Wielowarstwowy system rekrutacji dla szkół z interfejsem webowym i aplikacją mobilną - instrukcja instalacji systemu

Andrzej Westfalewicz, Filip Zyskowski

7 stycznia 2019

Spis treści

1 Wstęp

Niniejszy dokument zawiera opisy instalacji potrzebne do wdrożenia wielowarstwowego systemu rekrutacji dla szkół z interfejsem webowym i aplikacją mobilną. Konfiguracja i instalacja kolejnych elementów systemu są opisane w poszczególnych punktach. Każdy opis procesu został opisany w prosty, przystępny sposób, a ukończenie wszystkich punktów z tej instrukcji praktycznie pozwoli na uruchomienie całego systemu. W instrukcji wskazane są również elementy, które nie zostały tutaj opisane, ponieważ jest wiele sposobów na ich implementację, bądź nie zależą od kodu systemu - te punkty również powinny zostać wykonane, by cieszyć się pełną funkcjonalnością i wygodą systemu.

2 Aplikacja mobilna

Zanim użytkownicy będą mogli skorzystać z aplikacji mobilnej, trzeba ją najpierw skompilować tzn. zamienić kod źródłowy z folderu /src/RecruitMe.MobileApp do jednego pliku, a następnie zainstalować na urządzeniu. Cały proces składa się z paru kroków, które zostaną przedstawione poniżej.

2.1 Kompilacja aplikacji do pliku APK

Do procesu kompilacji aplikacji wymagane jest zainstalowanie co najmniej kilku narzędzi programistycznych, w tym Node.js oraz NativeScript CLI. Pełna instrukcja instalacji wymaganych rzeczy na stronie NativeScript Docs w artykule *CLI Setup* (https://docs.nativescript.org/start/quick-setup#full-setup). Interesuje nas podpunkt **Full Setup**.

Po przejściu całego poradnika instalacji potrzebnych narzędzi możemy przejść do procesu kompilacji.
Aby tylko skompilować aplikację lokalnie, by sprawdzić na swoim telefonie jak działa aplikacja, uruchamiamy okno konsoli w folderze src/RecruitMe.MobileApp i wpisujemy:

```
tns run android
```

Proces kompilacji może potrwać nawet kilka minut i zakończy się automatyczną instalacją oraz uruchomieniem aplikacji na podłączonym do komputera urządzeniu z systemem Android lub do wirtualnej maszyny Androida zainstalowanej wcześniej. Jeżeli chcemy tylko otrzymać plik APK, w ekran konsoli wpisujemy:

```
tns build android
```

W obu przypadkach w folderze src/RecruitMe.MobileApp/platforms/android/app/build/outputs/apk/debug/ znajdziemy plik APK naszej aplikacji w formacie debug.

Jeżeli chcemy skompilować naszą aplikację do pliku, który będzie można potem rozpowszechniać w sklepach z aplikacjami (np. Google Play), należy

przeprowadzić proces kompilacji w trybie *release*. Aby to zrobić, potrzebny jest certyfikat - klucz prywatny potwierdzający, że to my jesteśmy właścicielami naszej aplikacji. Szczegółowe instrukcje dotyczące tego jak taki certyfikat stworzyć, znajdują się w artykule *Publishing for Android* na stronie NativeScript Docs (https://docs.nativescript.org/tooling/publishing/publishing-android-apps).

Po stworzeniu certyfikatu możemy przystąpić do kompilacji w trybie release. Uruchamiamy okno konsoli w folderze src/RecruitMe.MobileApp i wpisujemy:

```
tns build android --release
    --key-store-path <ścieżka-do-certyfikatu>
    --key-store-password <hasło-certyfikatu>
    --key-store-alias <alternatywna-nazwa-certyfikatu>
    --key-store-alias-password <hasło-altern.-nazwy-certyfikatu>
```

Wszystkie cztery wartości otrzymamy po stworzeniu certyfikatu w poprzednim kroku. Stworzony plik znajdziemy w folderze src/RecruitMe.MobileApp/platforms/android/app/build/outputs/apk/release.

