

Projet Web

Site de recettes - *Papilles*



Romain SIMON – 2021

Présentation du projet

Pour mon projet, j'ai décidé reprendre le site que j'avais réalisé lors du projet d'Architecture et Développement Web au semestre 3, et de l'adapter en version Symfony et VueJs.

Comme thème, j'avais choisi de faire un site web contenant des recettes de cuisine que j'ai appelé 'Papilles'. Pour m'aider j'ai utilisé un template HTML5 que j'ai trouvé sur le site freehtml5.co qui propose des templates gratuits et libres d'utilisation.

Sur le site, les recettes sont regroupées par type pour que l'utilisateur puisse trouver plus facilement ce qu'il recherche. Les différents types de recettes sont : entrées, plats et desserts. Il est aussi possible de se créer un compte et de se connecter au site. Il y aussi la possibilité de réserver des leçons de cuisine des recettes proposées.

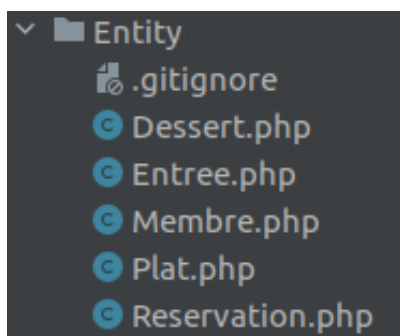
Les administrateurs du site peuvent créer, modifier ou supprimer des recettes et des membres sur une page spécifique qui n'est accessible que par eux.

Outils et méthodologies utilisés

Dans un premier temps, j'ai modifié mon site PHP en projet Symfony à l'aide du logiciel 'PHP Storm'. Pour réaliser cela j'ai eu besoin au fur et à mesure du projet de '*composer require*' plusieurs éléments :

- *twig/string-extra* : pour sa fonction '*u.truncate*' qui m'a permis de faire un plus bel affichage,
- *security* : pour la connexion d'un membre au site,
- *easycorp/easyadmin-bundle* : pour la création de la partie administrateur,
- *api* : pour la création de l'api que j'utilise pour la partie VueJS.

Mon projet utilise un total de 5 entités que voici :



J'ai fait le choix de créer une entité par catégorie de recettes pour faciliter leur affichage et leur gestion. Ainsi qu'une entité pour stocker les membres et une entité pour les réservations. On peut retrouver plusieurs relations entre les entités, par exemple il y a une relation ManyToMany entre une réservation et une entrée.

Sans compter les Controller créés automatiquement par les bundles EasyAdmin et Security, je n'utilise qu'un seul Controller qui gère le routage et les données à fournir à chacune de mes pages.

De même, sans compter les Form créés automatiquement, j'en utilise 2, un pour la création de membre et un pour les réservations.

La base de données et une base de données locale en MYSQL.

Tout le code CSS et plusieurs scripts JavaScript étaient fournis dans le template que j'ai utilisé. Je n'y ai effectué que très peu de modifications.

C'est tout pour le projet Symfony, quant au projet VueJS comme précisé plus tôt j'ai créé une API contenant toutes les données de mon site via Symfony. J'utilise les données de cette API pour afficher les données sur mon site VueJS. Je me connecte à l'api grâce à Axios.

Pour la partie VueJS, j'ai utilisé **vue cli** pour créer mon projet et j'ai programmé avec l'IDE **Webstorm**.

```
import axios from 'axios'

export default() => {
  return axios.create({
    baseURL: 'http://127.0.0.1:8000/api/'
  })
}
```

Ensuite je n'ai plus qu'à, dans le composant où j'ai besoin des données, utiliser la fonction *get* de Axios pour récupérer les données de l'api ; comme ceci :

```
mounted() {
  Axios().get( url: 'desserts').then(response => this.desserts = response.data['hydra:member'])
  Axios().get( url: 'entrees').then(response => this.entrees = response.data['hydra:member'])
  Axios().get( url: 'plats').then(response => this.plats = response.data['hydra:member'])
},
```

