

# PayGreen

## Étape 1 : l'analyse

- Définition des variables privées.
- Récupération d'une nouvelle instance de la classe si elle n'existe pas (fonction getInstance).
- Attribution des valeurs aux variables UI, CP et HOST par le constructeur de la classe.

Authentification au serveur PayGreen :

- Récupération de l'adresse IP du client avec l'index ADDR de la variable serveur.
- Récupération d'un jeton par le protocole OAuth par la fonction requestApi et ses autres appels de méthodes en fonction des conditions.
- Récupération des URL d'autorisation avec les fonctions d'authentifications.

Modification de la fonction refundOrder pour avoir une condition « if else » au lieu de « if if » :

```
if (!empty($pid)) {  
    $datas = array('pid' => $pid);  
    if ($amount != null) {  
        $datas['content'] = array('amount' => $amount * 100);  
    }  
    return $this->requestApi('refund', $datas);  
} else return false;
```

Récupération des informations du compte utilisateur :

- Soulagement de la charge de travail du code en mettant en premier les conditions « == false » dans la fonction getAccountInfos.

```
if ($account == false) {  
    return false;  
}  
  
if ($this->isContainsError($account)) {  
    return $account->error;  
}
```

Étapes du paiement sécurisé avec la récupération des informations de paiement et d'un jeton unique à partir de la fonction `getRoundingInfo` jusqu'à la fonction `createTokenize`.

Les fonctions privées récupèrent des informations des utilisateurs, retournent des URL et modifient des informations dans la base de données.

### Amélioration :

L'amélioration du code pourrait se faire sur plusieurs fichiers en fonction des étapes afin de soulager sa lecture et compréhension.