom-site-template-develop.git
    hosts:
      - wordpress-develop.test

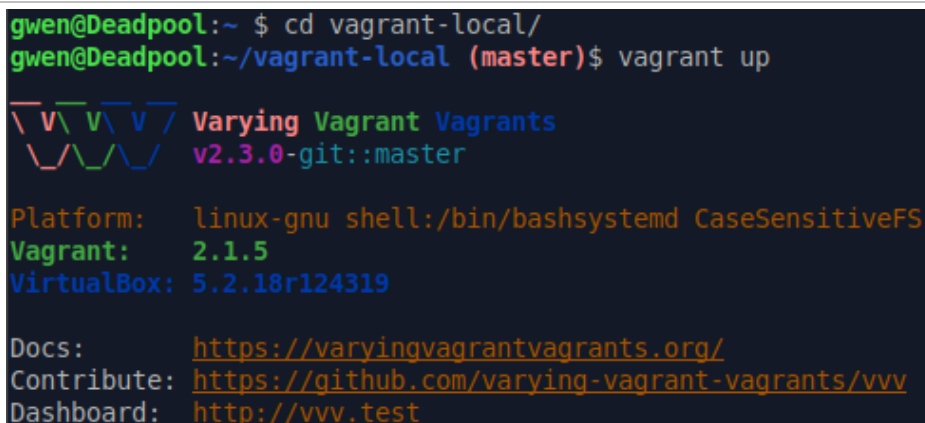
  # The following commented out site configuration will create a standard WordPress
  # site in www/example-site/ available at http://my-example-site.dev.
  # Remember, whitespace is significant! Tabs and spaces mean different things

  #example-site:
  #  repo: https://github.com/Varying-Vagrant-Vagrants/custom-site-template.git
  #  hosts:
  #    - my-example-site.test

  my-wordpress-site:
    repo: https://github.com/Varying-Vagrant-Vagrants/custom-site-template.git
    hosts:
      - my-wordpress-site.test

  doggy-treats:
    repo: https://github.com/Varying-Vagrant-Vagrants/custom-site-template.git
    hosts:
      - doggy-treats.test
```

Fichier de VVV à modifier pour créer des sites WordPress supplémentaires



