

# Exercices

---

## Q1

---

Créer votre projet Symfony en full avec : `symfony new reduc --full`

Pour forcer la version de Symfony : `symfony new reduc --full --version=5.4`

## Q2

---

Créer une page d'accueil dans le site symfony avec `symfony console make:controller`

Appeler le controller `Home`

Créer une page de contact dans le site symfony dont le controller sera nommé `Contact`

## Q3

---

Créer un formulaire nommé Reduction avec `symfony console make:form` et avec 3 champs (inputs) nommés `nomProduit`, `prixInitial` et `pourcentageReduction`

## Q4

---

Afficher le formulaire en 1. le créant à partir du `FormType` `$formulaire = $this->createForm(ReductionType::class);`, transférez le `$formulaire` vers la vue (le mettre dans le array du `renderForm`) et afficher le dans Twig avec la fonction `Twig form(...)`

## Q5

---

Afficher un bouton submit dans le formulaire avec le type `SubmitType`

## Q6

---

Afficher une nouvelle vue `success.html.twig` lorsque le formulaire est soumis Vous pouvez utiliser la méthode `handleRequest` pour récupérer les données de l'objet `Request` et vous pouvez utiliser la méthode `isSubmitted(): bool` pour savoir si les données sont dans l'objet `Request`

## Q7

---

Récupérer les données envoyées à partir du formulaire avec la méthode `getData(): array` Calculer le prix final après la réduction et afficher le dans la vue (je vous rappelle qu'il faut le transmettre avant)

## Mise en situation #1

---

Expression du besoin : Faire la page de contact en affichant un formulaire avec les champs suivant `email`, `objet`, `message` pour que le visiteur puisse entrer son adresse email (pour qu'il puisse être recontacté), l'objet du son message et le contenu de son message. Le champs message devra être un champs multiligne. A la soumission du formualire, une nouvelle vue devra s'afficher, confirmant que le message a bien été envoyé et que le visiteur sera recontacté au plus vite à l'adresse email "l'adresse email que le visiteur a saisie"

1. Créer un `FormType` avec ``symfony console make:form``
2. Créer le formulaire à partir du `FormType` dans le controller
3. Passer le formulaire à la vue
4. Afficher le formulaire dans la vue
5. Autowirer l'objet `Request`
6. Utiliser `handleRequest(Request)` pour que le formulaire puisse chercher ses données dans l'objet `Request`
7. Mettre un if pour faire un traitement spécifique au cas ou le formulaire trouve ses données dans `Request`
8. Récupérer les données saisies avec `getData(): array`
9. Transmettre les données à la vue
10. Afficher ces données dans la vue

## Mise en situation #2

---

Expression du besoin : Faire une page de newsletter dans laquelle une personne peut s'inscrire à une newsletter en entrant son adresse email et son prénom. Lorsque le visiteur valide le formulaire, afficher une nouvelle vue annonçant au visiteur qu'il a bien été inscrit à la newsletter. N'oubliez pas que vous pouvez demander à Symfony de pré-coder un controller avec `symfony console make:controller`.

## Q10

---

Relier votre site Symfony à votre base de données en modifiant la connectionString de la variable d'environnement `DATABASE_URL`

Ensuite, demandez à Doctrine de créer la base de données avec `symfony console doctrine:database:create`

## Q11

---

Créer une entité `User` pour gérer les utilisateurs avec `symfony console make:user`

Ensuite demander à Symfony de générer le code SQL associé : `symfony console make:migration`

Ensuite demandez à Doctrine d'exécuter la migration créée : `symfony console doctrine:migrations:migrate`