Michał Sobieralski

Informatyka Stosowana

st. stacjonarne, sem. VII

Konfiguracja serwera

Żeby aplikacje mogły działać, należy zainstalować aplikację sieciową na serwerze wraz z niezbędnymi technologiami. *Na potrzeby projektu został wynajęty serwer z systemem operacyjnym Ubuntu 16.04 LTS, choć aplikacja nie jest ograniczona wyłącznie do tej dystrybucji Linuxa* [1]. Pierwszym krokiem w celu poprawnej konfiguracji serwera jest zainstalowanie wszystkich technologii wymienionych w podrozdziale 2.2. pracy dyplomowej inżynierskiej:

* Apache/2.4.39
  + rozszerzenie mod\_php
* PHP 7.3
  + rozszerzenie gd
  + rozszerzenie mbstring
  + rozszerzenie pgsql (jeśli baza PostgreSQL)
* PostgreSQL 9.5 (altrenatywnie można zainstalować inną bazę danych)
* phpPgAdmin 5.1 (opcjonalnie)
* Composer 1.7.3

Po zainstalowaniu potrzebnych programów należy utworzyć projekt szkielet aplikacji sieciowej (pusty projekt Laravel’a). W tym celu z poziomu katalogu */var/www/html* należy wywołać komendę:

sudo composer create-project laravel/laravel Courier "5.6.\*" --prefer-dist

Po utworzeniu domyślnego projektu, wewnątrz katalogu */var/www/html/Courier* należy nadpisać plik composer.json (opisujący zależności, które wchodzą w skład projektu) plikiem o tej nazwie załączonym do pracy. (Plik composer.json znajduje się wewnątrz archiwum sieciowa-28.08.2019.rar, wewnątrz katalogu Courier.)

Następnie z poziomu katalogu projektu (Courier) należy wywołać polecenie:

composer update

, w celu aktualizacji zależności projektu.

Potem należy wgrać (nadpisać, jeśli trzeba; do katalogu Courier) pliki aplikacji sieciowej załączone do pracy.

Pobierz framework Bootstrap dostarczający skompilowane pliki CSS oraz JavaScript. W tym celu przejdź na stronę <https://getbootstrap.com/>, wybierz opcję Download. W następnym oknie, pod zakładką: "Compiled CSS and JS" - wybierz opcję Download.

Zawartość pobranego archiwum wypakuj do katalogu *./Courier/public*.

Żeby serwer HTTP miał dostęp do plików aplikacji, należy nadać katalogom odpowiednie uprawnienia. Z poziomu katalogu projektu *Courier* należy wykonać polecenia:

sudo chown -R root:www-data storage

sudo chown -R root:www-data bootstrap/cache

sudo chown -R root:www-data storage/app/barcodes

sudo chown -R root:www-data storage/app/signatures

chmod -R 775 storage

chmod -R 775 bootstrap/cache

chmod -R g+w storage/app/barcodes

chmod -R g+w storage/app/signatures

Utworzona aplikacja sieciowa działa w protokole HTTPS, co oznacza, że do jej obsługi należy zainstalować i skonfigurować na serwerze certyfikat SSL.

Żeby uruchomić aplikację bez certyfikatu SSL, należy w pliku *../Courier/app/Http/Kernel.php* dodać komentarz na początku następującej linii:

//\App\Http\Middleware\HttpsProtocol::class,

Niektóre funkcjonalności aplikacji korzystają z zewnętrznych API. Wyznaczanie optymalnej trasy jest możliwe dzięki Openrouteservice API, a komunikacja oraz powiadomienia mailowe korzystają ze skrzynki pocztowej użytkownika. Aby serwis mógł poprawnie działać, należy utworzyć konto i pobrać darmowy klucz do API z Openrouteservice oraz utworzyć skrzynkę pocztową (np. na Gmailu).

Poza tym, aplikacja sieciowa nie będzie działać, bez odpowiedniej konfiguracji zainstalowanych programów. Poniższe kroki pozwolą rozwiązać ten problem.

Po pierwsze - w pliku */etc/php/7.3/apache2/php.ini* należy znaleźć i ustawić następujące zmienne:

memory\_limit = 256M

cgi.fix\_pathinfo=0

upload\_max\_filesize = 64M

oraz odkomentować następujące linie (tzn. usunąć średnik) jeśli używana baza to PostgreSQL:

extension=pdo\_pgsql

extension=pgsql

Po drugie - należy skonfigurować ustawienia Apache2, by móc wyświetlić stronę serwisu. W katalogu */etc/apache2/sites-available/* należy umieścić załączone do pracy pliki *site.conf* oraz *site-ssl.conf* (jeśli zainstalowano certyfikat SSL), a następnie edytować ich zawartość poprzez zastąpienie [NAWIASÓW KWADRATOWYCH] poprawnymi danymi. Jeśli zainstalowano phpPgAdmin, w katalogu */etc/apache2/conf-available/* należy nadpisać plik *phppgadmin.conf* plikiem załączonym do pracy.

