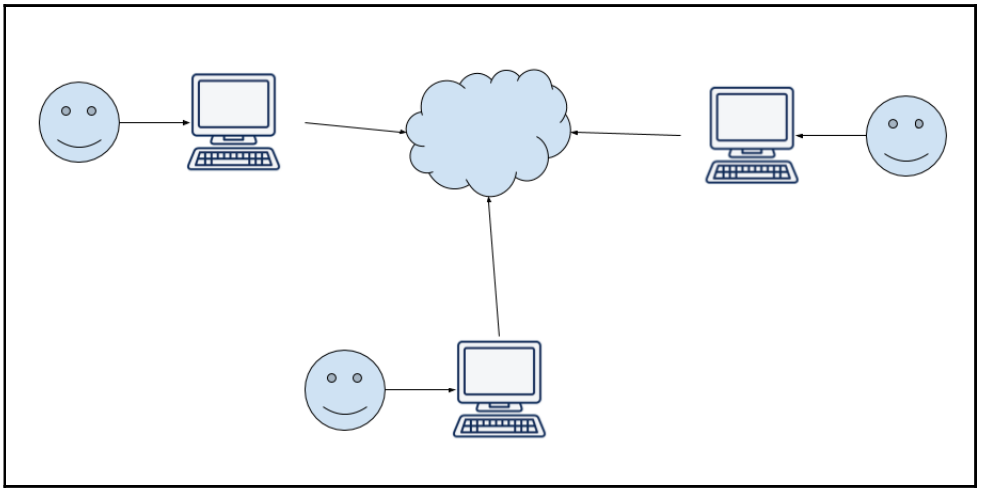
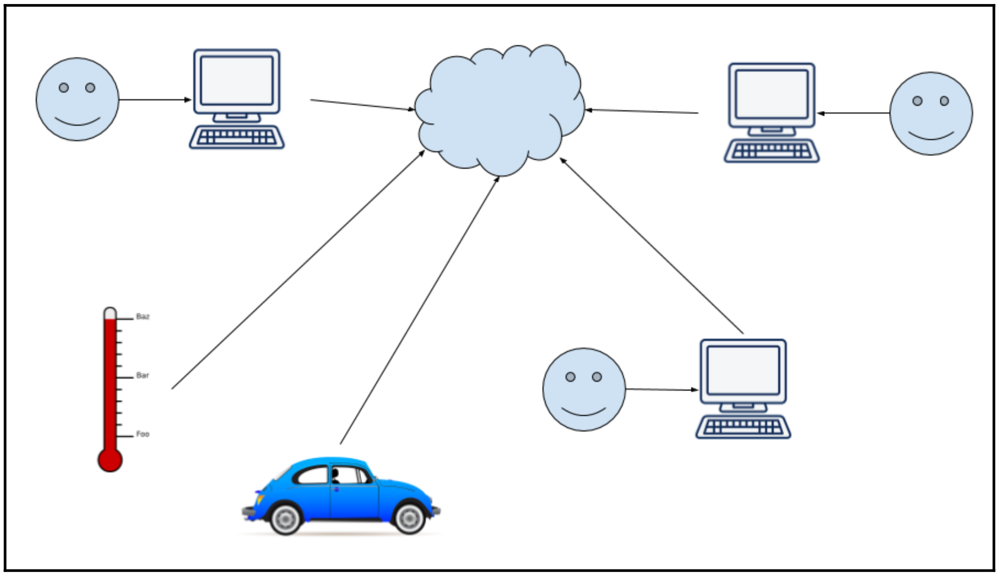
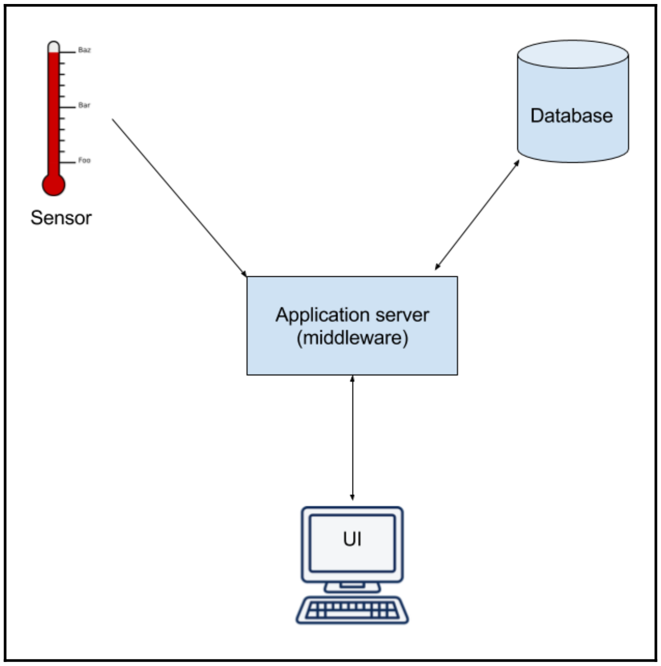
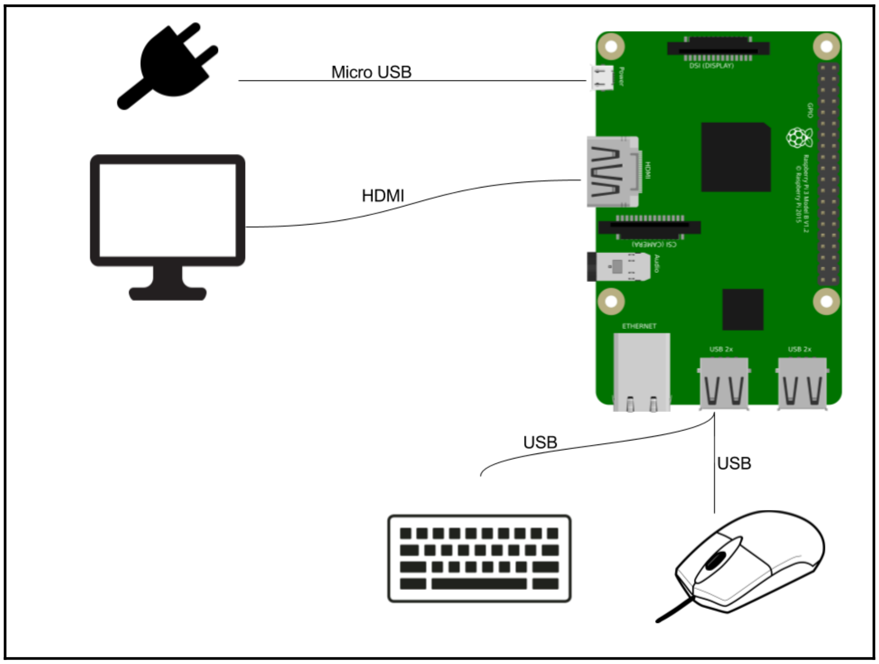
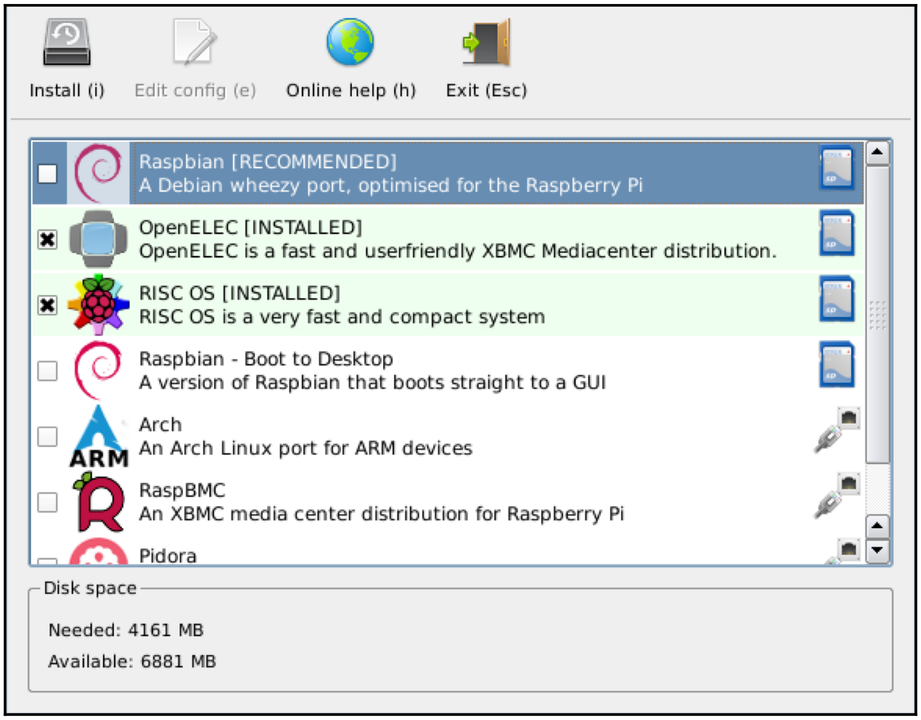
Chapter 1: Getting Started on the Raspberry Pi






Baz
Bar
Foo
Sensor
Database
Application server (middleware)
UI


Micro USB
HDMI
USB
USB



Raspbian [RECOMMENDED]
A Debian wheezy port, optimised for the Raspberry Pi

[✓] OpenELEC [INSTALLED]
OpenELEC is a fast and userfriendly XBMC Mediacenter distribution.

[✓] RISC OS [INSTALLED]
RISC OS is a very fast and compact system

Raspbian - Boot to Desktop
A version of Raspbian that boots straight to a GUI

Arch
An Arch Linux port for ARM devices

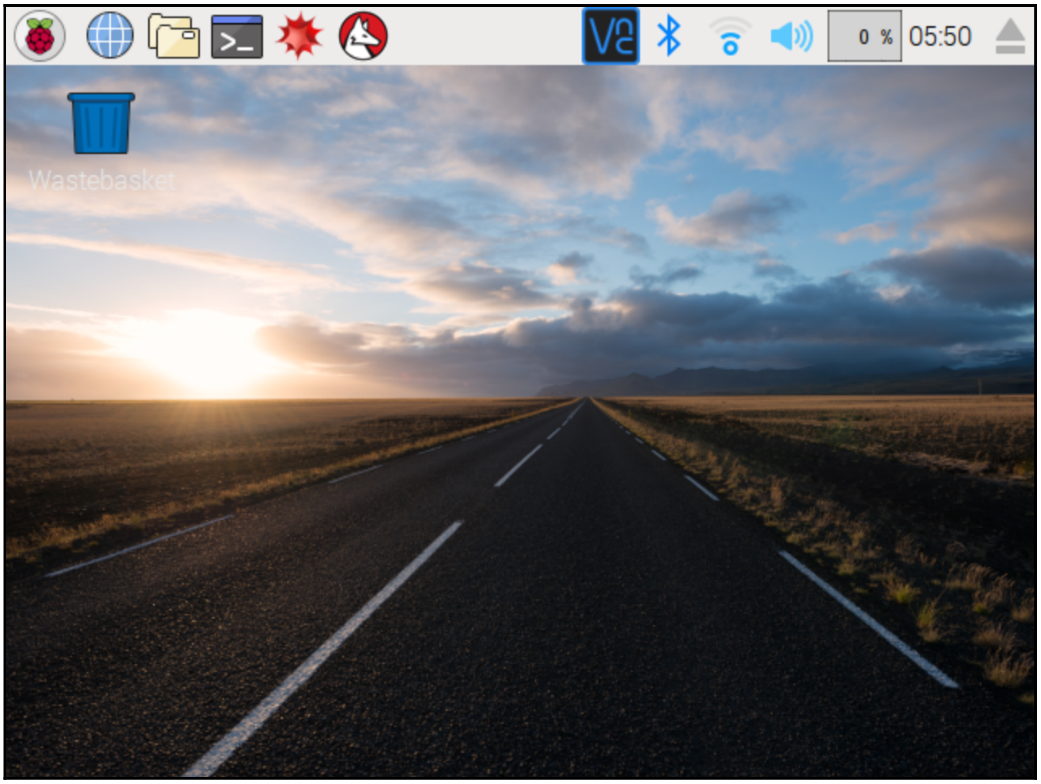
RaspBMC
An XBMC media center distribution for Raspberry Pi

