

Prise en main de Symfony 5.4 Les formulaires

Navires Partie 2

Objectifs :

Développer une petite application qui va gérer un les navires.
Versionner le projet
Créer des contrôleurs
Créer et valider des formulaires

Environnement :

- ✓ Serveur apache : celui de wamp
- ✓ Base de données : mysql en local (wamp)
- ✓ IDE : netbeans
- ✓ virtualHost : navire.sio
- ✓ PHP 8.1
- ✓ Symfony 5.4
- ✓ Mysql 8.

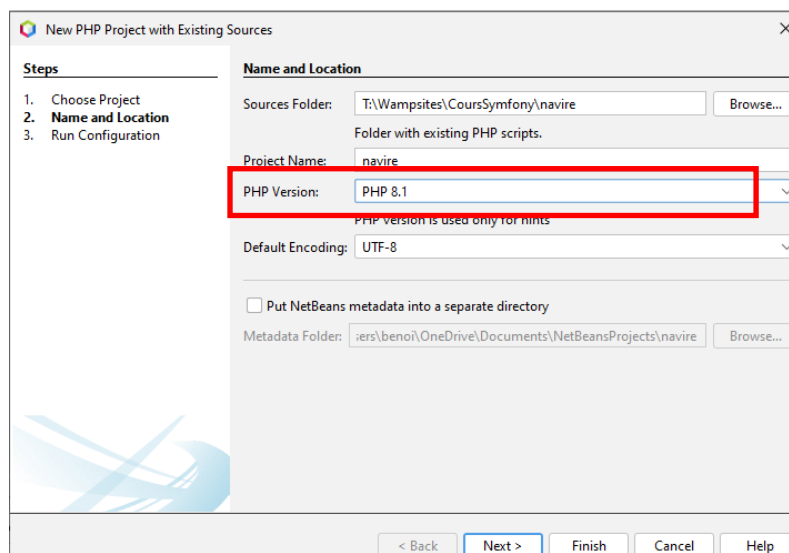
Partie 1 : MISE EN PLACE DU PROJET

Vous allez créer un nouveau projet appelé navire à l'aide de composer.

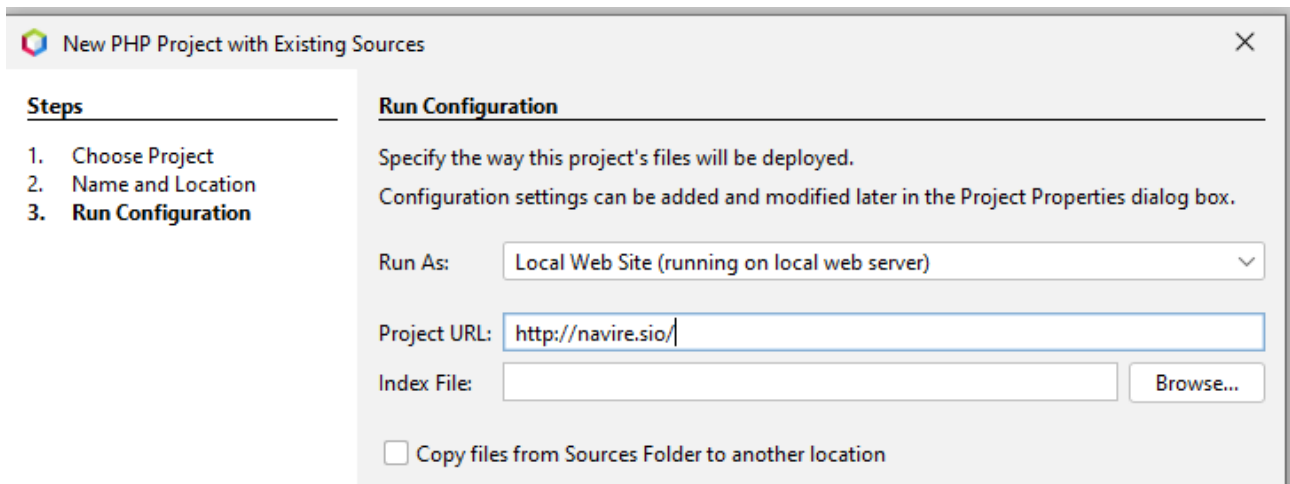
- ✓ Ouvrez une fenêtre de terminal
- ✓ Positionnez-vous sur le dossier parent du dossier contenant l'application
- ✓ Exécutez la commande :

```
symfony> symfony new navire --version=5.4
```

