

Instalacja Eclipse/ADT

1. Ściągnij i zainstaluj środowisko Eclipse IDE for Java EE developers.
2. W środowisku Eclipse przejdź do Help > Install New Software...
3. W polu *Work with:* wklej poniższy link do pluginu ADT (Android Development Tools)
<https://dl-ssl.google.com/android/eclipse/>
4. Odznacz poniższą opcję (to pozwoli zaoszczędzić wiele czasu ;))

☐ Contact all update sites during install to find required software

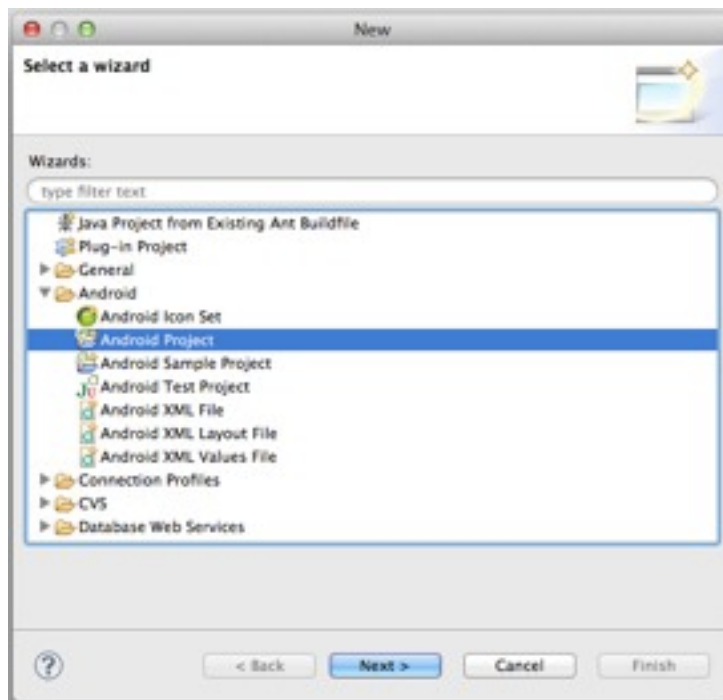
5. Rozpocznij instalację plugin'u ADT
6. Po zainstalowaniu plugin'u należy zrestartować środowisko Eclipse
7. Po restarcie zostaniemy poproszeni o wskazanie ścieżki gdzie ma zostać ściągnięte środowisko Android SDK. Należy pamiętać aby w ściąganych pakietach zaznaczyć również wirtualne urządzenie Android.

Ściągnij PhoneGap'a

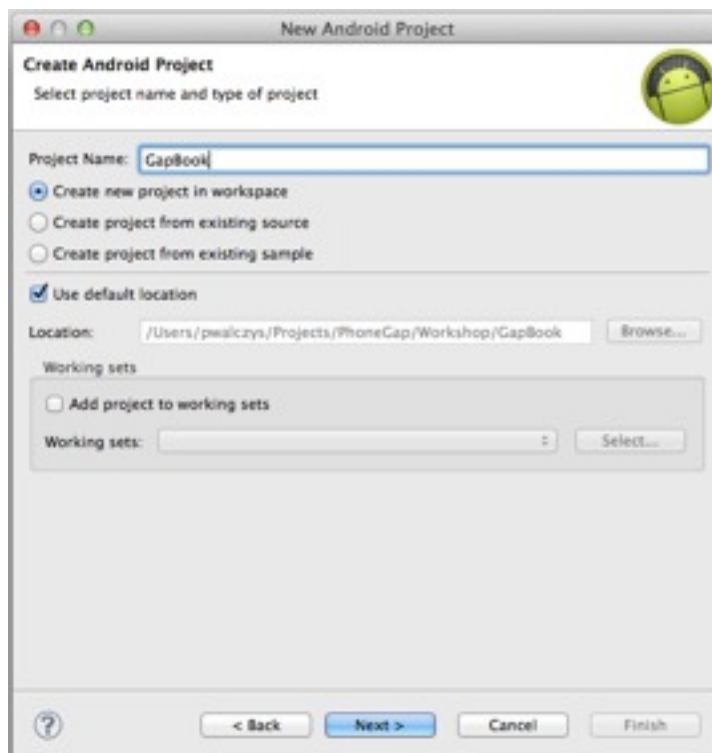
1. Najnowsze wydania technologii PhoneGap są zawsze dostępne pod poniższym adresem:
<http://phonegap.com/download>
2. Rozpakuj ściągnięty zip

