

# Documentation Technique

## CCI APPRO MOBILE (PPE4)

<b>Cahier des charges</b>	<b>2</b>
Résumé	2
Restriction	2
<b>Prérequis d'installation</b>	<b>2</b>
<b>Installation du backend</b>	<b>3</b>
Installation des logiciels requis	3
Configuration de mariadb	3
Création de la base de données	3
Création de l'utilisateur	3
Affectation des droits à l'utilisateur	3
Installation de l'application web contenant l'API	4
Paramétrage du fichier de configuration	4
Installation des packages	4
Installation de la base de données	4
Changement des droits et propriétaires de fichiers	5
Configuration d'apache	5
Par la suite, entrez la commande suivante pour activer la configuration :	6
<b>Installation du frontend</b>	<b>7</b>
Pré-requis	7
Installation d'Android studio	7
Installation de node.js	8
Variables d'environnement	8
Installation du projet	9
Lancement de l'application	9
<b>Diagrammes</b>	<b>10</b>
Diagramme de contextualisation	10
<b>Les données</b>	<b>11</b>
Bibliothèque de données	11
Modèle conceptuel des données	12
Endpoints de l'API	13

# Cahier des charges

## Résumé

Le projet concerne l'extension de l'application web de gestion de fourniture de la CCI Campus Strasbourg, CCI APPRO, par la création d'une application mobile cross-plateforme permettant de faciliter la gestion des commandes aux services généraux.

## Restriction

L'application doit être cross-plateforme et connectée à la base de données CCI APPRO par le biais d'une API. L'API doit respecter la norme PSR-12. Le projet doit être sur GitLab avec des mises à jour régulières.

## Prérequis d'installation

Système d'exploitation : Ubuntu 20.04.1 LTS

Framework PHP : Laravel 7

Application nécessaire :

- php7.3
- apache2
- mariadb-server
- composer

Extension de php nécessaire :

- php-bcmath
- php-json
- php-mb
- php-tokenizer
- php-XML
- php-openssl
- php-pdo
- php-ctype

# Installation du backend

## Installation des logiciels requis

Vous pouvez installer les logiciels listés au dessus grâce à la commande suivante :

```
sudo apt install php7.3 apache2 mariadb-server composer  
php7.3-bcmath php7.3-json php7.3-mb php7.3-tokenizer php7.3-xml  
php7.3-openssl php7.3-pdo php7.3-ctype
```

## Configuration de mariadb

Pour réaliser les manipulations nécessaires il faut accéder au CLI de mariadb. Pour cela, entrez la commande suivante :

```
sudo mariadb
```

### Création de la base de données

Dans l'interface de mariadb, entrez la commande suivante afin de créer la base de données de l'application (vous pouvez changer le nom de la base de données en remplaçant 'cci\_appro' par le nom voulu).

```
CREATE DATABASE cci_appro;
```

### Création de l'utilisateur

Pour créer l'utilisateur de l'application, entrez la commande suivante en indiquant le mot de passe voulu (vous pouvez changer le nom de l'utilisateur en remplaçant 'laravel' par le nom voulu).

```
CREATE USER laravel IDENTIFIED BY '<mot de passe>';
```

### Affectation des droits à l'utilisateur

Pour que l'utilisateur possède les droits nécessaires sur la base de données, entrez la commande suivante :

```
GRANT ALL PRIVILEGES ON cci_appro.* TO 'laravel';
```

Enfin vous pouvez quitter mariadb en entrant la commande suivante :

```
exit
```

## Installation de l'application web contenant l'API

### Paramétrage du fichier de configuration

A la racine du projet, vous trouverez un fichier nommé '.env.example'. Copiez le avec le nom '.env' à l'aide de la commande suivante :

```
cp .env.exemple .env
```

Puis, dans le fichier '.env' modifiez les paramètres suivants :

**APP\_URL**=lien qui sera utilisé pour accéder à l'application.

**DB\_DATABASE**=nom de la base ('cci-appro' par défaut, défini auparavant).

**DB\_USERNAME**=nom d'utilisateur ('laravel' par défaut, défini auparavant).

**DB\_PASSWORD**=mot de passe de l'utilisateur (défini auparavant).

### Installation des packages

Exécutez les commandes suivantes dans le dossier racine de l'application.