Pour ce qui est de la structure, j'ai placé le header de ma page web dans le fichier `App.vue`, comme cela il sera présent en haut de toutes mes pages de mon site.

```
<template>
  <div id="fh5co-page">
    <section id="fh5co-header">
      <div class="container">
        <nav role="navigation">
          <ul class="pull-left left-menu">
            <li><router-link to="/entree">Entrees</router-link></li>
            <li><router-link to="/plat">Plats</router-link></li>
            <li><router-link to="/dessert">Desserts</router-link></li>
            <li><router-link to="/reservation">Reservation</router-link></li>
          </ul>
          <h1 id="fh5co-logo"><router-link to="/">Papilles<span>.</span></router-link></h1>
          <ul class="pull-right right-menu">
            <li><router-link to="/login">S'identifier</router-link></li>
            <li class="fh5co-cta-btn"><router-link to="/signin">Créer un compte</router-link></li>
          </ul>
        </nav>
      </div>
    </section>
    <router-view/>
  </div>
</template>

<script>
</script>
```

Comme on peut le voir sur la capture d'écran, j'utilise `<router-link>` pour gérer le routage de mes différentes pages web. Toute la configuration des routes se trouvent dans le dossier *router*, dans le fichier *index.js*. Aussi, j'utilise le même code CSS que dans mon projet Symfony. Code que j'importe dans le fichier *main.js*.



Pour que mon site s'affiche correctement et de manière similaire à celui de mon projet Symfony j'utilise 11 composants.

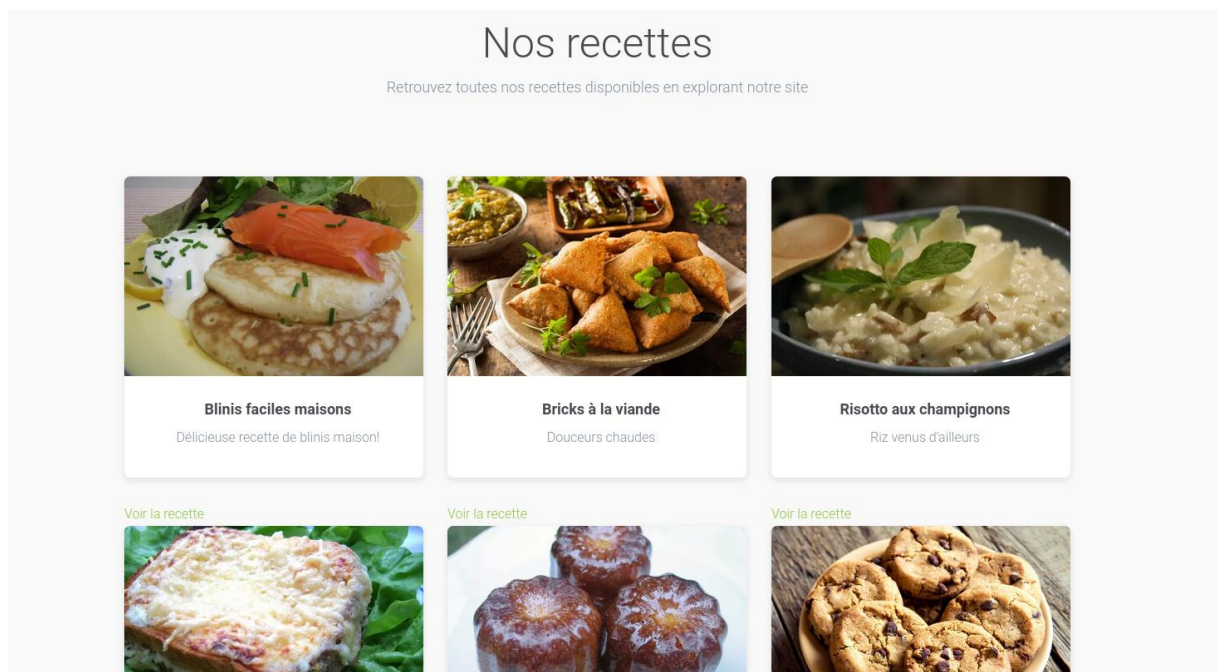
Il faut savoir que mon site avec VueJS est moins complet que mon site avec Symfony, comme l'objectif du site VueJS était d'afficher dynamiquement les données récupérées via l'api Symfony. Par exemple, sur mon site VueJS il n'y a pas de partie administrateur. Cependant, toutes les données du site s'affichent correctement.