Nowy projekt dla platformy Android

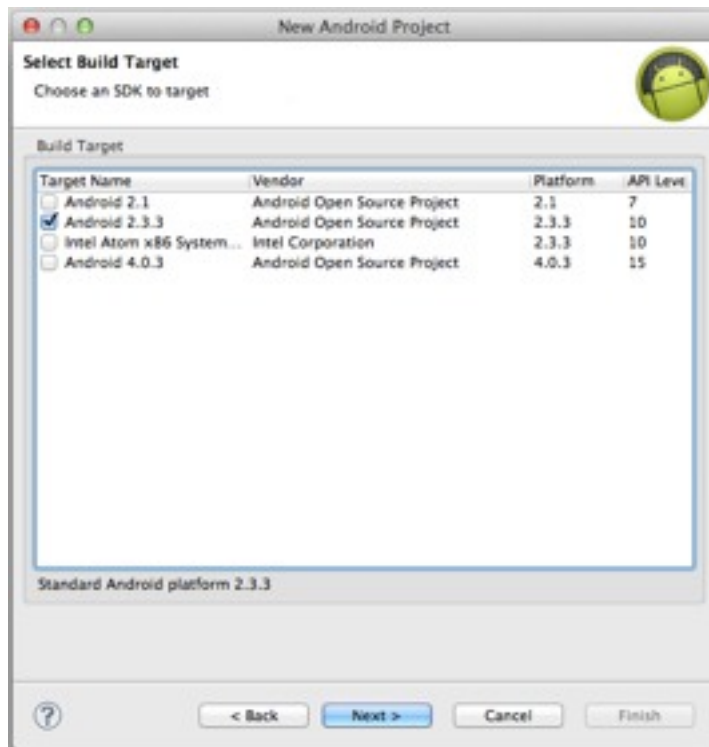
1. W środowisku Eclipse przejdź do File > New > Other... znajdź sekcję Android i wybierz opcję Android Project.



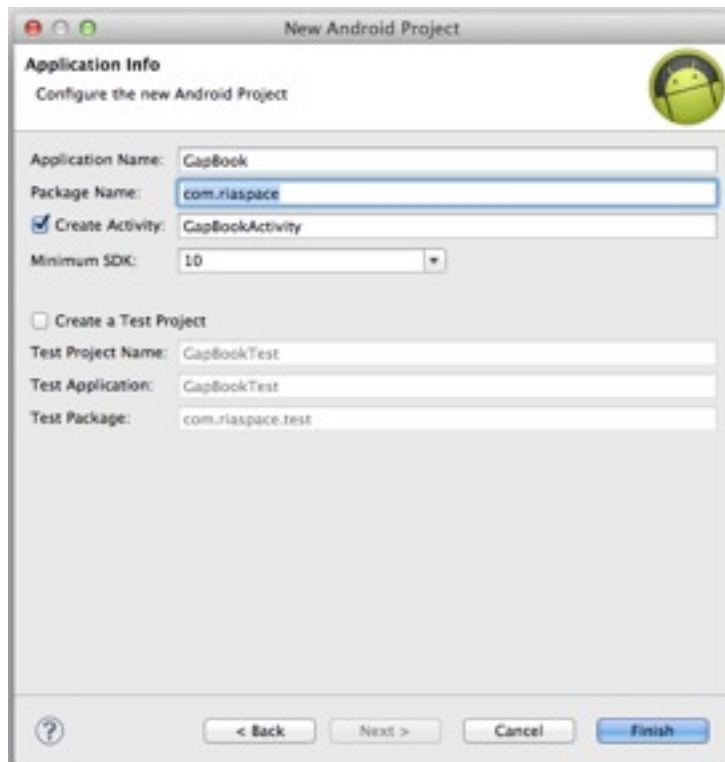
2. Kliknij przycisk Next aby rozpocząć kreator nowego projektu.
3. Wprowadź nazwę projektu np. GapBook i naciśnij przycisk Next



4. Wybierz wersję docelowej platformy:



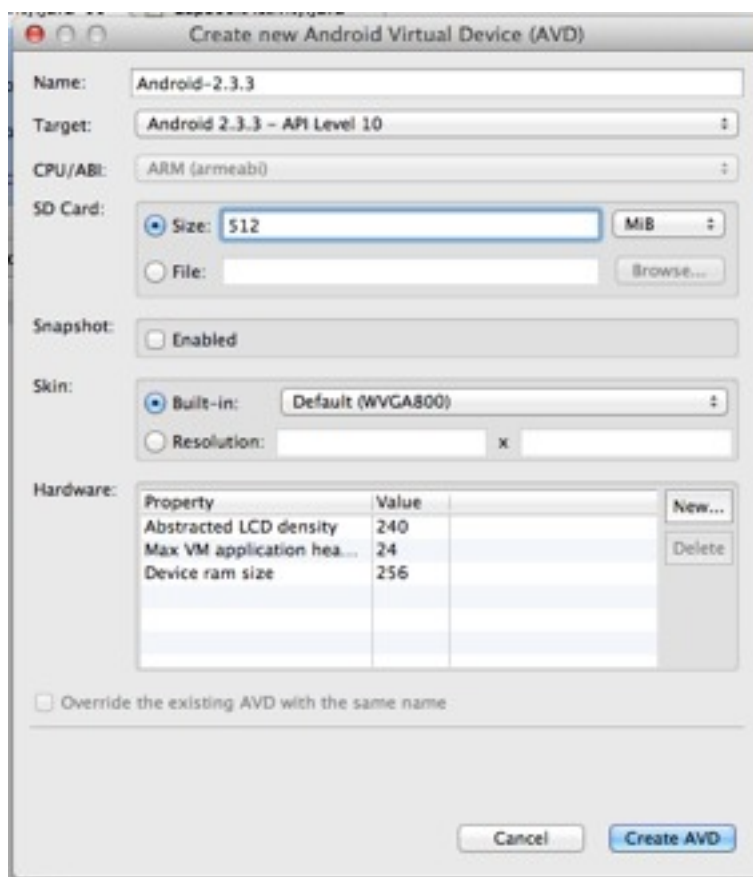
5. Wpisz nazwę pakietu aplikacji, z reguły jest to odwrócona nazwa domeny, tak jak w moim przypadku com.riaspacer gdzie riaspace.com to adres mojego bloga.



