

Instrukcja instalacji aplikacji (lokalnie)

Poniższa instrukcja opisuje proces instalacji aplikacji lokalnie. Przedstawione kroki dotyczą instalacji na systemie Windows, jednak dla systemów Linuxowych wygląda to bardzo podobnie. Podane kursywą komendy wykonujemy w cmd / terminalu.

1. Instalacja wymaganego oprogramowania

- a. Python (<https://www.python.org/>)

2. Instalacja pomocnicznego oprogramowania

- a. XAMPP – przydatne narzędzie do zarządzania usługami sieciowymi, m.in. MySQL i Apache, których uruchomienie będzie potrzebne do działania aplikacji. (<https://www.apachefriends.org/>)

3. Instalacja opcjonalnego oprogramowania, acz niezbędnego do obsługi funkcji asynchronicznego i okresowego wysyłania maili. Aplikacja włączy się bez tego, ale nie będzie w stanie wysyłać powiadomień mailowych

- a. Erlang OTP (<https://www.erlang.org/>)
- b. RabbitMQ (<https://www.rabbitmq.com/>)

4. Pobieramy kod źródłowy

- a. Pobieramy kod źródłowy aplikacji i umieszczamy go w wybranej lokalizacji. Wykorzystać można narzędzia takie jak wget, git clone, wrzucić plik przez FTP, pendriva lub pobrać ZIP repozytorium z GitHuba i wypakować.

5. Utworzenie wirtualnego środowiska

python -m venv <nazwa>

6. Uruchomienie stworzonego środowiska

<nazwa>\Scripts\activate

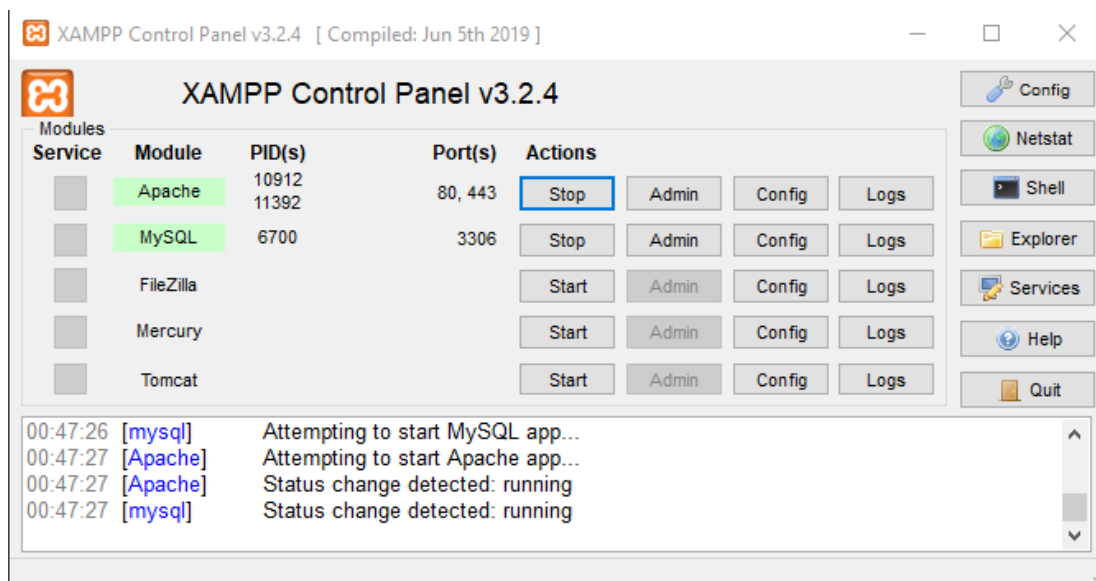
7. Instalacja wymaganych bibliotek

- a. Przechodzimy do głównego katalogu aplikacji (organize-application\scheduler_project) i wydajemy komendę:

```
pip install -r requirements.txt
```



8. Stworzenie pustej bazy danych

- a. W celu utworzenia pustej bazy przechodzimy do programu XAMPP i uruchamiamy usługi MySQL oraz Apache



- b. Przechodzimy pod adres <http://localhost/phpmyadmin/> (domyślnie) i tworzymy nową bazę o nazwie scheduler_database

Bazy danych

 Utwórz bazę danych 

9. Zainicjowanie bazy danych

- a. Stworzoną przed chwilą bazę trzeba wypełnić danymi niezbędnymi do poprawnego funkcjonowania aplikacji. Robimy to za pomocą komend:

```
python manage.py makemigrations  
python manage.py migrate
```

10. Uruchomienie aplikacji

- a. W zasadzie aplikacja jest już gotowa do uruchomienia i możemy to zrobić. Instalację i konfigurację komponentów odpowiedzialnych za powiadomienia mailowe możemy przeprowadzić później. Aplikację uruchamiamy za pomocą poniższej komendy. Istotne jest wybranie nieużywanego portu, pod którym aplikacja będzie nasłuchiwać zapytań.

```
python manage.py runserver 0.0.0.0:<port>
```

- b. Poprawne uruchomienie aplikacji powinno być zwieńczone komunikatem

```
June 11, 2021 - 01:01:51  
Django version 3.2, using settings 'scheduler_project.settings'  
Starting development server at http://0.0.0.0:80/  
Quit the server with CTRL-BREAK.
```

- c. Po ujrzeniu tego komunikatu możemy przejść na zawartą w nich stronę i rozpocząć korzystanie z aplikacji. Chcąc uzyskać dostęp do aplikacji spoza lokalnego systemu, należy odblokować odpowiedni port w firewallu.

