

# RESTful Services

Funshine

# Teil 1: Einfacher REST- Service



# Create Android Project

Android Studio

## Configure your new project

Application name:

Company domain:

Package name:  [Edit](#)

Include C++ support

Project location:  ...

Cancel

Previous

Next

Finish



# Target Android Devices

Android Studio

## Select the form factors your app will run on

Different platforms may require separate SDKs

Phone and Tablet

Minimum SDK

Lower API levels target more devices, but have fewer features available.

By targeting API 16 and later, your app will run on approximately **95.2%** of the devices that are active on the Google Play Store.

[Help me choose](#)

Wear

Minimum SDK

TV

Minimum SDK

Android Auto

Cancel

Previous

Next

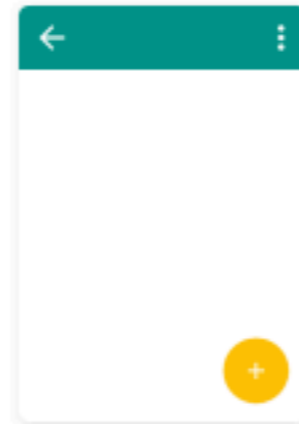
Finish



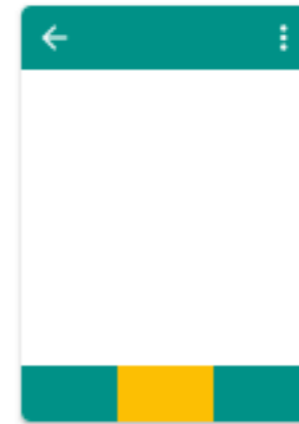
# Add an Activity to Phone and Tablet

Android Studio

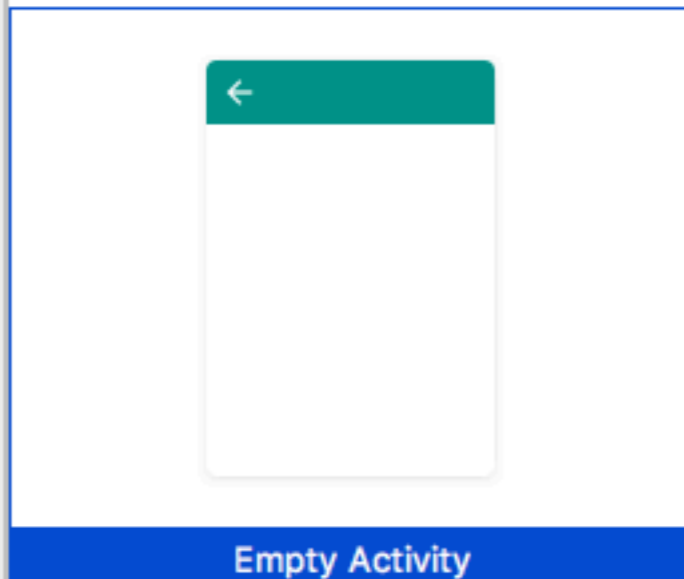
Add No Activity



Basic Activity



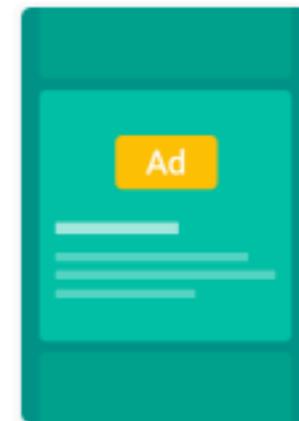
Bottom Navigation Activity



Empty Activity



Fullscreen Activity



Google AdMob Ads Activity

Cancel

Previous

Next

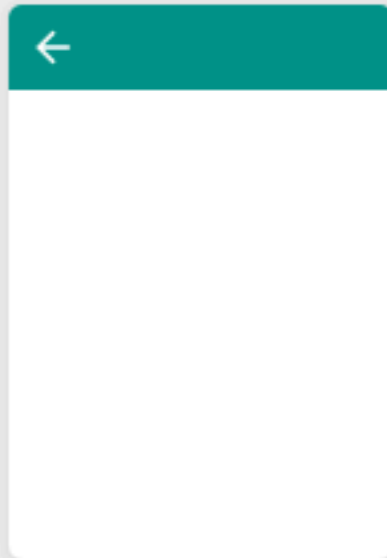
Finish



# Configure Activity

Android Studio

Creates a new empty activity



Activity Name:

WeatherActivity

Generate Layout File

Layout Name:

activity\_weather

Backwards Compatibility (AppCompat)

If true, a layout file will be generated

Cancel

Previous

Next

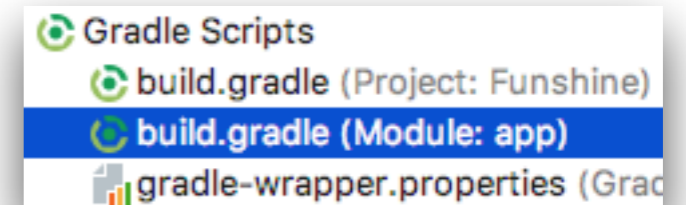
Finish

# Hinzufügen von Volley

```
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(dir: 'libs', include: ['*.jar'])
    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'
}
```



Bei sämtlichen http-Anfragen ist Volley zu verwenden, außer bei großen Files und Videostreaming

# Anlegen der Konstanten

```
package at.htl.funshine;
```

```
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;
```

```
public class WeatherActivity extends AppCompatActivity {
```