Pour compenser l'absence du '*composer require twig/string-extra*' sur VueJS, j'ai utilisé un filter **truncate** dont j'ai trouvé le code sur internet, qui me permet de réduire la taille d'un *string* lors de son affichage.

```
filters: {  
  truncate: function (text, length, suffix) {  
    if (text.length > length) {  
      return text.substring(0, length) + suffix;  
    } else {  
      return text;  
    }  
  },  
}
```

Résultats obtenus

Pour ce qui est des pages principales du site (*Accueil, Entrées, Plats, Desserts, Réservation*) l’affichage est quasiment identique que ce soit pour le site avec Symfony ou le site avec VueJS. Une partie de la page d’accueil de mon site est déjà visible sur ce rapport à la 1ère page ; si on descend dans la page d’accueil on peut trouver 6 recettes (2 de chaque catégorie) avec comment information leur nom et une courte description.



Il est possible de naviguer entre les pages en cliquant sur les mots-clés (*Entrées, Plats, etc....*) en haut de chaque page. Et pour retourner sur la page d'accueil il suffit de cliquer sur le nom du site.

Lorsqu'un utilisateur clique sur *Desserts* par exemple, il est redirigé vers une page qui affichera toutes les recettes de desserts présentes dans la base de données, triée par leur ID. Il en va de même pour les autres catégories de recette.



Gâteau au chocolat fondant

Recette rapide d'un gâteau

1/ Préchauffez votre four à 180°C (thermostat 6). Dans une casserole, faites fondre le chocolat et le beurre coupé en morceaux à feu très doux. ...

[Voir la recette](#)



Crumble aux pommes

Dessert fruité et croquant!

1/ Préchauffer le four à 210°C (thermostat 7). 2/ Peler, évider et découper les pommes en cubes grossiers, les répartir dans un plat allant au ...

[Voir la recette](#)



Crepes

Spécialité française pour petits et grands

1/ Mettez la farine dans un saladier avec le sel et le sucre. 2/ Faites un puits au milieu et versez-y les œufs. 3/ Commencez à mélanger doucement...

[Voir la recette](#)



C'est à ce moment là qu'interviennent les filtres *truncate* pour n'afficher que les 150 premiers caractères de chaque recette. Pour accéder à la recette complète il suffit à l'utilisateur de cliquer sur *Voir la recette*, il sera alors redirigé vers une page qui contient toutes les informations de la base de données concernant la recette demandée.

Recette de Gâteau au chocolat fondant

Temps de préparation : 10 min et temps de cuisson : 30 min



Ingrédients : - 200g de chocolat noir - 100g de beurre - 3 oeufs - 50g de farine - 100g de sucre

Recette :

1/ Préchauffez votre four à 180°C (thermostat 6). Dans une casserole, faites fondre le chocolat et le beurre coupé en morceaux à feu très doux.

2/ Dans un saladier, ajoutez le sucre, les oeufs, la farine. Mélangez.

3/ Ajoutez le mélange chocolaubeurre. Mélangez bien.

4/ Beurrez et farinez votre moule puis y versez la pâte à gâteau.

5/ Faites cuire au four environ 20 minutes.

Si l'utilisateur veut effectuer une réservation, ce qui est possible uniquement avec le site Symfony, il doit choisir un des *packages* proposés.

The image shows four package cards for cooking classes. The 'DEMI-JOURNÉE' card is highlighted with a green border and a 'Prix réduit' tag. Each card has a title, a price per person, a description, a list of items, and a 'Choisir' button.