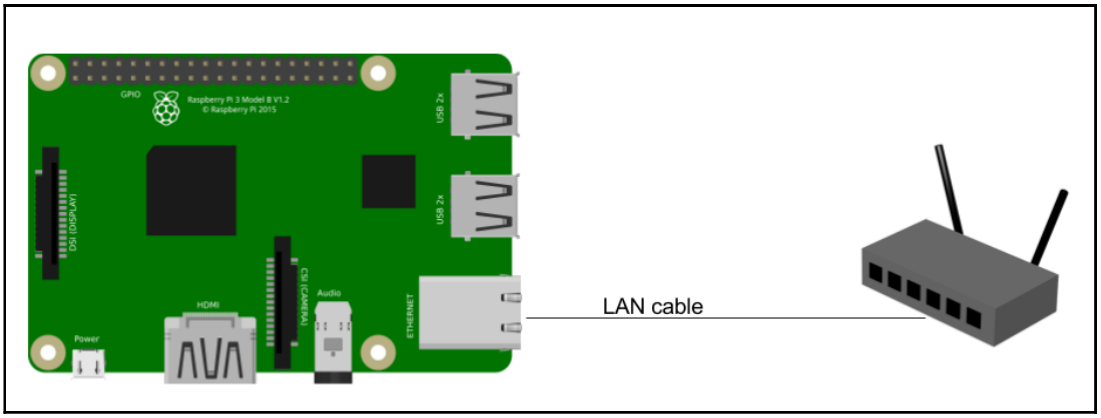
Pidora

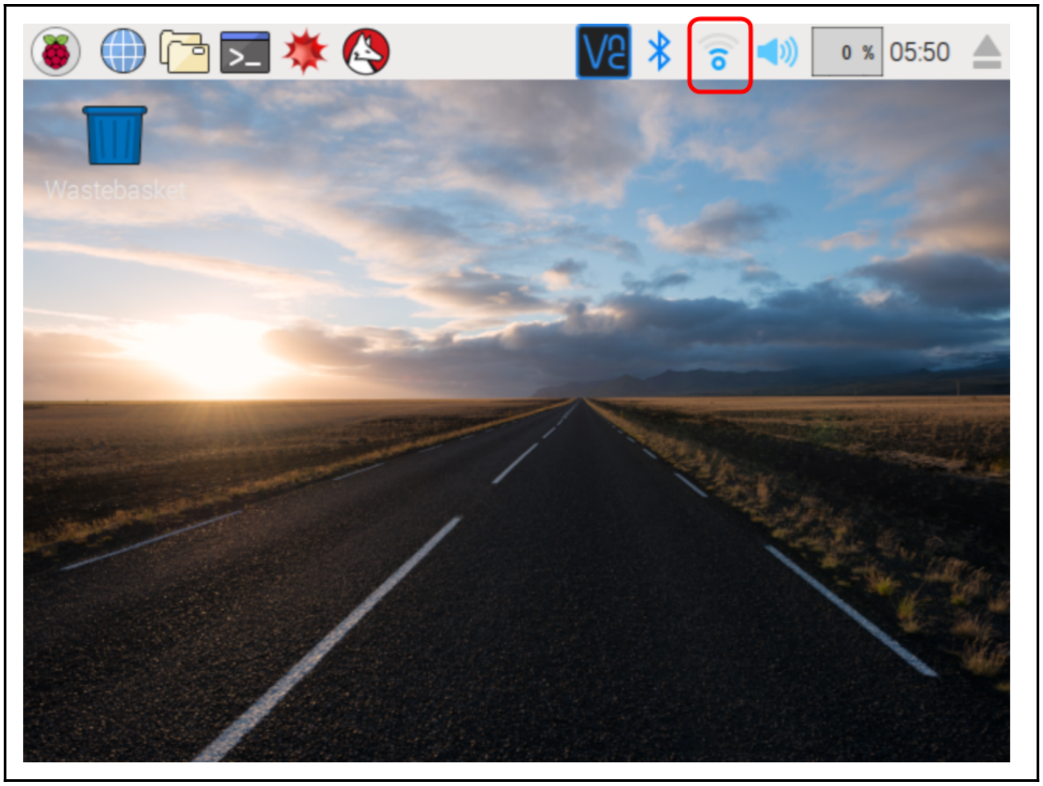
Disk space

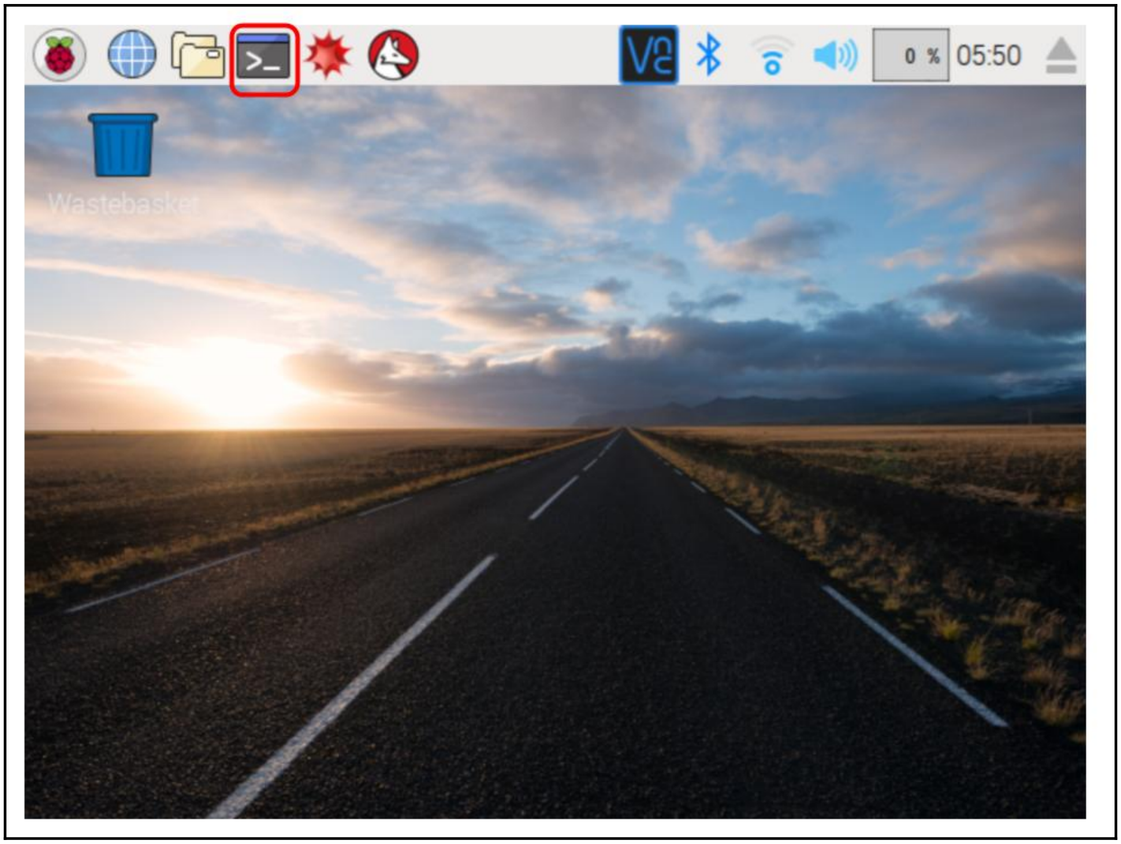
Needed: 4161 MB

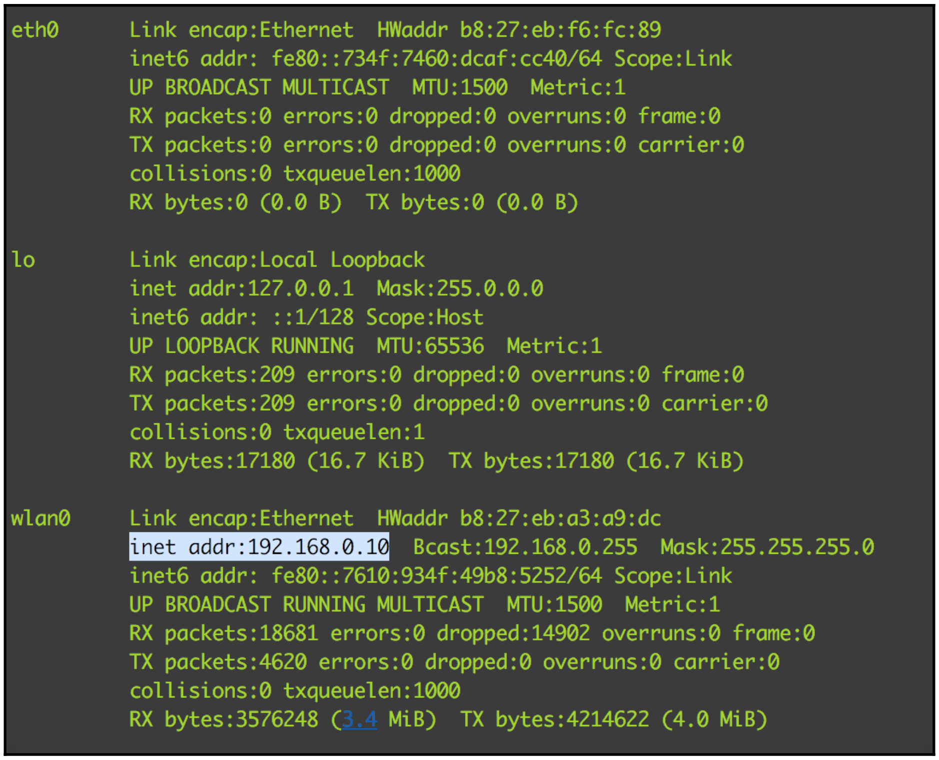
Available: 6881 MB


0 %
05:50
Wastebasket

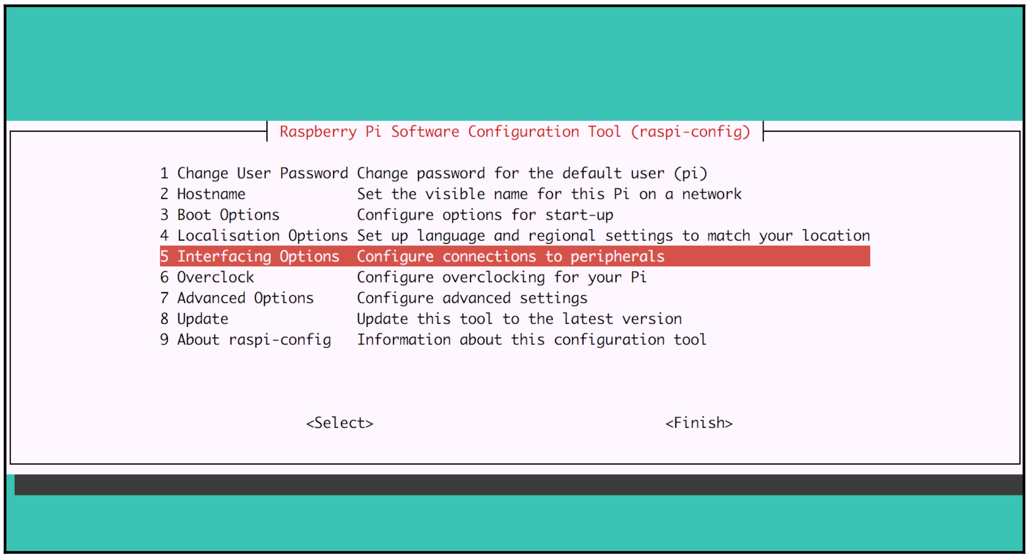

GPIO
Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
CSI (CAMERA)
Audio
HDMI
Power
ETHERNET
LAN cable


0 %
05:50
Wastebasket

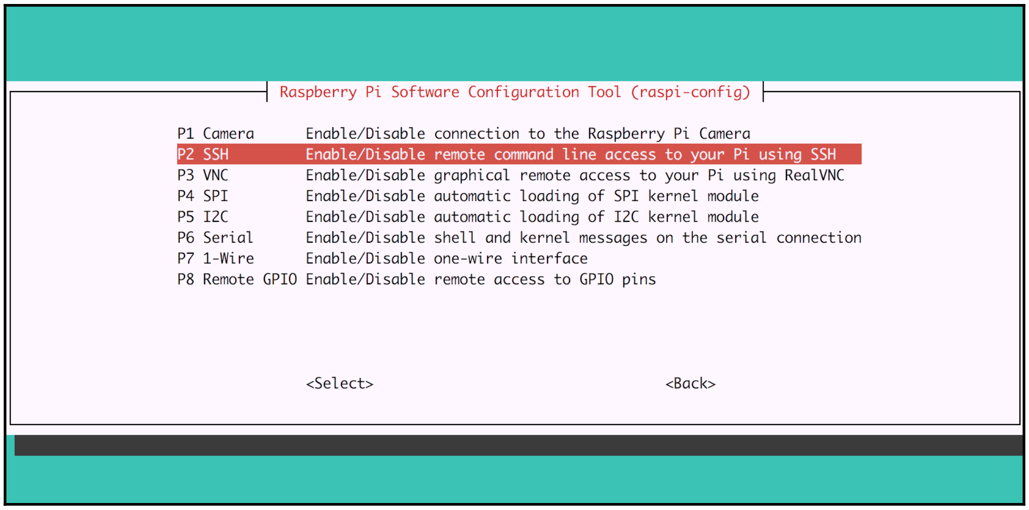

0 %
05:50
Wastebasket



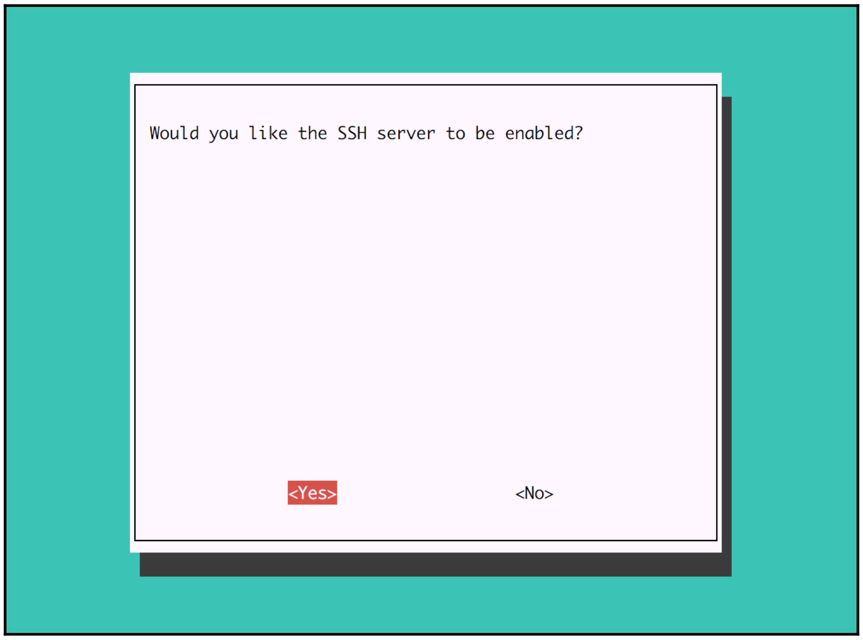
```
eth0      Link encap:Ethernet  HWaddr b8:27:eb:f6:fc:89
          inet6 addr: fe80::734f:7460:dcaf:cc40/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:209 errors:0 dropped:0 overruns:0 frame:0
          TX packets:209 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:17180 (16.7 KiB)  TX bytes:17180 (16.7 KiB)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:a3:a9:dc
          inet addr:192.168.0.10  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::7610:934f:49b8:5252/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:18681 errors:0 dropped:14902 overruns:0 frame:0
          TX packets:4620 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3576248 (3.4 MiB)  TX bytes:4214622 (4.0 MiB)
```



```
Raspberry Pi Software Configuration Tool (raspi-config)

1 Change User Password Change password for the default user (pi)
2 Hostname             Set the visible name for this Pi on a network
3 Boot Options         Configure options for start-up
4 Localisation Options Set up language and regional settings to match your location
5 Interfacing Options  Configure connections to peripherals
6 Overclock            Configure overclocking for your Pi
7 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config   Information about this configuration tool

                <Select>                     <Finish>
```



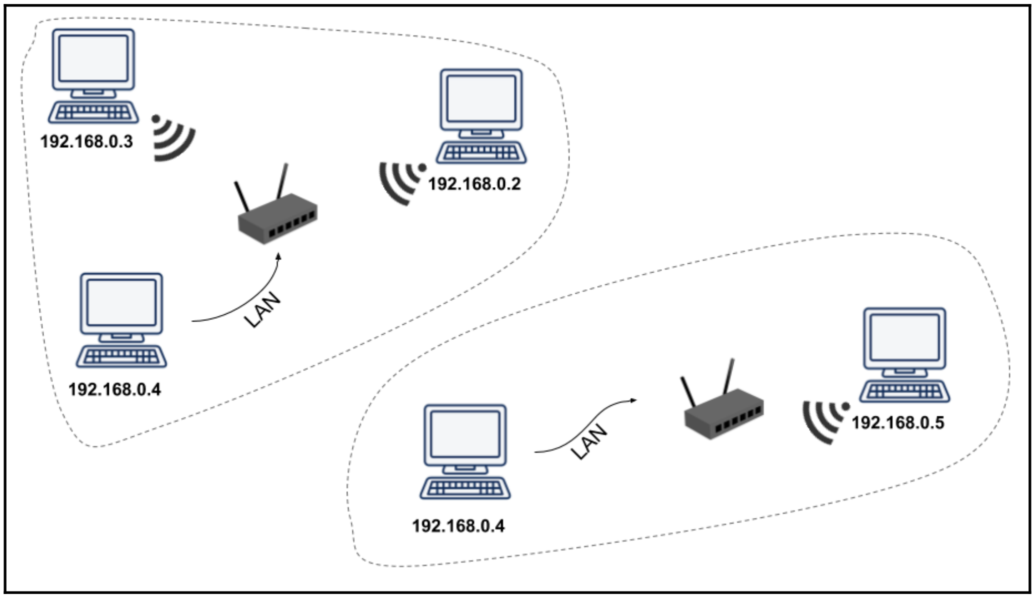
```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using SSH
P3 VNC         Enable/Disable graphical remote access to your Pi using RealVNC
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial connection
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins

               <Select>                                   <Back>
```

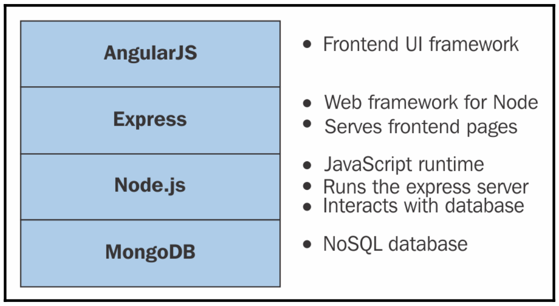


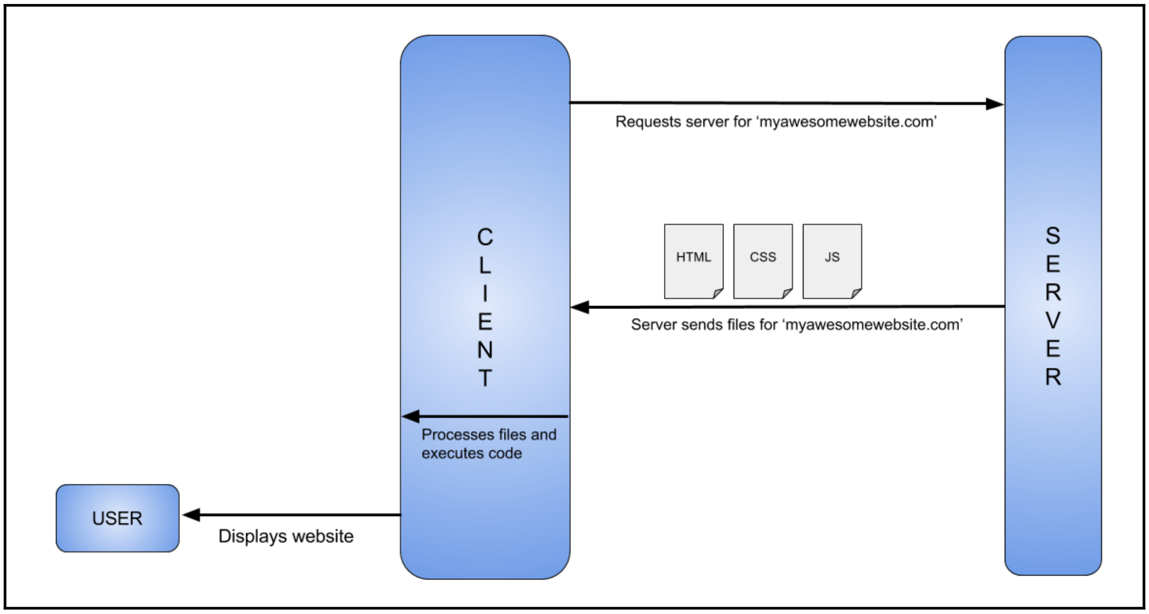
```
Would you like the SSH server to be enabled?

            <Yes>                    <No>
```

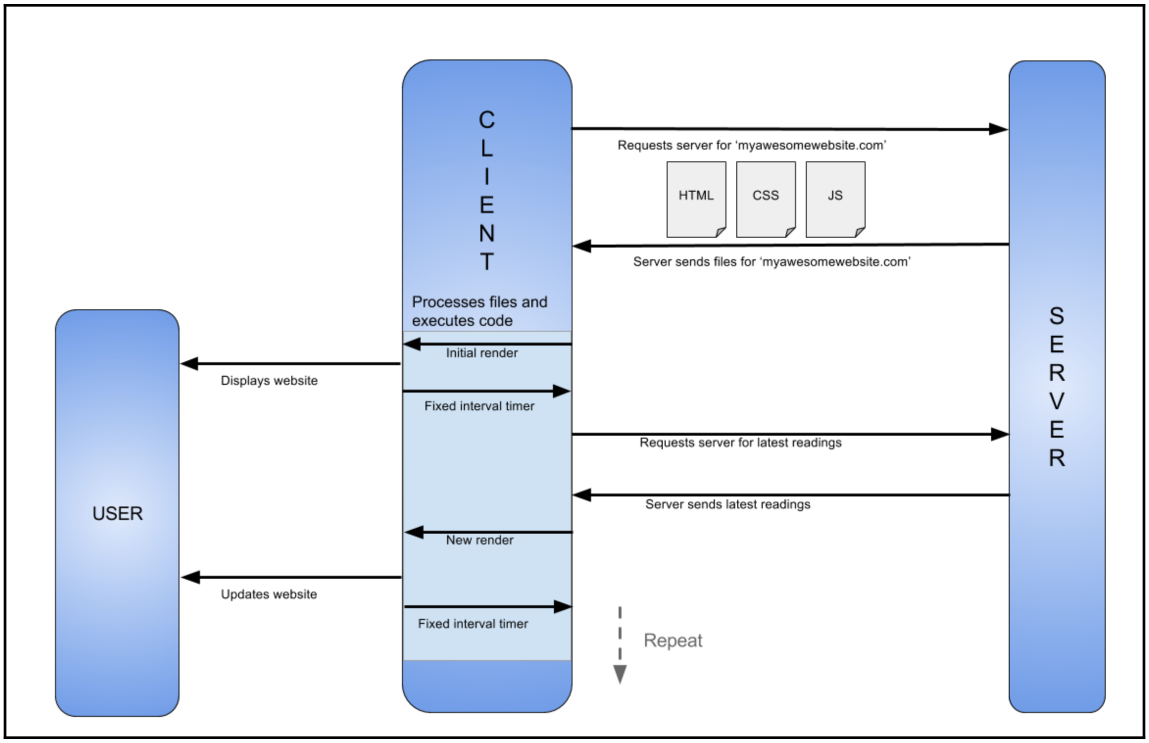
```
pi@raspberrypi:~ $
```

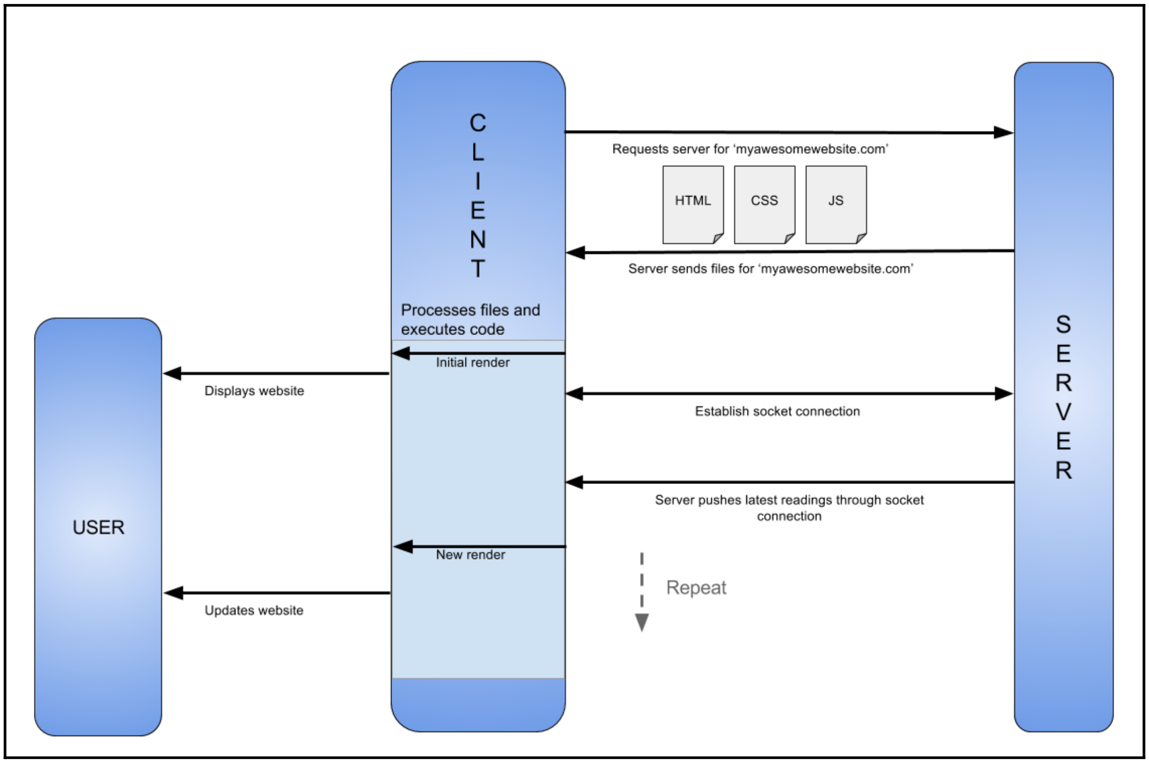
Chapter 2: Getting Up-and-Running with Web Development on the Raspberry Pi

