IoT Workshop

Vincent Claes

Servaas Tilkin

DE HOGESCHOOL
MET HET NETWERK

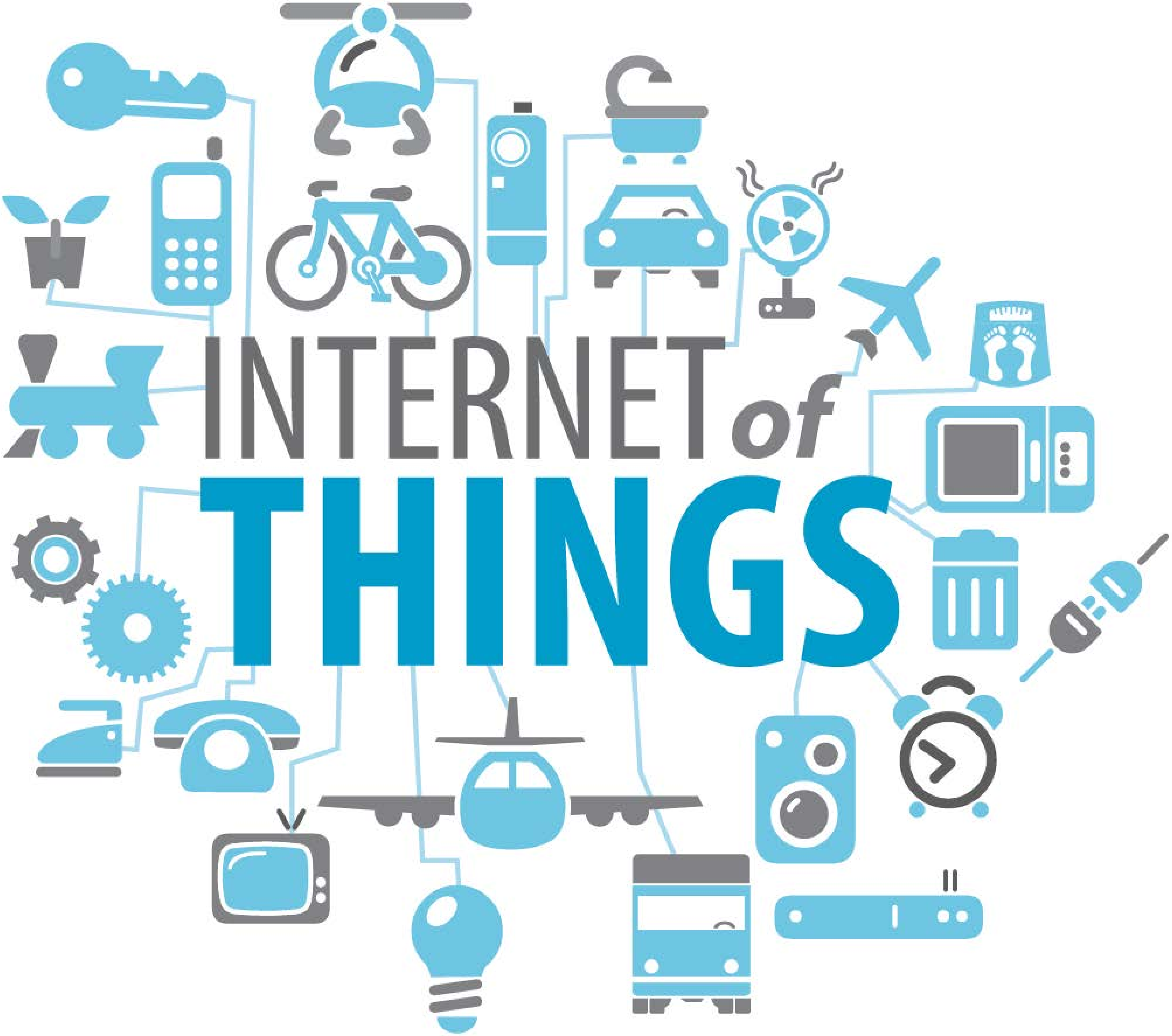
Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Vandaag

