

Objectif général

Dans ce TD, vous allez intégrer un paiement en ligne avec la solution **Up2Pay e-Transactions du Crédit Agricole** en mode **redirection vers la page de paiement sécurisée de la banque**.

Le principe général est le suivant :

1. le site prépare une commande ;
2. le site envoie l'utilisateur vers la page de paiement de la banque ;
3. la banque traite le paiement ;
4. l'utilisateur revient sur le site avec un message visuel de succès, d'annulation ou d'échec ;
5. en parallèle, le **serveur de la banque** envoie un retour serveur à serveur permettant de **valider réellement le statut du paiement**. Ce retour est le plus fiable pour mettre à jour la base de données. Up2Pay décrit bien ce fonctionnement avec redirection client + notification automatique IPN.

<https://www.ca-moncommerce.com/espace-client-mon-commerce/up2pay-e-transactions/ma-documentation/>

Données techniques à utiliser

Les identifiants de test fournis pour ce TD sont :

- **PBX_SITE** : 3277512
- **PBX_IDENTIFIANT** : 38023694
- **PBX_RANG** : 001
- **PBX_RUF1** : POST
- **Clé HMAC** :
E7DD686B8817CD0A6772BBB0C744705A6C3814444C15337FF7878EAFDC1CF4BA67ABAC9E92C8BA5C000C187DAA22CFA9C3182D94C22F69698982A285EBAB8846

Point très important sur le numéro de commande

Pour ce TD, votre numéro de commande doit **obligatoirement commencer** par :

25gpX-

X étant votre numéro de serveur

Exemples attendus :

- 25gpX-1
- 25gpX-2
- 25gpX-3

Dans un premier temps, pour tester rapidement l'intégration, vous pourrez utiliser :

```
$numeroCommande = '25gpX-' . uniqid();
```

Puis, dans un second temps, vous devrez remplacer cette logique par le **vrai numéro de commande enregistré en base**, par exemple :

```
$numeroCommande = '25gpX-' . $commande['id'];
```

Organisation attendue

Vous devez créer :

- un **contrôleur** : c-paiement.php
- une **vue** : v-paiement.php

Puis vous devrez ajouter les pages de retour de la banque.

Dérroulement du TD

Étape 1 – Comprendre le fonctionnement global

Avant de coder, il faut bien distinguer **deux types de retour** :

1. Retour navigateur

Après le paiement, la banque redirige l'utilisateur vers votre site. Ce retour sert principalement à **afficher un message** :

- paiement accepté ;
- paiement annulé ;
- paiement refusé.

Ce retour est utile pour l'expérience utilisateur, mais il ne doit pas être considéré comme la seule source fiable pour valider le paiement.

2. Retour serveur à serveur

La banque envoie aussi un appel HTTP directement à votre serveur via la variable **PBX_REPONDRE_A**.

C'est ce mécanisme qui permet d'enregistrer automatiquement le résultat du paiement dans la base de données. La documentation de test Up2Pay précise bien que les informations de paiement sont envoyées à votre site pour la mise à jour automatique de l'état de la commande par IPN, tandis que le client est redirigé vers votre page de confirmation.

Conclusion importante pour ce TD :

- le **retour utilisateur** sert à afficher ;
 - le **retour serveur** sert à enregistrer réellement le paiement.
-

Étape 2 – Créer la page /paiement/

L'objectif est de rendre accessible une page de test via l'URL :

/paiement/

Cette page doit :

- préparer les variables nécessaires à l'appel de l'API ;
- construire le formulaire HTML ;
- envoyer automatiquement l'utilisateur vers la plateforme de paiement Crédit Agricole.

La plateforme de test Up2Pay utilise une URL de préproduction pour initier les transactions, distincte de la production.

Étape 3 – Premier objectif : faire partir un paiement avec un numéro de commande temporaire

Dans cette première phase, vous ne gérez pas encore la vraie commande.

Vous devez simplement réussir à appeler l'API avec un identifiant temporaire unique :

```
$numeroCommande = '25gp1-' . uniqid();
```

Travail demandé

Dans c-paiement.php, préparer les variables utiles, par exemple :

- site ;
- rang ;
- identifiant ;
- montant ;
- devise ;
- numéro de commande ;
- email client ;
- URL de retour succès ;
- URL de retour annulation ;
- URL de retour refus ;
- URL de réponse automatique ;
- horodatage ;
- chaîne signée ;
- empreinte HMAC.

Attention, c'est à vous de choisir les informations renvoyées par la banque. Pensez à sélectionner les éléments qui vous semblent pertinents pour votre client.

Résultat attendu

Quand l'étudiant ouvre /paiement/, il doit être redirigé vers la page de paiement de test de la banque et pouvoir simuler un paiement.

Étape 4 – Mettre en place les pages de retour utilisateur

Une fois le paiement réalisable, il faut gérer le retour côté navigateur.

Les variables à exploiter sont :

- \$pbx_effectue
- \$pbx_annule
- \$pbx_refuse

L'idée est d'avoir **un contrôleur** qui lit le retour, puis affiche **une des trois vues possibles** :

- paiement effectué ;
- paiement annulé ;
- paiement refusé.

Ce qu'on attend

Créer par exemple :

- c-retour-paiement.php
- v-paiement-ok.php
- v-paiement-annule.php
- v-paiement-refuse.php

ou bien un seul contrôleur avec une vue variable.

Important

À ce stade, on **n'enregistre pas encore définitivement** le paiement grâce au retour navigateur seul.

On affiche seulement l'état retourné à l'utilisateur.

Étape 5 – Deuxième objectif : remplacer `uniqid()` par le vrai numéro de commande

Une fois le premier test fonctionnel, il faut passer à une intégration plus réaliste.

Cette fois, le numéro de commande transmis à la banque doit correspondre à une **vraie commande enregistrée dans votre base de données**.

Exemple :

1. insertion de la commande dans la table commande ;
2. récupération de son identifiant ;
3. envoi du paiement avec :

```
$numeroCommande = '25gp1-' . $idCommande;
```

Pourquoi cette étape est indispensable

Le numéro de commande envoyé à la banque est ce qui vous permettra ensuite de :

- retrouver la commande concernée ;
- enregistrer le paiement dans la table paiement ;
- mettre à jour le statut de la commande.

Résultat attendu

Le paiement doit désormais être lié à une vraie commande et non plus à un identifiant de test isolé.

Étape 6 – Mettre en place le retour serveur de la banque

C'est l'étape la plus importante du TD.

La variable **PBX_REPONDRE_A** permet de fournir à la banque l'URL que son serveur devra appeler pour vous transmettre le résultat du paiement. Up2Pay indique que ce retour automatique permet la mise à jour de l'état de la commande par IPN.

Rôle de ce fichier

Ce script doit :

1. récupérer les données envoyées par la banque ;
2. identifier la commande concernée ;
3. enregistrer la trace du paiement dans la table paiement ;
4. mettre à jour le statut de la commande.

Exemple de logique attendue

- si le paiement est accepté :
 - insertion d'un paiement avec statut accepte
 - mise à jour de la commande en payee
- si le paiement est annulé :
 - insertion d'un paiement avec statut annule
 - mise à jour éventuelle de la commande en annulee
- si le paiement est refusé :
 - insertion d'un paiement avec statut refuse
 - mise à jour éventuelle de la commande en refusee

Très important

Le **retour serveur** doit être considéré comme la source principale pour la base de données. Le retour navigateur, lui, sert surtout à informer le client.

PS : Vous devez donc créer une table paiement et ajouter un champ statut dans votre table commande.