Afin de récupérer les packages php entrez la commande suivante :

```
composer install
```

Pour récupérer les packages javascript entrez la commande suivante :

```
npm install
```

Pour compiler le javascript, entrez la commande suivante :

```
npm run build
```

### Installation de la base de données

Afin de créer les tables entrez les commandes suivantes :

```
php artisan migrate:install
```

```
php artisan migrate
```

## Changement des droits et propriétaires de fichiers

Positionnez-vous à la racine de l'application à l'aide de la commande 'cd'

Affectez l'utilisateur d'apache aux fichiers et dossiers nécessaires à l'aide de la commande suivante :

```
sudo chown -R www-data:www-data public storage/logs storage/app  
storage/framework
```

Affectez les droits nécessaires aux fichiers et dossiers avec la commande suivante :

```
sudo chmod -R 775 public storage/logs storage/app  
storage/framework
```

## Configuration d'apache

Accédez au dossier de configuration d'apache à l'aide de la commande suivante :

```
cd /etc/apache2
```

Une fois dans le dossier de configuration, copiez le fichier d'exemple de configuration avec la commande suivante :

```
sudo cp sites-available/000-default.conf sites-available/ci-appro.conf
```

Dans le fichier copié, ajoutez les paramètres suivants :

```
ServerAdmin Nom de domaine de l'application.  
DocumentRoot <chemin absolu de l'application>/public/  
DirectoryIndex index.php  
  
<Directory <chemin absolu de l'application>/public/>  
    Options Indexes FollowSymLinks MultiViews  
    AllowOverride All  
    Require all granted  
</Directory>
```

Le fichier doit ressembler à ça :

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    ServerName n-w3.ddns.net
    DocumentRoot /var/www/cci-appro/public/
    DirectoryIndex index.php

    <Directory /var/www/cci-appro/public>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Par la suite, entrez la commande suivante pour activer la configuration :

```
sudo a2ensite cci-appro.conf
```

Puis entrez les commandes suivantes afin d'activer les modules et configuration d'apache :

```
sudo a2enmod proxy_fcgi setenvif
```

```
sudo a2enconf php7.3-fpm
```

Enfin, entrez la commande suivante pour qu'apache prenne en compte les modifications :

```
sudo systemctl reload apache2
```

# Installation du frontend

## Pré-requis

Pour pouvoir compiler l'application il vous faudra ces applications :

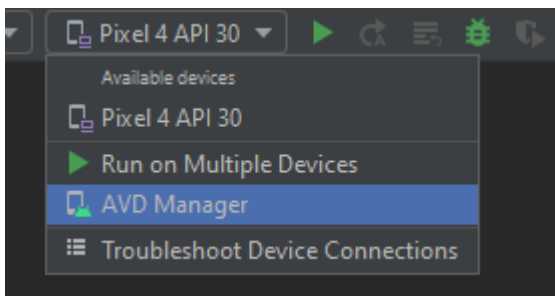
- Android Studio
  - Emulateur, à installer via Android Studio.
- node.js
  - ionic, à installer via npm
  - capacitor, à installer via npm

## Installation d'Android studio

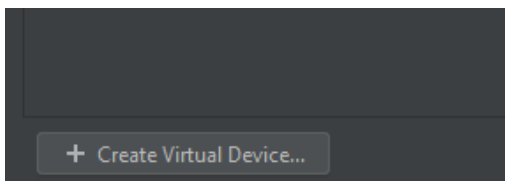
Depuis [le site d'Android Studio](#), téléchargez le logiciel. Puis exécutez-le.

Une fois sur l'interface du logiciel, créez un émulateur en suivant cette procédure :

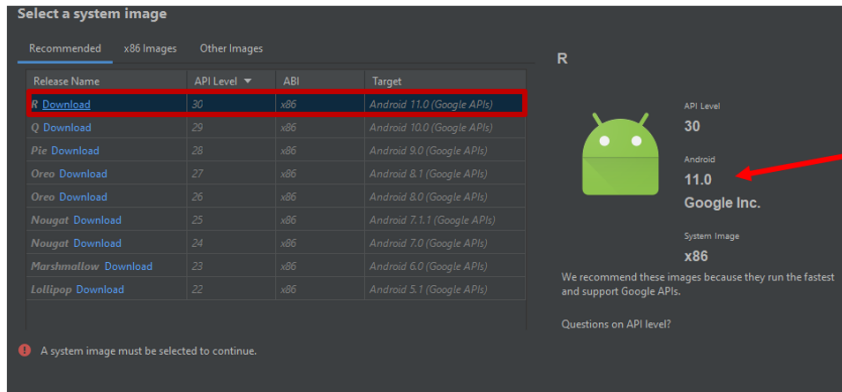
- Depuis la bar d'outils, cliquez sur le menu déroulant, puis sélectionnez « AVD Manager »



