T8 - PHP Frameworks

T-WEB-600

E-Commerce

Using Symfony

Symfony

Projet réalisé par
Clément Lacroix, Martin Lepoutre, & Quentin Tridon.

TABLE DES MATIERES

EPIMARKET – LE PROJET 3

ORGANISATION & ARBORESCENCE 4

Client 4

Admin 4

MOODBOARDS 5

TECHNOLOGIES UTILISÉES 6

PHP symfony 6

ReactJS 6

DESCRIPTION BASE DE DONNÉES 7

MÉTHODE D'ORGANISATION 9

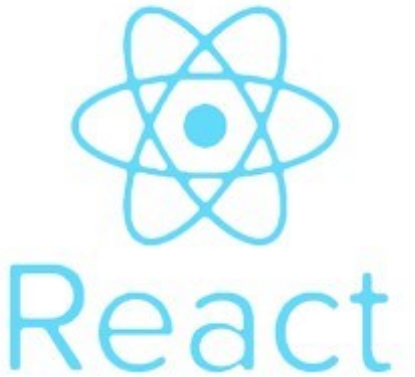
DIFFICULTÉS RENCONTRÉES 12

EPIMARKET – LE PROJET

Le projet consiste à créer un site marchand regroupant des offres discount sur des produits informatiques avec de nombreuses fonctionnalités.

Informarket est le site que nous avons mis en place. Nous y proposons la vente de barettes mémoires, cartes mères, cartes graphiques, etc…

Nous avons utilisé le **framework PHP symfony** côté serveur & **ReactJS** pour la partie front



Nous avons choisi la technologie **ReactJS** pour le front car les membres du groupe ont déjà utilisé ce langage pour de nombreux projets ou durant des missions professionnelles.

La technologie back étant imposé, nous avons donc réalisé ce dernier avec PHP symfony.

ORGANISATION & ARBORESCENCE

Notre site est divisé en 2 parties, une partie cliente, & une partie Admin. Nous allons aborder dans un premier temps la partie cliente puis nous verrons la partie Admin.

Client

Dans la partie Client, nous avons un header composé de 4 onglets :

- INFORMARKET → Cette page contient les « TOP SELLING ».
- PRODUCTS → Cette page contient une barre de recherche, et tous les produits mis en vente. Ainsi qu'un bouton catégorie, permettant de filtrer par composant spécifique.
- LOGIN → Cette page permet de se connecter en tant qu'utilisateur, mais aussi de créer son propre compte.
    - Une fois connecté, l'onglet Login devient l'onglet Profile, il est composé de 3 sous parties :
- Une page Profile ou l'on peut consulter ses informations utilisateur.
- Une page Orders ou on peut consulter ses commandes Un bouton Logout pour se déconnecter.
- CART → Cette page « CART » permet de commander les articles que l'on a ajouté à son panier. De regarder la quantité d'articles que l'on a commandé, le prix de ces derniers, etc… On ne peut commander que si on est « Log in »

Admin

Dans la partie admin nous disposons de 4 onglets qui sont les suivants :

- ORDERS → Cette page permet de suivre en temps & en heure les commandes passées par les utilisateurs
- USERS → Cette page permet d'administrer les utilisateurs, d'éditer leurs profiles & consulter leurs informations personnelles.
- PRODUCTS → Cette page permet de gérer les produits ainsi que les quantitées.
- PROFILE → Cet onglet permet de modifier les informations de l'utilisateur admin, et de se déconnecter.

MOODBOARDS

Pour le design nous avons utilisé l'outils WixLogo, qui nous a permis de créer un logo de manière simple & efficace.

Couleur 1	Couleur 2	Couleur 3
#EDEDED	#F35F40	#B2B2B1

Et la police que nous avons utilisée est la « Anton -400 ».