- IoT
- PCB
- Arduino IDE
- Web API
- Android App
- Fun

Concept

IOT

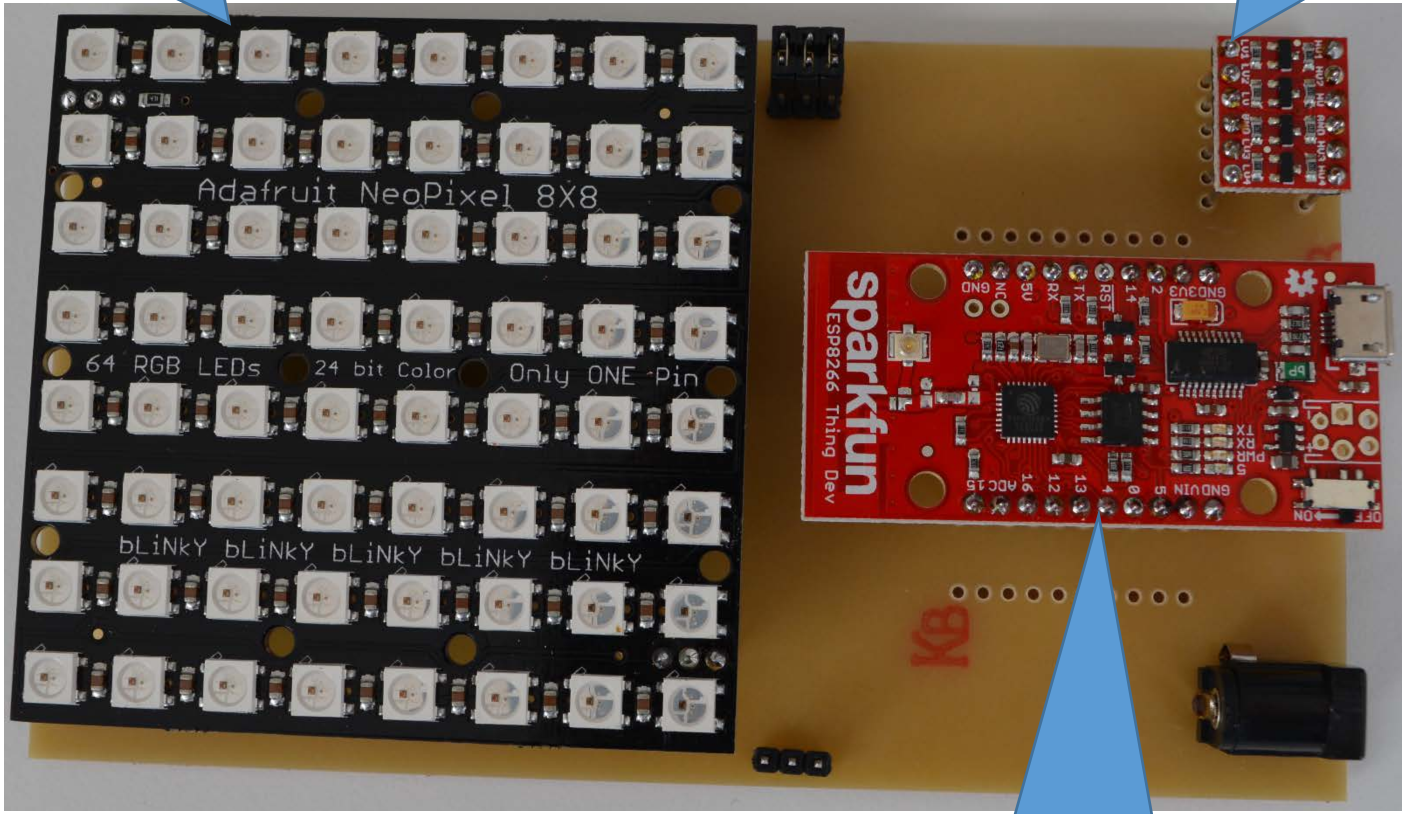

INTERNET of
THINGS

Hardware

PCB

8x8 RGB Matrix

Logic Level Convertor
Bi-Directional



ESP8266 Thing Dev Board

ESP8266 Thing Dev Board (WRL-13711)