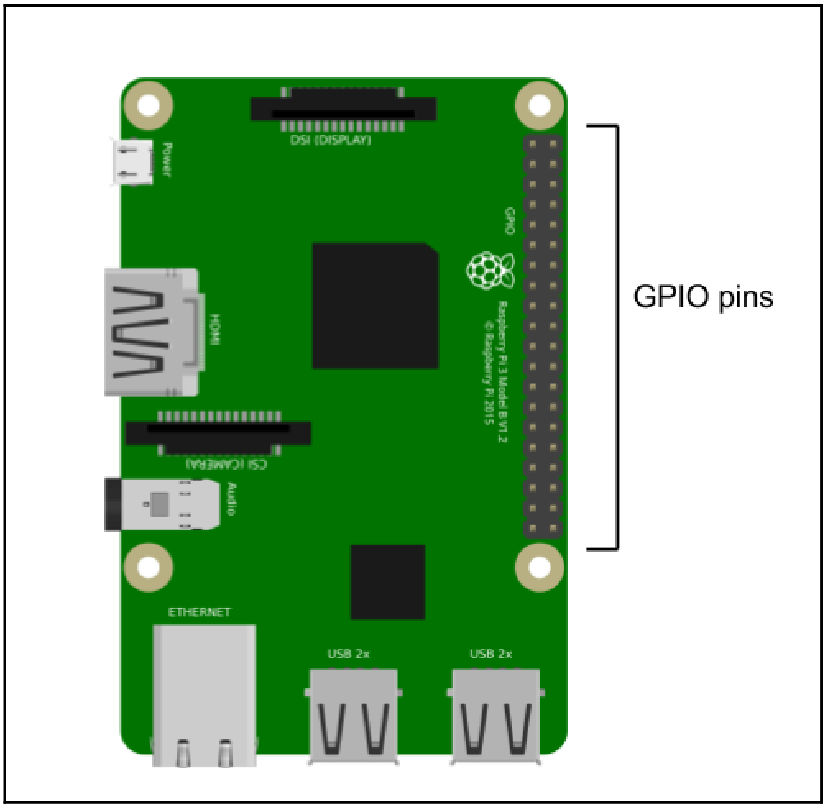


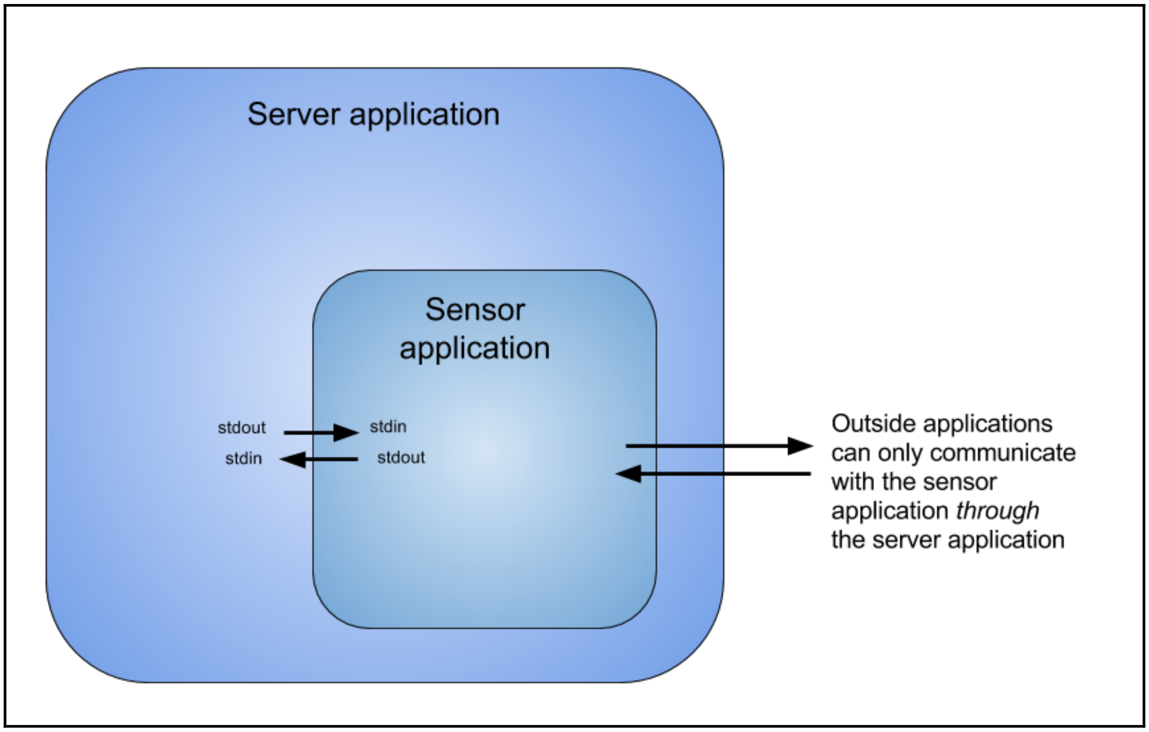


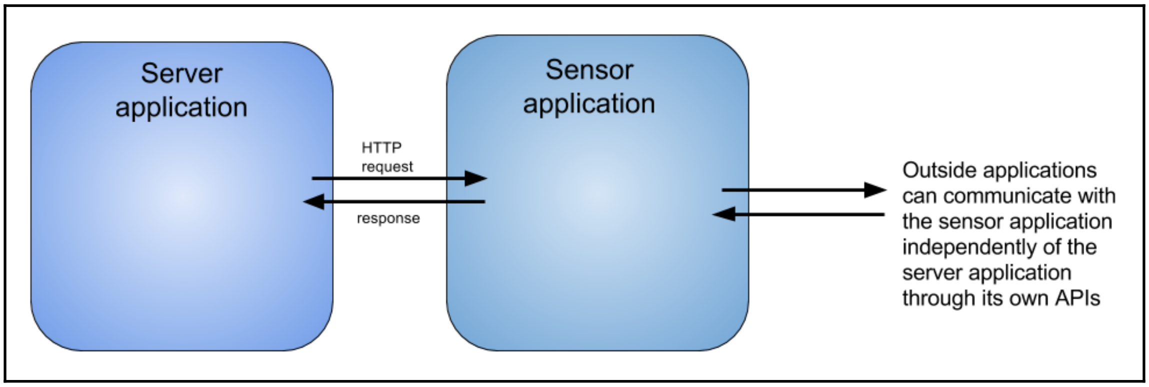

CLIENT
SERVER
Requests server for 'myawesomewebsite.com'
HTML
CSS
JS
Server sends files for 'myawesomewebsite.com'
Processes files and executes code
USER
Displays website

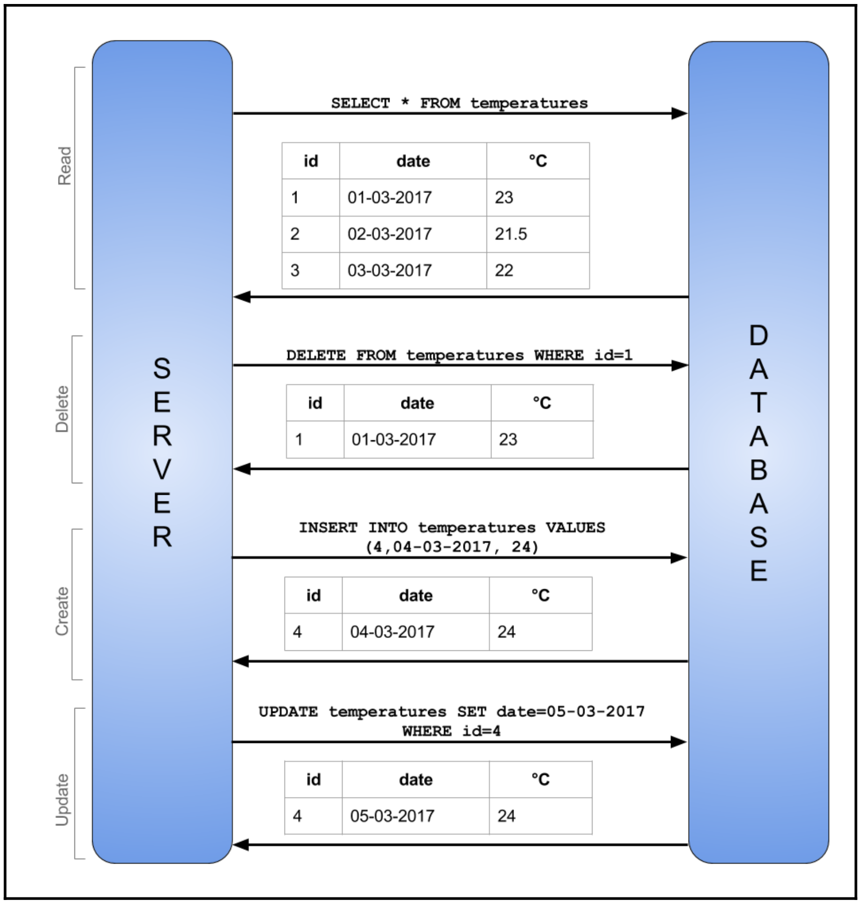

C L I E N T
Processes files and executes code
Requests server for 'myawesomewebsite.com'
HTML
CSS
JS
Server sends files for 'myawesomewebsite.com'
S E R V E R
USER
Initial render
Displays website
Fixed interval timer
Requests server for latest readings
Server sends latest readings
New render
Updates website
Fixed interval timer
Repeat


C L I E N T
Processes files and executes code
S E R V E R
USER
Requests server for 'myawesomewebsite.com'
HTML
CSS
JS
Server sends files for 'myawesomewebsite.com'
Initial render
Displays website
Establish socket connection
Server pushes latest readings through socket connection
New render
Repeat
Updates website


DSI (DISPLAY)
Power
GPIO
HDMI
Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
CSI (CAMERA)
Audio
ETHERNET
USB 2x
USB 2x
GPIO pins


Server application
Sensor application
stdout
stdin
stdin
stdout
Outside applications can only communicate with the sensor application *through* the server application


Server application
Sensor application
HTTP request
response
Outside applications can communicate with the sensor application independently of the server application through its own APIs



SELECT * FROM temperatures

id	date	°C
1	01-03-2017	23
2	02-03-2017	21.5
3	03-03-2017	22

DELETE FROM temperatures WHERE id=1

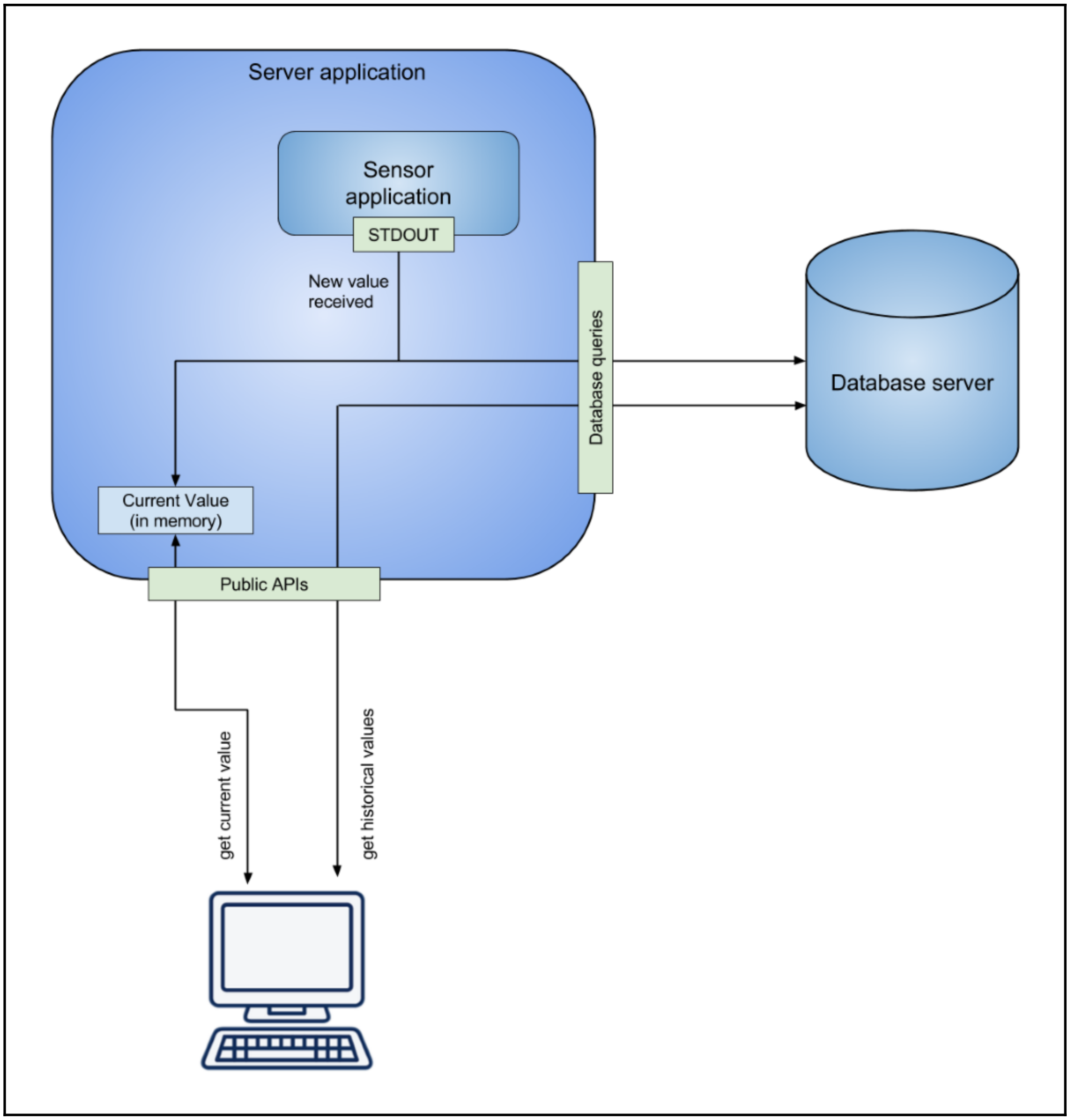
id	date	°C
1	01-03-2017	23

INSERT INTO temperatures VALUES (4,04-03-2017, 24)

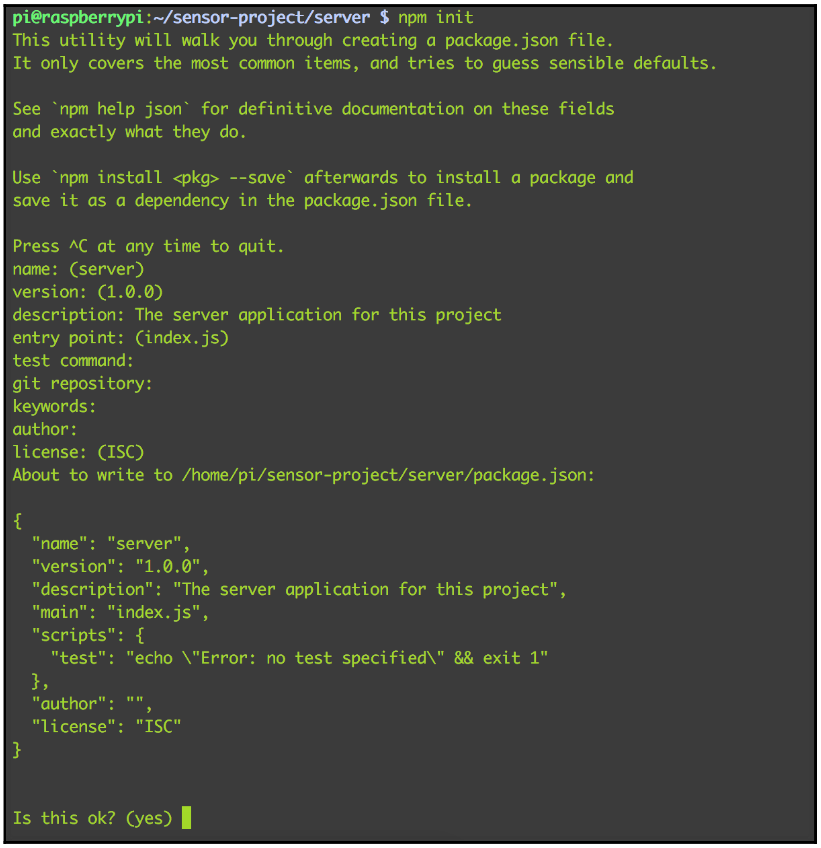
id	date	°C
4	04-03-2017	24

UPDATE temperatures SET date=05-03-2017 WHERE id=4

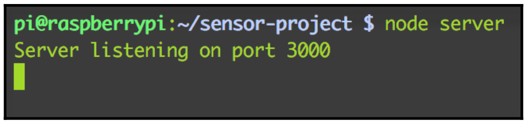
id	date	°C
4	05-03-2017	24


Server application
Sensor application
STDOUT
New value received
Database queries
Database server
Current Value (in memory)
Public APIs
get current value
get historical values

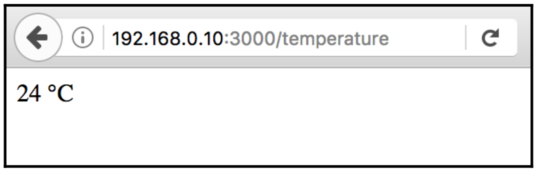
Chapter 3: Running a Node Server on the Pi



```
pi@raspberrypi:~/sensor-project/server $ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (server)
version: (1.0.0)
description: The server application for this project
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/pi/sensor-project/server/package.json:

{
  "name": "server",
  "version": "1.0.0",
  "description": "The server application for this project",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}


Is this ok? (yes)
```