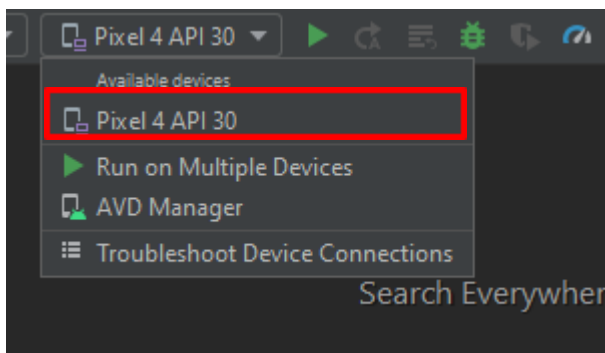
- Sur l'écran qui s'est ouvert, sélectionnez le bouton ci-dessous.



- Dans l'onglet « Phone » sélectionnez l'émulateur « Pixel 2 » (La plupart des émulateurs sont également compatibles).
- Ensuite sélectionnez la version 11 du système d'exploitation, en cliquant sur « Download ».



- Une fois le système d'exploitation téléchargé, sélectionnez « Next », puis « Finish ».
- Vous devriez apercevoir l'émulateur dans le menu déroulant.



## Installation de node.js

Installer node.js via [ce lien](#), la version LTS est conseillée.

## Variables d'environnement

Accédez à l'interface de gestion des variables d'environnement.

Si vous n'avez pas de variable d'environnement nommée « ANDROID\_HOME », créez la avec le chemin suivant, à adapter avec votre utilisateur.

- C:\Users\\AppData\Local\Android\Sdk

Ensuite, dans les variables d'environnement, vous devriez avoir le chemin suivant :

- C:\Program Files\nodejs\node\_modules\npm

Si vous ne l'avez pas ajoutez-le.



## Installation du projet

Une fois node.js installé et les variables d'environnement vérifiées, dans le dossier du projet, exécutez les commandes suivantes :

- `npm install -g @ionic/cli`
  - Installe l'interface de commande d'ionic
- `npm install @capacitor/core @capacitor/cli`
  - Installe capacitor
- `npm install`
  - Installe les dépendances du projet
- `ionic build`
  - Compile l'application
- `npx cap add android`
  - Prépare le dossier android

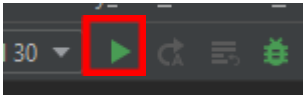
## Lancement de l'application

Pour lancer l'application exécutez la commande suivante :

- `ionic cap build android`

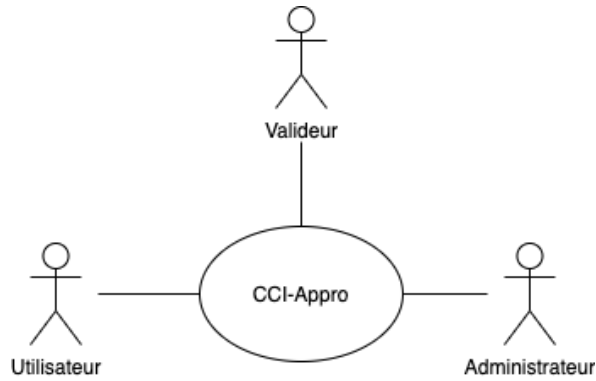
Elle va compiler l'application pour Android et ouvrir Android Studio.

Sur Android Studio, une fois que tous les processus se sont exécutés, cliquez sur le bouton encadré ci-dessous. L'émulateur va alors se lancer et ouvrir l'application.