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

```
@Override
```

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

```
}
```

```
}
```



# WeatherActivity.java

```
public class WeatherActivity extends AppCompatActivity {  
  
    final String LOG_TAG = AppCompatActivity.class.getSimpleName();  
    final String URL_BASE = "http://api.openweathermap.org/data/2.5/forecast";  
    final String URL_COORD = "?lat=48.2686066&lon=14.2493933";  
    final String URL_UNITS = "&units=metrics";  
    final String URL_API_KEY = "&APPID=VerwendeDeinenAPI-Key";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_weather);  
  
        final String url = URL_BASE + URL_COORD + 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);  
    }  
}
```

Hier wird das jsonRequest-Object erzeugt

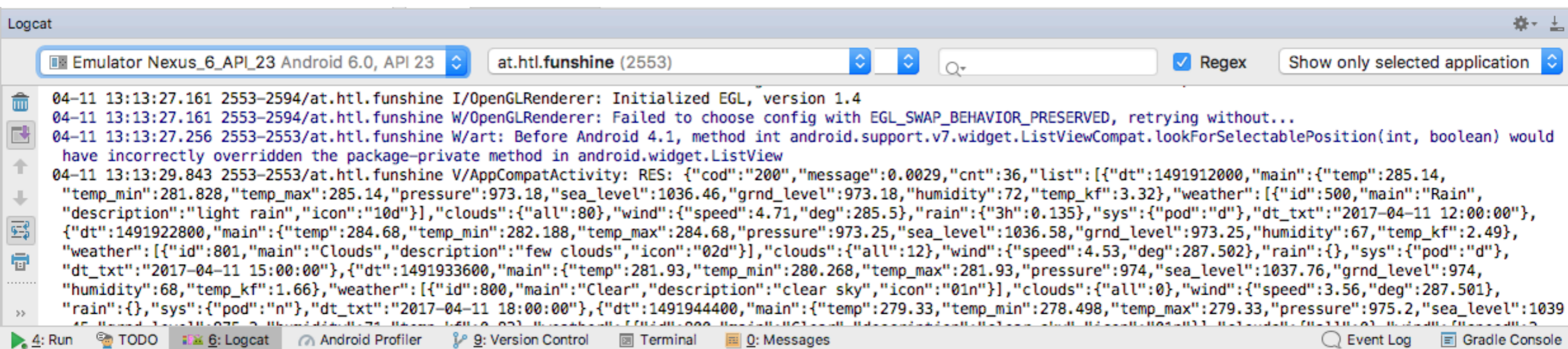
... und hier wird der Request abgesetzt

# 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" />
<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>
```

# Probelauf

- Nach dem Start der Application wird im LogCat das json-Objekt mit den Wetterdaten angezeigt



The screenshot shows the Logcat window in Android Studio. The top bar indicates the emulator is 'Emulator Nexus\_6\_API\_23 Android 6.0, API 23' and the application is 'at.htl.funshine (2553)'. The 'Regex' checkbox is checked, and the filter is set to 'Show only selected application'. The log output shows several messages, with the most prominent one being a JSON object representing weather data:

```
04-11 13:13:29.843 2553-2553/at.htl.funshine V/AppCompatActivity: RES: {"cod":"200","message":"0.0029","cnt":36,"list":[{"dt":1491912000,"main":{"temp":285.14,"temp_min":281.828,"temp_max":285.14,"pressure":973.18,"sea_level":1036.46,"grnd_level":973.18,"humidity":72,"temp_kf":3.32},"weather":[{"id":500,"main":"Rain","description":"light rain","icon":"10d"}],"clouds":{"all":80},"wind":{"speed":4.71,"deg":285.5},"rain":{"3h":0.135},"sys":{"pod":"d"},"dt_txt":"2017-04-11 12:00:00"}, {"dt":1491922800,"main":{"temp":284.68,"temp_min":282.188,"temp_max":284.68,"pressure":973.25,"sea_level":1036.58,"grnd_level":973.25,"humidity":67,"temp_kf":2.49},"weather":[{"id":801,"main":"Clouds","description":"few clouds","icon":"02d"}],"clouds":{"all":12},"wind":{"speed":4.53,"deg":287.502},"rain":{"sys":{"pod":"d"},"dt_txt":"2017-04-11 15:00:00"}, {"dt":1491933600,"main":{"temp":281.93,"temp_min":280.268,"temp_max":281.93,"pressure":974,"sea_level":1037.76,"grnd_level":974,"humidity":68,"temp_kf":1.66},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01n"}],"clouds":{"all":0},"wind":{"speed":3.56,"deg":287.501},"rain":{"sys":{"pod":"n"},"dt_txt":"2017-04-11 18:00:00"}, {"dt":1491944400,"main":{"temp":279.33,"temp_min":278.498,"temp_max":279.33,"pressure":975.2,"sea_level":1039.45,"grnd_level":975.2,"humidity":71,"temp_kf":0.83},"weather":[{"id":800,"main":"Clear","description":"clear sky","icon":"01n"}],"clouds":{"all":0},"wind":{"speed":3.56,"deg":287.501},"rain":{"sys":{"pod":"n"},"dt_txt":"2017-04-11 21:00:00"}]}
```

### [-] Request

Method  URL

**Body**

Request Body

### [-] Response

```
1. {
2.   "cod": "200",
3.   "message": 0.0462,
4.   "cnt": 40,
5.   "list":
6.     [
7.       {
8.         "dt": 1491858000,
9.         "main":
10.        {
11.          "temp": 285.14,
12.          "temp_min": 284.311,
13.          "temp_max": 285.14,
14.          "pressure": 967.58,
15.          "sea_level": 1030.32,
16.          "grnd_level": 967.58,
17.          "humidity": 82,
18.          "temp_kf": 0.83
19.        },
20.       "weather":
```





Noch  
Fragen?