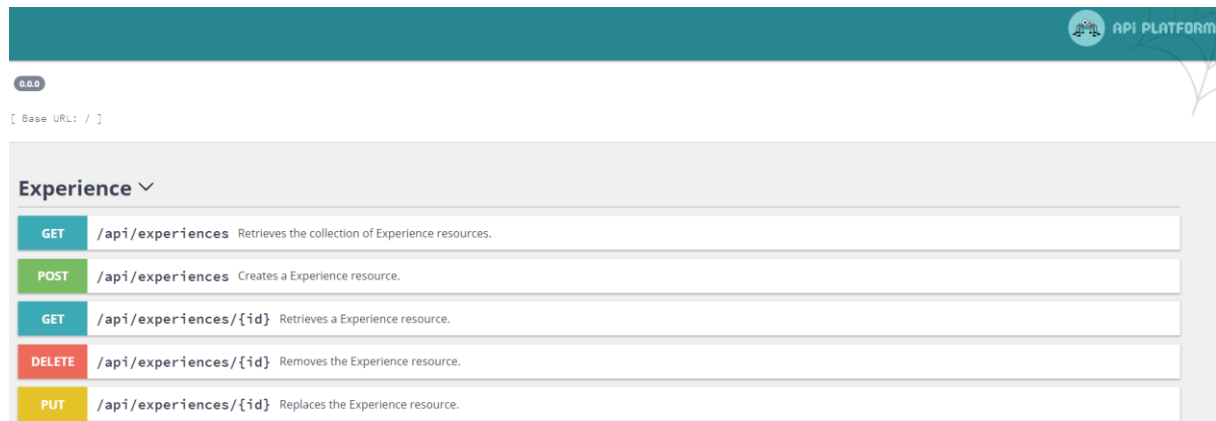


« Day 3 »

Api-Platform

Mettre à jour **composer** avec « api-platform/core », pour cela, taper
« **composer require api** »

Utiliser l'annotation `@ApiResource` sur chaque entité de votre modèle pour générer une API de votre modèle. Aller voir l'url `/api`.



Utilisez votre API de votre navigateur (« try it out » button), de votre terminal cloud9 (curl)

Les formulaires (créer/modifier) des données de vos pages

L'objectif de cette section est de vous permettre de créer ou modifier des données à l'aide de pages dédiées sur votre site. On parle alors de « formulaire ».

1. Dans votre terminal cloud9 « **composer require form** » puis

```
php bin/console make:form
```

Cette commande va créer un formulaire et l'associer à une classe de votre modèle.

2. Ensuite il faut appeler votre `FormType` dans un contrôleur dédié (créer un nouveau contrôleur pour toutes les opérations sur votre entité `Loisir`) et l'associer à une entité `loisir` « vide »

```
my_cv > src > Controller > LoisirController.php > PHP Intelephense > LoisirController > create
1  <?php
2
3  namespace App\Controller;
4
5  use Symfony\Component\HttpFoundation\Response;
6  use Symfony\Component\Routing\Annotation\Route;
7  use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
8  use App\Entity\Loisir;
9  use App\Form\LoisirType;
10
11 class LoisirController extends AbstractController
12 {
13     public function create()
14     {
15         $loisir = new Loisir();
16         $form = $this->createForm(LoisirType::class, $loisir);
17
18         return $this->render('loisir/create.html.twig', [
19             'entity' => $loisir,
20             'form' => $form->createView(),
21         ]);
22     }
23 }
24 }
```

3. Il ne vous reste plus qu'à créer une page dédiée pour votre nouvelle action dans template/Loisir/create.html.

```
my_cv ▸ templates ▸ loisir ▸ create.html.twig
1  <!DOCTYPE html>
2  <html>
3      <h1>Création d'un nouveau loisir</h1>
4      {{ form(form) }}
5  </html>
```

Le formulaire s'affiche mais ne sauvegarde pas encore vos données !



← → ↻ 🏠 ⓘ 127.0.0.1:8000/loisir/create

Création d'un nouveau loisir

Name

Commentaire

4. Pour cela, rajouter dans votre fichier create.html.twig la ligne :

```
5. {{ form_start(form, {'action': path('valid_loisir') }) }}
```

avant la balise form(form). Enfin faite l'action valide_loisir dans votre contrôleur des loisirs :

```
my_cv ▸ src ▸ Controller ▸ LoisirController.php ▸ PHP Intelephense ▸ LoisirController ▸ create
24  }
25
26  public function valid(Request $request)
27  {
28      $loisir = new Loisir();
29      $form = $this->createForm(LoisirType::class, $loisir);
30
31      $form->handleRequest($request);
32
33      if ($form->isSubmitted() && $form->isValid()) {
34          $loisir = $form->getData();
35
36          $entityManager = $this->getDoctrine()->getManager();
37          $entityManager->persist($loisir);
38          $entityManager->flush();
39
40          return $this->redirectToRoute('index');
41      }
42
43      return $this->render('loisir/create.html.twig', [
44          'entity' => $loisir,
45          'form' => $form->createView(),
46      ]);
47  }
48
49  }
```

On oublie pas de rajouter un bouton « save » dans le formtype :

```
my_cv ▸ src ▸ Form ▸ LoisirType.php ▸ PHP Intelephense ▸ LoisirType ▸ buildForm
4
5 use App\Entity\Loisir;
6 use Symfony\Component\Form\AbstractType;
7 use Symfony\Component\Form\FormBuilderInterface;
8 use Symfony\Component\OptionsResolver\OptionsResolver;
9 use Symfony\Component\Form\Extension\Core\Type\SubmitType;
10
11 class LoisirType extends AbstractType
12 {
13     public function buildForm(FormBuilderInterface $builder, array $options)
14     {
15         $builder
16             ->add('name')
17             ->add('commentaire')
18             ->add('save', SubmitType::class, [
19                 'attr' => ['class' => 'save'],
20             ])
21     ;
22 }
23
24 public function configureOptions(OptionsResolver $resolver)
25 {
26     $resolver->setDefaults([
27         'data_class' => Loisir::class,
28     ]);
29 }
30 }
```

A votre avis, que faut-il changer pour non plus créer de nouveaux loisirs mais les éditer ?
Un indice ? ...

```
public function edit($id)
{
    $entityManager = $this->getDoctrine()->getManager();
    $loisir = $entityManager->getRepository(Loisir::class)->findOneBy(['id' => $id]);
    $form = $this->createForm(LoisirType::class, $loisir);

    return $this->render('loisir/create.html.twig', [
        'entity' => $loisir,
        'form' => $form->createView(),
    ]);
}
```

Dernier point, en solo <https://symfony.com/doc/current/forms.html>

Regardez la section Form Validation et mettez la en application dans votre code.