Uwaga: Przed procesem kompilacji aplikacji należy zainstalować najpierw serwer i otrzymać jego adres. Następnie należy otworzyć plik src/RecruitMe.MobileApp/app/services/common/apiGateway.ts w dowolnym edytorze tekstu i zmienić w linii 9. pomiędzy cudzysłowami adres, na jaki nasza aplikacja będzie wysyłała żądania. W przeciwnym wypadku proces kompilacji będzie trzeba powtórzyć już z zmienionymi danymi. Niezmienienie danych skutkuje wadliwym działaniem całej aplikacji mobilnej.

2.2 Instalacja aplikacji na urządzeniu

Aby móc korzystać z specjalnie przygotowanej aplikacji mobilnej, należy ją wcześniej zainstalować na urządzeniu z systemem operacyjnym Android. W zależności od wybranej przez Organizatorów rekrutacji metody dystrybucji aplikacji mobilnej istnieją różne sposoby instalacji tego programu.

Przekazywanie pliku APK
Organizatorzy mogą się zdecydować udostępniać we własnym zakresie aplikację mobilną na systemy Android poprzez plik APK. Zainteresowani programem użytkownicy muszą ściągnąć go na swoje telefony i kliknąć na pobrany plik, by go zainstalować. W takim wypadku użytkownicy zmuszeni są zezwolić na **instalowanie aplikacji z nieznanych źródeł**. W każdym telefonie tą opcję włącza się inaczej, więc zalecane jest wyszukanie w Internecie gdzie znajduje się ta opcja w danym modelu telefonu.

Google Play lub inny sklep z aplikacjami
By potencjalnie zwiększyć zakres odbiorców aplikacji, istnieje opcja upublicznienia aplikacji w sklepie/pach z aplikacjami. Gdy już nasza aplikacja znajdzie się w takim sklepie, musimy ją wyszukać i kliknąć przycisk *Instaluj*. Po

pobraniu i zainstalowaniu aplikacji, możemy kliknąć *Otwórz*, by od razu przejść do aplikacji.

3 Aplikacja webowa

Aplikacja webowa nie wymaga instalacji - jedynym wymaganiem jest posiadanie zainstalowanej przeglądarki internetowej. Chociaż aplikacja powinna działać na dowolnej nowoczesnej przeglądarce internetowej, zalecane jest używanie przeglądarek Google Chrome lub Mozilla Firefox, za pomocą których aplikacja była tworzona i została na nich przetestowana. Podczas testów znalezione zostały niepożądane zachowania w przypadku niektórych przeglądarek:

- **Mozilla Firefox** - Jeśli w procesie konfiguracji (opisanym poniżej) zrezygnujemy z przekierowania użytkowników na połączenie bezpieczne (HTTPS) przeglądarka, ze względów bezpieczeństwa, odmówi aplikacji dostępu do kamerki urządzenia.
- **Microsoft Edge** - Ze względu na ograniczone funkcjonalności przeglądarki niektóre funkcjonalności mogą nie działać poprawnie np.: przeglądarka nie wspiera konstruktora `new File()`, który jest używany w zapisie zdjęcia zrobionego kamerką - tym samym ta funkcjonalność nie działa poprawnie.

4 Baza danych oraz serwer API

Instalacja tych dwóch komponentów jest ze sobą bardzo powiązana. Jest ona również dość złożona, więc została ona podzielona na dwie części: konfigurację, czyli dostosowanie systemu do naszych potrzeb oraz wdrożenie czyli upublicznienie systemu.

4.1 Konfiguracja

4.1.1 Potrzebne narzędzia

Absolutnym minimum do konfiguracji aplikacji jest dowolny edytor tekstowy, jednak zalecane użycie zintegrowanego środowiska programistycznego - IDE (np.: Visual Studio, Visual Studio Code), które ułatwi zdanie i wskaże ewentualne błędy. Dodatkowo, jeśli chcemy testować stworzoną konfigurację potrzebujemy narzędzi programistycznych:

- dotnet sdk (https://dotnet.microsoft.com/download),
- node oraz npm (https://nodejs.org/en/download/),
- MySQL Server (https://www.mysql.com/downloads/).