The screenshot shows a terminal window with the following text:

```
gwen@Deadpool:~ $ cd vagrant-local/
gwen@Deadpool:~/vagrant-local (master)$ vagrant up

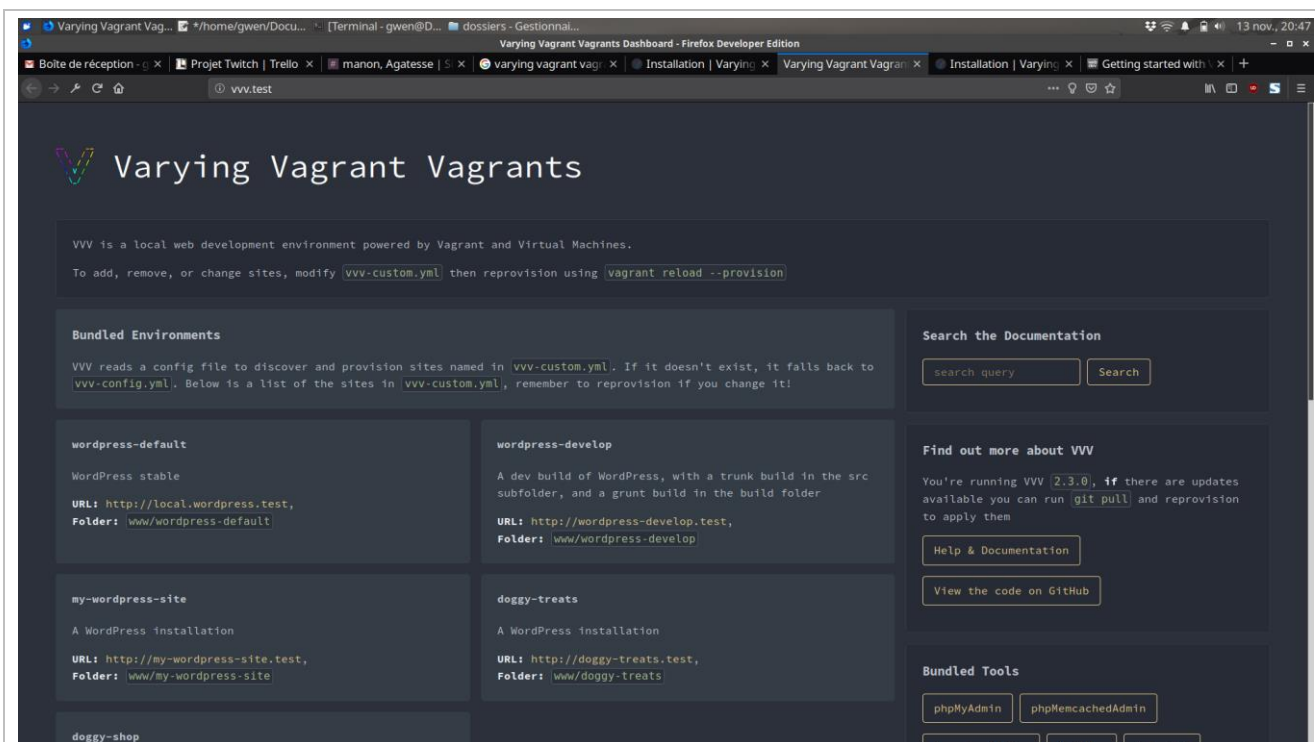
\ V V V / Varying Vagrant Vagrants
 \ ^ ^ ^ / v2.3.0-git::master

Platform: linux-gnu shell:/bin/bashsystemd CaseSensitiveFS
Vagrant: 2.1.5
VirtualBox: 5.2.18r124319

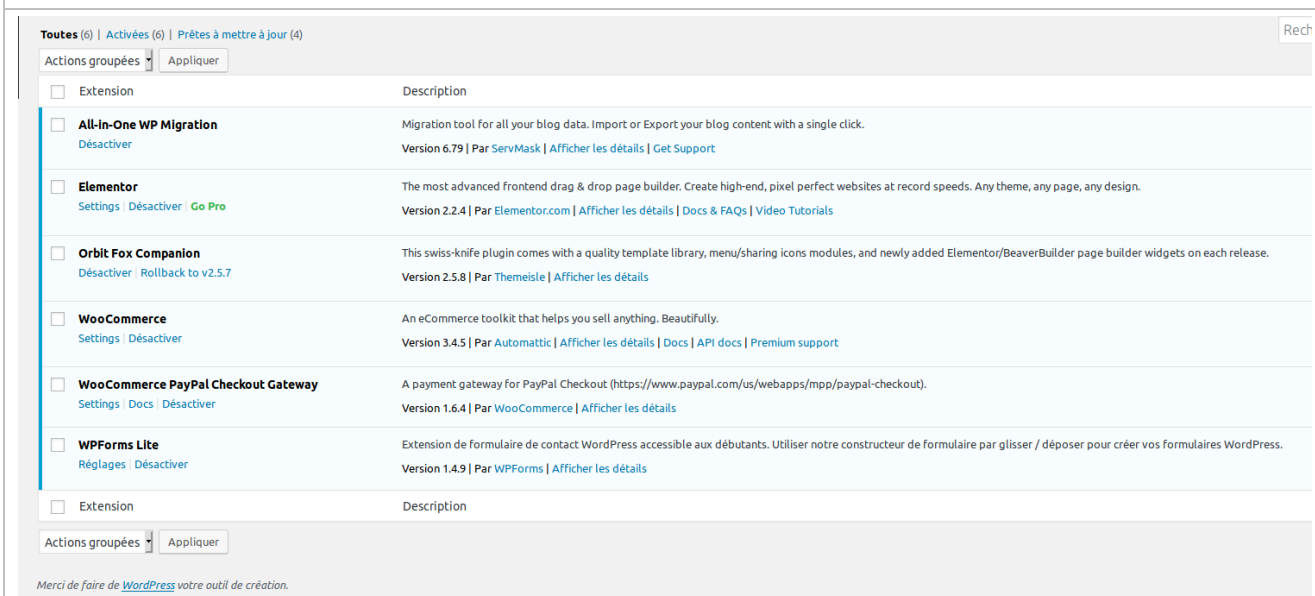
Docs: https://varyingvagrantvagrants.org/
Contribute: https://github.com/varying-vagrant-vagrants/vvv
Dashboard: http://vvv.test
```

La commande **vagrant up** permet lancer VVV

DOSSIER PROFESSIONNEL (DP)



La page d'accueil de VVV



Les extensions WordPress que j'ai utilisé

DOSSIER PROFESSIONNEL (DP)

Utilisateurs [Ajouter](#)

There are updates available for the following plugins: [Orbit Fox](#) and [WooCommerce](#).