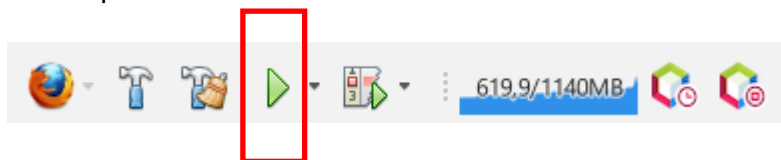
- ✓ Créez le virtualhost navire.sio qui va pointer sur le dossier navire/public
- ✓ Redémarrez votre DNS local
- ✓ Testez l'url <http://navire.sio/>



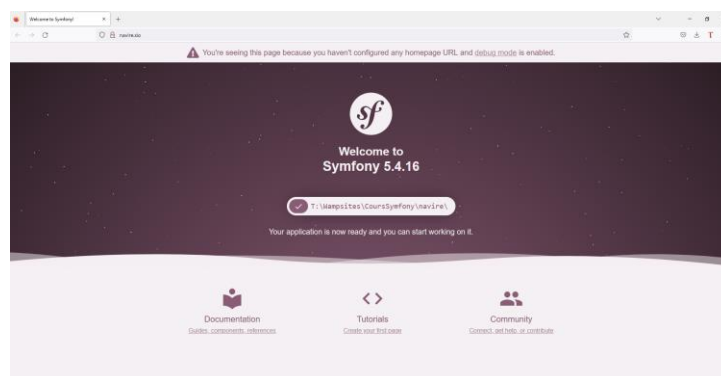
Vous allez maintenant créer le projet *with existing sources* dans netbeans :



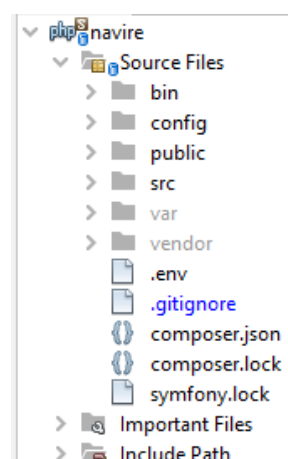
Vous pouvez tester l'url depuis netbeans :



Vous devriez arriver sur le projet vierge :



Et dans netbeans :



Vous installerez les packages suivants :



Rappel : on installe un package par la commande suivante à la racine du projet

Symfony composer require nomDuPackage

```
call symfony composer require annotations
call symfony composer require maker --dev
call symfony composer require symfony/apache-pack --dev
call symfony composer require twig
call symfony composer require logger
call symfony composer require profiler --dev
call symfony composer require debug --dev
call symfony composer require symfony/form
call symfony composer require symfony/validator
call symfony composer require symfony/orm-pack
call symfony composer require symfony/mime
call symfony composer require symfony/asset
call symfony composer require symfony/mailer
call symfony composer require sensio/framework-extra-bundle
call symfony composer require symfony/security-bundle
# call composer require symfony/webpack-encore-bundle
```

Vérifier les pré-requis de configuration et adaptez votre configuration si nécessaire :

symfony check:requirements

```
PS T:\Wampsites\CoursSymfony\navire> symfony check:requirements
```

```
Symfony Requirements Checker
```

```
~~~~~
```

