

## Setup database

Connettersi al database postgres come utente postgres ed eseguire i seguenti comandi SQL

```
create role groupmember with createdb createrole login password
'groupmember';
set role groupmember;
create database "PCTO";
```

Dopodiché, connettersi al database PCTO come groupmember ed eseguire il codice SQL dei file sotto docs/SQL nel seguente ordine: tables.sql, triggers.sql, permissions.sql

Se lo si desidera sono disponibili delle query SQL sotto docs/SQL/sampleData.sql per generare dei dati fittizi.

## Setup server

### Virtualenv

Dalla shell, navigare nella root directory del progetto eseguire i seguenti comandi bash

```
pip install virtualenv
virtualenv venv
source venv/bin/activate
pip install -r utilities/requirements.txt
```

### Variabili d'ambiente

Tutte le variabili d'ambiente necessarie sono contenute sotto utilities/env.txt

Le seguenti sono già settate di default:

- FLASK\_KEY\_PATH: il file contenente la flask secret key
- DB\_HOST: il dominio o l'indirizzo del server che hosta il database
- DB\_PORT: la porta attraverso la quale il DB\_HOST accetta le richieste al database

## Zoom

Per testare l'applicazione con questa feature è necessario [creare un'applicazione OAuth](#).

Finché l'applicazione non viene [pubblicata](#) sul [marketplace di zoom](#), tutte le feature fornite dall'[API di zoom](#) sono limitate al solo account del developer che possiede l'applicazione, pertanto la gestione dei meeting per le lezioni è possibile solo se l'email del docente che le organizza è effettivamente l'email del developer. Affinché l'applicazione possa comunicare con l'API di zoom, essa deve essere accessibile da remoto tramite opportuno port-forwarding.

### Configurazione OAuth app

Sotto la tab *App Credentials*, impostare l'attributo *Redirect URL for OAuth* all'indirizzo **pubblico** completo mappato sull'endpoint `/zoom_auth_code`

Sotto la tab *Scopes* includere 'View and manage all user meetings'

## Variabili d'ambiente

Le seguenti variabili d'ambiente sono necessarie per runnare l'app con il supporto per zoom e vanno settate **manualmente**

- `ZOOM_CLIENT_ID`: il codice identificativo dell'applicazione del marketplace di zoom
- `ZOOM_CLIENT_SECRET`: il codice segreto dell'applicazione
- `ZOOM_REDIRECT_URI`: l'indirizzo **pubblico** completo mappato sull'endpoint `/zoom_auth_code` usato dall'API di zoom per passare il codice di autorizzazione dell'utente nel momento in cui autorizza l'applicazione all'uso dei suoi dati.

## Avvio del server

I seguenti comandi sono da eseguire dalla root directory del progetto.

Dopo aver configurato le variabili d'ambiente sotto `utilities/env.txt`, attivarle eseguendo

```
source utilities/env.txt
```

Per avviare l'applicazione si distinguono i seguenti casi:

Per l'applicazione base (senza zoom):

```
python app/app.py
```

Per l'applicazione con zoom:

```
python app/app.py --zoom
```

Zoom richiede una connessione sicura, se non si possiede un certificato, avviare con:

```
python app/app.py --zoom --adhoc
```