6. Naciśnij przycisk Finish. Krok ten stworzy nam pusty projekt dla platformy Android.

7. W utworzonym projekcie stwórz dwa nowe katalogi: libs i assets/www

8. Do katalogu libs skopiuj plik lib/android/cordova-x.x.x.jar znajdujący się we wcześniej rozpakowanym pakiecie środowiska PhoneGap.
9. Dodaj skopiowany plik jar do Java Build Path naszego projektu. Aby to zrobić kliknij prawym przyciskiem myszy na projekcie GapBook następnie wybierz opcję Properties > Java Build Path > Libraries > Add JARs... znajdź i wybierz wcześniej dodany plik i naciśnij przycisk OK.
10. Do katalogu res naszego projektu skopiuj cały katalog lib/android/xml również znajdujący się we wcześniej rozpakowanym katalogu środowiska PhoneGap.
11. W projekcie przejdź do katalogu src i znajdź klasę GapBookActivity.
12. W klasie GapBookActivity zmień rozszerzaną klasę Activity na DroidGap.



13. Zmień zawartość metody onCreate tak aby cała klasa wyglądała następująco:

```
package com.riaspace;  
  
import org.apache.cordova.DroidGap;  
  
import android.os.Bundle;  
  
public class GapBookActivity extends DroidGap {  
    /** Called when the activity is first created. */  
}
```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    super.loadUrl("file:///android_asset/www/index.html");
}
}

```

14. W środowisku Android aby użytkownik miał dostęp do bardziej zaawansowanych funkcji lub sensorów urządzenia należy dodać kilka wpisów do dokumentu AndroidManifest.xml znajdującego się w katalogu głównym projektu. Poniżej przedstawiony jest cały dokument AndroidManifest.xml z zaznaczonymi na żółto dodatkowymi wpisami.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.riaspaces"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk android:minSdkVersion="10" />

    <supports-screens
        android:anyDensity="true"
        android:largeScreens="true"
        android:normalScreens="true"
        android:resizeable="true"
        android:smallScreens="true" />

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.VIBRATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.RECEIVE_SMS" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.GET_ACCOUNTS" />
    <uses-permission android:name="android.permission.BROADCAST_STICKY" />

    <uses-feature android:name="android.hardware.camera" />
    <uses-feature android:name="android.hardware.camera.autofocus" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".GapBookActivity"
            android:configChanges="orientation|keyboardHidden"
            android:label="@string/app_name" >
            <intent-filter>

```

```
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
```

```
<activity
    android:name="com.phonegap.DroidGap"
    android:configChanges="orientation|keyboardHidden"
    android:label="@string/app_name" >
    <intent-filter>
    </intent-filter>
</activity>
```

```
</application>
```

```
</manifest>
```

15. Aby uruchomić przykłady standardowo dostarczane ze środowiskiem PhoneGap skopiuj zawartość katalogu lib/android/example/assets/www do assets/www w naszym projekcie.
16. Aby uruchomić naszą aplikację kliknij prawym przyciskiem myszy na projekcie GapBook i wybierz opcję Run As > Android Application.
17. Jeżeli nie masz podpiętego telefonu do swojego komputera lub wcześniej skonfigurowanego emulatora środowisko Eclipse poprosi o utworzenie wirtualnego urządzenia Android. Należy nacisnąć przycisk OK i jeżeli pojawi się okienko Android Device Chooser anulować je przyciskiem Cancel.
18. Na ekranie powinniśmy zobaczyć okno Android Virtual Device Manager gdzie za pomocą przycisku New... możemy stworzyć nowe wirtualne urządzenie Android.
19. W oknie konfiguracji wirtualnego urządzenia Android można wybrać poniższe opcje i nacisnąć przycisk Create AVD:
20. Po utworzeniu wirtualnego urządzenia w oknie Android Virtual Device Manager wybieramy z listy nasze urządzenie i naciskamy przycisk Start...
21. Po wystartowaniu wirtualnego urządzenia zamykamy okno Android Virtual Device Manager i ponownie uruchamiamy naszą aplikację.

Jeżeli wszystko zostało wykonane poprawnie i w katalogu www mieliśmy skopiowaną przykładową aplikację powinniśmy zobaczyć poniższy wynik:

