

RESTful Services

Funshine

Teil 2: Zugriff auf Location API

manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="at.htl.funshine">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application android:allowBackup="true" android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true" android:theme="@style/AppTheme">
        <activity android:name=".WeatherActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Hinzufügen der Play-Services

```
dependencies{
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 25
5      buildToolsVersion "25.0.2"
6      defaultConfig {
7          applicationId "at.htl.funshine"
8          minSdkVersion 16
9          targetSdkVersion 25
10         versionCode 1
11         versionName "1.0"
12         testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
13     }
14     buildTypes {
15         release {
16             minifyEnabled false
17             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18         }
19     }
20 }
21
22 dependencies {
23     compile fileTree(include: ['*.jar'], dir: 'libs')
24     androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
25         exclude group: 'com.android.support', module: 'support-annotations'
26     })
27     compile 'com.android.support:appcompat-v7:25.3.1'
28     testCompile 'junit:junit:4.12'
29
30     compile 'com.google.android.gms:play-services:10.2.1'
31 }
32 }
```

Leider tritt dabei ein Fehler auf, der auf Versionsprobleme hinweist

All com.android.support libraries must use the exact same version specification (mixing versions can lead to runtime crashes). Found versions 25.3.1, 24.0.0. Examples include com.android.support:animated-vector-drawable:25.3.1 and com.android.support:mediarouter-v7:24.0.0 [more...](#) (%F1)

Abhilfe

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "at.htl.funshine"
        minSdkVersion 16
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    compile 'com.android.volley:volley:1.0.0'
    compile 'com.google.android.gms:play-services:10.2.1'
    compile 'com.google.android.gms:play-services-maps:10.2.1'
    compile 'com.google.android.gms:play-services-location:10.2.1'
}
```

Anstelle von den play-services werden die play-services-maps und play-services-location importiert

WeatherActivity.java 1

```
public class WeatherActivity extends AppCompatActivity
    implements GoogleApiClient.OnConnectionFailedListener,
               GoogleApiClient.ConnectionCallbacks,
               LocationListener {
```

Class 'WeatherActivity' must either be declared abstract or implement abstract method 'onConnectionFailed(ConnectionResult)' in 'OnConnectionFailedListener'

```
final String LOG_TAG = AppCompatActivity.class.getSimpleName();
final String URL_BASE = "http://api.openweathermap.org/data/2.5/forecast";
final String URL_COORD = "?lat="; //"?lat=48.2686066&lon=14.2493933";
final String URL_UNITS = "&units=metrics";
final String URL_API_KEY = "&APPID=5b59acdd3bf5119d2fd7f1f958ae01ec";
//final String URL_API_KEY = "&APPID=VerwendeDeinenAPI-Key";
```

```
private GoogleApiClient mGoogleApiClient;
private final int PERMISSION_LOCATION = 111;
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_weather);

    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addApi(LocationServices.API)
        .enableAutoManage(this, this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .build();
}
```

WeatherActivity.java 2

```
public void downloadWeatherData(Location location) {
    final String fullCoords = URL_COORD + location.getLatitude() + "&lon=" +
location.getLongitude();
    final String url = URL_BASE + fullCoords + URL_UNITS + URL_API_KEY;

    final JsonObjectRequest jsonRequest = new JsonObjectRequest(
        Request.Method.GET,
        url,
        null, // hier könnte man JSON-Daten im body mitschicken
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                Log.v(LOG_TAG, "RES: " + response.toString());
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Log.v(LOG_TAG, "Err: " + error.getLocalizedMessage());
            }
        });

    Volley.newRequestQueue(this).add(jsonRequest);
}
```

WeatherActivity.java 3

```
@Override
public void onConnected(@Nullable Bundle bundle) {
    if (ContextCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(
            this,
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, PERMISSION_LOCATION);
    } else {
        startLocationServices();
    }
}

@Override
public void onConnectionSuspended(int i) {
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
}