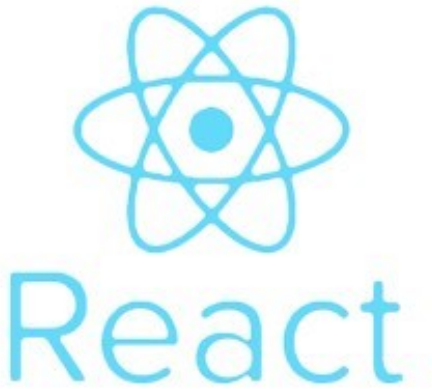
TECHNOLOGIES UTILISÉES

PHP symfony

Technologie imposée pour le projet, nous avons donc dû nous adapter et l'utiliser.

ReactJS

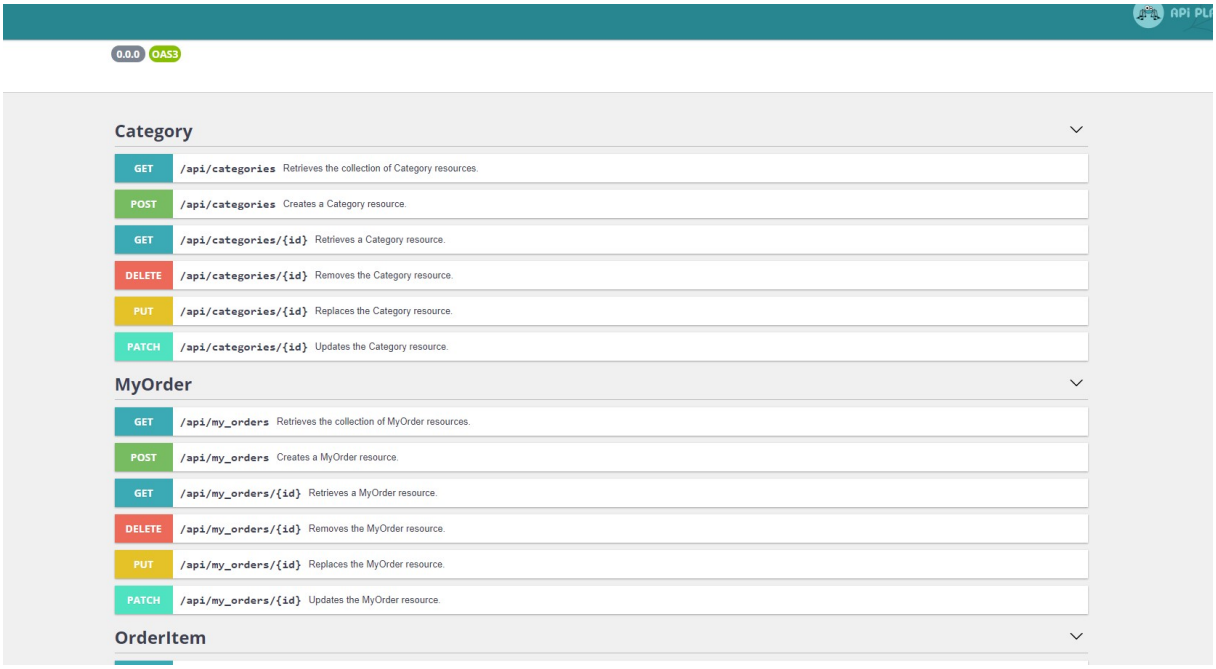
Technologie que nous avons choisis car nous maitrisions déjà le ReactJS, les délais étant courts nous avons préféré nous orienter vers un langage que nous connaissions déjà afin de gagner du temps et rendre un projet aux apparences les plus professionnelles possibles.



DESCRIPTION BASE DE DONNÉES

Routes avec API platform

Nous avons utilisé le package api platform pour générer des routes simples depuis notre base de données. (POST, GET, DELETE, PATCH...) Cet outil est très utile et nous a permis de gagner énormément de temps. Une doc complète de ces routes son disponibles à l'adresse 127.0.0.1 :8000/docs après avoir lancé le serveur local du backend. (Le port peut changer).