Arduino add-on available
80MHz

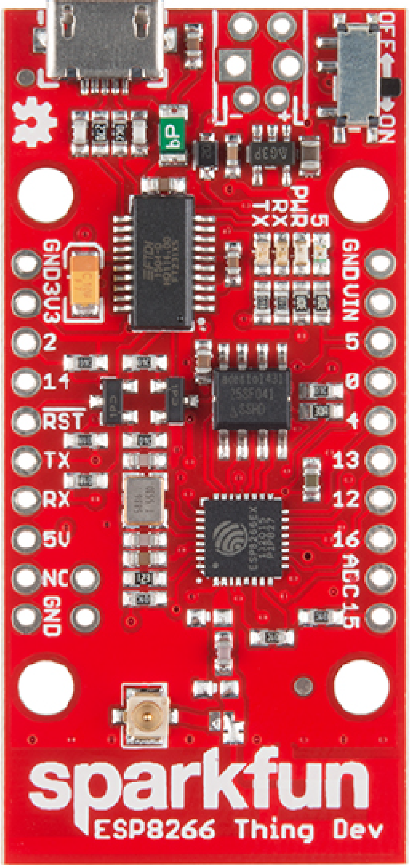
Name	Arduino
Power	ADC
GND	Serial
Control	Misc

MicroB USB

3.7-6V
Power connector

Power switch

GPIO0
On bootup will run program if high and bootloader if low
Tied to DTR to run bootloader when reset



	GND	GND
	3.3V	3V3
SDA	D2	2
SCL/SCLK	D14	14
	Reset	RST
TX	D7	TX
RX	D8	RX
	V_USB	5V
	Not Connected	NC
	GND	GND

GND	GND	
Vin	Vin	
5	D5	LED
0	D0	Used in reset
4	D4	
13	D13	MOSI
12	D12	MISO
16	D16	XPD
ADC	A0	10-bit 1V
15	D15	

u.fl antenna connector
Not Connected
To use swap the solder jumper

Jumpers on back
PWR-LED - cut to disconnect power LED
SLEEP-EN - Connect RST to XPD to enable sleep (disables programming)