Następnie w celu umieszczenia strony na serwerze należy w terminalu wykonać komendy:

sudo a2enmod ssl (jeśli mamy zainstalowany certyfikat SSL)

sudo a2dissite 000-default.conf

sudo a2dissite default-ssl.conf

sudo a2ensite site.conf

sudo a2ensite site-ssl.conf (jeśli mamy zainstalowany certyfikat SSL)

sudo a2enconf phppgadmin.conf

sudo a2enmod rewrite

sudo systemctl reload apache2

Kolejnym krokiem jest utworzenie oraz dostosowanie bazy danych. Konieczne będzie nadanie hasła użytkownikowi postgres:

su - postgres

psql

\password postgres

[PODAJ WŁASNE HASŁO]

[POWTÓRZ HASŁO]

... oraz utworzenie bazy danych serwisu:

CREATE DATABASE "ParcelDB" template template0;

\q

exit

Konfigurację PostgreSQL dokonuje się poprzez zmianę danych w plikach:

1) */etc/postgresql/9.5/main/postgresql.conf:*

Należy ustawić pole:

listen\_addresses = '\*'

2) */etc/postgresql/9.5/main/pg\_hba.conf:*

Należy na koniec pliku dodać linię:

host all all 0.0.0.0/0 md5

Jeśli zainstalowaliśmy phpPgAdmin, należy zmienić również linie w pliku */etc/phppgadmin/config.inc.php*:

$conf['servers'][0]['desc'] = 'PostgreSQL';

$conf['servers'][0]['host'] = 'localhost';

$conf['servers'][0]['port'] = 5432;

$conf['servers'][0]['sslmode'] = 'allow';

$conf['servers'][0]['defaultdb'] = 'ParcelDB';

$conf['extra\_login\_security'] = false;

Po wprowadzeniu tych zmian, trzeba zrestartować i aktywować serwer HTTP oraz bazę danych:

systemctl restart postgresql

systemctl restart apache2

systemctl enable postgresql

systemctl enable apache2

Na koniec z poziomu katalogu projektu *Courier* należy wywołać komendy:

php artisan migrate

composer dump-autoload

php artisan db:seed

php artisan passport:client --personal

[WCIŚNIJ ENTER POZOSTAWIAJĄC PUSTE POLE]

w celu utworzenia tabel w bazie danych oraz wypełnienia ich przykładowymi danymi.

Ostatnim krokiem jest konfiguracja zmiennych środowiskowych projektu. Z poziomu katalogu projektu *Courier* należy wywołać polecenie:

php artisan key:generate

Wygeneruje ono nowy unikalny klucz aplikacji APP\_KEY, służący do enkrypcji ciasteczek [2].

Z poziomu tego samego katalogu trzeba ustawić w pliku *.env* (katalogu *Courier*) następujące zmienne:

* APP\_URL - Domena dla aplikacji webowej (https://demotspservice.pl);
* DB\_CONNECTION - Jedno ze zdefiniowanych w *config/database.php* połączeń z odpowiednią bazą danych (pgsql);
* DB\_HOST - Adres IP serwera z bazą danych;
* DB\_PORT - Port wykorzystywany przez bazę danych (domyślny dla PostgreSQL: 5432);
* DB\_DATABASE - Nazwa bazy danych, którą wykorzystuje aplikacja (ParcelDB);
* DB\_USERNAME - Nazwa użytkownika łączącego się z bazą danych (postgres);
* DB\_PASSWORD - Hasło tego użytkownika;
* MAIL\_DRIVER - Protokół poczty (smtp);
* MAIL\_HOST - Host poczty (dla Gmail: smtp.googlemail.com);
* MAIL\_PORT - Port poczty (dla Gmail: 465);
* MAIL\_USERNAME - Nazwa użytkownika (dla Gmail: adres email użytkownika);
* MAIL\_PASSWORD - Hasło użytkownika do poczty;
* MAIL\_ENCRYPTION - Szyfrowanie poczty (ssl);
* ORS\_KEY - Klucz do Openrouteservice API (niezbędny do realizacji algorytmu wyznaczania optymalnej trasy).

To już wszystko, wystarczy uruchomić przeglądarkę i wpisać adres domeny, by skorzystać z aplikacji webowej.

Można zalogować się na konto administratora (login: admin; hasło: admin) lub kuriera (login: courier; hasło: courier).

W celu pobrania aplikacji mobilnej należy zalogować się do aplikacji sieciowej i wybrać opcję „Pobierz aplikację mobilną”. Uzyskany w ten sposób plik z rozszerzeniem .apk należy wgrać na urządzenie z systemem operacyjnym Android o minimalnej wersji 4.0.

Literatura:

1) Cytaty zaczerpnięte z pracy dyplomowej inżynierskiej (podrozdział 2.2.).

2) J. Bathman, APP\_KEY And You, https://tighten.co/blog/app-key-and-you [Data uzyskania dostępu: 03.07.2019].