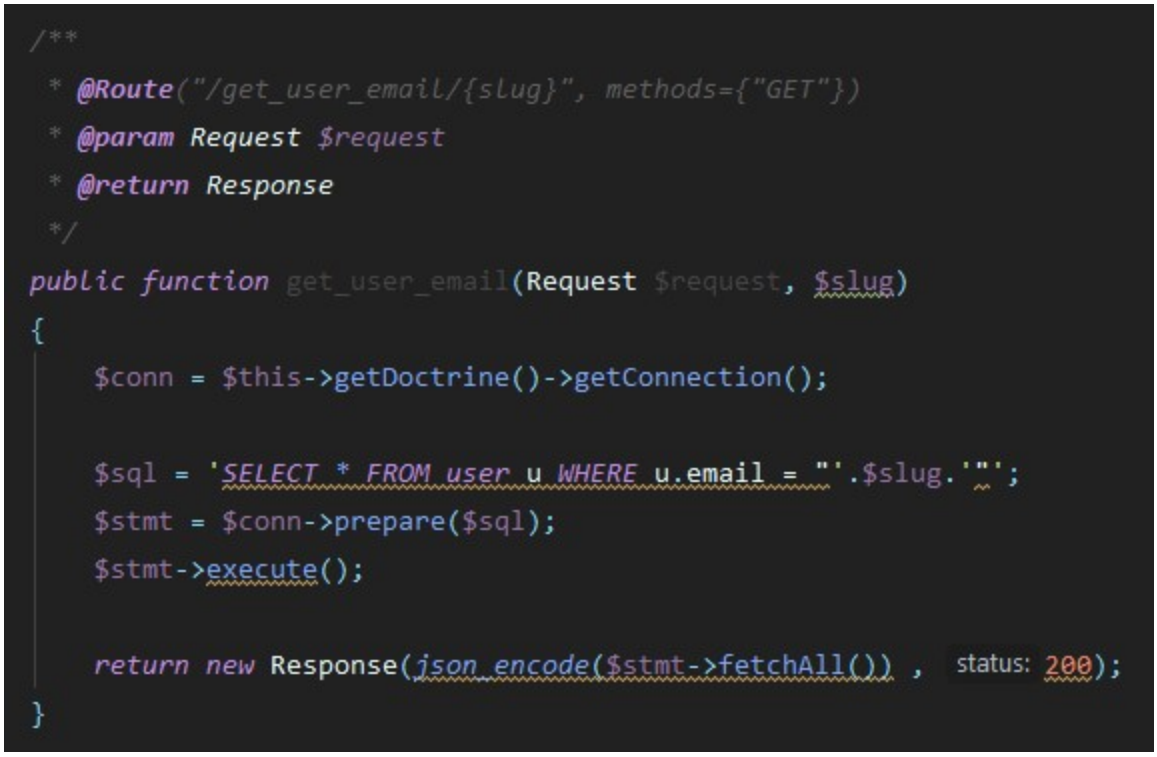
Les routes customisées.

Pour ce projet nous avons aussi dû créer des routes customisées sans l'outil API platform. Principalement pour la connexion, des requêtes avec jointure ou pour des requêtes avec le jwt.

Pour la connexion nous avons utilisé le package lexik_jwt_authentication qui nous a permis de créer la route de login qui prends en paramètre le username et le mot de passe en clair. Et nous avons fait une route de register à la main avec un encodage natif de symfony.

```
$new_user->setPassword($encoder->encodePassword($new_user, $password));
```

Les routes customisées sont créées dans le fichier ApiController. Chaque fonction de ce fichier est une route.



```php
/**
 * @Route("/get_user_email/{slug}", methods={"GET"})
 * @param Request $request
 * @return Response
 */
public function get_user_email(Request $request, $slug)
{
    $conn = $this->getDoctrine()->getConnection();

    $sql = 'SELECT * FROM user u WHERE u.email = "'.$slug.'"';
    $stmt = $conn->prepare($sql);
    $stmt->execute();

    return new Response(json_encode($stmt->fetchAll()) , status: 200);
}
```