PCB Antenna

Power (ESP8266 Thing)
Vin: 3.7V-6V
VUSB: 5V
VCC:3.3V @500mA
Max 12mA per I/O pin

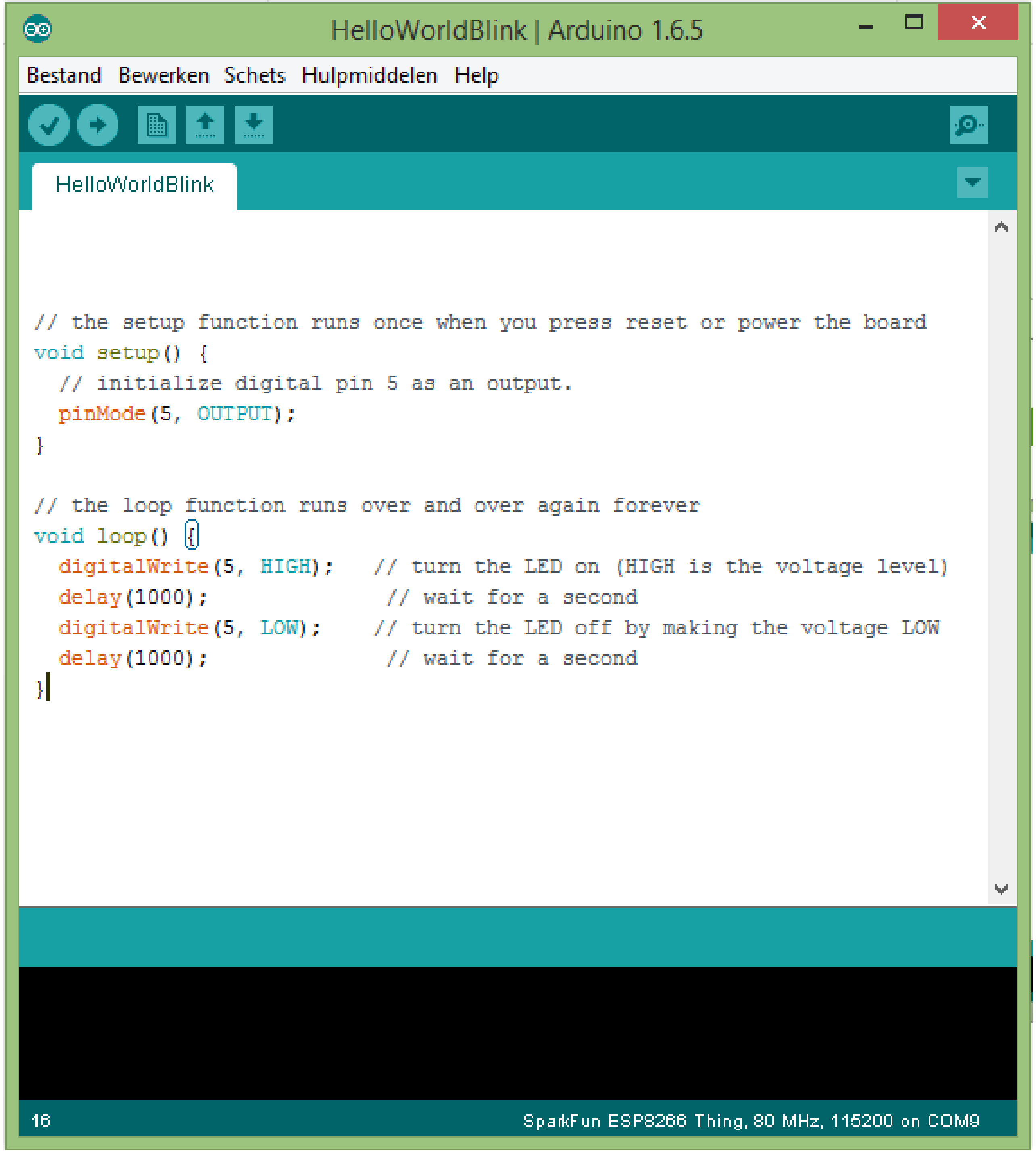
Typical Power (ESP8266 module)
Transmit 135-215mA
Receive 60-62mA
Standby 0.9mA
Deep sleep 10uA