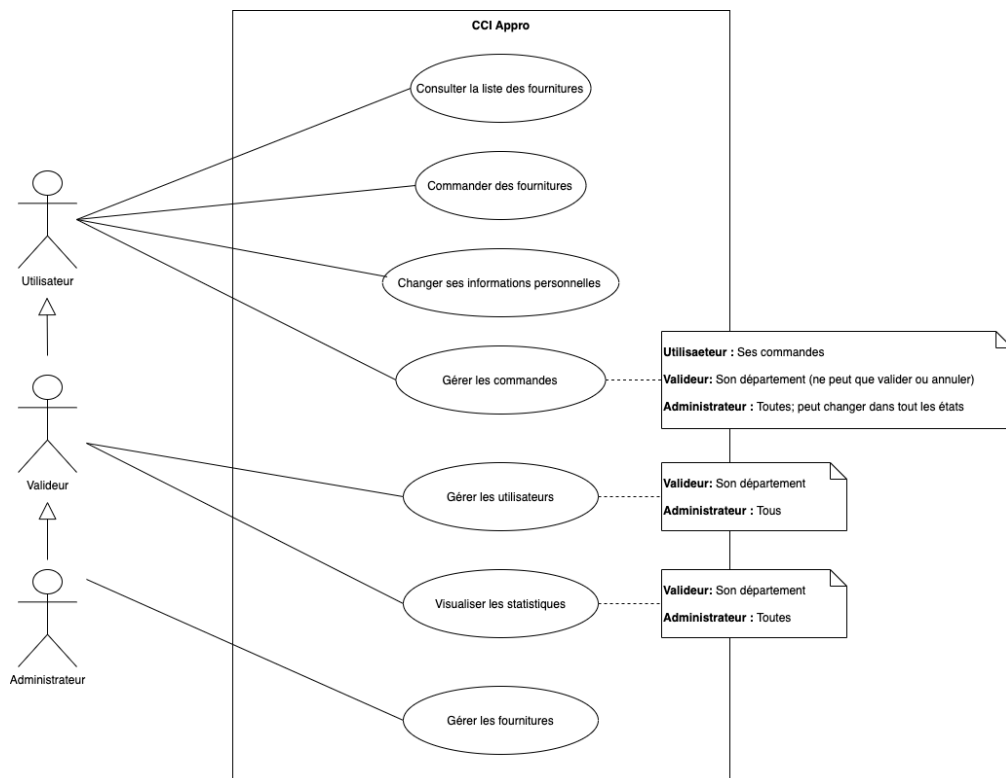


# Diagrammes

## Diagramme de contextualisation



## Diagramme de cas d'utilisation



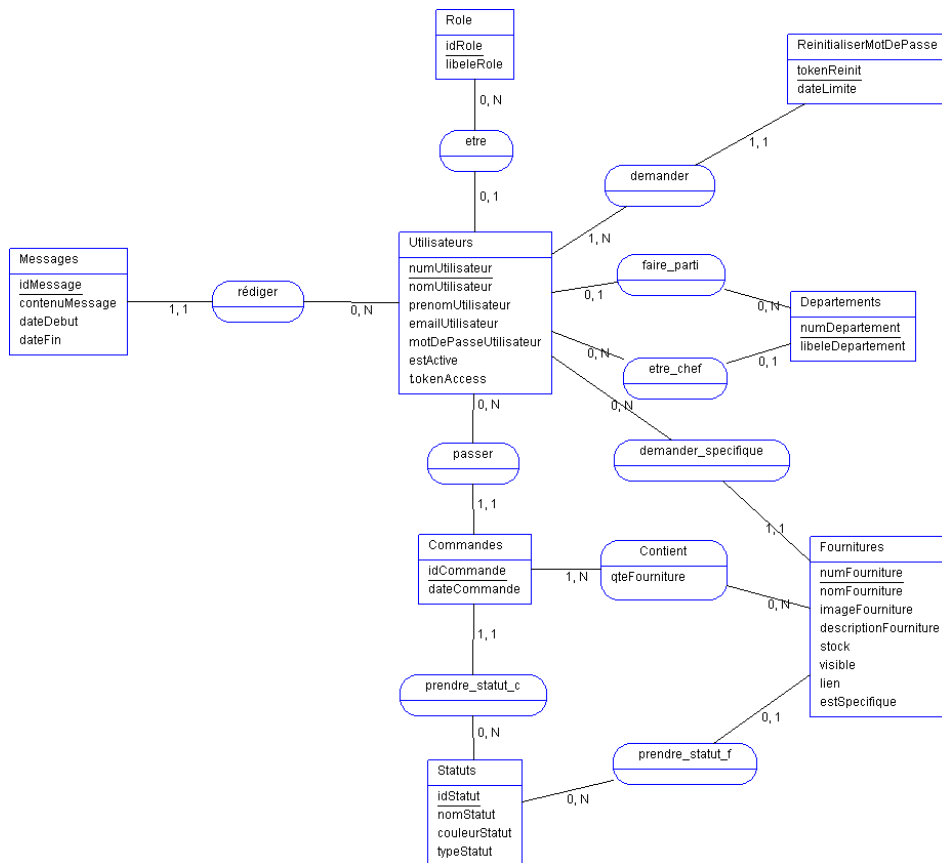
# Les données

## Bibliothèque de données

Nom du champs	Description	Type de champs / taille	Commentaire
<b>Utilisateurs</b>			
numUtilisateur	Numéro de l'utilisateur	E	Auto Incrémenté
nomUtilisateur	Nom de l'utilisateur	A=30	
prenomUtilisateur	Prénom de l'utilisateur	A=30	
emailUtilisateur	E-mail de l'utilisateur	A=120	
motDePasseUtilisateur	Mot de passe de l'utilisateur	A	Chiffré
tokenAccesUtilisateur	Token d'accès via l'API	A=80	Chiffré
estActive	Compte activé ou non	Booléen	
estAdministrateur		Booléen	
<b>Départements</b>			
numDepartement	Numéro du département	E	
libelleDepartement	Libellé du département	A=30	
<b>Fournitures</b>			
idFourniture	Id unique du Fourniture	E	Auto incrémenté
nomFourniture	Nom de la Fourniture	A=150	
imageFourniture	Nom de l'image avec extension	A=100	
descriptionFourniture		A	
stock	Stock des Fournitures	E	
visible	Visibilité des Fournitures	Booléen	
lien	Lien vers ressource externe	A	
estSpecifique	Demande (Fourniture) spécifique		Sera une vue sur MySQL

Commandes			
idCommande	Id unique de la commande	E	Auto incrémenté
dateCommande	Date et heure de la création de la commande	Date et heure	
Statut_commandes			
idStatutCommande	Id unique du Statut_commande	E	
nomStatutCommande	Nom du Statut_commande	A=30	
couleurStatutCommande	Couleur du statut	A=7	
Messages			
idMessage		E	
contenuMessage		A	
dateDebut		Date et heure	
dateFin		Date et heure	

### Modèle conceptuel des données



## Endpoints de l'API

Ressources	Description	URI	Méthode	Paramètres
Supplies	Liste de toutes les fournitures	/supplies	GET	page, search
	Récupérer une fourniture	/supply/{slug}	GET	
