



# OC PIZZA

## OCPizzapp

Dossier d'exploitation

Version 1.1

**Auteur**

FERRANDEAU MATHIEU

*Analyst-programmeur*



# TABLE DES MATIÈRES

<b>1 - Versions.....</b>	<b>3</b>
<b>2 - Introduction.....</b>	<b>4</b>
2.1 - Objet du document.....	4
2.2 - Références.....	4
<b>3 - Pré-requis.....</b>	<b>5</b>
3.1 - Système.....	5
3.2 - Bases de données.....	5
3.3 - Web-services.....	5
<b>4 - Procédure de déploiement.....</b>	<b>6</b>
4.1 - Déploiement de l'application web.....	6
4.1.1 - Composition de l'application web.....	6
4.1.2 - Variables d'environnement.....	6
4.1.3 - Configuration.....	7
4.1.4 - Déploiement.....	7
4.1.4.1 - Création de l'application :.....	7
4.1.4.2 - Configuration des variables d'environnement :.....	7
4.1.4.3 - Envoyer l'application sur le serveur :.....	7
4.1.5 - Vérifications.....	8
4.2 - Déploiement de la base de données.....	8
4.2.1 - Lancement des migrations.....	8
4.2.2 - Création d'un superutilisateur.....	8
4.2.3 - Import des données.....	8
4.2.4 - Vérifications.....	9
<b>5 - Procédure de démarrage / arrêt.....</b>	<b>10</b>
5.1 - Démarrage / arrêt sur le site Heroku.com.....	10
5.2 - Démarrage / arrêt en ligne de commande.....	10
<b>6 - Procédure de mise à jour.....</b>	<b>11</b>
<b>7 - Supervision/Monitoring.....</b>	<b>12</b>
<b>8 - Procédure de sauvegarde et restauration.....</b>	<b>13</b>
8.1 - Sauvegarde de la base de données.....	13
8.2 - Restauration de la base de données.....	13
<b>9 - Glossaire.....</b>	<b>14</b>



# 1 - VERSIONS

Auteur	Date	Description	Version
MF	13/11/2019	Création du document	1.0
MF	13/11/2019	Mise à jour du document	1.1



## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application OCPizzapp.

L'objectif du document est de fournir à l'équipe technique de la société OC PIZZA les informations essentielles pour une bonne utilisation de l'application, ainsi que les instructions à suivre pour le déploiement et la maintenance de celle-ci.

### 2.2 - Références

Pour de plus amples informations, se référer :

1. **DCT – 1.1** : Dossier de conception technique de l'application
2. **DCF – 1.1** : Dossier de conception fonctionnelle de l'application



## 3 - PRÉ-REQUIS

### 3.1 - Système

L'application web OCPizzapp est hébergé sur la plateforme « Heroku ».

L'application est liée au nom de domaine : « ocpizzapp.fr ».

### 3.2 - Bases de données

La SGBD utilisé par l'application est MySQL version : 8.0.

La base de données est hébergée sur les serveurs Heroku.

### 3.3 - Web-services

Le fonctionnement de l'application nécessite que les web-services suivants soit opérationnels:

- Google Maps API : L'utilisation de cette API nécessite une clé d'identification. La clé est la suivante: **'B>WF2FzyF>opguZQ7DqoJ(Uz'**.

Le service est payant en fonction du nombre de requêtes à cette API, il est nécessaire de veiller à ce que les règlements soient bien effectués pour éviter tout dysfonctionnement de l'application.

- Stripe : Pour cette API une commission est prise automatiquement sur chaque paiement effectués en ligne.

- Mobimel : En fonction du nombre de sms émis par mois, le sms est facturé à un certain montant (formule à choisir directement sur le site).



## 4 - PROCÉDURE DE DÉPLOIEMENT

### 4.1 - Déploiement de l'application web

#### 4.1.1 - Composition de l'application web

L'application OCPizzapp est construite sous la forme d'une archive ZIP contenant les répertoires suivants:

- **OCPizzapp** : contient les fichiers de configuration de l'application et de Django
- **Procfile** : contient les instructions pour démarrer l'application
- **Authentification** : contient les fichiers liés au package « Authentification »
- **Ventes** : contient les fichiers liés au package « Vente »
- **Production** : contient les fichiers liés au package « Production »
- **Templates** : contient les fichiers HTML de l'application
- **Static** : Contient les fichiers .CSS et .JS de l'application
- **requirements.txt** : contient les librairies nécessaires pour que l'application fonctionne
- **Docs** : contient la documentation de l'application
- **README.md** : fichier expliquant comment lancer et le fonctionnement de l'application

Le fichier **Procfile** contient les instructions exécutées par Heroku pour démarrer l'application.

Le contenu de ce fichier, pour que l'application fonctionne, doit au moins contenir :

```
web: gunicorn ocpizzapp.run :app
```

#### 4.1.2 - Variables d'environnement

L'application nécessite la configuration des variables d'environnement suivantes:

Nom	Valeur	Obligatoire	Description
ENV	<b>PRODUCTION</b>	Oui	Indique à l'application qu'il faut utiliser la configuration de production et non celle de développement
SECRET_KEY	'B>4R2DzyF>op gwJQ7DqoJ(Uz'	Oui	Nécessaire au démarrage de Django. Cette clé est la clé de production et ne doit figurer nulle part ailleurs que sur le



			serveur de production.
--	--	--	------------------------

### 4.1.3 - Configuration

Voici les différents fichiers de configuration :

- **Procfile** : fichier contenant les commandes utilisées par Heroku pour le déploiement de l'application
- **requirements.txt** : fichier contenant la liste des librairies à installer sur le serveur Heroku pour que l'application fonctionne correctement.
- **OCPizzapp/Ocpizzapp/settings.py** : fichier de configuration de l'application Django

### 4.1.4 - Déploiement

Heroku propose différentes méthodes pour déployer une application. Nous conseillons cependant d'effectuer un déploiement en ligne de commande en utilisant « Heroku CLI ».