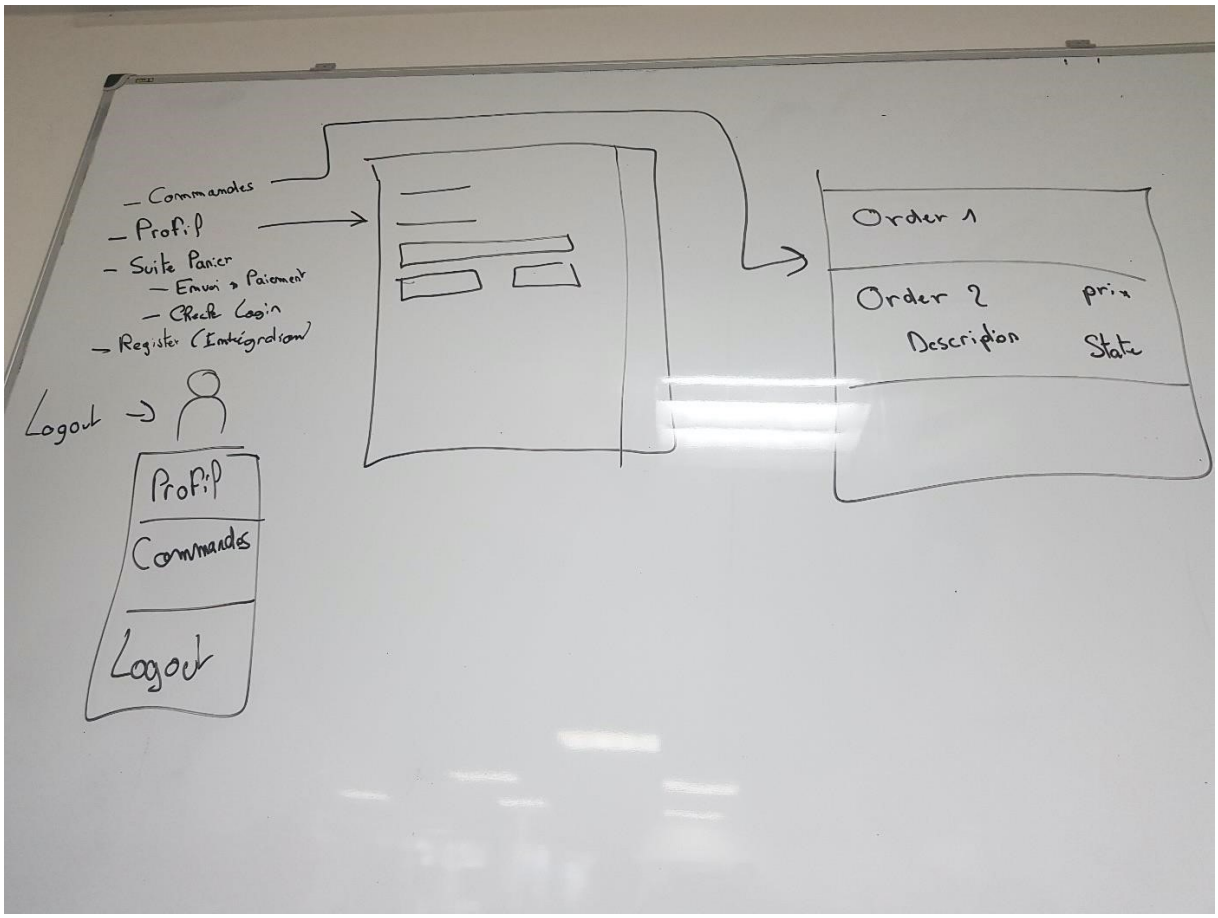
Par exemple la route get_user_email qui permet de récupérer les infos d'un utilisateur via son email. On appelle cette route via l'adresse 127.0.0.1 :3000/get_user_email/{email} email étant une variable. On voit bien que la requête SQL est construite avec le paramètre de la route. On retourne un objet Reponse avec le résultat de la requête.