Domyślne dane do logowania admina to **superuser:super**. Po zainicjowaniu aplikacji zalecane jest zmienienie hasła w zakładce Profil > Zmiana hasła.

11. Uruchomienie środowiska do wysyłania powiadomień mailowych – i innych ewentualnych operacji asynchronicznych/periodycznych w kodzie aplikacji.

- a. Uruchamiamy brokera RabbitMQ poprzez uruchomienie skryptu `rabbitmq-server.bat`

domyślna lokalizacja: `C:\Program Files\RabbitMQ Server\rabbitmq_server-3.8.16\sbin`

- b. Możliwe, że w procesie instalacji został już uruchomiony proces aplikacji `erl`, który będzie blokował utworzenie nowej instancji RabbitMQ. Przechodzimy wówczas do menedżera zadań i zatrzymujemy proces o nazwie `erl` i ponownie uruchamiamy plik `rabbitmq-server.bat`. Operacja powinna zakończyć się sukcesem:

```
## ##      RabbitMQ 3.8.16
## ##
##### Copyright (c) 2007-2021 VMware, Inc. or its affiliates.
##### ##
##### Licensed under the MPL 2.0. Website: https://rabbitmq.com

Doc guides: https://rabbitmq.com/documentation.html
Support:    https://rabbitmq.com/contact.html
Tutorials:  https://rabbitmq.com/getstarted.html
Monitoring: https://rabbitmq.com/monitoring.html

Logs: c:/Users/Administrator/AppData/Roaming/RabbitMQ/log/rabbit@44F4390641AAA36.log
      c:/Users/Administrator/AppData/Roaming/RabbitMQ/log/rabbit@44F4390641AAA36_upgrade.log

Config file(s): c:/Users/Administrator/AppData/Roaming/RabbitMQ/advanced.config

Starting broker... completed with 0 plugins.
```

- c. Uruchamiamy dwa kolejne okienka `cmd` lub terminala i przechodzimy w nich do głównego katalogu aplikacji – tam, gdzie mamy plik `manage.py`. Następnie w jednym z okienek wydajemy komendę:

`celery -A scheduler_project worker -l info -pool=solo`

```

----- celery@44F4390641AAA36 v5.0.5 (singularity)
-----
*****
***** Windows-10-10.0.14393-SP0 2021-06-09 16:11:07
*** --- * ---
** ----- [config]
** ----- .> app: scheduler_project:0x16e9601e6d0
** ----- .> transport: amqp://guest:**@localhost:5672//
** ----- .> results: disabled://
*** --- * --- .> concurrency: 1 (solo)
***** ----- .> task events: OFF (enable -E to monitor tasks in this worker)
*****
----- [queues]
. > celery exchange=celery(direct) key=celery

[tasks]
. schedule.tasks.send_email_organizer
. schedule.tasks.send_mail_register
. schedule.tasks.send_notification_all
. schedule.tasks.send_notification_organizer
. schedule.tasks.send_poll_notification
. schedule.tasks.send_poll_notification_cron

[2021-06-09 16:11:07,316: INFO/MainProcess] Connected to amqp://guest:**@127.0.0.1:5672//
[2021-06-09 16:11:07,379: INFO/MainProcess] mingle: searching for neighbors
[2021-06-09 16:11:08,441: INFO/MainProcess] mingle: all alone
[2021-06-09 16:11:08,477: WARNING/MainProcess] c:\users\administrator\comarch\lib\site-pa
Warning: Using settings.DEBUG leads to a memory
leak, never use this setting in production environments!
warnings.warn('Using settings.DEBUG leads to a memory
[2021-06-09 16:11:08,477: INFO/MainProcess] celery@44F4390641AAA36 ready.

```

- d. W drugim okienku wydajemy komendę dla kolejnego niezbędnego komponentu, czyli celery beat:

celery -A scheduler_project beat -l INFO

```

celery beat v5.0.5 (singularity) is starting.
-----
LocalTime -> 2021-06-11 01:18:14
Configuration ->
. broker -> amqp://guest:**@localhost:5672//
. loader -> celery.loaders.app.AppLoader
. scheduler -> celery.beat.PersistentScheduler
. db -> celerybeat-schedule
. logfile -> [stderr]@%INFO
. maxinterval -> 5.00 minutes (300s)
[2021-06-11 01:18:14,767: INFO/MainProcess] beat: Starting...

```

- e. W samej aplikacji natomiast należy zdefiniować odpowiednie ustawienia serwera SMTP, z którego mają być wysyłane powiadomienia. Można to zrobić po zalogowaniu się na konto admina w zakładce Ustawienia