[Begin updating plugins](#) | [Dismiss this notice](#)

Nouvel utilisateur créé. [Modifier l'utilisateur](#)

Tous (3) | Administrateur (1) | Customer (1) | Shop manager (1)

Actions groupées Appliquer Changer de rôle pour... Changer 3 éléments

<input type="checkbox"/>	Identifiant	Nom	Adresse de messagerie	Rôle	Articles
<input type="checkbox"/>	admin	Admin ADMIN	admin@local.test	Administrateur	0
<input type="checkbox"/>	customer1	Customer CUSTOMER	customer1@local.test	Customer	0
<input type="checkbox"/>	manager	Shop MANAGER	manager@local.test	Shop manager	0
<input type="checkbox"/>	Identifiant	Nom	Adresse de messagerie	Rôle	Articles

Actions groupées Appliquer Changer de rôle pour... Changer 3 éléments

Les utilisateurs que j'ai créés

phpMyAdmin

Recent Favorites

- New
- doggyshop
- New
- wp_commentmeta
- wp_comments
- wp_links
- wp_options
- wp_postmeta
- wp_posts
- wp_termmeta
- wp_terms
- wp_term_relationships
- wp_term_taxonomy
- wp_usermeta
- wp_users
- wp_wc_download_log
- wp_wc_webhooks
- wp_woocommerce_api_keys
- wp_woocommerce_attribute_taxonomies
- wp_woocommerce_downloadable_product...
- wp_woocommerce_log
- wp_woocommerce_order_itemmeta
- wp_woocommerce_order_items
- wp_woocommerce_payment_tokenmeta
- wp_woocommerce_payment_tokens
- wp_woocommerce_sessions
- wp_woocommerce_shipping_zones
- wp_woocommerce_shipping_zone_location
- wp_woocommerce_shipping_zone_method
- wp_woocommerce_tax_rates
- wp_woocommerce_tax_rate_locations
- doggytoys
- doggytreats
- information_schema
- mysql
- mywordpresstest
- performance_schema
- wordpressdefault
- wordpressdevelop
- wordpress_unit_tests

Server: localhost - Database: doggyshop - Table: wp_users

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)

SELECT * FROM wp_users

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Options

	ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status
<input type="checkbox"/>	1	admin	\$P\$BjKfKxjzVR9f0y5NAU0BH65gh7bqb.	admin	admin@local.test		2018-10-04 10:14:45		
<input type="checkbox"/>	2	manager	\$P\$B56N5dIEYrvX0KTH5zQ2cb8jdulZ0	manager	manager@local.test		2018-11-13 20:05:49	1542139552:\$P\$BnBr6uIOPhpZ3mZx5dZADH7mw2iwD.	