Package	Price per person	Description	Items
1 RECETTE	€12	Apprenez une recette	1 plat de votre choix Ingrédients & ustensiles fournis
REPAS COMPLET	€35	Apprenez à cuisiner un repas complet de votre choix	1 entrée 1 plat 1 dessert Ingrédients & ustensiles fournis
DEMI-JOURNÉE	€65	Passez la demi-journée à cuisiner avec l'un de nos chefs, qui cuisinera pour vous votre dîner	Repas cuisiné par notre chef 3 recettes de votre choix à apprendre Ingrédients & ustensiles fournis
JOURNÉE COMPLÈTE	€150	Passez la journée à cuisiner avec l'un de nos chefs, qui cuisinera pour vous votre déjeuner et votre dîner	2 repas cuisinés par notre chef 6 recettes de votre choix à apprendre Dégustation de vins en fin de journée Ingrédients et ustensiles fournis

Lorsqu'il clique sur le bouton *Choisir*, s'il n'est pas connecté il est redirigé vers la page de Connexion, sinon il peut effectuer sa réservation.

The image shows a screenshot of the 'Papilles' reservation form. The background is a dark wood texture with tomatoes and basil. The form is overlaid on the right side. It includes a navigation bar at the top with links: Entrées, Plats, Desserts, and Réservez. Below the navigation bar, the user is greeted with 'Bonjour SIMON Romain' and a 'Déconnexion' link. The form has a date and time selector, a dropdown menu for selecting items, and an 'Envoyer' button.

Navigation: Entrées, Plats, Desserts, Réservez

Header: Papilles

User: Bonjour SIMON Romain, Déconnexion

Date: [Date selector]

Heure: [Time selector]

Personne: [Person selector]

Entrées:

- Salade facile maison
- Velouté de champignons
- Millefeuille de la mer
- Foie gras maison

Plats:

- Lasagnes à la bolognaise
- Boeuf bourguignon rapide
- Noix de Saint-Jacques
- Hachis Parmentier

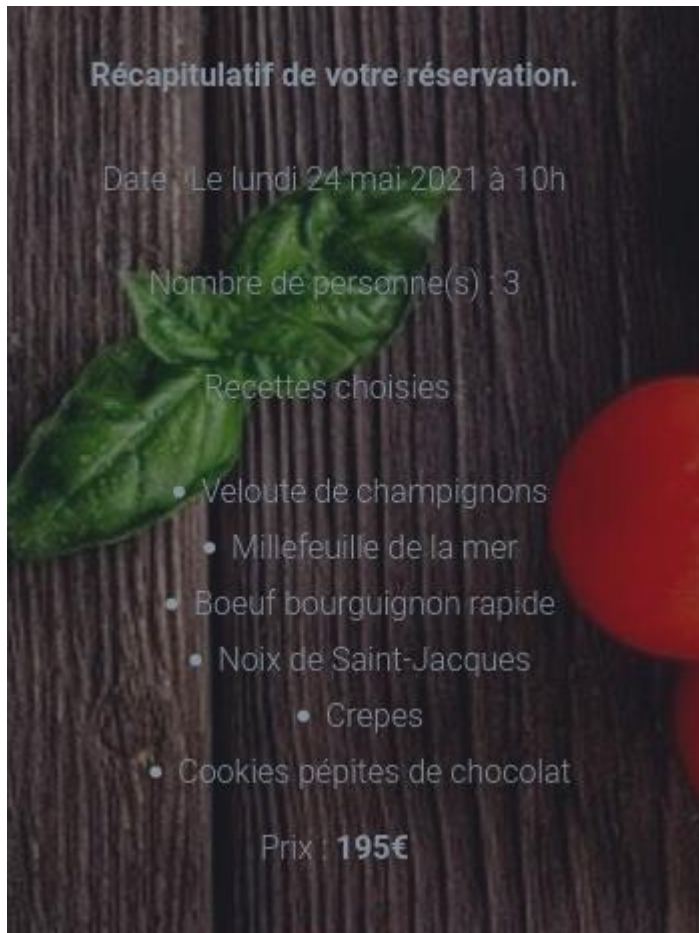
Desserts:

- Gâteau au chocolat fondant
- Crumble aux pommes
- Crêpes
- Cookies pépites de chocolat

Envoyer

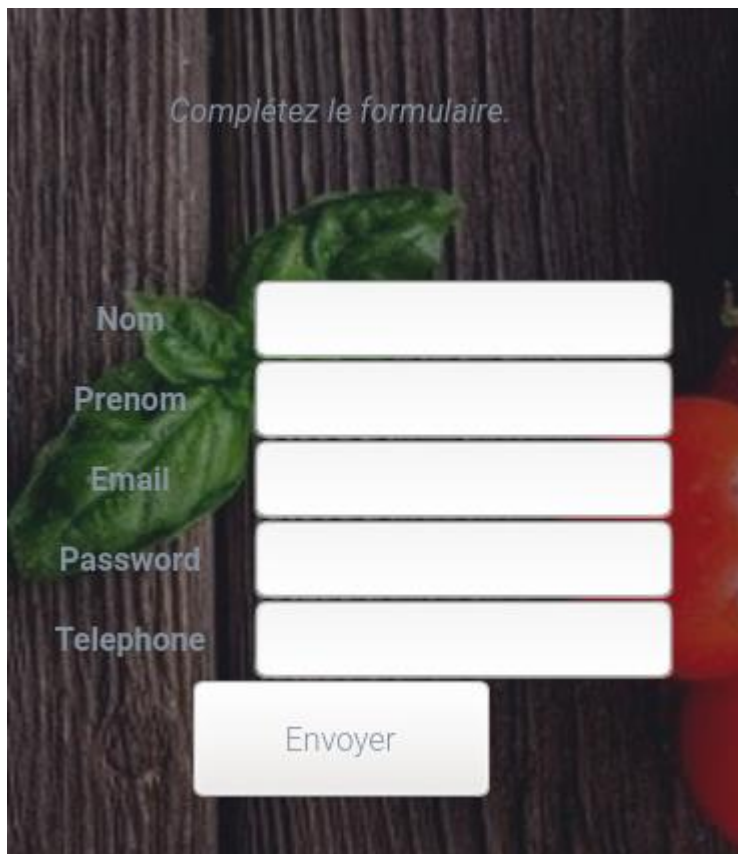
On peut voir que l'utilisateur est connecté car son nom est affiché en haut à droite de l'écran, et il a aussi la possibilité de se déconnecter.

L'utilisateur doit donc remplir/choisir les éléments nécessaire pour sa réservation dont une date qui est stockée dans l'entité Réservation. Une fois que tout est remplie et que l'utilisateur valide, il est renvoyé vers une page avec un récapitulatif de sa réservation.



On peut voir, comme il a été demandé dans le sujet, que la date est affichée en français.

Si l'utilisateur n'est pas connecté au site, il peut se créer un compte via un formulaire relativement classique et simple.

The image shows a registration form overlaid on a background of a wooden surface with fresh basil leaves and a tomato. The form is titled "Complétez le formulaire." in a light gray font. It contains five input fields, each preceded by a label: "Nom", "Prenom", "Email", "Password", and "Telephone". The "Password" field has a red underline. Below the input fields is a white button with the text "Envoyer" in gray.

La création d'un compte et la connexion sont disponibles sur le site Symfony et sur le site VueJS, à la différence que sur le site Symfony les mots de passes vont être encodés. Donc il n'est possible de se connecter au site qu'avec des comptes créés sur leurs sites respectifs.

Pour terminer la présentation, nous allons voir la partie Administrateur du site qui n'est disponible qu'avec le projet Symfony. Pour y accéder il y a un petit *Admin* cliquable dans le footer de chaque page. Une fois cliquée, l'utilisateur est redirigé vers la page de connexion (s'il n'est pas déjà connecté) et si le compte utilisé n'a pas le rôle *admin* alors une page avec marquée 'Access Denied' apparait. Si l'utilisateur connecté est un admin il peut alors accéder à la partie Admin qui est un *CRUD* crée grâce au bundle EasyAdmin.