LEDs
Power: Red
User (pin 5): Blue
RX: Yellow
TX: Green

Wi-Fi
802.11 b/g/n
Wi-Fi Direct (P2P) soft AP

Embedded Software

ARDUINO IDE

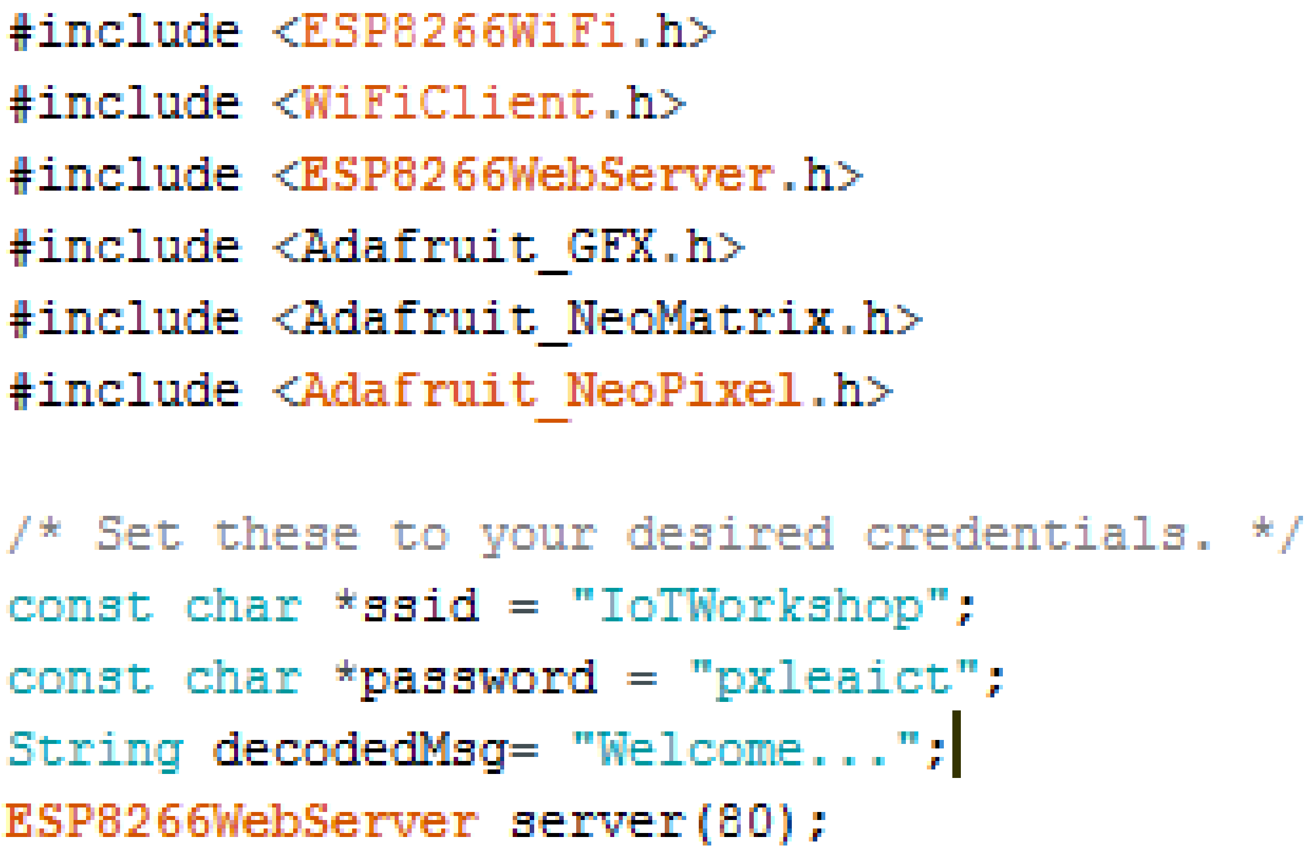


```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 5 as an output.
  pinMode(5, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(5, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(5, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

16   SparkFun ESP8266 Thing, 80 MHz, 115200 on COM9

IoTWorkshop.ino



```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <Adafruit_GFX.h>
#include <Adafruit_NeoMatrix.h>
#include <Adafruit_NeoPixel.h>