MÉTHODE D'ORGANISATION

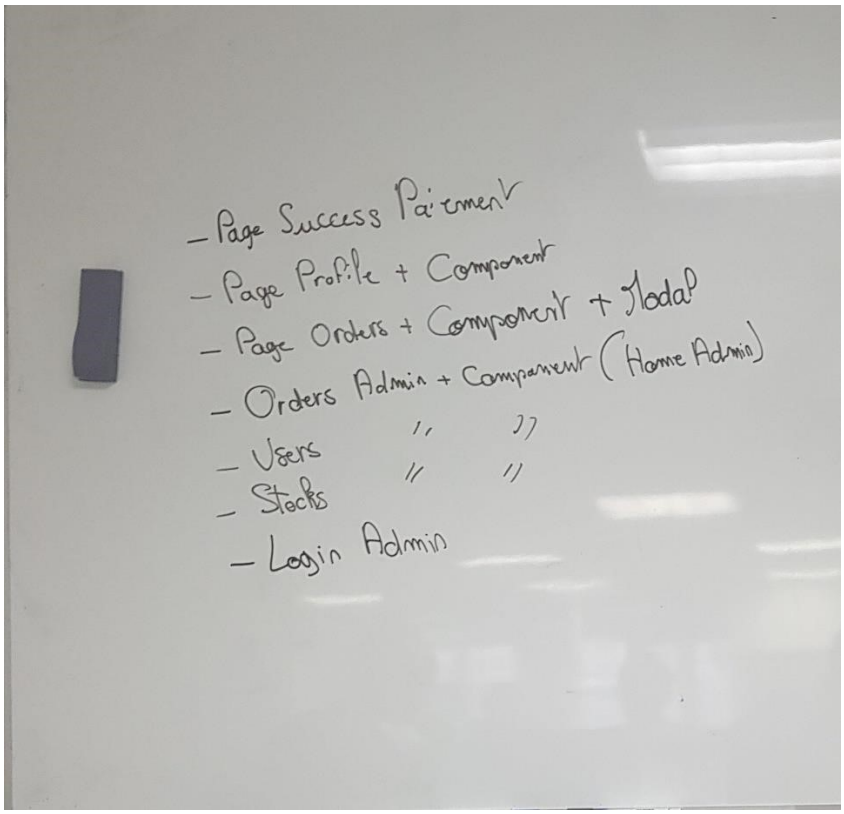
Nous avons utilisé l'outils Trello afin de s'organiser au mieux et de répartir les tâches.

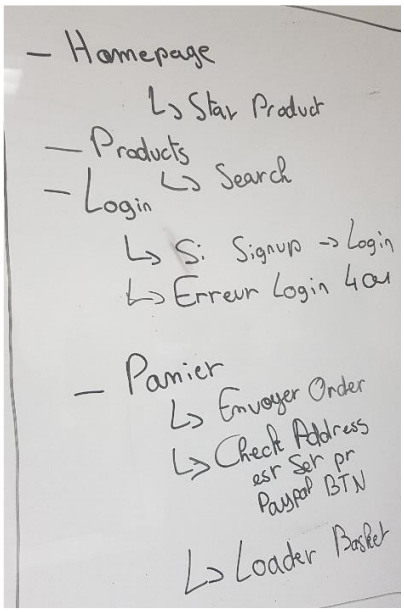
Nous avons réparti les tâches de la manière suivante :

- Martin s'est occupé de la base de données.
- Clément & Quentin du Front.