```
> PHP is using the following php.ini file:
```

```
C:\wamp64\bin\php\php8.1.0\php.ini
```

```
> Checking Symfony requirements:
```

```
.....WW.....
```

```
[OK]
```

```
Your system is ready to run Symfony projects
```

```
PS T:\Wampsites\CoursSymfony\navire> symfony check:requirements
```

Vous allez maintenant réaliser votre premier commit et votre premier push sur un repository distant.

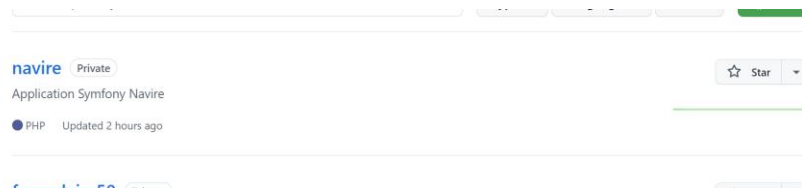
Vérifiez le fichier .gitignore pour ne pas versionner le dossier nbproject :

```
###> symfony/framework-bundle ###
/.env.local
/.env.local.php
/.env.*.local
/config/secrets/prod/prod.decrypt.private.php
/public/bundles/
/var/
/vendor/
###< symfony/framework-bundle ###
/nbproject/
```



Vous utiliserez votre outil préféré pour réaliser cette opération.
Vous utiliserez votre plateforme distante préférée pour pousser votre code

Et vous devriez avoir un repository sur github ou gitlab :



Mon repository est privé, le vôtre sera **public**.

Partie 2 : L'INTERFACE GRAPHIQUE

Vous prévoyez d'utiliser bootstrap 4 ou 5



L'utilisation du webpack encore serait un plus !!!

L'interface devra présenter :

- ✓ Une barre de navigation dans la partie supérieure de l'interface
- ✓ Un corps où seront affichées les informations demandées
- ✓ Un pied de page où vous afficherez les informations légales, ...



Le reste de l'interface est à votre convenance. Vous pourrez utiliser un thème donné ou votre propre thème.

Le problème n'est pas de créer une interface graphique sophistiquée, il est de rendre l'application fonctionnelle. Vous ne consacrerez donc l'essentiel de votre temps sur l'aspect fonctionnel de l'application

- ✓ La barre de navigation devra comporter une photo en haut à droite,
- ✓ Un formulaire de recherche
