

Apache Cordova – aplikacje hybrydowe dla systemu Android

Apache Cordova pojawił się w 2008 roku i od razu zdobył szersze zainteresowanie entuzjastów i wielkich korporacji. Jego przeznaczeniem jest tworzenie aplikacji będących połączeniem technologii webowych z natywnymi interfejsami. Oparte o platformę Node.js wielosystemowe narzędzie wspomaga wytwarzanie oprogramowania na platformy mobilne w myśl zasady „jeden kod, wiele platform”.

WSTĘP

W chwili obecnej rynek urządzeń mobilnych jest zdominowany przez 3 systemy operacyjne. Mowa tutaj o platformie Android firmy Google, iOS stworzonym przez Apple oraz próbującym zdobyć coraz to większe udziały w rynku Windows Phone spod szyldu Microsoft. Każdy z tych systemów jest inny, zarówno pod względem zewnętrznym, jak i procesu tworzenia aplikacji dla niego. I tak, Android korzysta z języka Java, iOS z Objective-C, a system Windows Phone z języka C# lub C++ z dodatkowymi bibliotekami. Taka sytuacja sprawia, że firmy chcąc stworzyć aplikację przeznaczoną na każdy z tych systemów, muszą większość kodu pisać od nowa, w innym języku programowania. W rzeczywistości powstają trzy różne aplikacje przeznaczone na odmienne systemy posiadające taką samą funkcjonalność. Jeśli dodać do tego systemy niszowe takie jak Blackberry, Firefox OS, Bada czy WebOS, stworzenie i rozwój aplikacji wieloplatformowej stanowi nie lada wyzwanie, zarówno pod względem utrzymania kodu, jak i skompletowania zespołu programistów specjalizujących się w danych platformach mobilnych.

Projekt Apache Cordova jest w pewnym sensie odpowiedzią na tę bolączkę. Stanowi on zbiór interfejsów programistycznych dających dostęp do natywnych funkcji urządzenia mobilnego, takich jak czujniki, system plików czy zarządzanie opcjami systemu operacyjnego. Obsługa tego wszystkiego odbywa się z poziomu języka JavaScript. Dzięki temu rozwiązaniu większość kodu może być w niezmienionej formie skonwertowana na aplikację przeznaczoną na inny system mobilny, niż ten, na który pierwotnie tworzone. Rozwiązanie to znacząco przyspiesza proces tworzenia wieloplatformowej aplikacji, redukując przy tym konieczność wgłębiania się w niuanse konkretnej platformy i jej języka programowania. Oczywiście takie rozwiązanie nie jest idealne w każdym przypadku. Istnieją typy aplikacji, które ze względów np. wydajnościowych należy pisać w języku programowania przeznaczonym na daną platformę.

Początki projektu Apache Cordova, jak wspomniano na początku, datują się na rok 2008, kiedy to firma Nitobi Software stworzyła projekt Phonegap, umożliwiając tworzenie natywnych aplikacji mobilnych z wykorzystaniem raczkującej wówczas technologii HTML5. Zespół programistów postawił sobie za cel stworzenie środowiska, w którym technologie HTML5 i CSS3 będą odpowiedzialne za warstwę prezentacji, natomiast logika aplikacji zostanie oparta o język JavaScript. W tamtych czasach HTML5 nie posiadał tak szerokiego dostępu do niższych warstw aplikacji takich jak geolokalizacja, dostęp do kamery czy żyroskopu. Aby osiągnąć założony cel, zespół musiał skonstruować pomost pomiędzy warstwą sprzętową a warstwą prezentacji. Osiągnięto to poprzez stworzenie czegoś na wzór szablonu aplikacji natywnej i umieszczenie w jej Web View kodu aplikacji internetowej. Do komunikacji między sobą użyto mechanizmu Foreign Function Interface (FFI). Dzięki temu kod oparty o HTML5 mógł wywoływać kod aplikacji natywnej, w jakiej de facto został uruchomiony. Ukończeniem tego procesu było stworzenie mechanizmu wtyczek opartych o FFI, przez co cały projekt mógł być dowolnie rozszerzalny o nowe funkcjonalności.

W 2009 roku podczas O'Reilly Media's 2009 Web 2.0 Conference spośród 80 tysięcy zgłoszonych aplikacji Phonegap znalazł się w finałowej piątce, ostatecznie zdobywając nagrodę People's Choice Award jako oprogramo-

wanie o otwartym kodzie źródłowym służące do tworzenia wieloplatformowych mobilnych aplikacji. Wówczas Phonegap wspierał platformy iOS oraz Android, z planami wsparcia dla Nokia, Palm, Blackberry, Windows.

Rok 2011 przyniósł wielkie zmiany w dalszym rozwoju projektu. Firma Nitobi Software z początkiem października 2011 roku została przejęta przez Adobe. Aby projekt mógł być nadal rozwijany na zasadach wolnego oprogramowania, Adobe zdecydowało o przekazaniu go do fundacji Apache. Jako że nazwa Phonegap była zastrzeżona dla Nitobi, a po jej nabyciu prawa do posługiwania się tą nazwą przejęło Adobe, postanowiono o nadaniu mu nowej nazwy Callback. Ostatecznie nazwa ta została zmieniona na Apache Cordova, na cześć ulicy, przy której mieściła się siedziba firmy Nitobi. Od tego momentu projekt jest stale rozwijany, dostosowując się do dynamicznego rynku urządzeń mobilnych.

PRZYGOTOWANIE SYSTEMU

Przed rozpoczęciem pracy z Apache Cordova konieczne jest przygotowanie systemu operacyjnego. Za przykład posłużą nam dwa systemy: Windows 7 oraz Ubuntu. Proces instalacji i konfiguracji w każdym z nich jest nieco inny. Późniejsza praca z zainstalowanymi narzędziami jest zbliżona. Osoby korzystające z systemu Windows nie są tak zaznajomione z linią poleceń jak użytkownicy Linuksa, jednak po kilku chwilach spędzonych z terminalem na pewno opanują techniki takiej pracy.

Prawidłowo skonfigurowany system zawiera oprócz Cordova również biblioteki programistyczne języka Java, środowisko uruchomieniowe systemu Android wraz jego bibliotekami i emulatorami urządzeń.

KONFIGURACJA UBUNTU

Pierwszym krokiem jest instalacja serwera Node.js. Odbywa się to poprzez wydanie komendy:

```
sudo apt-get install nodejs
```

W systemie Linux Ubuntu po zainstalowaniu pakietów nodejs należy ręcznie utworzyć dowiązanie stałe. W przeciwnym wypadku praca z Apache Cordova będzie niemożliwa. Dowiązanie tworzymy, wykonując polecenie:

```
sudo ln -s /usr/bin/nodejs /usr/bin/node
```

Wraz z instalacją Node.js zostanie zainstalowany manager pakietów npm. Aby sprawdzić, czy instalacja przebiegła pomyślnie, wydajemy polecenia:

```
nodejs -v
//v0.10.25
```

a następnie:

```
npm -v
//v0.10.25
```