<input type="checkbox"/>	3	customer1	\$P\$B0WMOzt3MhiGICZJOjBAICZiQNm81	customer1	customer1@local.test		2018-11-13 20:08:50	1542139731:\$P\$B5j8ZcbFFZ6q54vA3F8BjZg5nvKEL1	

Check all With selected: Edit Copy Delete Export

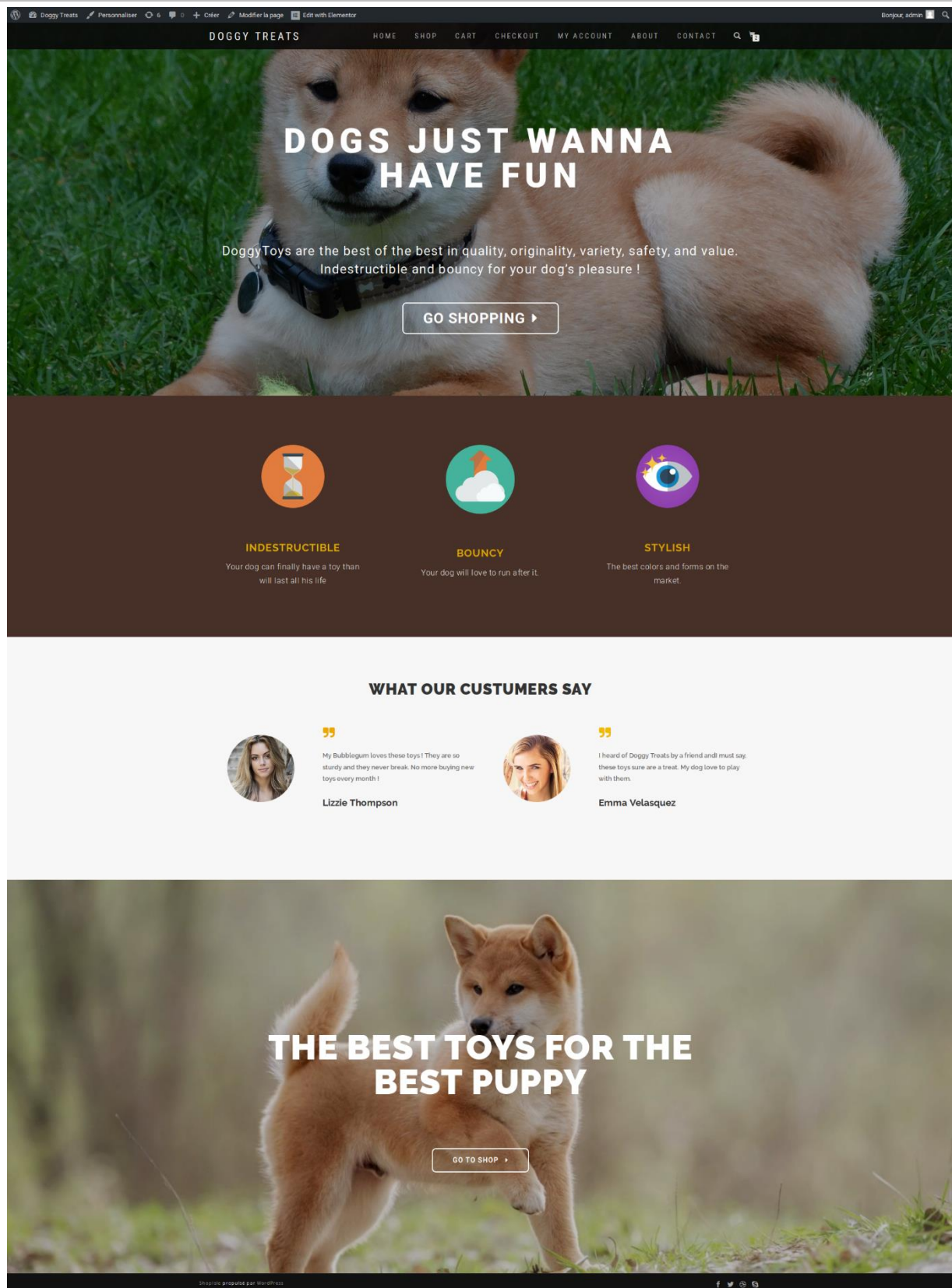
Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

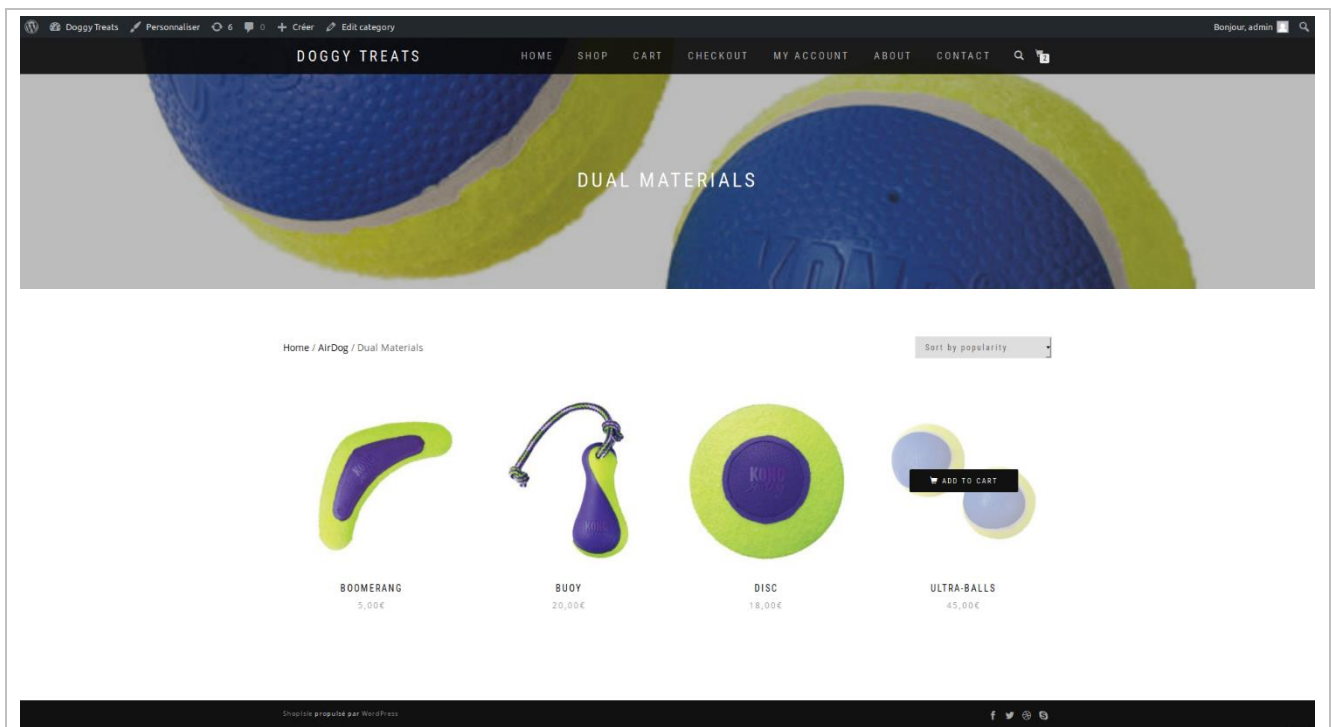
Les utilisateurs sont bien présents dans la base de données

DOSSIER PROFESSIONNEL (DP)

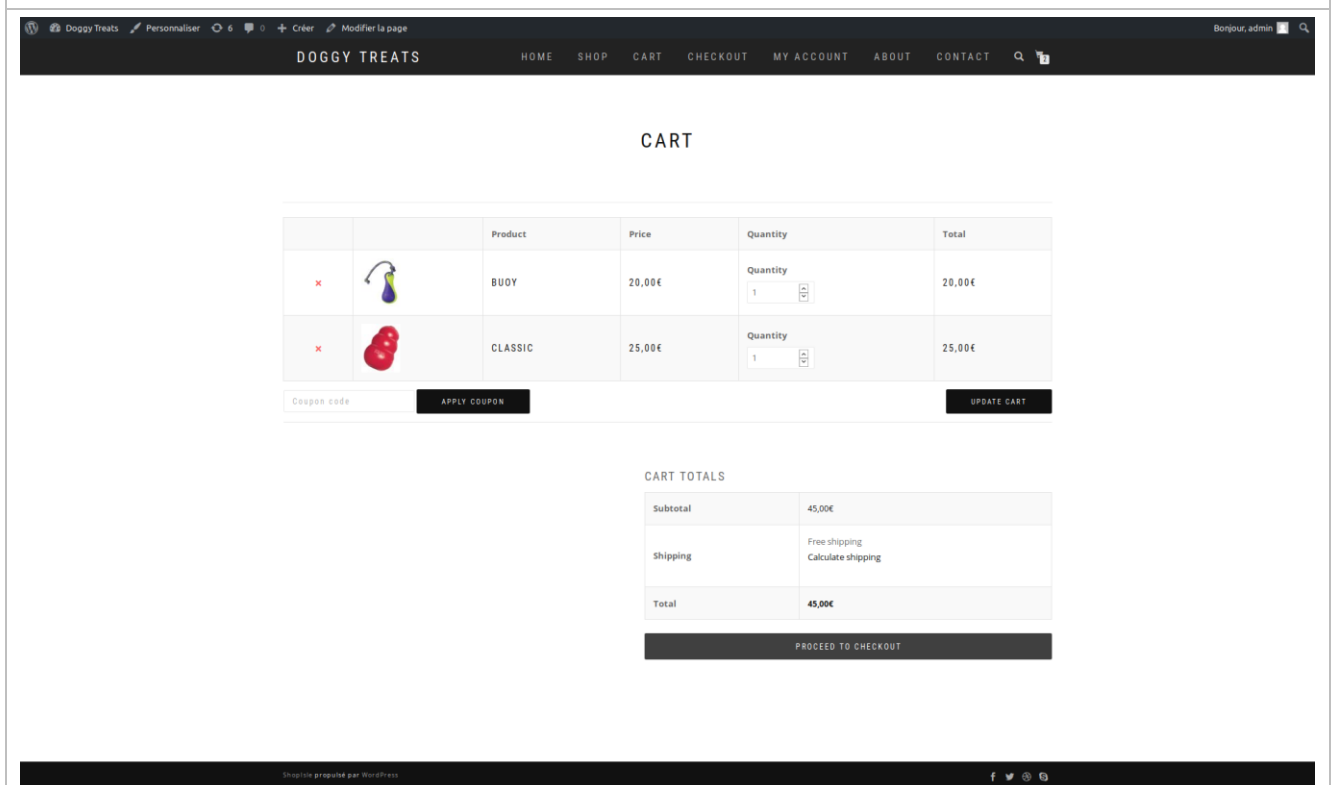


Page d'accueil de mon e-commerce

DOSSIER PROFESSIONNEL (DP)



Page d'une des catégories et ses articles



Page du panier

Documents relatifs à API JDBC



Structure de mon projet SpringBoot

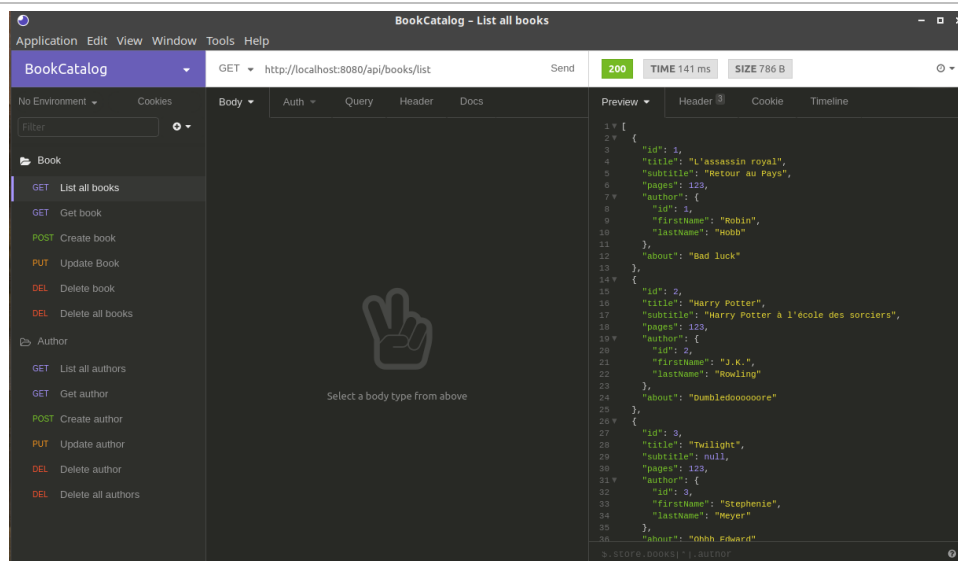
DOSSIER PROFESSIONNEL (DP)

```
1 DROP TABLE IF EXISTS author CASCADE ;
2 CREATE TABLE IF NOT EXISTS author (
3     author_id serial primary key,
4     first_name character varying(255) NOT NULL,
5     last_name character varying(255) NOT NULL
6 );
7
8 DROP TABLE IF EXISTS book;