4.1.2 Strony dla użytkowników niezalogowanych

W systemie zostały przygotowany strony ogólnodostępne: strona główna, "o szkole" i "regulamin". Ich zawartość można zmienić edytując odpowiadające im pliki w folderze src/RecruitMe.Web/ClientApp/components/staticpages. Domyślnie są one wypełnione przykładowym tekstem "lorem ipsum".

4.1.3 Treść wysyłanych maili

Aby zmienić treść wiadomości email wysyłanych do użytkowników należy zmodyfikować pola w klasie EmailContentConfiguration znajdującej się w folderze src/RecruitMe.Logic/Configuration/. Możliwe jest zmienienie tytułów i treści wszystkich maili wysyłanych przez system.

- RegisteredTitle, RegisteredBody odpowiadają tytułowi i treści wiadomości wysyłanej po zarejestrowaniu się do systemu,
- EmailVerifiedTitle, EmailVerifiedBody odpowiadają tytułowi i treści wiadomości wysyłanej po potwierdzeniu adresu email,
- LoginRemindedTitle, LoginRemindedBody odpowiadają tytułowi i treści wiadomości wysyłanej kiedy użytkownik używa funkcjonalności przypomnienia loginu
- ResetPasswordTitle, ResetPasswordBody odpowiadają tytułowi i treści wiadomości kiedy użytkownik chce zresetować hasło.

4.1.4 Certyfikat SSL

Z uwagi na dane osobowe przesyłane pomiędzy aplikacjami, połączenie musi być zabezpieczone, a użytkownicy powinni używać protokołu HTTPS przy łączeniu się z serwerem. Do aplikacji dołączony jest testowy certyfikat SSL wygenerowany na potrzeby rozwoju i testów aplikacji. Nie należy, jednak go używać w środowisku produkcyjnym. Wykupiony certyfikat należy umieścić w formacie .pfx w folderze src/RecruitMe.Web/ssl/ z nazwą aspnetapp.pfx. Hasło, którym zaszyfrowany jest plik, należy umieścić w pliku appsettings.json w folderze src/RecruitMe.Web/ pod kluczem SslCertificatePassword.

4.1.5 Pliki konfiguracyjne

W pliku appsettings.json znajdują się dane konfigurujące zachowanie aplikacji. Poza wspomnianym powyżej hasłem do certyfikatu należy zmienić:

- ConnectionString - wartość "password" należy zmienić z domyślnej ("Tester!123") na wybrane przez siebie hasło,
- UseHttpsRedirection - jeśli dostarczony certyfikat jest zaufany (tzn. jest wystawiony przez Urząd Certyfikacji, angl: CA), należy zmienić wartość na "True" - wówczas użytkownicy będą przekierowywani na połączenie

