

## Étape 1: Configuration de l'environnement

---

Installer Symfony:

- Assurez-vous que PHP, Composer, et Symfony CLI sont installés.
- Créez trois nouveaux projets Symfony:

```
symfony new notification-service --full
symfony new billing-service --full
symfony new content-service --full
```

## Étape 2: Création des microservices

---

### Notification Service

- Configuration de la base de données:

Modifiez le fichier `.env` pour configurer l'accès à la base de données.

- Créer une entité Notification:

Utilisez la commande `make:entity` pour créer l'entité Notification. Ajoutez des champs comme type : `email_recipient`, `message`, `sujet`.

- Créer un contrôleur pour gérer les notifications:

Créez un contrôleur avec `make:controller`.

Votre tâche consiste à créer un endpoint qui, lorsqu'il est appelé, envoie un email à un destinataire spécifié et enregistre les détails de l'email (destinataire, sujet, message) dans la base de données.

- Retournez les réponses en JSON.

### Billing Service

- Configuration de la base de données:

Modifiez le fichier `.env` pour configurer l'accès à la base de données.

- Créer une entité Facture:

Utilisez la commande `make:entity` pour créer l'entité Facture.  
Ajoutez des champs comme `amount`, `due_date`, `customer_email`.

- Créer un contrôleur pour gérer les factures:

Créez un contrôleur avec `make:controller`.  
Ajoutez des actions pour créer, lire, mettre à jour et supprimer

- Retournez les réponses en JSON.

## Order Service

- Configuration de la base de données

Modifiez le fichier `.env` pour configurer l'accès à la base de données.

- Créer une entité Commande:

Utilisez la commande `make:entity` pour créer l'entité Commande.  
Ajoutez des champs comme `product_id`, `customer_email`, `quantity`, `total_price`.

- Créer un contrôleur pour gérer les commandes:

Créez un contrôleur avec `make:controller`.  
Ajoutez des actions pour créer, lire, mettre à jour, et supprimer des commandes

- Retournez les réponses en JSON.

## Etape 3 : Communication entre les microservices

### Communication entre Order Service et Billing Service :

Order Service enregistre les informations sur une commande reçue sous forme d'une requête via Postman et envoie une requête à Billing Service pour la création de facture.

```
$response = $this->client->request('POST', 'http://billing-service.local/create-invoice',
    'json' => [
        'amount' => $data['total_price'],
        'due_date' => (new \DateTime('+30 days'))->format('Y-m-d'),
        'customer_email' => $data['customer_email']
    ],
    []);
```

## Communication entre Billing Service et Notification Service

Une fois qu'une facture a été générée, il faut faire une requête à Notification Service pour envoyer une notification à l'adresse mail du client.

```
$notificationResponse = $this->client->request('POST', 'http://notification-service.local/send',
    'json' => [
        'sujet' => 'Billing',
        'recipient' => 'customer@example.com',
        'message' => 'Your invoice has been created.'
    ],
    []);
```

## Critères à respecter

- Envoyer le projet avant 14 Juillet 23h59
- Toute traces copier coller de GPT ou autres IA générative entrainera des points négatifs (Appuyez-vous sur le bon utilisation des ces outils IA)
- Le projet est à faire en trinôme mais si vous êtes que deux dans l'équipe il faut faire le Billing service et Notification Service
- Il faut m'envoyer un lien vers votre repository github. (Il faut simplement d'une seule repo github pour les 3 microservices)
- Le mail contenant le lien github doit être intitulé : **[Projet Final] - [Classe 3] - [Nom + Prénom Eleve 1] / [Nom + Prénom Eleve 2] / [Nom + Prénom Eleve 3]**
- Pour toute question sur le projet privilégier l'adresse de contact : [bkirouch@parisnanterre.fr](mailto:bkirouch@parisnanterre.fr)

## Lien utiles

- <https://github.com/symfony/demo>
- <https://symfony.com/doc/current/index.html>