

Routing E-Commerce

Dans notre cas du site E-Commerce nous utilisons le routeur de Symfony 6 voir ci-dessous

RoutingConfigurator nous permet de configurer nos routes que nous allons créer nos routes

Ce qui a été souligné en rouge sont les routes.

Nous retourneront du JSON pour l'API avec le code comme cela :

```
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;

use Symfony\Component\HttpFoundation\Response;

use Symfony\Component\Routing\Annotation\Route;

class EcommerceController extends AbstractController

{

    /**

    * @Route("/login", name="login")

    */

    public function login(): Response

    {

        $login = "La logique sera ici";

        return $this->json(['login' => $login]);

    }

    /**

    * @Route("/signup", name="signup")

    */

    public function signup(): Response

    {

        return $this->json([]);

    }

}
```

```
}

/**

* @Route("/users", name="users")

*/

public function users(): Response

{

    return $this->json([]);

}

/**

* @Route("/products", name="products")

*/

public function products(): Response

{

    return $this->json([]);

}

/**

* @Route("/products/{id}", name="product")

*/

public function product($id): Response

{

    return $this->json([]);

}

/**

* @Route("/profil/{id}", name="profile")

*/

public function profile($id): Response

{
```

```

        return $this->json([]);
    }

    /**
     * @Route("/profil/{id}/carts", name="carts")
     */
    public function profileCartsList($id): Response
    {
        // Votre logique pour récupérer un panier d'un profil en fonction de l'id

        return $this->json([]);
    }

    /**
     * @Route("/profil/{id}/cart/{id}", name="cart")
     */
    public function profileCart($id): Response
    {
        // Votre logique pour récupérer un panier d'un profil en fonction de l'id

        return $this->json([]);
    }
}

```