bezpieczne, nawet jeśli pierwsze połączenie było po kanale niezabezpieczonym,

- BusinessConfiguration:Email - konto poczty elektronicznej, której chcemy aby były wysyłane wiadomości do użytkowników. Potrzebny jest dostęp SMTP to wskazanego konta.
- BusinessConfiguration:EmailPassword - hasło to powyższego konta,
- BusinessConfiguration:LowestRegistrationDate - najniższa możliwa data urodzenia jaką może mieć użytkownik, aby mógł się zarejestrować w systemie,
- BusinessConfiguration:HighestRegistrationDate - najwyższa możliwa data urodzenia jaką może mieć użytkownik, aby mógł się zarejestrować w systemie,
- BusinessConfiguration:BaseAddress - adres publiczny serwera, po którym będą się łączyć użytkownicy.
- PaymentConfiguration:Id - unikatowy numer sklepu w serwisie Dotpay
- PaymentConfiguration:RegistrationFee - kwota jaką kandydaci muszą uiścić w serwisie Dotpay
- PaymentConfiguration:Currency - waluta określająca *RegistrationFee*; dostępne wartości: *PLN*, *EUR*, *USD*, *GBP*, *JPY*, *CZK*, *SEK*, *UAH*, *RON*, *NOK*, *BGN*, *CHF*, *HRK*, *HUF*, *RUB*
- PaymentConfiguration:PIN - kod PIN do unikatowego numeru sklepu - dostępny w panelu Dotpay
- PaymentConfiguration:AuthToken - token autoryzacji do konta Dotpay; to ciąg znaków w systemie *Base64*, który po odkodowaniu sprowadza się do następującego napisu: *<nazwa użytkownika>:<hasło do systemu Dotpay>* (znaki < oraz > muszą być pominięte, zostały one użyte tylko dla czytelności)
- PaymentConfiguration:UseProductionServer - parametr określający czy aplikacja do połączenia ma korzystać z serwera produkcyjnego Dotpay; wartość *false* będzie oznaczała, że wszystkie zapytania do serwisu Dotpay będą przechodziły przez serwer testowy - należy wtedy w pola **PaymentConfiguration:PIN** oraz **PaymentConfiguration:AuthToken** przekazać dane testowe

4.2 Wdrożenie

4.2.1 Potrzebne narzędzia

Do wdrażania i hostowania systemu użyty został Docker. Jest to narzędzie dzięki któremu możemy zdefiniować środowisko wykonawcze aplikacji, usuwając w ten sposób zależność od sytemu operacyjnego i zainstalowanych bibliotekach.

4.2.2 Wymagania sprzętowe

Wymagania sprzętowe są w pełni zależne od wykorzystania systemu w trakcie rejestracji - aplikacja, z której będzie korzystać 100 użytkowników ma o wiele mniejsze wymagania od aplikacji z której korzysta 3000 osób. Dlatego też zaleca się monitorowanie wolnych zasobów systemu w czasie jego pracy - należy również pamiętać, że przed kluczowymi datami, np.: koniec okresu rejestracji, ogłoszenie wyników, ruch w systemie znacznie wzrośnie a z nim zapotrzebowanie na zasoby. Dla aplikacji, których liczba użytkowników nie przekroczy 1000 sugerujemy serwer: o co najmniej 32GB wolnego miejsca na dysku twardym, 8GB pamięci RAM oraz dowolny nowoczesny procesor o co najmniej 2 rdzeniach.

4.2.3 Utworzenie sieci

Pierwszym krokiem jest utworzenie wirtualnej sieci dla kontenerów dockera. Sprowadza się to do wykonania polecenia:

```
docker network create --driver=bridge recruit-me-network
```

4.2.4 Utworzenie bazy

Następnie wewnątrz sieci uruchamiamy MySQL Server. Poniższy skrypt kolejno: pobiera obraz serwera bazy danych, startuje go wewnątrz sieci i przegląda logi w poszukiwaniu wygenerowanego hasła.

```
docker pull mysql/mysql-server
docker run -p 3306:3306 --net=recruit-me-network
    --name=rm-mysql -d mysql/mysql-server
docker logs rm-mysql 2>&1 | grep GENERATED
```

Oczekiwanym wynikiem ostatniej komendy jest wiersz podobny do poniższego zawierającego hasło. Jeśli nie otrzymujemy takiego wyniku należy odczekać chwilę, aż baza danych się uruchomi. W przypadku niepowodzenia informacje o błędach możemy sprawdzić poleceniem `docker logs rm-mysql`.

```
[Entrypoint] GENERATED ROOT PASSWORD: 6IbLUwAaKXEG[aBc0MaONz
```

Następnie logujemy się do serwera za pomocą komendy.

```
docker exec -it rm-mysql mysql -uroot -p
```

Zostaniemy zapytani o hasło, które znaleźliśmy w logach. Po zalogowaniu wykonujemy poniższy skrypt, w którym podmieniamy hasło "Tester!123" na zgodne z hasłem w appsettings.json.

```
ALTER USER 'root'@'localhost' IDENTIFIED BY 'Tester!123';
CREATE USER 'recruitme'@'%' IDENTIFIED BY 'Tester!123';
GRANT ALL PRIVILEGES ON *.* TO 'recruitme'@'%';
quit
```

W ten sposób skonfigurowana baza danych jest gotowa na połączenia. Można się do niej podłączyć używając programów typu MySQL Workbench i konfigurować według własnych potrzeb. Katalog bazy aplikacji zostanie stworzony przy pierwszym włączeniu aplikacji.

