

# OC Pizza

## OCPizzapp

Dossier d'exploitation

Version v0.0

### **Auteur**

Amaury Bois  
*Developpeur*

# 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 -Avant le déploiement.....	6
4.1.1 -Dossiers et fichiers.....	6
4.1.2 -Variables d'environnement.....	6
4.1.3 -Configuration.....	7
4.2 -Déploiement de l'Application Web.....	8
4.2.1 -Création de l'application.....	8
4.2.2 -Création des variables d'environnement.....	8
4.2.3 -Déploiement de l'application.....	8
4.2.4 -Déploiement de la base de donnée.....	8
<b>5 -Procédure de démarrage / arrêt.....</b>	<b>10</b>
<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>
<b>9 -Glossaire.....</b>	<b>14</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Amaury Bois	24/09/20	Création du document	V0.0

## 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 les informations liés à la bonne utilisation de l'application aux services techniques de OC Pizza.

### 2.2 - Références

Pour de plus amples informations, se référer :

1. **DCF – v0.1** : Dossier de conception fonctionnelle de l'application
2. **DCT – v0.1** : Dossier de conception technique de l'application

## 3 - PRÉ-REQUIS

### 3.1 - Système

L'application sera hébergée sur la plateforme Heroku.

Le nom de domaine sera : « ocpizzapp.herokuapp.com »

### 3.2 - Bases de données

Heroku se chargera d'héberger la base de donnée. Nous utiliserons Posgresql v12.

### 3.3 - Web-services

Les web services suivants sont nécessaires à l'application :

**Google Map API :** L'utilisation de ce service nécessite une clé d'application. Voici la clé utilisée par OCPizzapp : **Vm)7Im/?Q]{CSjIk)058~YPZB&?|**

Le service étant payant, il est important de veiller au règlement des factures afin d'assurer le bon fonctionnement de l'application.

# 4 - PROCÉDURE DE DÉPLOIEMENT

## 4.1 - Avant le déploiement

### 4.1.1 - Dossiers et fichiers

Le fichier .ZIP qui vous sera remis contiendra les éléments suivants :

**ocpizzapp\_project** : Dossier contenant le “coeur” de l'application (paramètres etc.)

**order\_management** : Dossier contenant l'application de gestion des commandes

**direction\_management** : Dossier contenant l'application de gestion direction

**authentification** : Dossier contenant l'application d'authentification

**Procfile** : Fichier nécessaire au bon fonctionnement de l'application sur Heroku

**requirements.txt** : Fichier contenant les modules nécessaires à l'application

**README.md** : Fichier contenant diverses informations sur l'application (versionnage etc.)

**.gitignore** : Fichier contenant les fichiers ignorés par Git

**manage.py** : Fichier nécessaire au fonctionnement de l'application.

### 4.1.2 - Variables d'environnement

Voici les variables d'environnement nécessaires au fonctionnement de l'application :

Nom	Valeur	Obligatoire	Description
ENV	PRODUCTION	oui	Indique à l'application qu'elle ne doit pas utiliser le mode debug en production
SECRET_KEY	Ogczi7gBK3z2T jblF7j5kOz1pkO e8uVO	oui	Clé secrète utilisée par Django
GMAP_KEY	Vm)7lm/?Q] {CSjlk)058~YPZ B&?	oui	Clé de Google Map API (voir 3.3)

Définissez les variables d'environnement nécessaires comme ceci :

```
$ heroku config:set NOM=VALEUR
```

Exemple avec **ENV** :

```
$ heroku config:set ENV=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 modules à installer pour assurer le fonctionnement de l'application. Heroku installe ces modules lors du déploiement.
- **ocpizzapp\_project/settings.py** : fichier de configuration l'application Django

## 4.2 - Déploiement de l'Application Web

Il existe plusieurs façons de déployer une application sur Heroku. Nous recommandons le déploiement en ligne de commande à l'aide de "Heroku CLI" (qu'il sera nécessaire d'installer).

Le projet doit être sous Git et être sur la branche "master".

### 4.2.1 - Création de l'application

Une fois Heroku CLI installé et connecté au compte Heroku d'OC Pizza, créez l'application en utilisant la commande suivante :

```
$ heroku create ocpizzapp
```

### 4.2.2 - Création des variables d'environnement

Comme vu plus haut, ajoutez les variables d'environnement de cette façon :

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

```
$ heroku config:set SECRET_KEY=Ogczi7gBK3z2TjblF7j5kOz1pkOe8uVO
```

```
$ heroku config:set GMAP_KEY=Vm)7Im/?QJ{(CSjIk)058~YPZB&?|
```

Vous pouvez vérifier vos saisies avec la commande suivante :

```
$ heroku config
```

### 4.2.3 - Déploiement de l'application

Déployez simplement l'application avec la commande :

```
$ git push heroku master
```

Vous pourrez vérifier le bon déploiement de l'application quand les lignes suivantes apparaîtront :

```
remote : https://ocpizzapp.herokuapp.com/ deployed to Heroku
```

```
remote : verifying deploy... done
```

Afin de s'en assurer, connectez vous au site <https://ocpizzapp.herokuapp.com/>

### 4.2.4 - Déploiement de la base de donnée

Heroku se chargera de créer une base de donnée vierge puis d'effectuer les migrations lors du déploiement de l'application.

Si toutefois les migrations ne s'effectueraient pas, utilisez la commande suivante :

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

Il est également possible de remplir la base de donnée avec un dump créé sur la version de développement.

Voici la commande de création du dump :



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

Voici la commande permettant d'envoyer ce dump vers la base de donnée de Heroku :

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

Enfin, il est possible de vérifier le statut de la base de donnée en s'y connectant via ligne de commande :

```
$ heroku pg:psql
```

## 5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

Comme pour le déploiement, il existe plusieurs façons de démarrer ou d'arrêter une application sur Heroku. Une fois de plus, nous recommandons l'arrêt/démarrage en ligne de commande à l'aide de "Heroku CLI".

Commande de mise en marche de l'application :

```
$ heroku run
```

Commande d'arrêt de l'application :

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

## 6 - PROCÉDURE DE MISE À JOUR

Un mode maintenance est disponible sur Heroku pour les procédures de mise à jour. Ce dernier bloque l'accès à l'application aux utilisateurs tout en leur affichant un message informatif.

Vous pouvez activer ce mode avec la commande suivante :

```
$ heroku maintenance:on
```

Puis le désactiver avec :

```
$ heroku maintenance:off
```

## 7 - SUPERVISION/MONITORING

Heroku propose des outils de supervisions que vous pourrez trouver à cette adresse (une fois l'application créée):

<https://dashboard.heroku.com/apps/ocpizzapp/metrics>

Cette fonctionnalité est cependant payante.

## 8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

L'élément le plus important à sauvegarder est évidemment la base de donnée et son contenu.

Heroku propose une fonctionnalité pour cela.

Sauvegardez votre base de donnée avec la commande suivante :

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

Restaurez la avec cette commande :

```
$ heroku pg:backups:restore <nom sauvegarde> DATABASE_URL --app ocpizzapp
```

## 9 - GLOSSAIRE