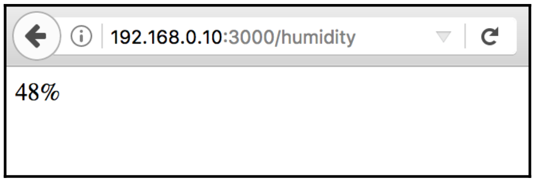


```
pi@raspberrypi:~/sensor-project $ node server
Server listening on port 3000
```

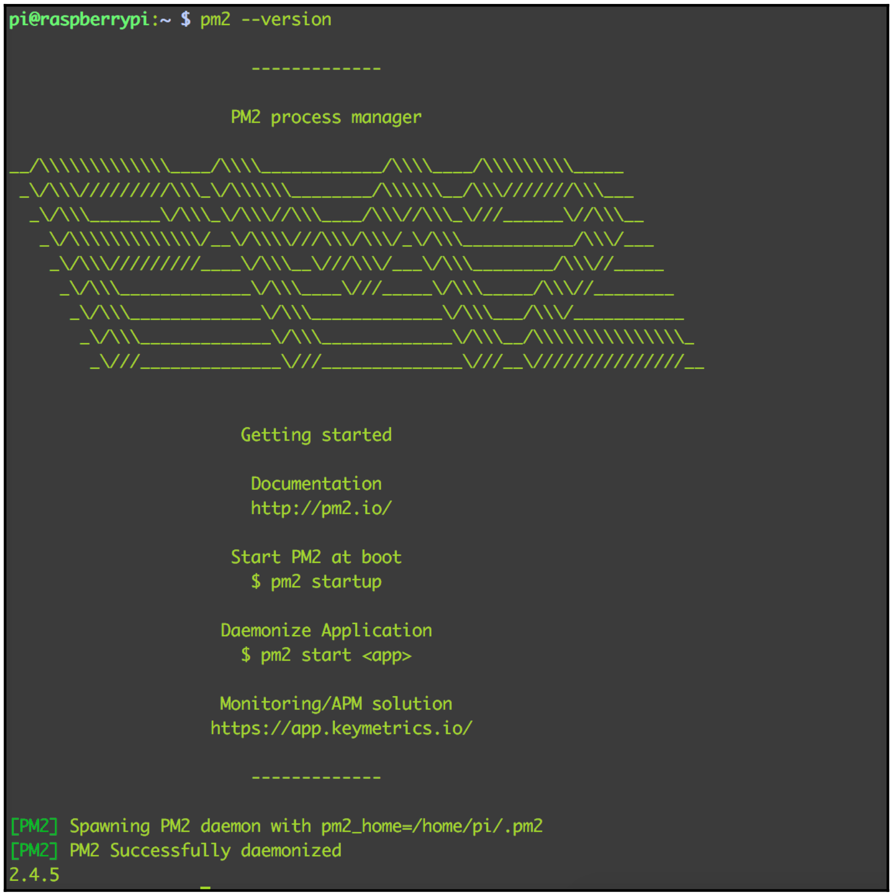


```
192.168.0.10:3000/temperature
```

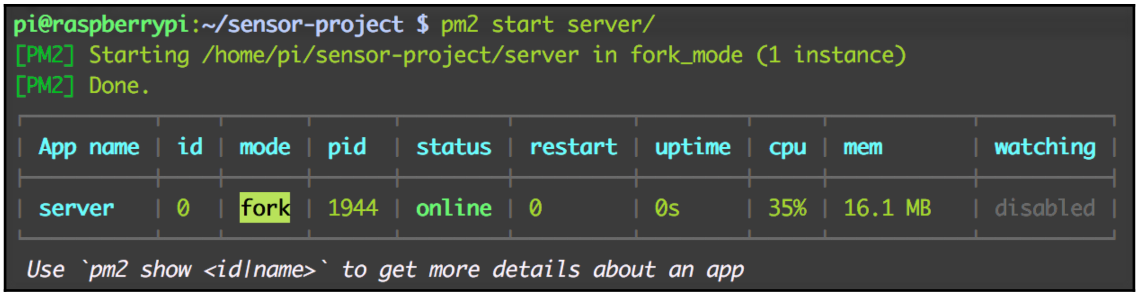
24 °C



```
192.168.0.10:3000/humidity
48%
```

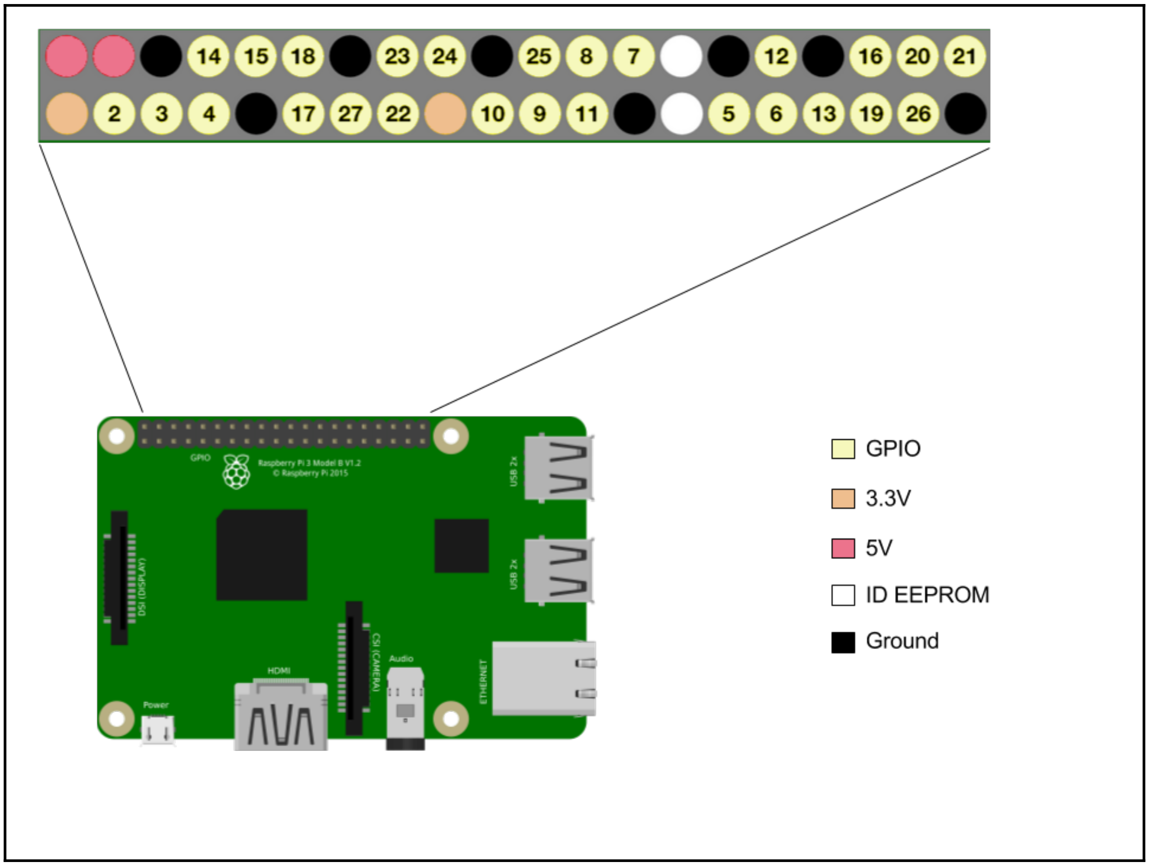


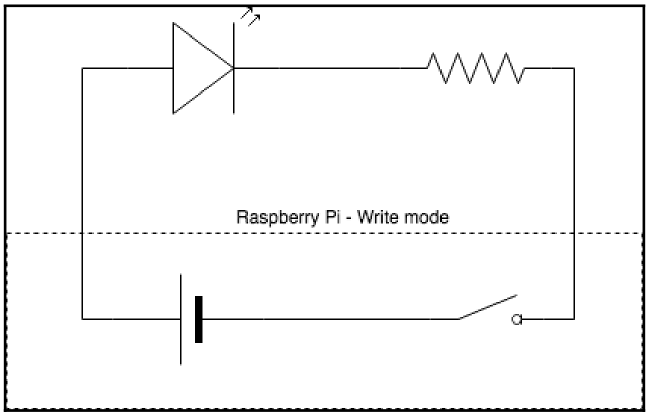
```
pi@raspberrypi:~ $ pm2 --version

                        -------------

                    PM2 process manager

__/\\\\\\\\\\\\\____/\\\\____________/\\\\____/\\\\\\\\\_____
 _\/\\\/////////\\\_\/\\\\\\________/\\\\\\__/\\\///////\\\___
  _\/\\\_______\/\\\_\/\\\//\\\____/\\\//\\\_\///______\//\\\__
   _\/\\\\\\\\\\\\\/__\/\\\\///\\\/\\\/_\/\\\___________/\\\/___
    _\/\\\/////////____\/\\\__\///\\\/___\/\\\________/\\\//_____
     _\/\\\_____________\/\\\____\///_____\/\\\_____/\\\//________
      _\/\\\_____________\/\\\_____________\/\\\___/\\\/___________
       _\/\\\_____________\/\\\_____________\/\\\__/\\\\\\\\\\\\\\\_
        _\///______________\///______________\///__\///////////////__


                       Getting started

                        Documentation
                        http://pm2.io/

                      Start PM2 at boot
                        $ pm2 startup

                    Daemonize Application
                      $ pm2 start <app>

                   Monitoring/APM solution
                 https://app.keymetrics.io/

                        -------------

[PM2] Spawning PM2 daemon with pm2_home=/home/pi/.pm2
[PM2] PM2 Successfully daemonized
2.4.5
```

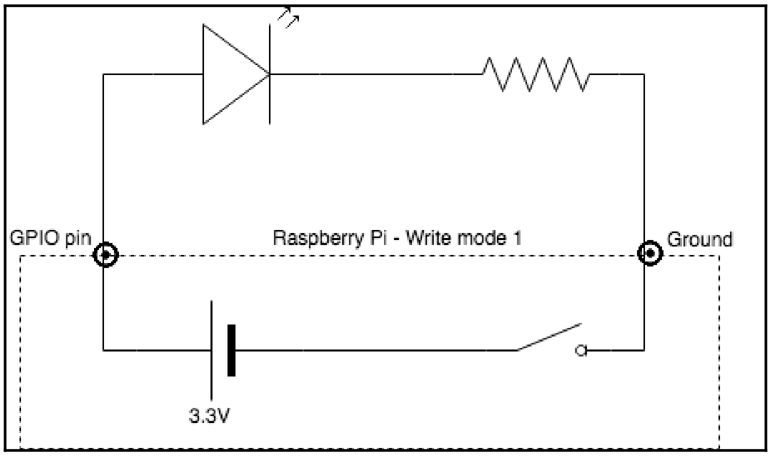


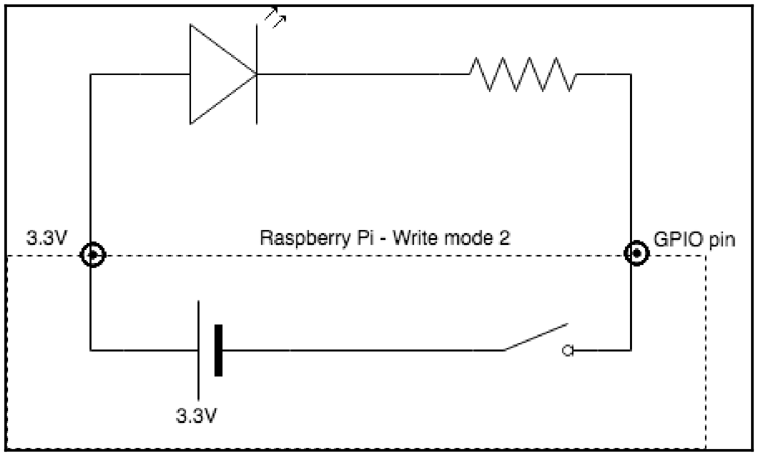
```
pi@raspberrypi:~/sensor-project $ pm2 start server/
[PM2] Starting /home/pi/sensor-project/server in fork_mode (1 instance)
[PM2] Done.
┌──────────┬────┬──────┬──────┬────────┬─────────┬────────┬─────┬─────────┬──────────┐
│ App name │ id │ mode │ pid  │ status │ restart │ uptime │ cpu │ mem     │ watching │
├──────────┼────┼──────┼──────┼────────┼─────────┼────────┼─────┼─────────┼──────────┤
│ server   │ 0  │ fork │ 1944 │ online │ 0       │ 0s     │ 35% │ 16.1 MB │ disabled │
└──────────┴────┴──────┴──────┴────────┴─────────┴────────┴─────┴─────────┴──────────┘
 Use `pm2 show <id|name>` to get more details about an app
```

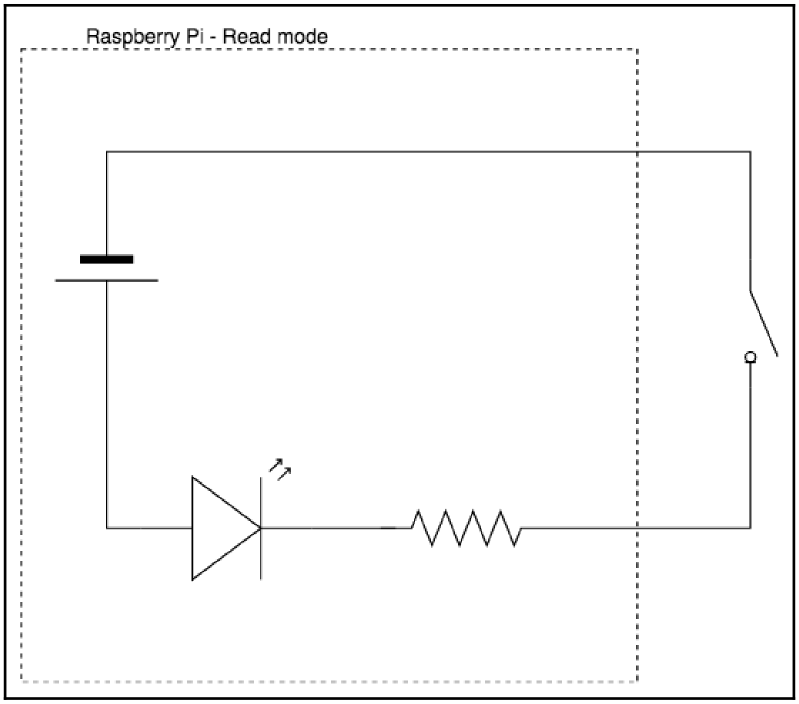
Chapter 4: Extracting Information from the GPIO Pins