4.2.5 Serwer API

Aby uruchomić serwer API najpierw należy utworzyć jego obraz. Dzieje się to w krokach zdefiniowanych w pliku Dockerfile. Aby uruchomić proces należy będąc w folderze src/ wywołać komendę:

```
docker build -t rm-api-image .
```

Proces kompilacji może być czasochłonny, zwłaszcza za pierwszym razem. Kiedy się skończy, możemy otrzymany obraz uruchomić poleceniem:

```
docker run -d --net=recruit-me-network --name rm-api
    -p 80:80 -p 443:443 --expose=80 --expose=443 rm-api-image
```

Po chwili serwer powinien być gotowy na przyjmowanie połączeń co można sprawdzić poleceniem logs:

```
docker logs rm-api
```

Przykładowy wynik polecenia:

```
Now listening on: https://0.0.0.0:443
Now listening on: http://0.0.0.0:80
Application started. Press Ctrl+C to shut down.
```

4.2.6 Czynności po wdrożeniu

Pierwszą czynnością po wdrożeniu systemu, którą należy wykonać natychmiast po pierwszym włączeniu aplikacji jest zmiana hasła administratora z domyślnego `ZMIEN_TO_HASLO`. Dokonuje się tego analogicznie jak hasło resetują użytkownicy, email z linkiem do zmiany hasła przyjdzie na adres wskazany w appsettings.json. Po zmianie hasła można przystąpić do wprowadzania danych: nauczycieli, egzaminów i ich kategorii.

5 Elementy nie pokryte instrukcją

5.1 Publikacja aplikacji w sklepie Google Play

Organizator, chcąc upublicznić aplikację mobilną w sklepie Google Play, musi uiścić opłatę rejestracyjną konta deweloperskiego. Z tej racji tworzenie takiego konta oraz publikacja tej aplikacji nie jest pokryta w tej instrukcji.

5.2 Konfiguracja sieci serwera

Konfiguracja sieci do której podłączony jest serwer zależy w pełni od osoby instalującej system. Oczywiście, aby system był użytkowane przez potencjalnych kandydatów należy wykupić domenę internetową, aby ukryć adres IP przed użytkownikiem.
Konfiguracja sieci również ma bardzo duży wpływ na bezpieczeństwo systemu. Zaleca się skonfigurowanie jej tak, aby baza danych nie była dostępna spoza sieci lokalnej. Dodatkowo, aby zapewnić płynne działanie systemu, powinno się zastosować usługi blokujące ataki typu odmowy usługi (DDoS) lub brute force.

5.3 Założenie konta produkcyjnego Dotpay

Do poprawnego działania systemu, administrator jest zobowiązany do założenia konta produkcyjnego w serwisie Dotpay. Samodzielne utworzenie konta zwiększa bezpieczeństwo i zmniejsza prawdopodobieństwo dostania się go w niepowołane ręce. Administrator nadal jest zobowiązany dostarczyć do aplikacji wszystkie potrzebne dane, jakie posiada właściciel konta w serwisie Dotpay.

6 Historia dokumentu

Autor	Data	Wersja	Wprowadzone zmiany
Andrzej Westfalewicz	04.01.2020	v0.1	Pierwsza wersja dokumentu
Andrzej Westfalewicz	05.01.2020	v0.1	Poprawki pierwszej wersji dokumentu
Andrzej Westfalewicz, Filip Zyskowski	06.01.2020	v0.1	Dodanie informacji o integracji z Dotpay i o aplikacji mobilnej