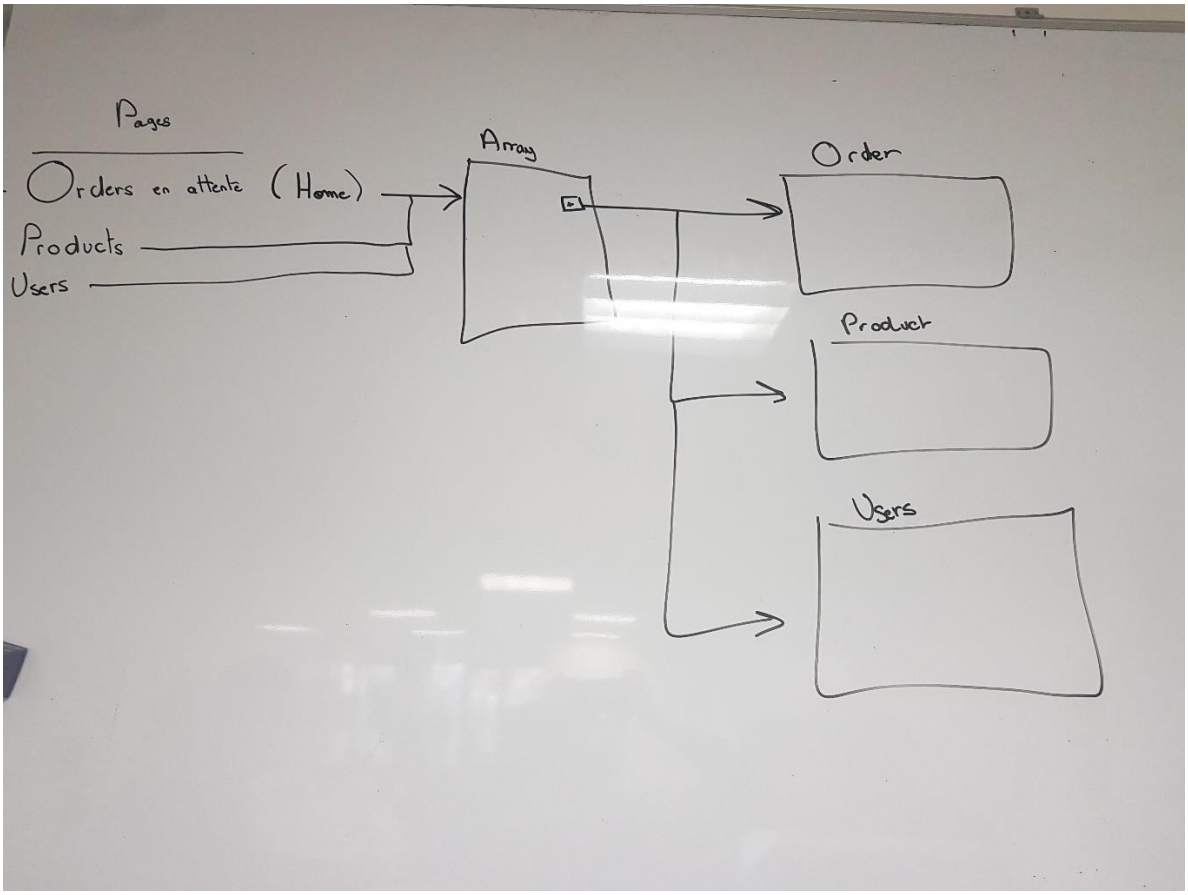


Pour ce qui est de l'architecture, quelques photos de nos idées posées sur tableaux. Afin d'organiser les idées, l'arborescence etc…





« Arborescence client »



« Création de la partie Admin et de ses composants. »

DIFFICULTÉS RENCONTRÉES

Délais très courts. 3 semaines pour réaliser un projet d'une telle envergure pour des personnes d'un tel niveau était un défi relativement dur.

Ce projet a nécessité plusieurs heures de travails en dehors de nos horaires de cours, notamment le week-end. La charge de travail étant relativement énorme, nous avons dû faire des sacrifices & des concessions.

Cependant, la répartition de la charge du travail & les connaissances des membres du groupe ont facilité la réalisation de ce dernier même s'il reste incomplet nous sommes fiers d'avoir aboutis à un tel résultat dans des délais ci restreints.