	Ajouter une fourniture	/supply	POST	image, name, stock, description, is_visible
	Mettre à jour une fourniture	/supply/{id}	PUT	image, name, stock, description, is_visible
	Supprimer une fourniture	/supply	DELETE	supply_id
Users	Récupérer l'utilisateur courant	/user	GET	
Departments	Récupérer tous les départements (pour l'inscription)	/departments/offline	GET	
	Récupérer tous les départements	/departments	GET	page, search
	Récupérer un département	/department/{id}	GET	
	Ajouter un département	/department	POST	label, user_id
	Mettre à jour un département	/department/{id}	PUT	label, user_id
	Supprimer un département	/department	DELETE	department_id
Cart	Récupérer le panier de l'utilisateur courant	/user/cart	GET	
	Ajouter une fourniture au panier	/user/cart	POST	supply_id, quantity
	Ajouter une fourniture spécifique au panier	/user/cart/specific	POST	supply_names, supply_urls

	Mettre à jour la quantité d'une fourniture	/user/cart	PUT	supply_id, quantity
	Supprimer une fourniture du panier	/user/cart	DELETE	supply_id
	Valider le panier	/user/cart/validate	GET	
Statuses	Récupérer la liste des statuts	/statuses	GET	
	Récupérer un statut	/status/{id}	GET	
Orders	Liste des commandes de l'utilisateur courant	/user/orders	GET	page
	Liste de toutes commandes	/orders	GET	page
	Récupérer une commande	/order/{id}	GET	
	Mettre à jour une commande	/order/{id}	PUT	status_<id_supply>
Auth	Enregistrer un utilisateur	/auth/register	POST	lastname, firstname, email, department_id, password, password_confirmation
	Se connecter	/auth/login	POST	email, password
	Confirmer l'email d'un utilisateur	/auth/confirm-email	POST	token
	Demande de réinitialisation de mot de passe	/auth/password/forgot	POST	email
	Réinitialiser le mot de passe	/auth/password/reset	POST	token, password, password_confirmation