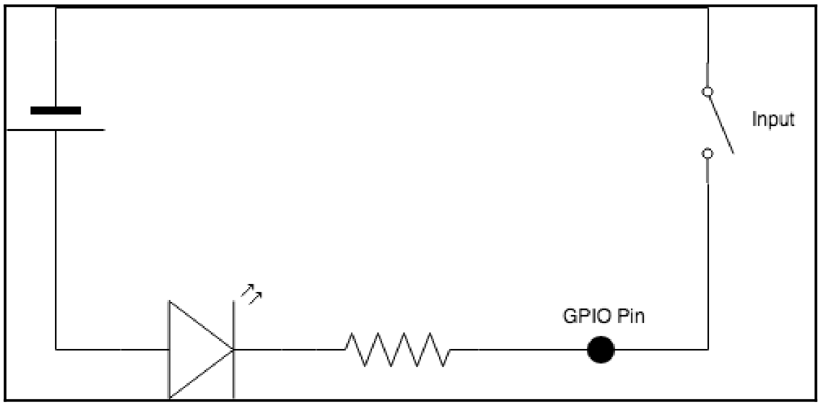


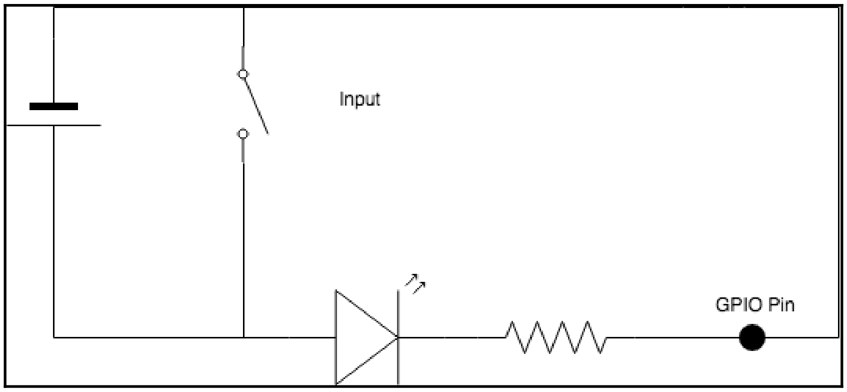

Raspberry Pi - Write mode

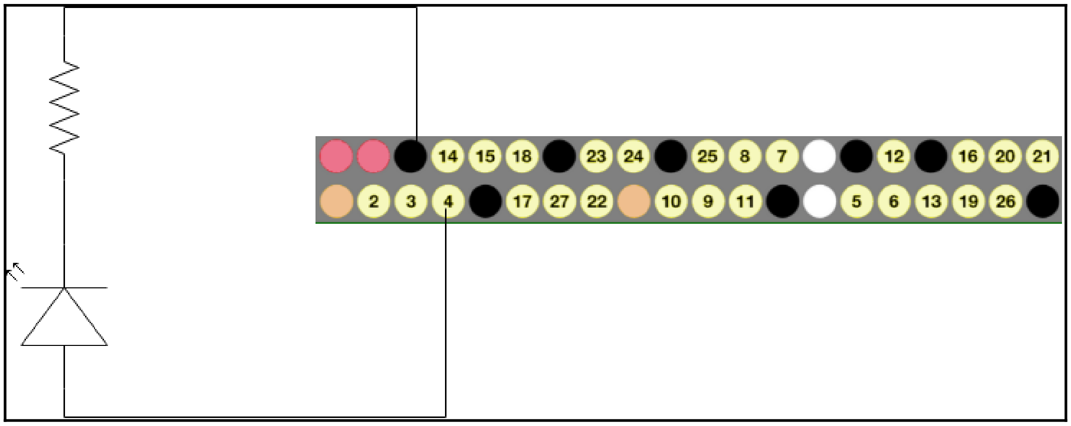

GPIO pin
Raspberry Pi - Write mode 1
Ground
3.3V

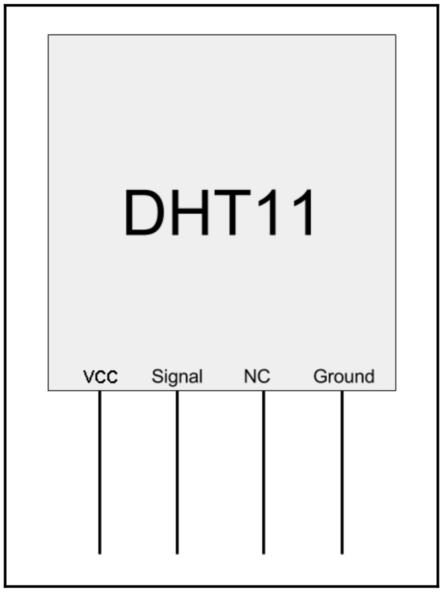

3.3V
Raspberry Pi - Write mode 2
GPIO pin
3.3V

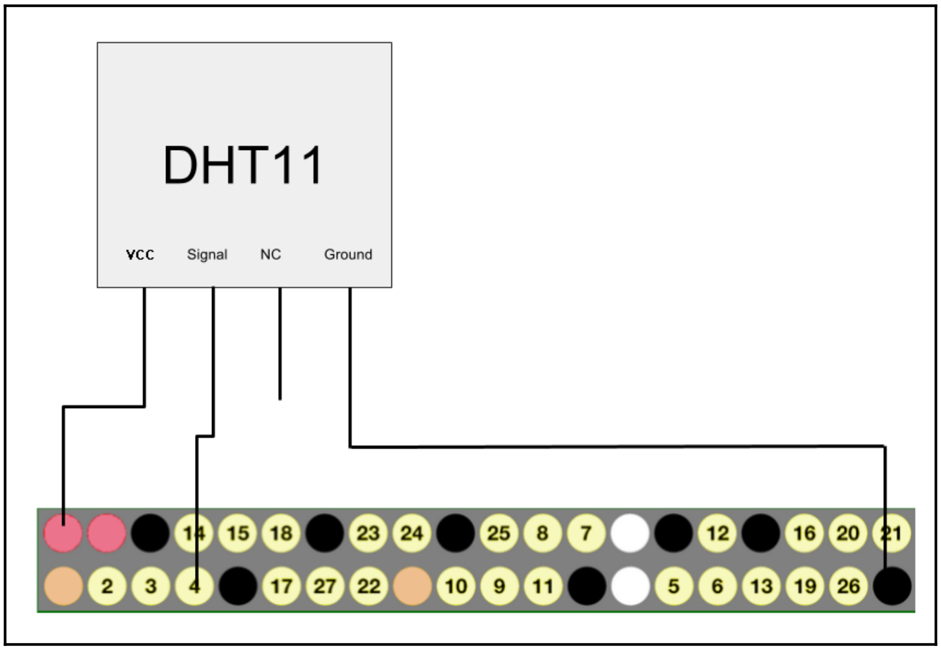

Raspberry Pi - Read mode

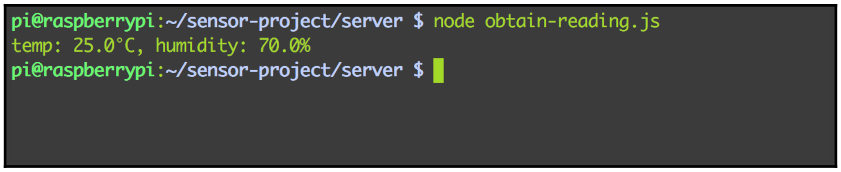

Input
GPIO Pin


Input
GPIO Pin


14 15 18 23 24 25 8 7 12 16 20 21
2 3 4 17 27 22 10 9 11 5 6 13 19 26

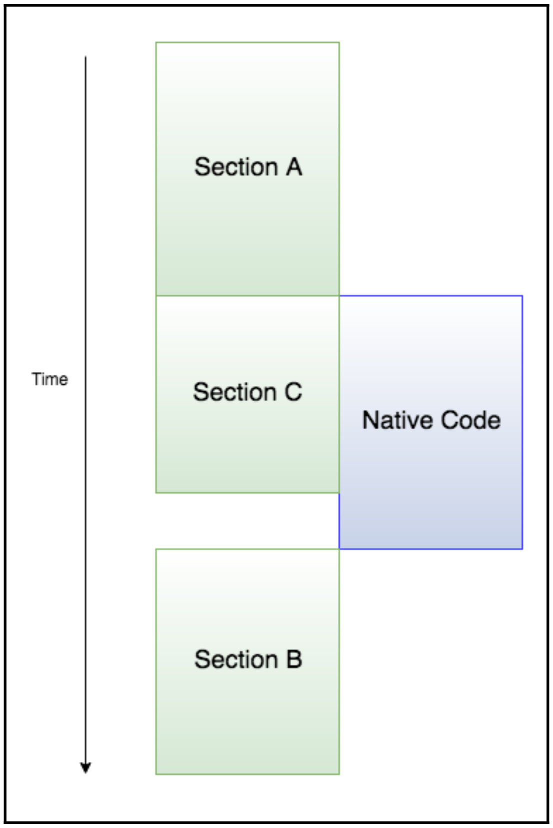

DHT11
VCC
Signal
NC
Ground

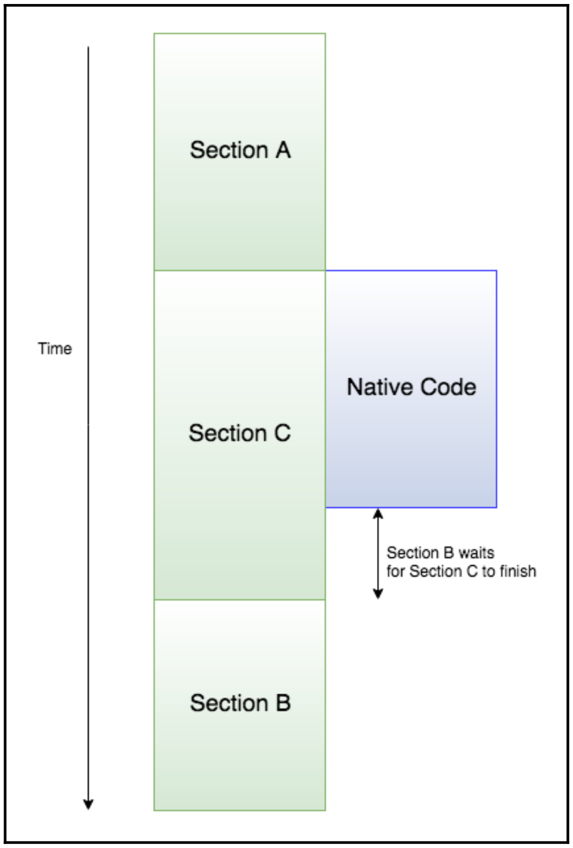

DHT11
VCC
Signal
NC
Ground
14
15
18
23
24
25
8
7
12
16
20
21
2
3
4
17
27
22
10
9
11
5
6
13
19
26

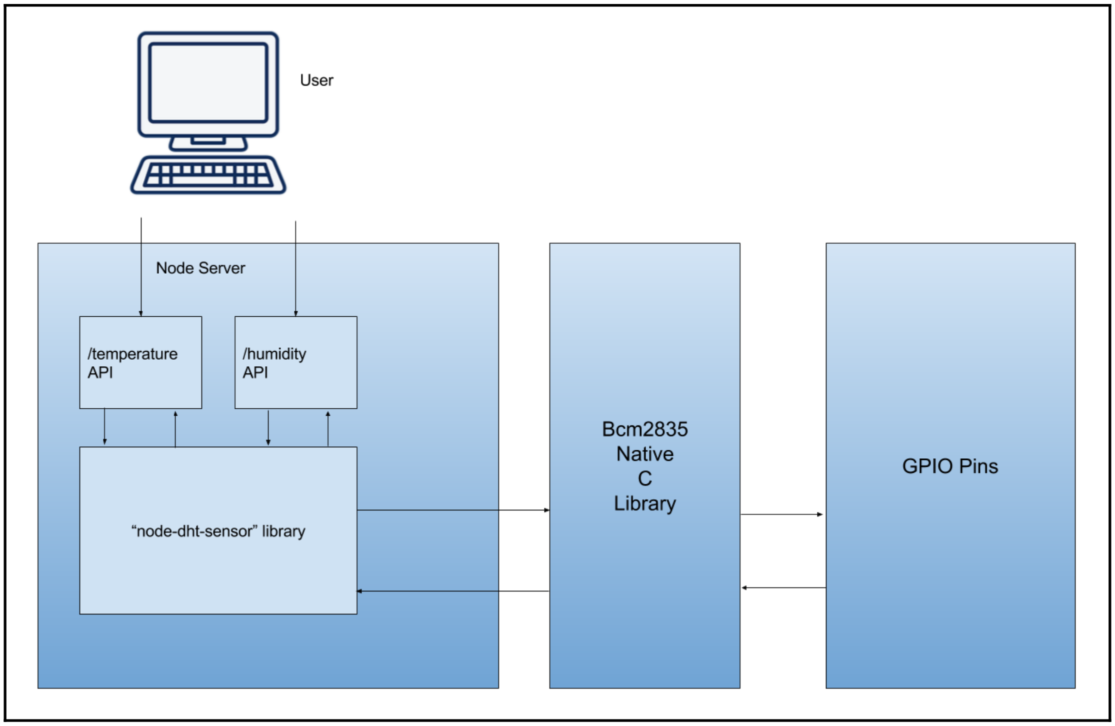


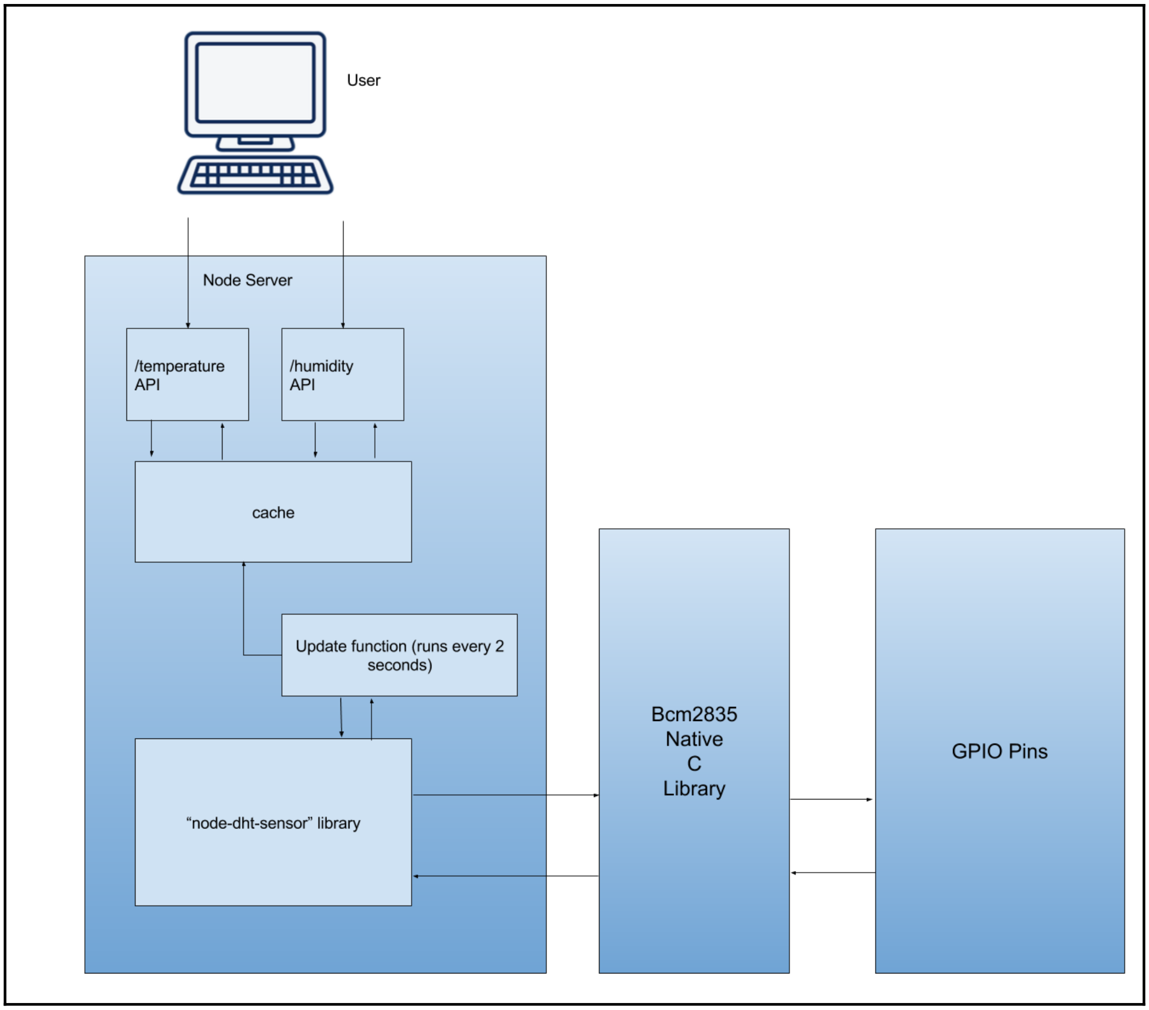
```
pi@raspberrypi:~/sensor-project/server $ node obtain-reading.js
temp: 25.0°C, humidity: 70.0%
pi@raspberrypi:~/sensor-project/server $
```

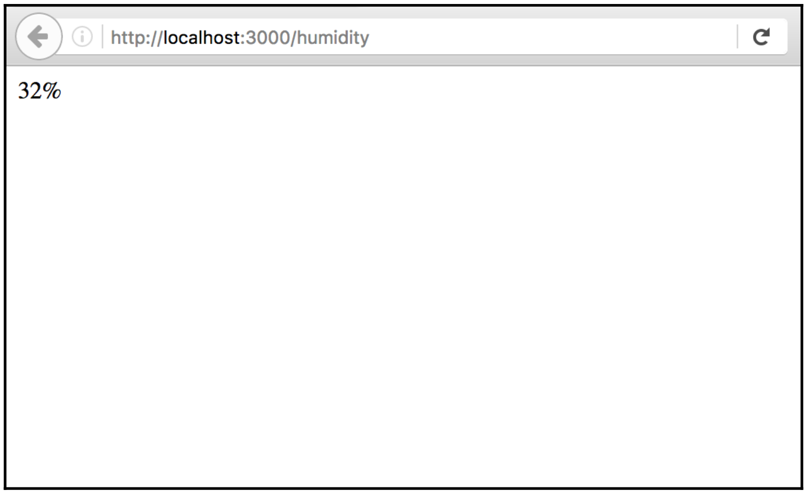
Chapter 5: Retrieving Sensor Readings from the Server




Time
Section A
Native Code
Section C
Section B waits
for Section C to finish
Section B

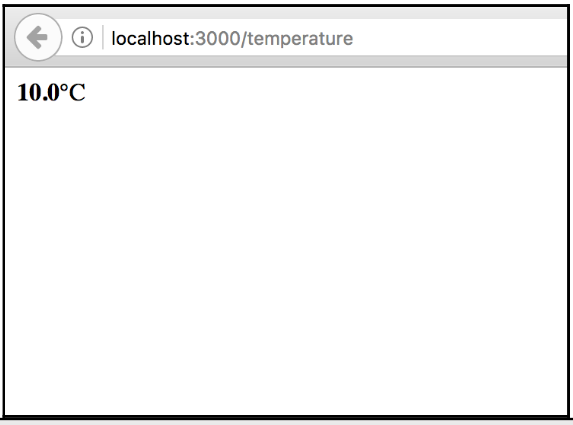

User
Node Server
/temperature API
/humidity API
"node-dht-sensor" library
Bcm2835 Native C Library
GPIO Pins


User
Node Server
/temperature API
/humidity API
cache
Update function (runs every 2 seconds)
"node-dht-sensor" library
Bcm2835 Native C Library
GPIO Pins

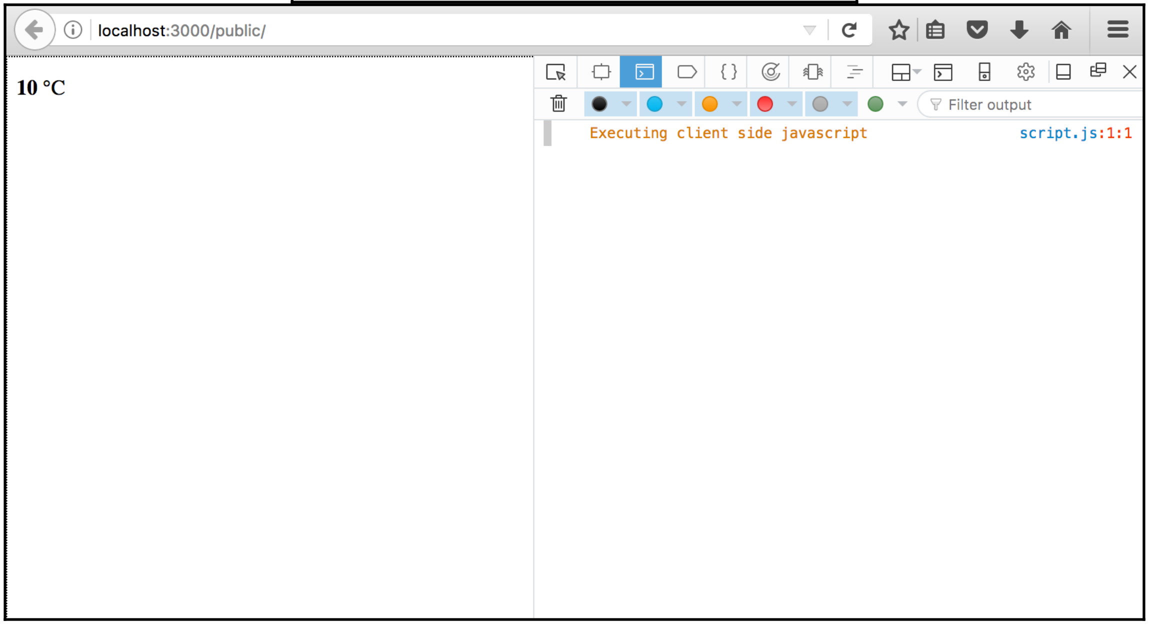


```
32%
```

Chapter 6: Creating a Web Page to Display Sensor Data

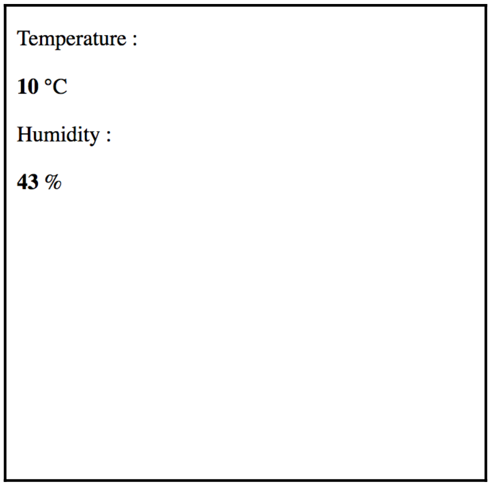

localhost:3000/temperature

10.0°C


localhost:3000/public/

10 °C

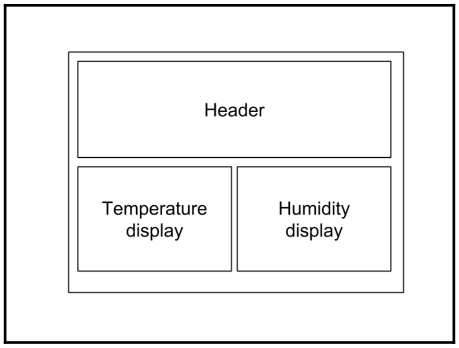
```
Executing client side javascript                    script.js:1:1
```