- ✓ Vous prévoyez un block *footer* dans le fichier base.html.twig pour le pied de page
- ✓ Un icône dans l'onglet du navigateur

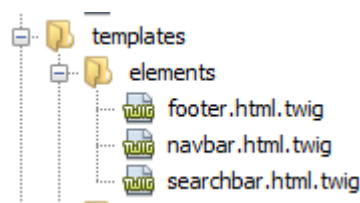


Vous installerez les ressources nécessaires à bootstrap dans les bons dossiers

Les liens figureront dans le fichier base.html.twig

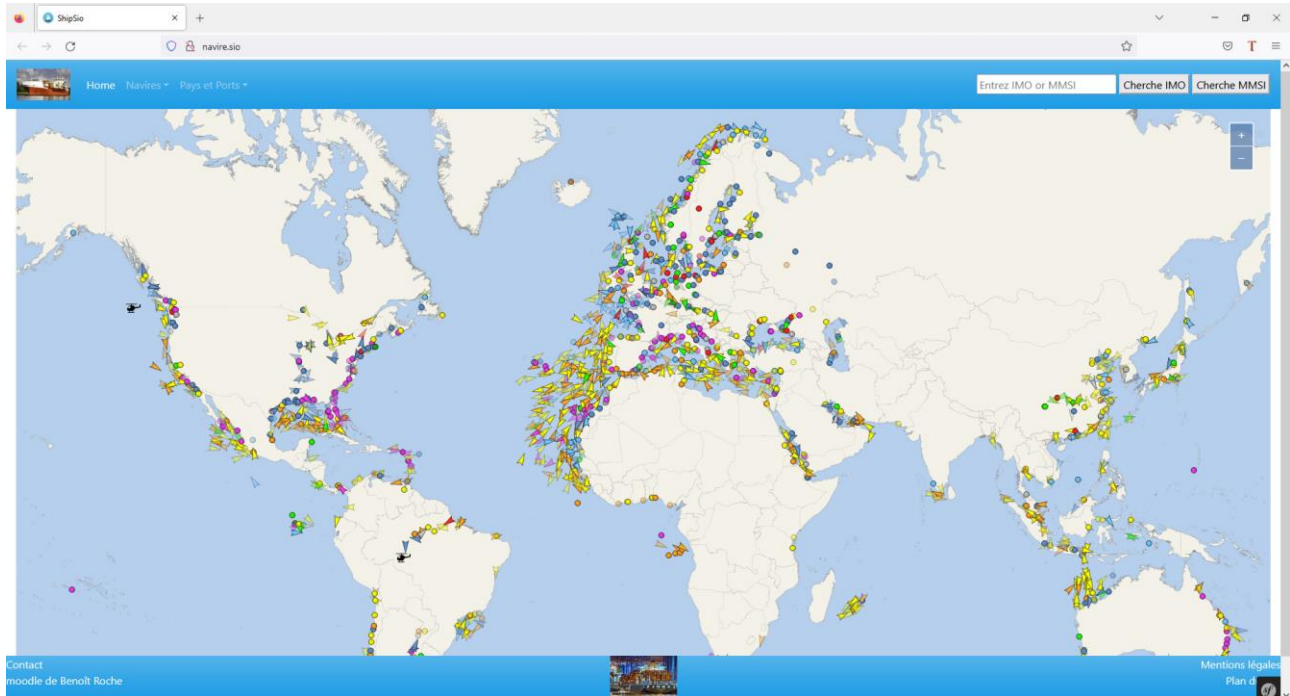
Vous placerez toutes vos ressources dans le dossier public/assets :

Dans le dossier templates vous allez créer un dossier éléments qui contiendra les fichiers suivants :

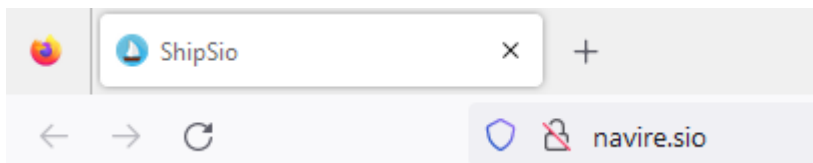


- ✓ Le fichier search.html.twig
- On verra plus tard

Exemple de homepage :



La carte : voir <https://www.vesselfinder.com/fr> pour afficher une carte en temps réel



Contact : voir plus loin formulaire de contact
Moodle de Benoît Roche ... vous pouvez prévoir un autre site !



Mentions légales et plan du site : liens morts pour l'instant

Partie 3 : FORMULAIRE DE DEMANDE D'INFORMATIONS

Vous allez maintenant gérer un formulaire de contact.

Ce formulaire s'ouvrira lorsque l'on cliquera su le lien contact dans le footer :



Pour cela, vous allez :

- ✓ Créer une entity Message
- ✓ Créer un formulaire
- ✓ Créer le contrôleur MessageController
- ✓ Créer le template twig
- ✓ Créer un service pour : enregistrer la demande en BD et envoyer un mail de confirmation à la personne qui a demandé un renseignement
- ✓ Afficher ce formulaire à l'aide d'une méthode du contrôleur MessageController
- ✓ Mettre à jour le lien contact dans le footer du template de base (footer)
- ✓ Envoyer le mail

Votre formulaire ressemblera à ceci :

Nom

Prénom

Mail

Message

Que voulez-vous nous dire ?

Bravo pour votre travail, il vous reste maintenant à complexifier les formulaires grâce aux données stockées en base

