

Documentation API

Définition d'une API:

Une API (Application Program Interface) est un outil qui permet de communiquer de manière standardisée et sécurisée pour permettre à un logiciel de faire un lien vers un autre. Elle peut échanger des informations ou déclencher des actions sur un autre logiciel. Par exemple une API pour un service financier qui permettrait de pouvoir avoir accès aux données bancaires des utilisateurs, après leur consentement, permettant de faciliter l'analyse des dépenses de l'utilisateur.

Fonctionnement d'une API :

Pour pouvoir fonctionner correctement, l'API a besoin de créer un lien entre les deux logiciels qui se fait généralement par le protocole HTTP avec le fonctionnement de client-serveur avec une application qui fait la demande et une autre qui répond. Les données sont généralement formatées en JSON (JavaScript Object Notation) ou XML (Extensible Markup Language).

Les API de RestoWeb :

Pour notre projet, nous pouvons retrouver deux types d'API:

1. L'API "commande_en_attente". Cette API renvoie du contenu JSON pour RestoSwing.
2. Les API "commende_accepter", "commende_refuser" et "commende_terminer" qui font des modification sur la base de donnée restoweb.

Fonctionnement de l'API qui revoi du contenu JSON :

La première API "commande_en_attente" est une API qui va permettre d'alimenter le client RestoSwing en lui envoyant les commandes qui sont en attente. Pour faire cela, il a fallu exécuter une requête SQL qui a récupéré toutes les commandes qui avait le statut "en attente".

Il a fallu également créer une variable mesLignescommande et une classe Lignecommande qui à repris chaque information de lignecommande de la base de donn  .

Par la suite, Nous avons mis un foreach pour que chaque r  sultat renvoyer soit mit dans un tableau en s'appuyant sur la classe Lignecommande. Nous avons donc eu des tableaux de tableau avec les informations g  n  rales de la commande qui est ensuite lignecommande qui avait plusieurs tableaux    cause des diff  rents produits.

Par la suite nous avons préparé la réponse pour le serveur en transformant le tableau au format JSON

```
( $json = json_encode(array(1=>"bleu",2=>"blanc",3=>"rouge")) ); ) en en  
encapsulant la réponse avec un entête qui était le header avec par exemple  
header("Content-type: application/json; charset=utf-8");
```

Fonctionnement des API qui modifie la base de donnée:

Les autres API étaient plus simples car elles ne nécessitent pas d'envoi de contenu et de transformation au format JSON.

Pour ces API nous nous sommes connectés à la base de données à l'aide de fonctions que nous avons déjà créées. Nous avons utilisé des requêtes SQL qui effectuaient un UPDATE sur les statuts de la commande pour pouvoir actualiser le suivi. Après ces requêtes, un catch, qui a été mis en place, pour que si une erreur est détectée, la requête s'arrête et nous renvoie le message d'erreur qui génère MySQL.

Ces API seront par la suite utilisées par le client RestoSwing qui sera utilisé par le restaurateur pour pouvoir informer les clients sur l'avancement ou non de leur commande.