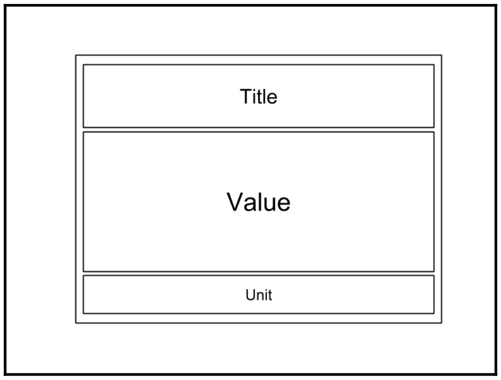

Temperature :

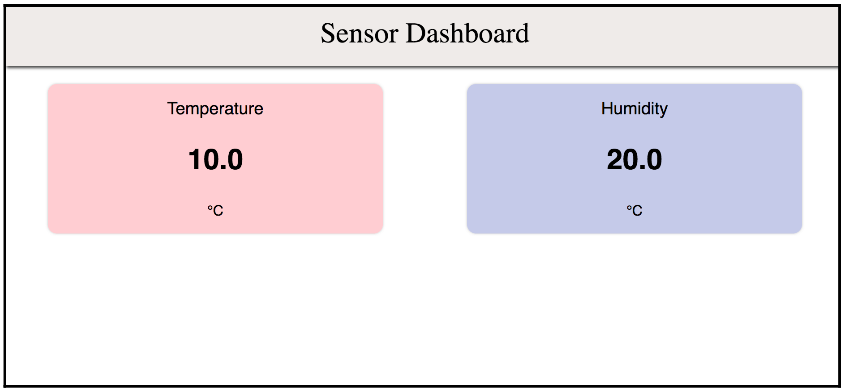
10 °C

Humidity :

43 %







Sensor Dashboard

Temperature

10.0

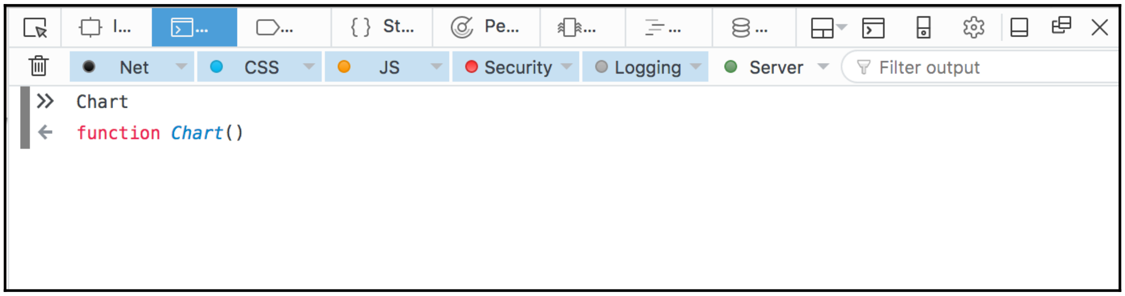
°C

Humidity

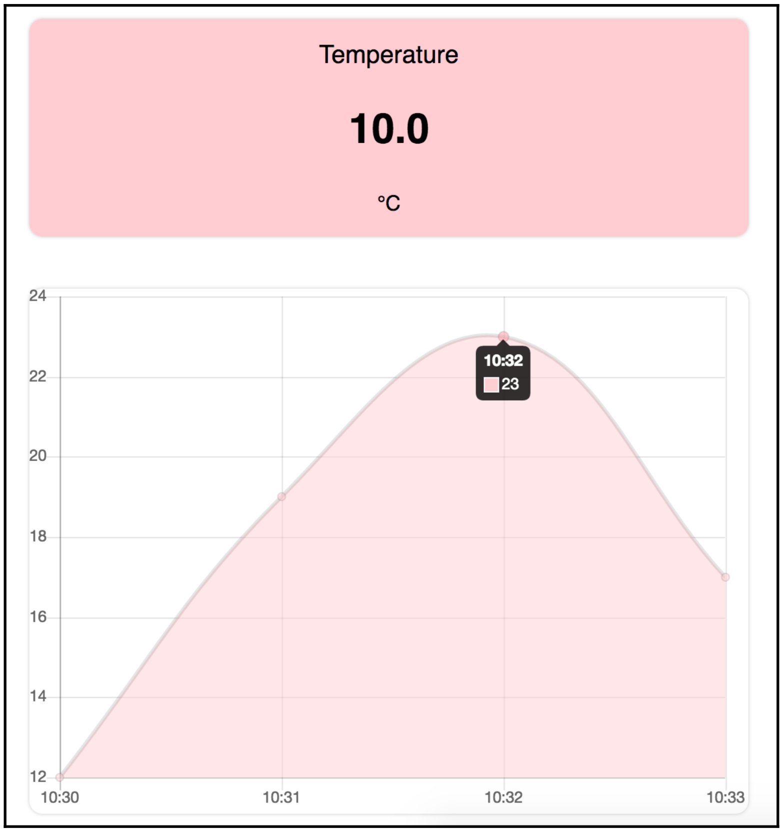
20.0

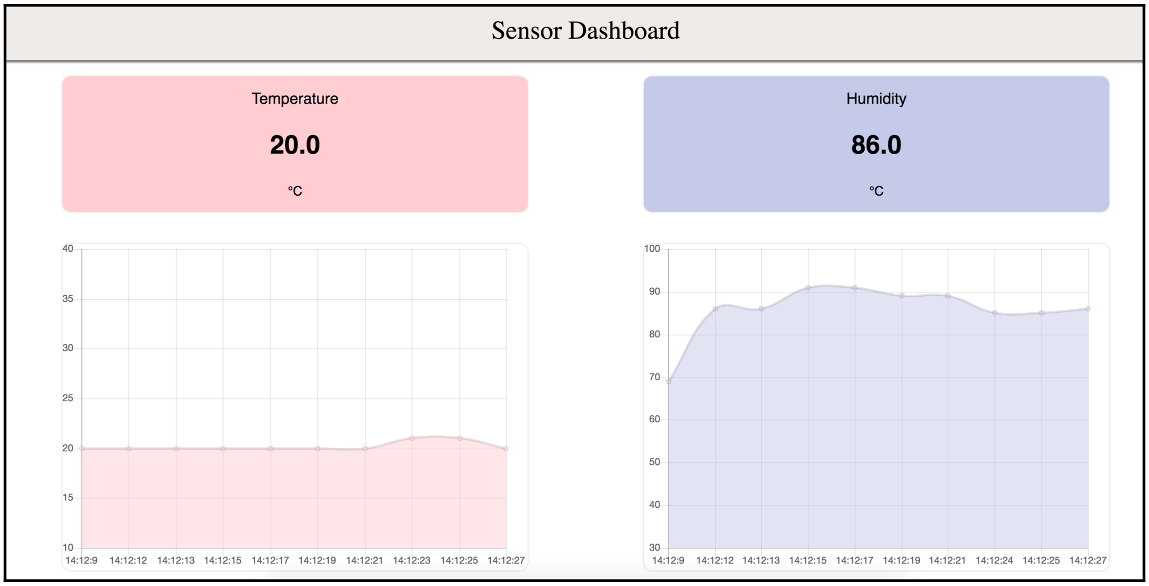
°C

Chapter 7: Enhancing Our UI - Using Interactive Charts

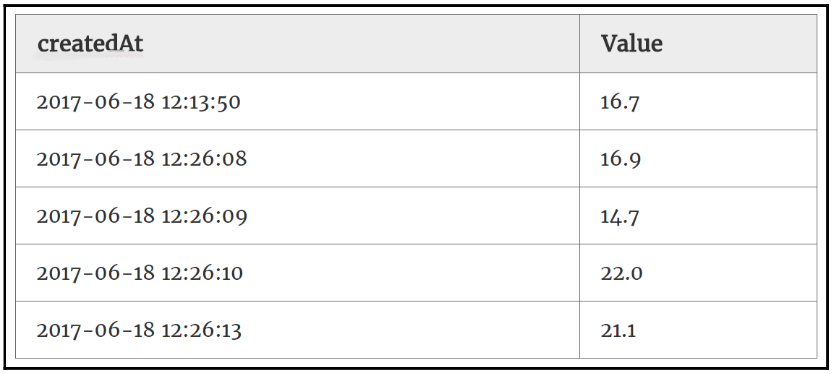


```
>> Chart
<- function Chart()
```

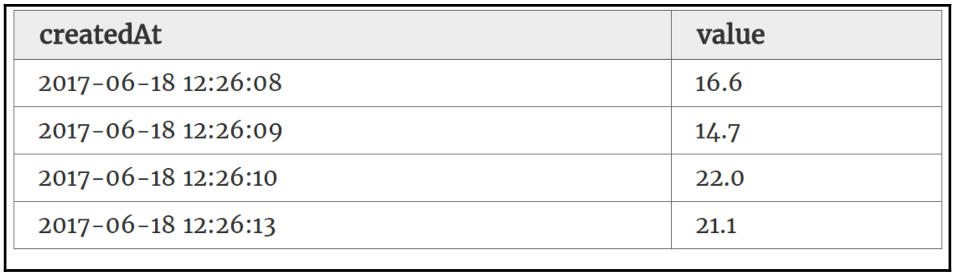



Sensor Dashboard
Temperature
20.0
°C
Humidity
86.0
°C
40
35
30
25
20
15
10
14:12:9
14:12:12
14:12:13
14:12:15
14:12:17
14:12:19
14:12:21
14:12:23
14:12:25
14:12:27
100
90
80
70
60
50
40
30
14:12:9
14:12:12
14:12:13
14:12:15
14:12:17
14:12:19
14:12:21
14:12:24
14:12:25
14:12:27

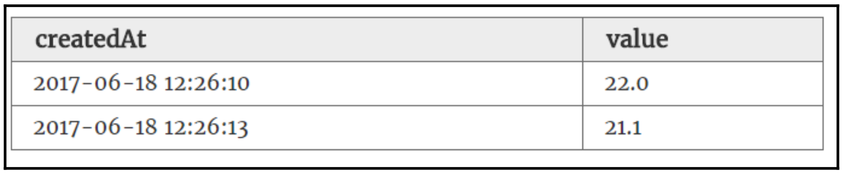
Chapter 8: SQLite - The Fast and Portable Database



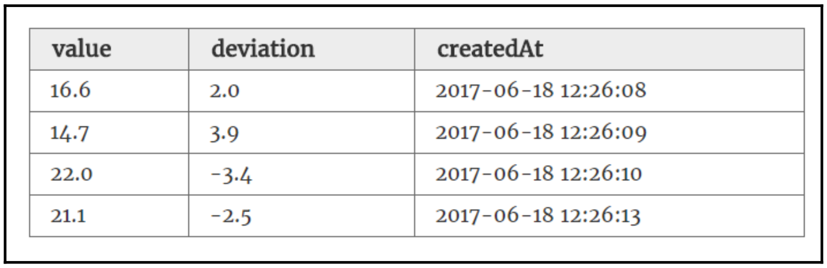
createdAt	Value
2017-06-18 12:13:50	16.7
2017-06-18 12:26:08	16.9
2017-06-18 12:26:09	14.7
2017-06-18 12:26:10	22.0
2017-06-18 12:26:13	21.1



createdAt	value
2017-06-18 12:26:08	16.6
2017-06-18 12:26:09	14.7
2017-06-18 12:26:10	22.0
2017-06-18 12:26:13	21.1

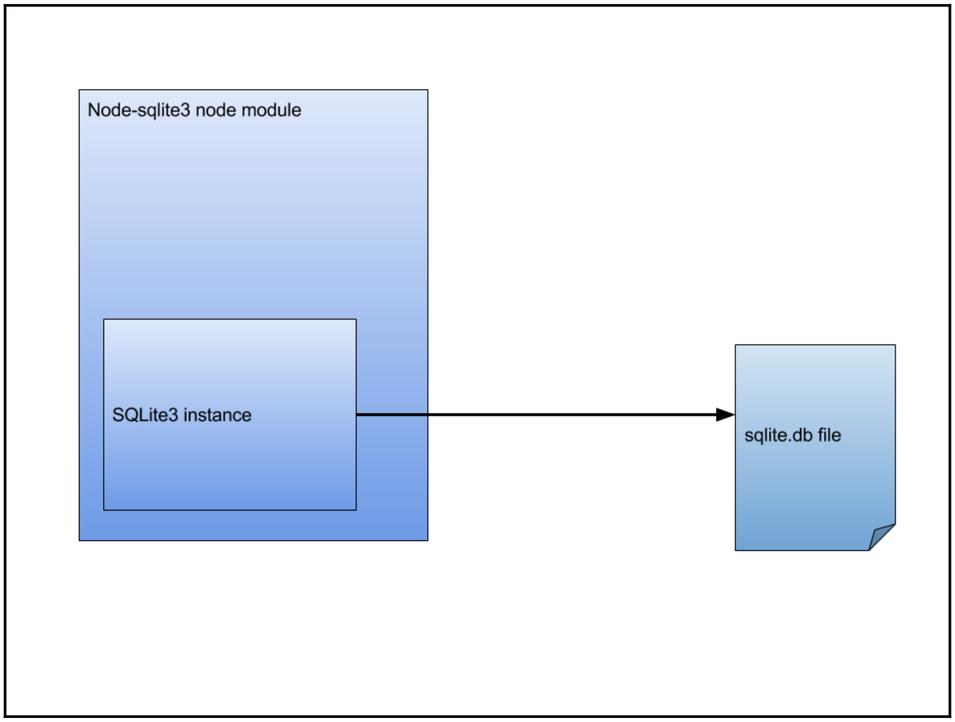


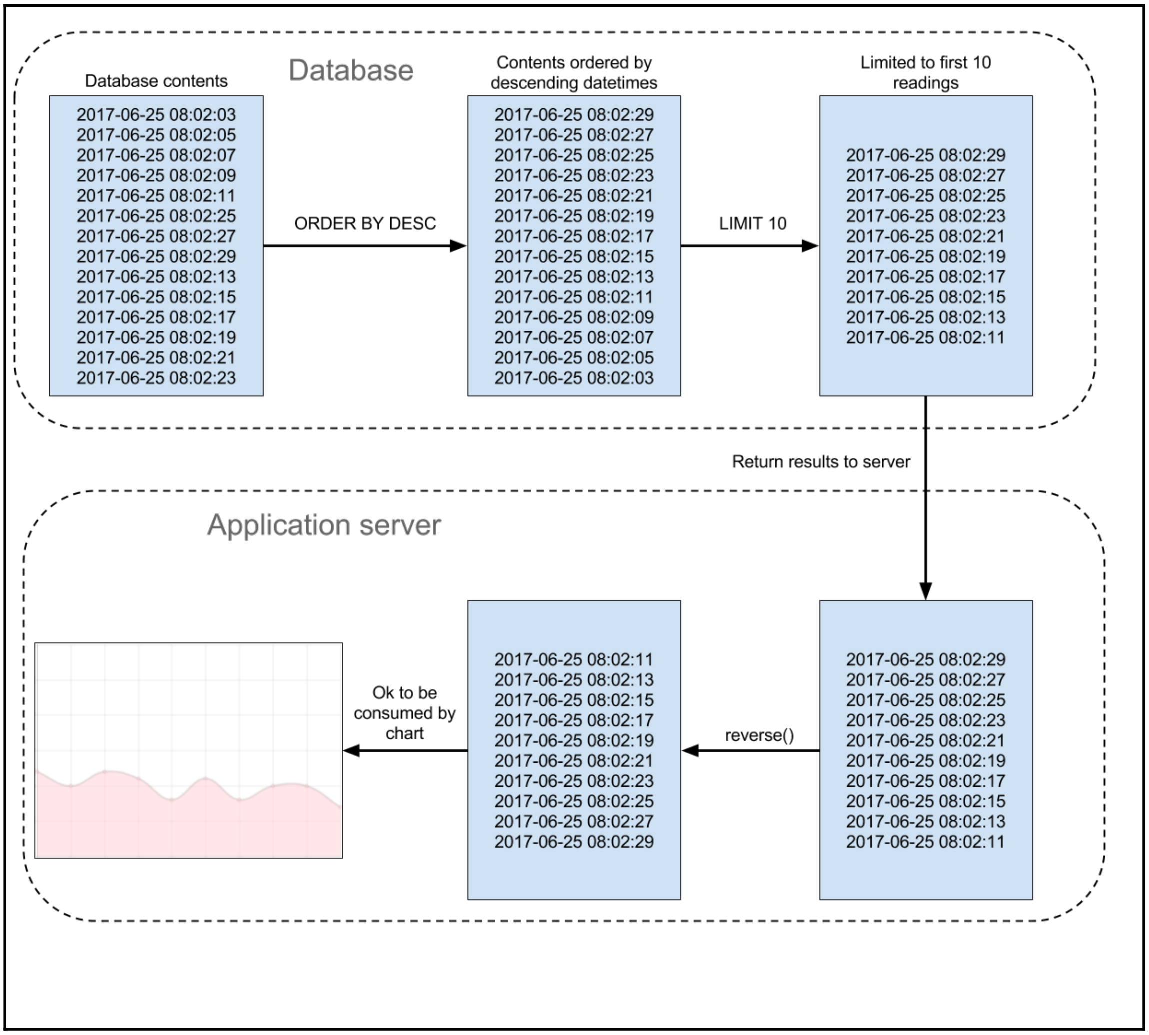
createdAt	value
2017-06-18 12:26:10	22.0
2017-06-18 12:26:13	21.1