Les étapes présentées ci-dessous nécessitent que « Heroku CLI » soit installé, que le projet soit suivi avec « Git » et il faut être sur la branche « master » du projet.

#### 4.1.4.1 - Création de l'application :

Une fois « Heroku CLI » installé, il faut se rendre à la racine du projet de l'application et entrer la commande suivante :

```
$ heroku create ocpizzapp
```

#### 4.1.4.2 - Configuration des variables d'environnement :

Les variables d'environnement se définissent via la ligne de commande de la façon suivante:

```
$ heroku config:set ENV='PRODUCTION'
```

```
$ heroku config:set SECRET_KEY='XXXXXXXXX'
```

Pour vérifier que les variables ont bien été configurées, utilisez la commande suivante:

```
$ heroku config
```



#### 4.1.4.3 - Envoyer l'application sur le serveur :

Une fois les étapes précédentes réalisées, il suffit d'effectuer un « Push » sur le serveur:

```
$ git push heroku master
```

#### 4.1.5 - Vérifications

Une fois le déploiement terminé, les lignes suivantes doivent apparaître dans le terminal :

```
remote : https://ocpizzapp.fr/ deployed to Heroku  
remote : Verifying deploy... done.
```

Pour s'assurer du bon déploiement de l'application, rendez-vous à l'adresse [www.ocpizzapp.fr](http://www.ocpizzapp.fr) afin de vérifier que celle-ci soit bien en ligne.





## 4.2 - Déploiement de la base de données

Le déploiement de la base de données sur la plateforme Heroku nécessite que l'application ait été déployée comme présenté précédemment.

De la même manière que pour le déploiement de l'application, celui de la base de données est présenté ici en utilisant la ligne de commande et « Heroku CLI ».

### 4.2.1 - Lancement des migrations

Avant d'importer les données en elle-même, il est nécessaire de créer et configurer les tables de la base de données via la commande suivante :

```
$ heroku run python manage.py migrate
```

### 4.2.2 - Création d'un superutilisateur

Une fois la base de données créée, il faut ajouter un super utilisateur via la commande :

```
$ heroku run python manage.py createsuperuser
```

### 4.2.3 - Import des données

Il est possible d'importer les données de la version de développement de l'application sur le serveur de production. Pour cela, il faut créer un dump de la base de données de développement, puis l'importer sur le serveur Heroku.

Création d'un dump de la base de données:

```
$ ./manage.py dumpdata ocpizzapp > ocpizzapp/dumps/ocpizzapp.json
```

Import des données dans la base :

```
$ heroku run python manage.py loaddata ocpizzapp/dumps/ocpizzapp.json
```

### 4.2.4 - Vérifications

La vérification de l'état de la base de données peut se faire directement via le site d'Heroku dans la rubrique « Ressources » du projet.

Il faut ensuite se rendre dans la partie « Add-ons » puis cliquer sur « Heroku MySQL ».

On retrouve dans cet espace les informations sur la base de données telles que : son état (active ou non), le nombre de lignes et de tables ou encore l'espace utilisé.



## 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

La gestion de l'application peut se faire de deux manières, soit depuis le site Heroku soit en ligne de commande.

### 5.1 - Démarrage / arrêt sur le site Heroku.com

Une fois déployée sur Heroku, l'application est par défaut en ligne.

Afin de la désactiver il faut se rendre dans la rubrique « Settings », puis passer le statut « Maintenance Mode CLI » à « ON ».

Il suffira de faire la démarche inverse pour rétablir l'application.

### 5.2 - Démarrage / arrêt en ligne de commande

Afin de démarrer ou arreter l'application en ligne de commande, il faut utiliser les commandes suivantes :

Mise en marche de l'application :

```
$ heroku run
```

Arrêt de l'application :

```
$ heroku ps:scale web=0
```



## 6 - PROCÉDURE DE MISE À JOUR

La mise à jour de l'application nécessite de passer celle-ci en maintenance.

La mise en mode maintenance peut s'effectuer directement via le tableau de bord Heroku (vu dans la section précédente), soit en ligne de commande via la commande suivante :

```
$ heroku maintenance:on
```

Lorsque le mode maintenance est activé, un message s'affiche lors de la connexion à l'application et les utilisateurs n'ont plus accès à celle-ci.

Une fois la mise à jour effectuée, l'application peut être réactivée avec la commande suivante :

```
$ heroku maintenance:off
```



## 7 - SUPERVISION/MONITORING

La plateforme Heroku fournit des outils de monitoring dans la rubrique « Metrics » de l'application.

Plus d'informations sont disponibles dans la documentation Heroku disponible à l'adresse suivante : <https://devcenter.heroku.com/categories/monitoring-metrics>

Nos recommandations de suivi à mettre en place a minima sont les suivantes :

- Rapprochement de la limite de stockage de la base de données.
- Utilisation/Saturation de la mémoire vive
- Utilisation/Saturation du processeur



## 8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

Le processus de sauvegarde et de restauration de l'application passe essentiellement par la sauvegarde des données de celle-ci. Heroku propose cette fonctionnalité.

### 8.1 - Sauvegarde de la base de données

```
$ heroku pg:backups:capture --app ocpizzapp
```

### 8.2 - Restauration de la base de données

```
$ heroku pg:backups:restore <nom_sauvegarde> DATABASE_url --app ocpizzapp
```



## 9 - GLOSSAIRE

<b>DJANGO</b>	Framework permettant la réalisation d'application web Python
<b>SGBD</b>	Système de gestion de base de données