/* Set these to your desired credentials. */
const char *ssid = "IoTWorkshop";
const char *password = "pxleaict";
String decodedMsg= "Welcome...";
ESP8266WebServer server(80);
```

Software 2 Hardware

WEB API

Try it in your browser!

- http://192.168.4.1/msg?msg=Hello World
- http://192.168.4.1/pix?pix=00
- http://192.168.4.1/rgb?rgb=255/255/255
- http://192.168.4.1/clear

Applicatie

ANDROID APP

Making web calls

- Opletten met *Android UI thread*
  - Wat is een thread?
  - Waarom opletten?

- Verschillende manieren
  - **Volley** bibliotheek
  - Zelf doen met threads, calls, callbacks, …

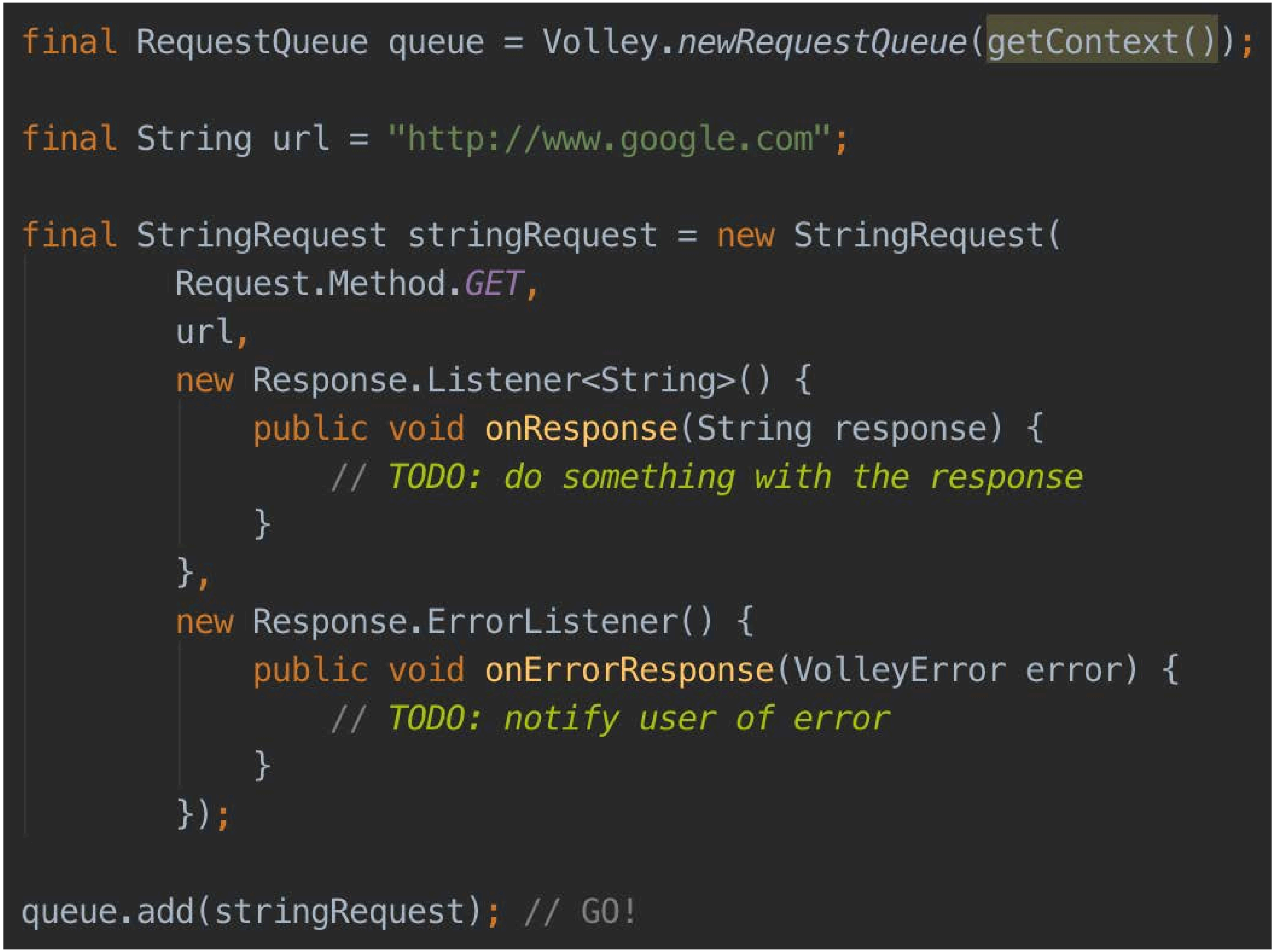
Making web calls

- Permissies

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

- Library
  - *app > Open Module Settings*
  - *Dependencies tab > + > Library Dependency*
  - *”com.android.volley” > OK*

Volley



```
final RequestQueue queue = Volley.newRequestQueue(getContext());

final String url = "http://www.google.com";

final StringRequest stringRequest = new StringRequest(
        Request.Method.GET,
        url,
        new Response.Listener<String>() {
            public void onResponse(String response) {
                // TODO: do something with the response
            }
        },
        new Response.ErrorListener() {
            public void onErrorResponse(VolleyError error) {
                // TODO: notify user of error
            }
        });

queue.add(stringRequest); // GO!
```

Volley

- Meer info:
  - http://developer.android.com/training/volley/simple.html
  - Vragen!

Playtime

FUN

Opdracht