9 CREATE TABLE IF NOT EXISTS book (
10    book_id serial primary key,
11    title character varying(255) NOT NULL,
12    subtitle character varying(255),
13    pages integer NOT NULL,
14    author_id bigint NOT NULL references author(author_id) ON DELETE CASCADE,
15    about text NOT NULL
16 );
```

Script de création de mes tables en base de données

```
1 INSERT INTO author(first_name,last_name)
2 VALUES
3     ('Robin','Hobb'),
4     ('J.K.','Rowling'),
5     ('Stephenie','Meyer'),
6     ('Georges','Martin');
7
8 INSERT INTO book(title,subtitle,pages, author_id, about)
9 VALUES
10    ('L'assassin royal','Retour au Pays',123, 1,'Bad luck'),
11    ('Harry Potter','Harry Potter à l'école des sorciers',123, 2,'Dumbledoooooooo'),
12    ('Twilight',null,123, 3,'Ohhh Edward'),
13    ('A song of ice and fire','Game of thrones',123, 4,'Kill them all'),
14    ('L'assassin royal','Le fou',123, 1,'Bad luck again');
```

Script d'insertion de mes données



Le logiciel Insomnia sur lequel j'ai testé mon API

DOSSIER PROFESSIONNEL (DP)

```
12  @Repository
13  @Transactional
14  public class BookRepositoryImpl implements BookRepository {
15
16      private final JdbcTemplate jdbcTemplate;
17
18      private final String INSERT = "INSERT INTO book(title, subtitle, pages, author_id, about) VALUES (?, ?, ?, ?, ?)";
19      private final String SELECT_ALL = "SELECT * FROM book INNER JOIN author ON book.author_id = author.author_id";
20      private final String SELECT_BY_ID = "SELECT * FROM book INNER JOIN author ON book.author_id = author.author_id WHERE book_id = ?";
21      private final String DELETE_BY_ID = "DELETE FROM book WHERE book_id = ?";
22      private final String DELETE_ALL = "DELETE FROM book";
23      private final String UPDATE_TITLE = "UPDATE book SET title = ?, subtitle = ?, pages = ?, author_id = ?, about = ? WHERE book_id = ?";
24
25      public BookRepositoryImpl(JdbcTemplate jdbcTemplate) { this.jdbcTemplate = jdbcTemplate; }
26
27
28
29      @Override
30      public List<Book> findAllBooks() { return this.jdbcTemplate.query(SELECT_ALL, new BookMapper()); }
31
32
33
34      @Override
35      public void createBook(Book book) {
36          this.jdbcTemplate.update(INSERT, book.getTitle(), book.getSubtitle(), book.getPages(), book.getAuthor(), book.getAbout());
37      }
38
39
40      @Override
41      public Book findBookById(Long id) {
42          return this.jdbcTemplate.queryForObject(SELECT_BY_ID, new Object[] {id}, new BookMapper());
43      }
44
45      @Override
46      public void deleteBookById(Long id) { this.jdbcTemplate.update(DELETE_BY_ID, id); }
47
48
49
50      @Override
51      public void deleteAllBooks() { this.jdbcTemplate.update(DELETE_ALL); }
52
53
54      @Override
55      public void updateBook(Long id, Book book) {
56          this.jdbcTemplate.update(UPDATE_TITLE, book.getTitle(), book.getSubtitle(), book.getPages(), book.getAuthor(), book.getAbout(), id);
57      }
58  }
59
60
```

Classe s'occupant des requêtes SQL via JDBC

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)