Designation	Description	Ingrédients	Recette	Tempspreparation	Tempscuisson	Image	
Blinis faciles maisons	Délicieuse recette de blinis maison!	- 1 yaourt blanc - 1 oeuf - 1 pot de yaourt de farine - 1/2 sac...	1/ Mélanger tous les ingrédients ensemble 2/ Puis laisser...	5	20		Éditer Supprimer
Velouté de champignons	Velouté onctueux qui réchauffe	- 3 cuillères à soupe de beurre - 1 oignon - 250g de champignon...	1/ Faire fondre dans une casserole 3 cuillères à soupe de beur...	10	40		Éditer Supprimer
Millefeuille de la mer	Délicieuse et vite préparée!	- 2 boîtes de crabe - 8 tranches de saumon fumé - 4 petites éch...	1/ Egouttez le crabe. Ouvrez une boîte vide des deux côtés afin...	20	0		Éditer Supprimer
Foie gras maison	Rapide et facile!	- 600g d'un foie gras d'oie - Sel - Poivre	1/ Laissez-le foie 1 heure à température ambiante, puis écarterz...	40	15		Éditer Supprimer
Cake jambon olive	Un cake salé parfait!	- 250g de farine - 250g de jambon fumé - 150g de gruyère rapé - ...	1/ Préchauffez le four à 180°C (thermostat 6). Coupez le jambo...	30	45		Éditer Supprimer
Carpaccio de Saint-Jacques	Unique et savoureux!	- 4 belles Saint-Jacques sans le corail, très fraîches - 1 mang...	1/ Commencer par bien rincer les saint-jacques et les essuyer d...	15	10		Éditer Supprimer
Bricks à la viande	Douceurs chaudes	- 200g de viande hachée - Menthe en feuille - 1 grosse pomme de...	1/ Hacher l'oignon, le faire revenir pendant 5 minutes dans une...	25	10		Éditer Supprimer

7 résultats

« Précédent 1 Suivant »

Ici on voit la gestion des Entrées, mais c'est aussi possible d'ajouter, modifier ou supprimer des Plats, Desserts ou Membre. Une fois que l'administrateur a terminé d'administrer il lui suffit de se déconnecter avec le bouton *Logout* et il sera redirigé vers la page d'accueil du site.

Conclusion

J'ai bien aimé pouvoir reprendre le site que j'avais créé pour le PHP car cela m'a permis de vraiment voir plusieurs manières différentes de créer un site internet. Que ce soit en PHP 'page-par-page', en PHP avec un modèle MVC, avec Symfony ou avec VueJS.

Pour moi, la méthode la plus simple et pratique à utiliser est Symfony notamment pour la base de données intégrée et les entités pour la gérer, ainsi que la possibilité d'installer des bundles qui facilitent la création et le paramétrage du site.

Tandis que, pour moi, le site le plus compliqué à faire était celui en VueJS, j'ai moins bien compris ce cours/module et c'est pourquoi sur mon site VueJS il y a moins de fonctionnalités que sur mon site Symfony.

Il y a quelques fonctionnalités auxquelles je n'ai pensé que trop tard, ou que je n'ai pas réussi à implémenter, que j'aurais bien aimé ajouter à mon site. Comme par exemple, la possibilité pour un membre connecté d'ajouter une recette à ses favoris pour que ce soit plus facile pour lui de la retrouver, ou encore lors de la création d'une réservation de limiter le nombre de recettes qu'un utilisateur peut choisir par catégorie.

Pour finir, je suis content du travail que j'ai réalisé et mon site correspond bien à l'idée que j'avais en tête lorsque je l'ai créé, même si je n'ai pas réussi à implémenter tout ce que je voulais, je trouve que le résultat est plaisant visuellement et fonctionnel.