@Override
public void onLocationChanged(Location location) {
    downloadWeatherData(location);
}
```

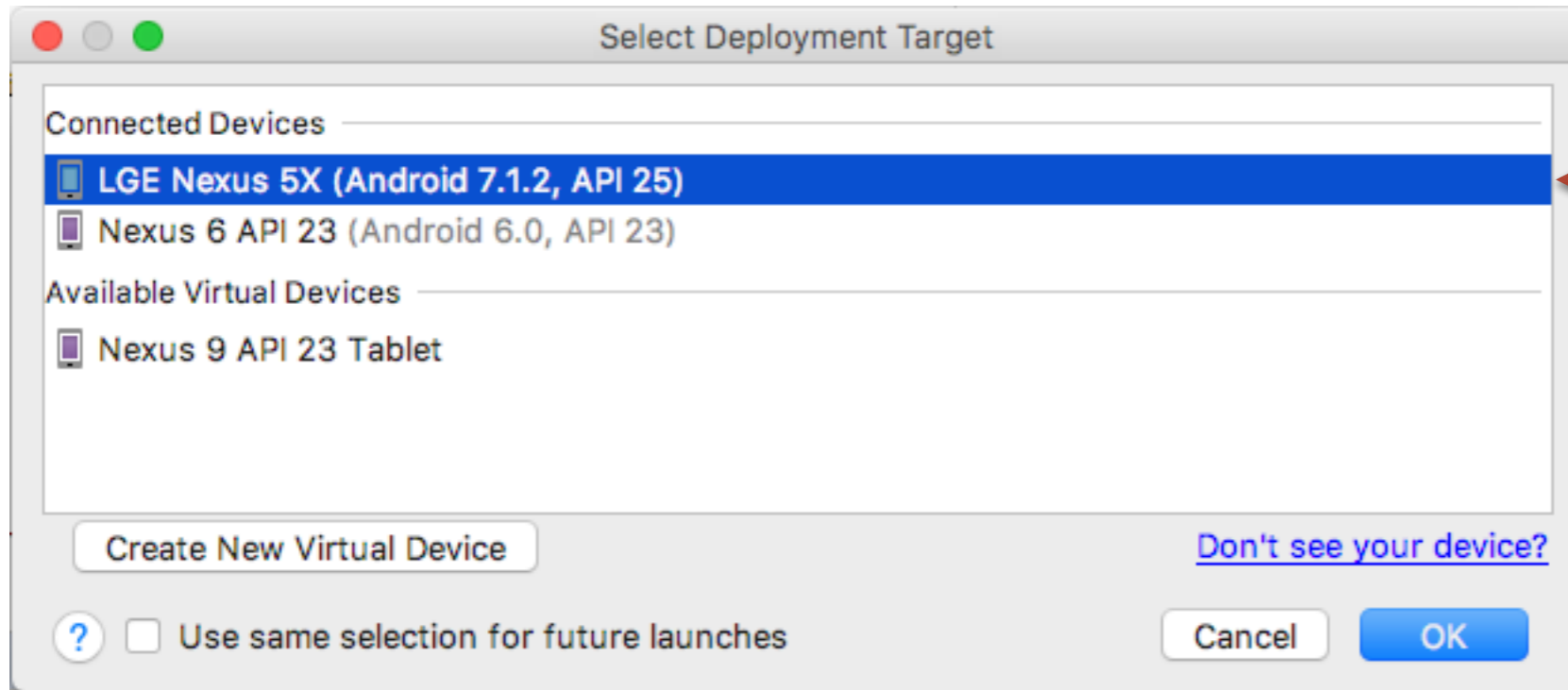

WeatherActivity.java 4

```
public void startLocationServices() {
    try {
        LocationRequest req = LocationRequest
            .create()
            .setPriority(LocationRequest.PRIORITY_LOW_POWER);
        LocationServices.FusedLocationApi
            .requestLocationUpdates(mGoogleApiClient, req, this);
    } catch (SecurityException exception) {
    }
}

@Override
public void onRequestPermissionsResult(int requestCode,
    @NonNull String[] permissions,
    @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    switch (requestCode) {
        case PERMISSION_LOCATION: {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                startLocationServices();
            } else {
                // show a dialog saying something like, "I can't run your location dummy - you
                denied permission!"
                Toast.makeText(
                    this,
                    "I can't run your location dummy - you denied permission!"
                    , Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

Probelauf



Der Probelauf wird NICHT am Emulator durchgeführt, da es hier zu Problemen kommen kann



Der momentane Standort ist ersichtlich



Noch
Fragen?