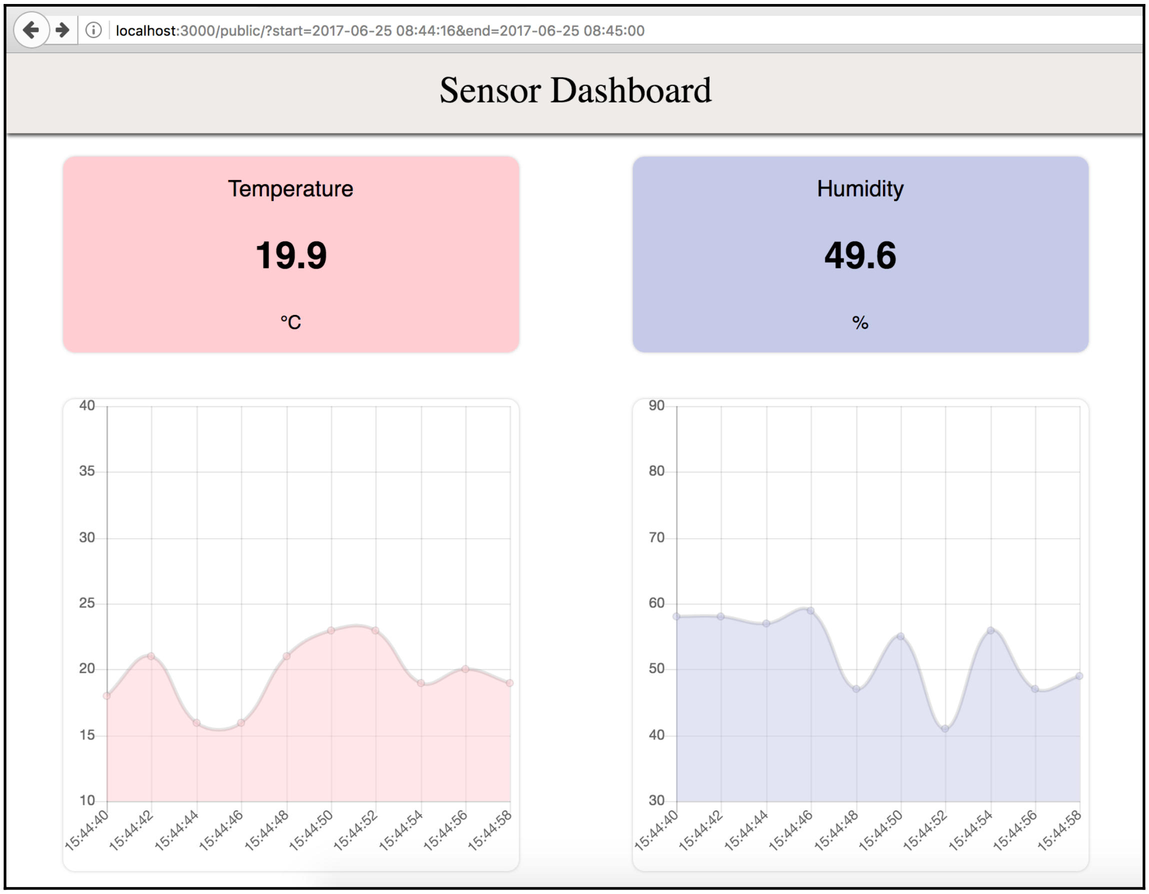


value	deviation	createdAt
16.6	2.0	2017-06-18 12:26:08
14.7	3.9	2017-06-18 12:26:09
22.0	-3.4	2017-06-18 12:26:10
21.1	-2.5	2017-06-18 12:26:13

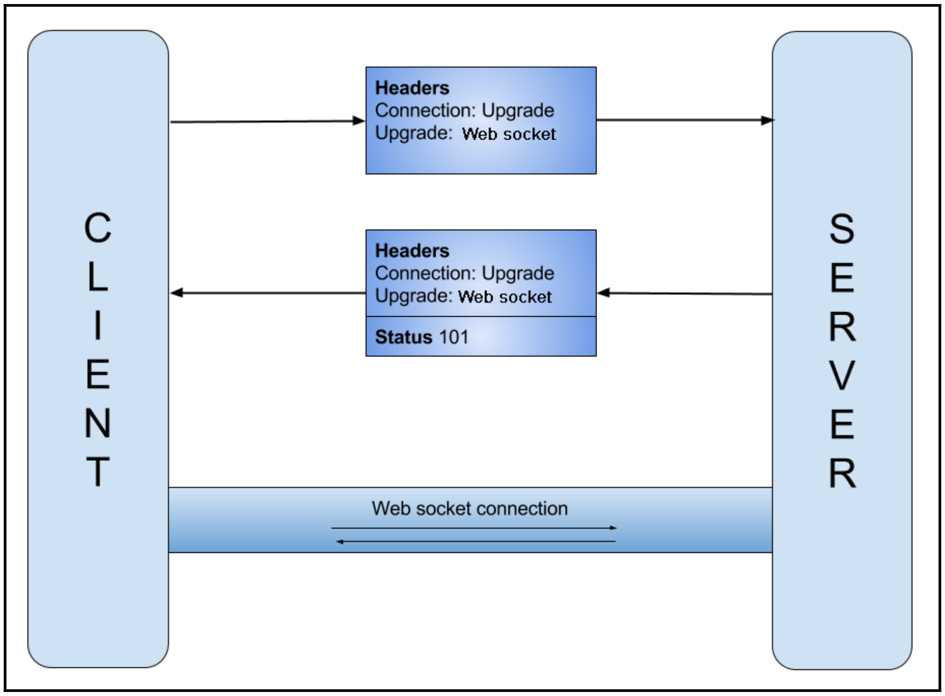
Chapter 9: Integrating SQLite into Our Application

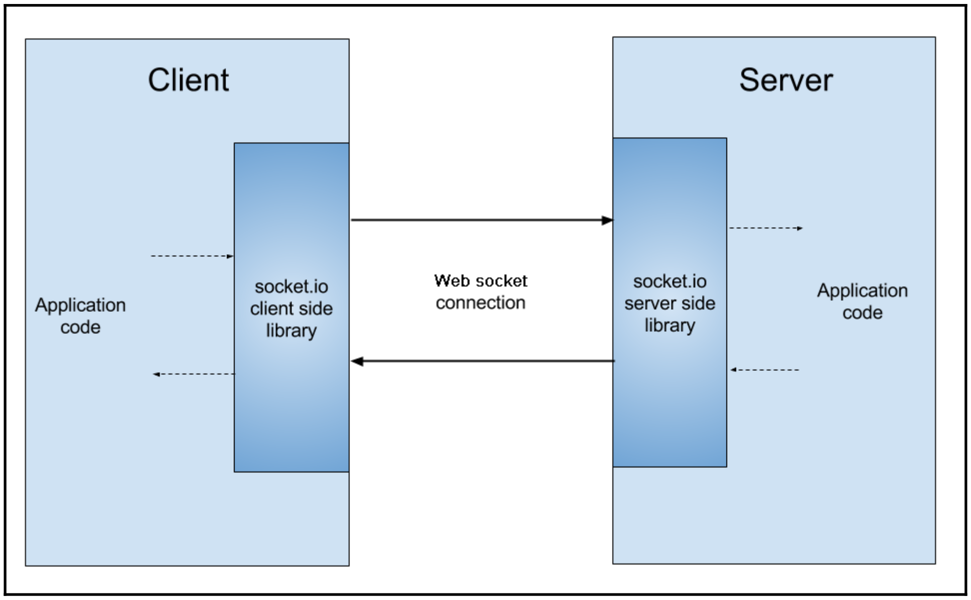


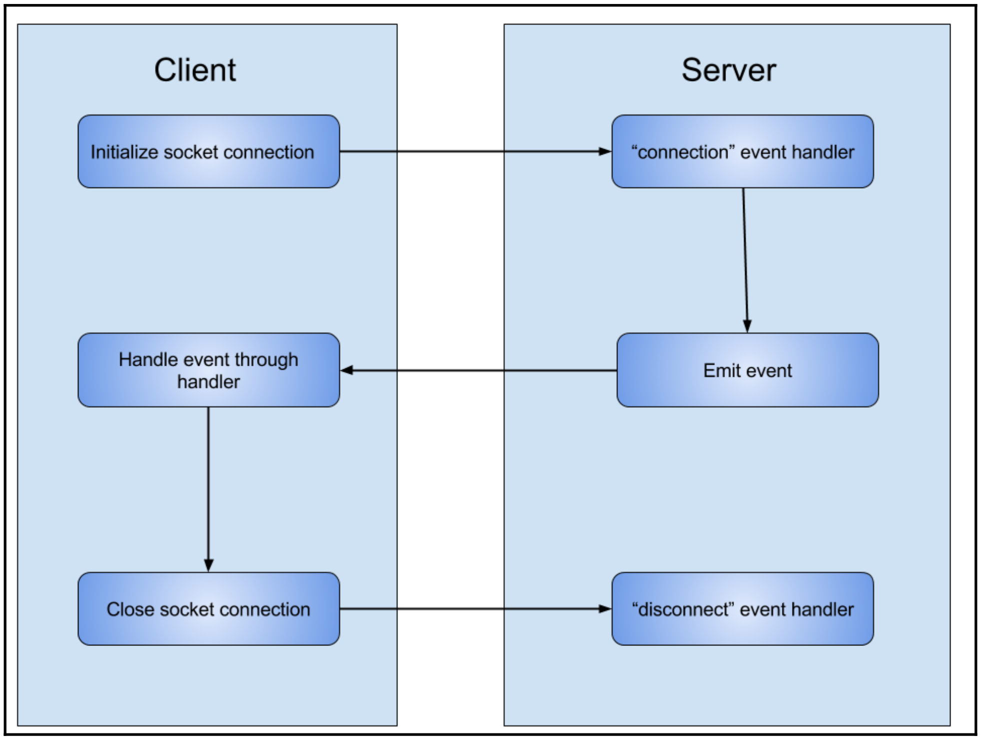

Database
Database contents
2017-06-25 08:02:03
2017-06-25 08:02:05
2017-06-25 08:02:07
2017-06-25 08:02:09
2017-06-25 08:02:11
2017-06-25 08:02:25
2017-06-25 08:02:27
2017-06-25 08:02:29
2017-06-25 08:02:13
2017-06-25 08:02:15
2017-06-25 08:02:17
2017-06-25 08:02:19
2017-06-25 08:02:21
2017-06-25 08:02:23
ORDER BY DESC
Contents ordered by descending datetimes
2017-06-25 08:02:29
2017-06-25 08:02:27
2017-06-25 08:02:25
2017-06-25 08:02:23
2017-06-25 08:02:21
2017-06-25 08:02:19
2017-06-25 08:02:17
2017-06-25 08:02:15
2017-06-25 08:02:13
2017-06-25 08:02:11
2017-06-25 08:02:09
2017-06-25 08:02:07
2017-06-25 08:02:05
2017-06-25 08:02:03
LIMIT 10
Limited to first 10 readings
2017-06-25 08:02:29
2017-06-25 08:02:27
2017-06-25 08:02:25
2017-06-25 08:02:23
2017-06-25 08:02:21
2017-06-25 08:02:19
2017-06-25 08:02:17
2017-06-25 08:02:15
2017-06-25 08:02:13
2017-06-25 08:02:11
Return results to server
Application server
2017-06-25 08:02:29
2017-06-25 08:02:27
2017-06-25 08:02:25
2017-06-25 08:02:23
2017-06-25 08:02:21
2017-06-25 08:02:19
2017-06-25 08:02:17
2017-06-25 08:02:15
2017-06-25 08:02:13
2017-06-25 08:02:11
reverse()
2017-06-25 08:02:11
2017-06-25 08:02:13
2017-06-25 08:02:15
2017-06-25 08:02:17
2017-06-25 08:02:19
2017-06-25 08:02:21
2017-06-25 08:02:23
2017-06-25 08:02:25
2017-06-25 08:02:27
2017-06-25 08:02:29
Ok to be consumed by chart

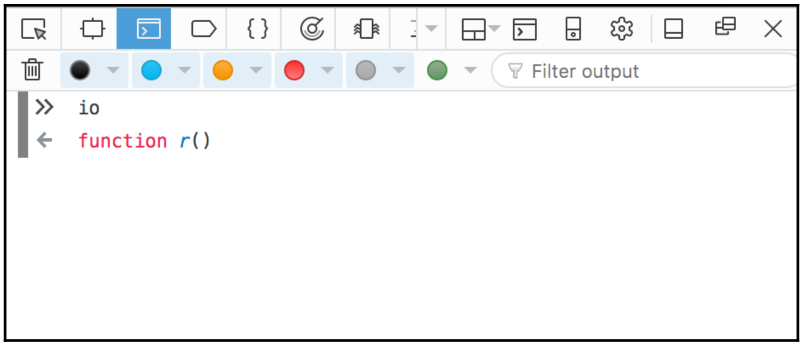

localhost:3000/public/?start=2017-06-25 08:44:16&end=2017-06-25 08:45:00
Sensor Dashboard
Temperature
19.9
°C
Humidity
49.6
%
40
35
30
25
20
15
10
15:44:40
15:44:42
15:44:44
15:44:46
15:44:48
15:44:50
15:44:52
15:44:54
15:44:56
15:44:58
90
80
70
60
50
40
30

Chapter 10: Making our Application Real Time with Web Sockets

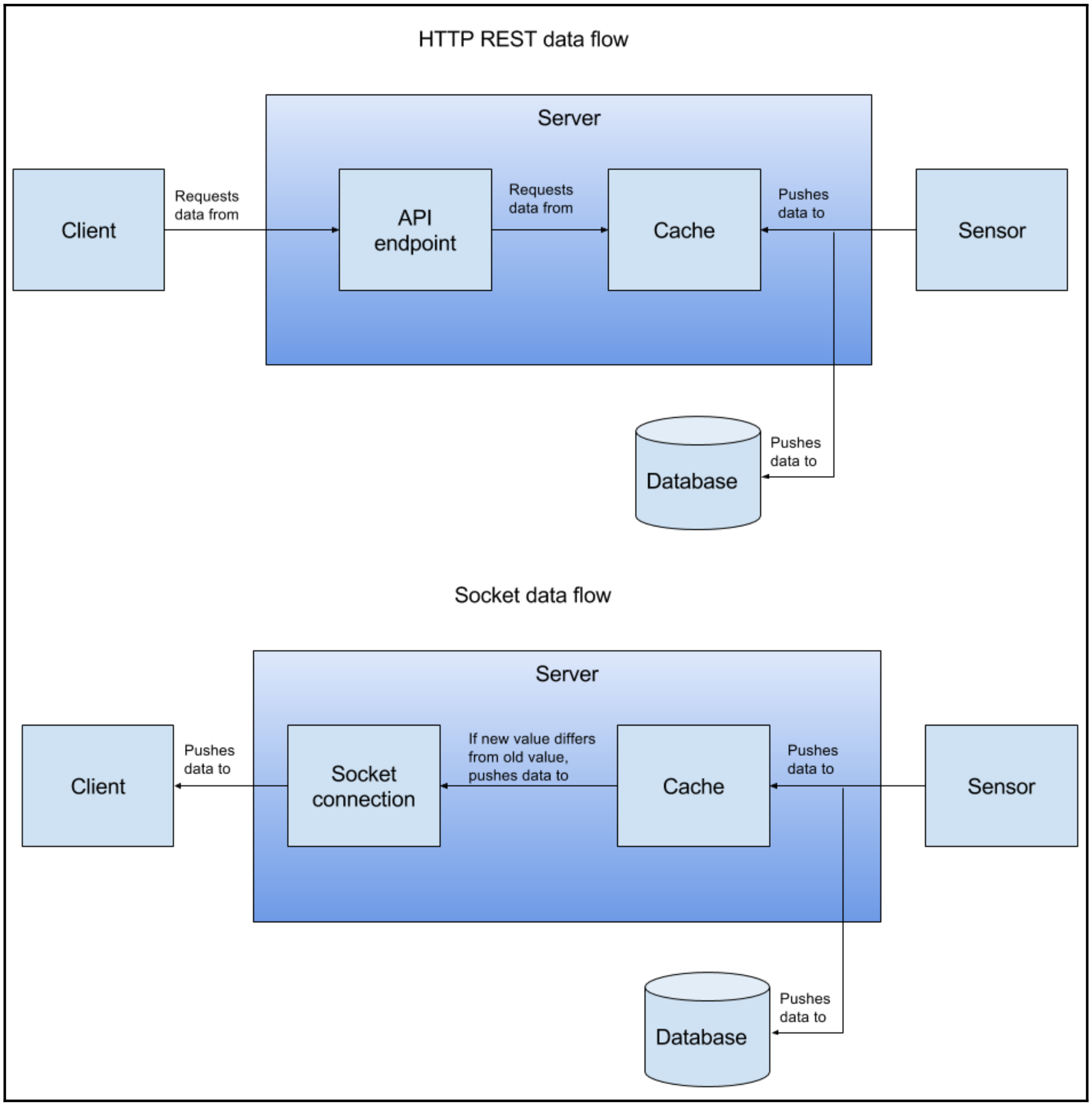


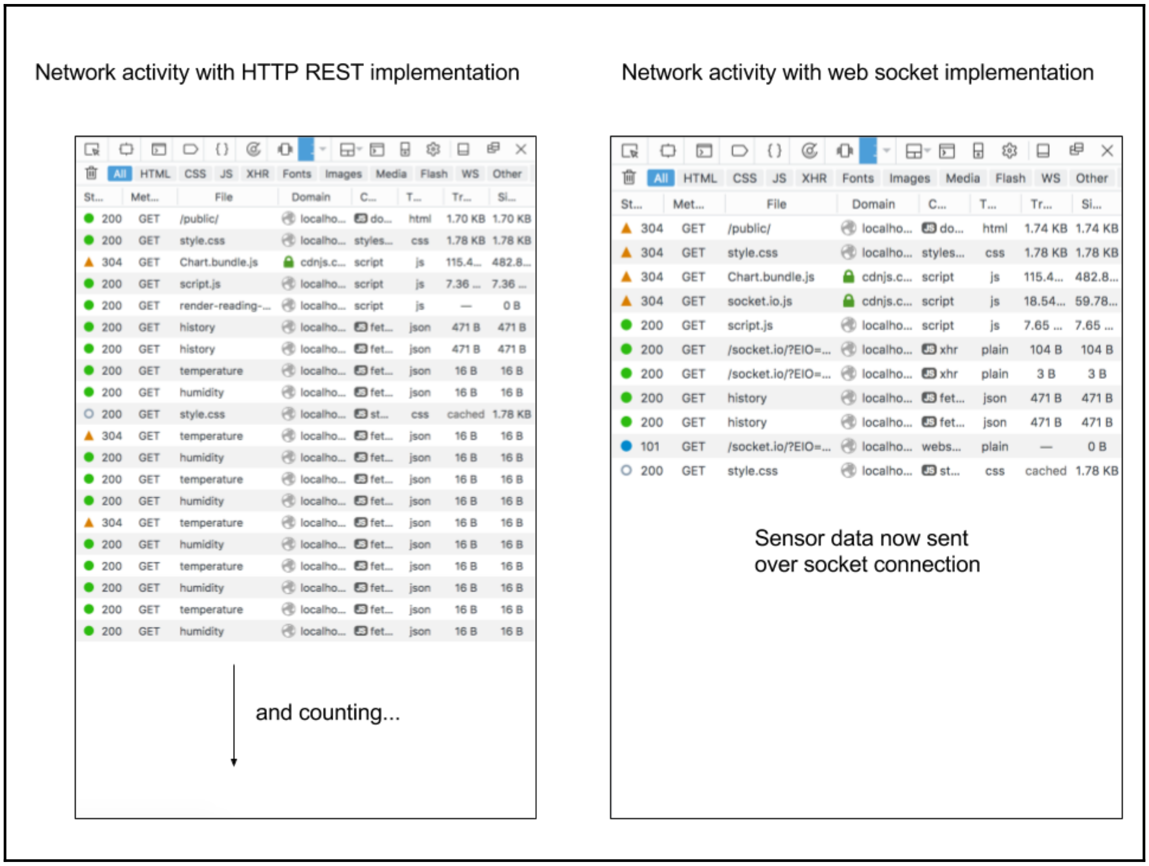

Client
Server
Application code
socket.io client side library
Web socket connection
socket.io server side library
Application code


Client
Server
Initialize socket connection
"connection" event handler
Handle event through handler
Emit event
Close socket connection
"disconnect" event handler



```
>> io
<- function r()
```


HTTP REST data flow
Server
Client
Requests data from
API endpoint
Requests data from
Cache
Pushes data to
Sensor
Database
Pushes data to
Socket data flow
Server
Client
Pushes data to
Socket connection
If new value differs from old value, pushes data to
Cache
Pushes data to
Sensor
Database
Pushes data to


Network activity with HTTP REST implementation
Network activity with web socket implementation
All HTML CSS JS XHR Fonts Images Media Flash WS Other
and counting...
Sensor data now sent over socket connection

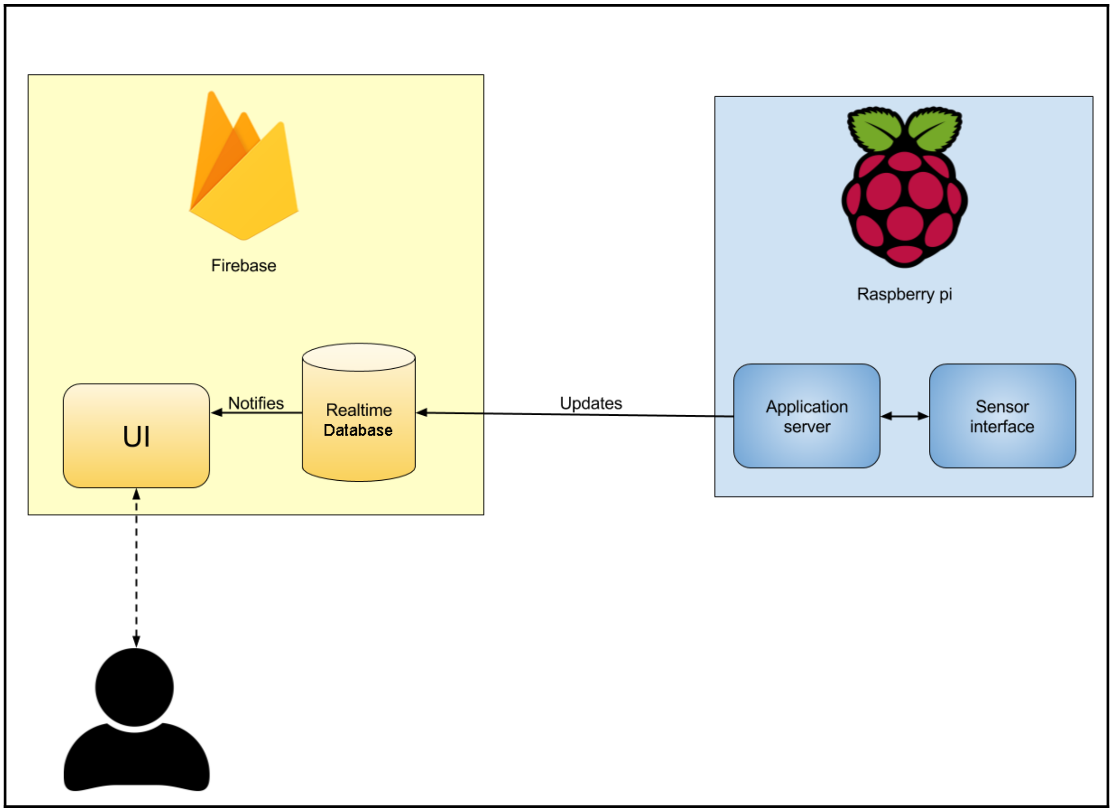
Network activity with HTTP REST implementation

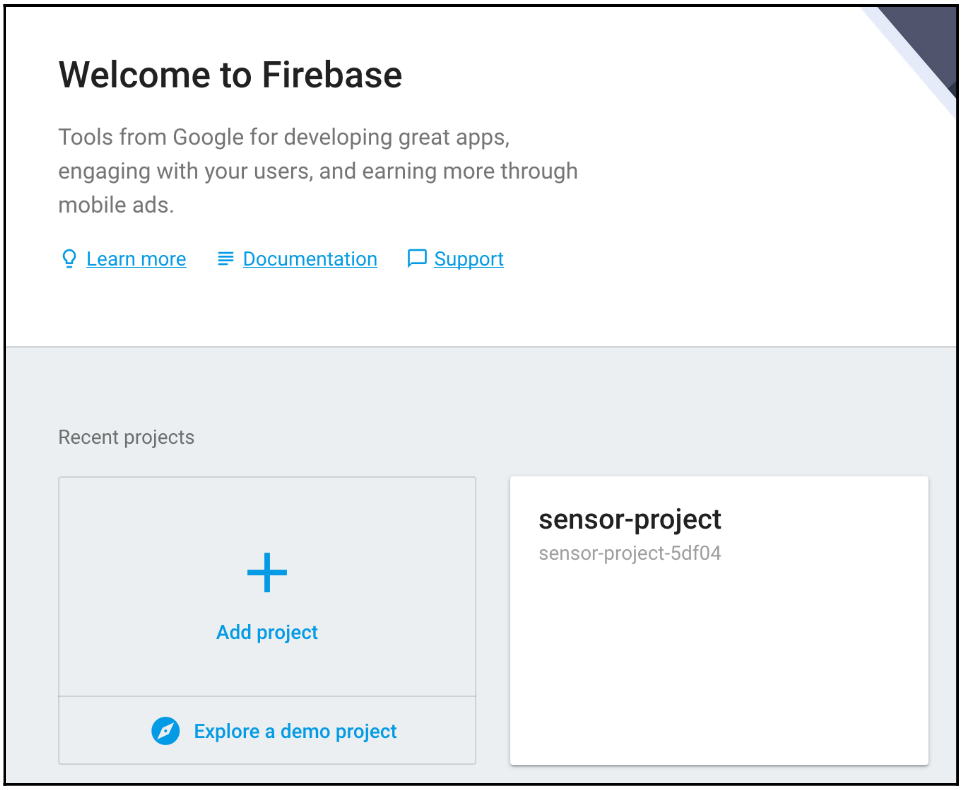
St...	Met...	File	Domain	C...	T...	Tr...	Si...
● 200	GET	/public/	localho...	do...	html	1.70 KB	1.70 KB
● 200	GET	style.css	localho...	styles...	css	1.78 KB	1.78 KB
▲ 304	GET	Chart.bundle.js	cdnjs.c...	script	js	115.4...	482.8...
● 200	GET	script.js	localho...	script	js	7.36 ...	7.36 ...
● 200	GET	render-reading-...	localho...	script	js	—	0 B
● 200	GET	history	localho...	fet...	json	471 B	471 B
● 200	GET	history	localho...	fet...	json	471 B	471 B
● 200	GET	temperature	localho...	fet...	json	16 B	16 B
● 200	GET	humidity	localho...	fet...	json	16 B	16 B
○ 200	GET	style.css	localho...	st...	css	cached	1.78 KB
▲ 304	GET	temperature	localho...	fet...	json	16 B	16 B
● 200	GET	humidity	localho...	fet...	json	16 B	16 B
● 200	GET	temperature	localho...	fet...	json	16 B	16 B
● 200	GET	humidity	localho...	fet...	json	16 B	16 B
▲ 304	GET	temperature	localho...	fet...	json	16 B	16 B
● 200	GET	humidity	localho...	fet...	json	16 B	16 B
● 200	GET	temperature	localho...	fet...	json	16 B	16 B
● 200	GET	humidity	localho...	fet...	json	16 B	16 B
● 200	GET	temperature	localho...	fet...	json	16 B	16 B
● 200	GET	humidity	localho...	fet...	json	16 B	16 B

Network activity with web socket implementation

St...	Met...	File	Domain	C...	T...	Tr...	Si...
▲ 304	GET	/public/	localho...	do...	html	1.74 KB	1.74 KB
▲ 304	GET	style.css	localho...	styles...	css	1.78 KB	1.78 KB
▲ 304	GET	Chart.bundle.js	cdnjs.c...	script	js	115.4...	482.8...
▲ 304	GET	socket.io.js	cdnjs.c...	script	js	18.54...	59.78...
● 200	GET	script.js	localho...	script	js	7.65 ...	7.65 ...
● 200	GET	/socket.io/?EIO=...	localho...	xhr	plain	104 B	104 B
● 200	GET	/socket.io/?EIO=...	localho...	xhr	plain	3 B	3 B
● 200	GET	history	localho...	fet...	json	471 B	471 B
● 200	GET	history	localho...	fet...	json	471 B	471 B
● 101	GET	/socket.io/?EIO=...	localho...	webs...	plain	—	0 B
○ 200	GET	style.css	localho...	st...	css	cached	1.78 KB

Chapter 11: Deploying our application to Firebase

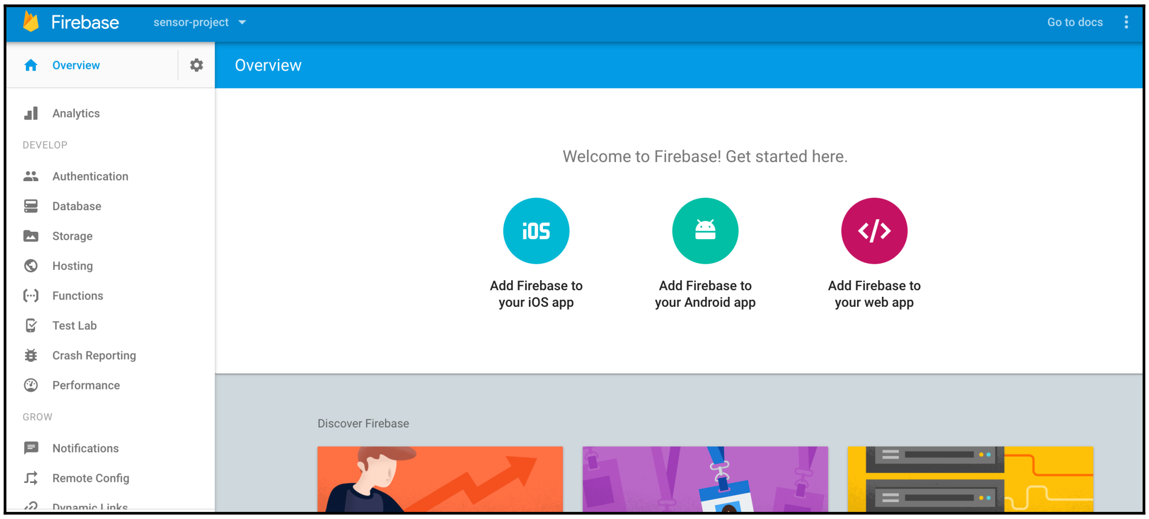



Welcome to Firebase

Tools from Google for developing great apps, engaging with your users, and earning more through mobile ads.

Recent projects

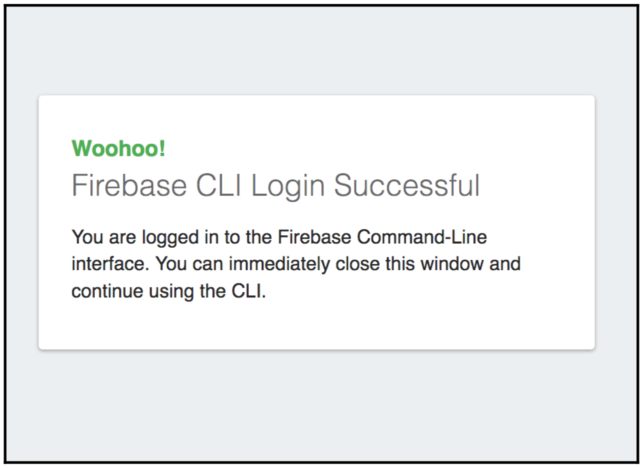
sensor-project
sensor-project-5df04



Overview

Welcome to Firebase! Get started here.

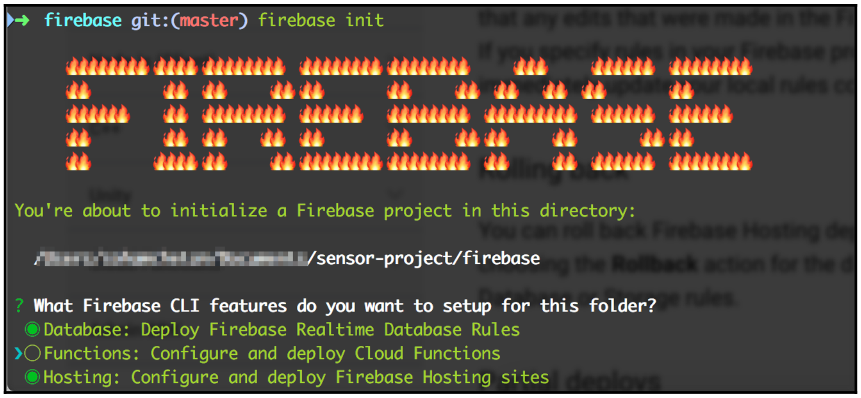
iOS
Add Firebase to your iOS app
Add Firebase to your Android app
Add Firebase to your web app
Discover Firebase



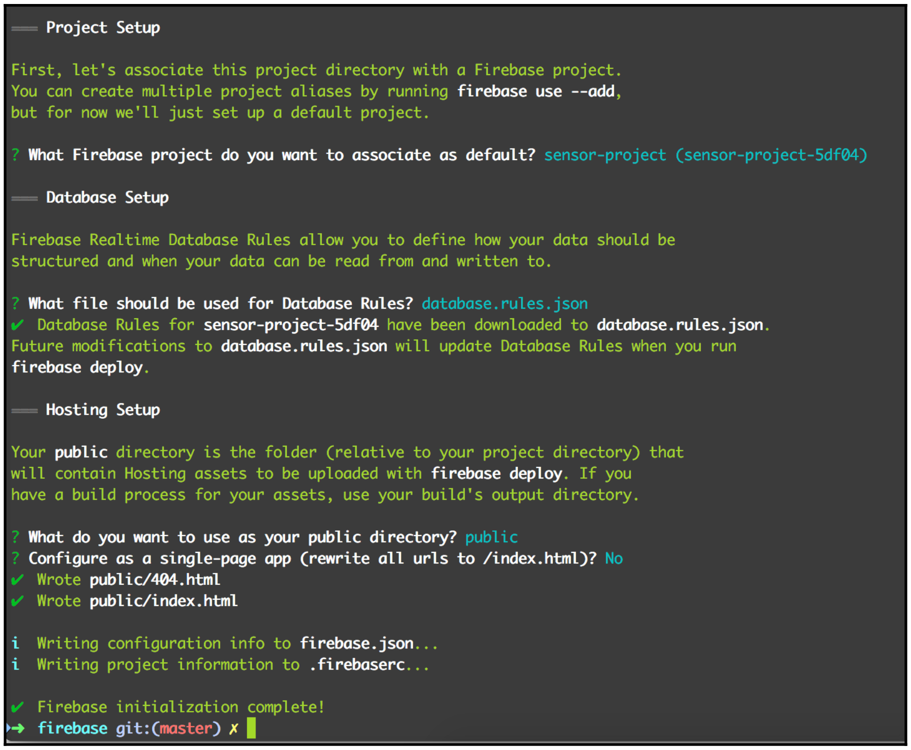
Woohoo!

Firebase CLI Login Successful

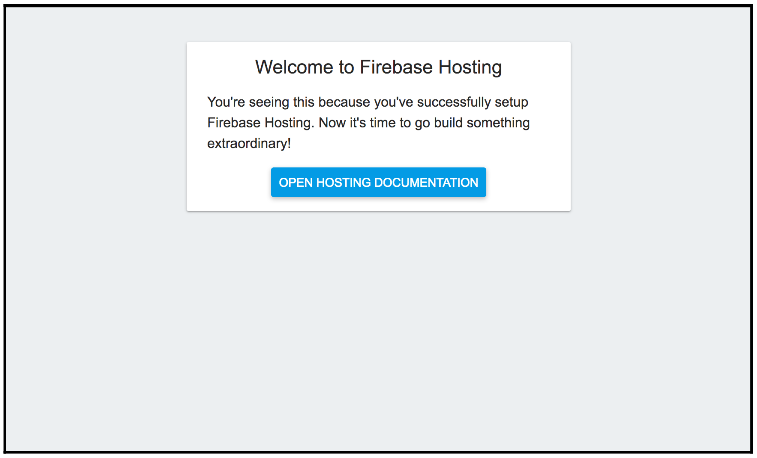
You are logged in to the Firebase Command-Line interface. You can immediately close this window and continue using the CLI.



```
➜ firebase git:(master) firebase init

You're about to initialize a Firebase project in this directory:

  [redacted]/sensor-project/firebase

? What Firebase CLI features do you want to setup for this folder?
 ◉Database: Deploy Firebase Realtime Database Rules
❯◯Functions: Configure and deploy Cloud Functions
 ◉Hosting: Configure and deploy Firebase Hosting sites
```

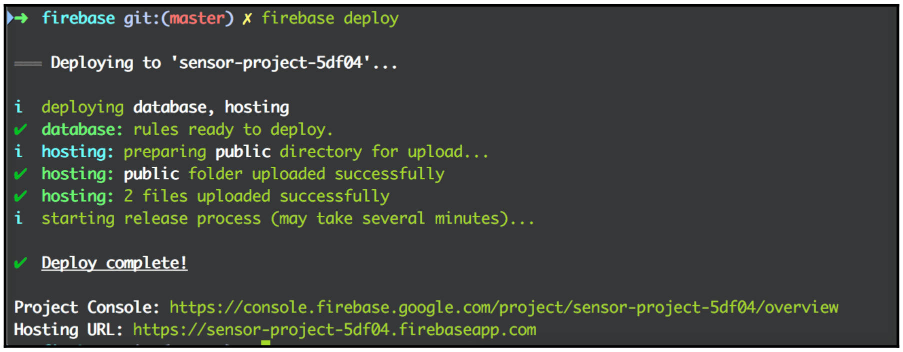


```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? What Firebase project do you want to associate as default? sensor-project (sensor-project-5df04)

=== Database Setup

Firebase Realtime Database Rules allow you to define how your data should be
structured and when your data can be read from and written to.

? What file should be used for Database Rules? database.rules.json
✔  Database Rules for sensor-project-5df04 have been downloaded to database.rules.json.
Future modifications to database.rules.json will update Database Rules when you run
firebase deploy.

=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? No
✔  Wrote public/404.html
✔  Wrote public/index.html

i  Writing configuration info to firebase.json...
i  Writing project information to .firebaserc...

✔  Firebase initialization complete!
➜  firebase git:(master) ✗
```

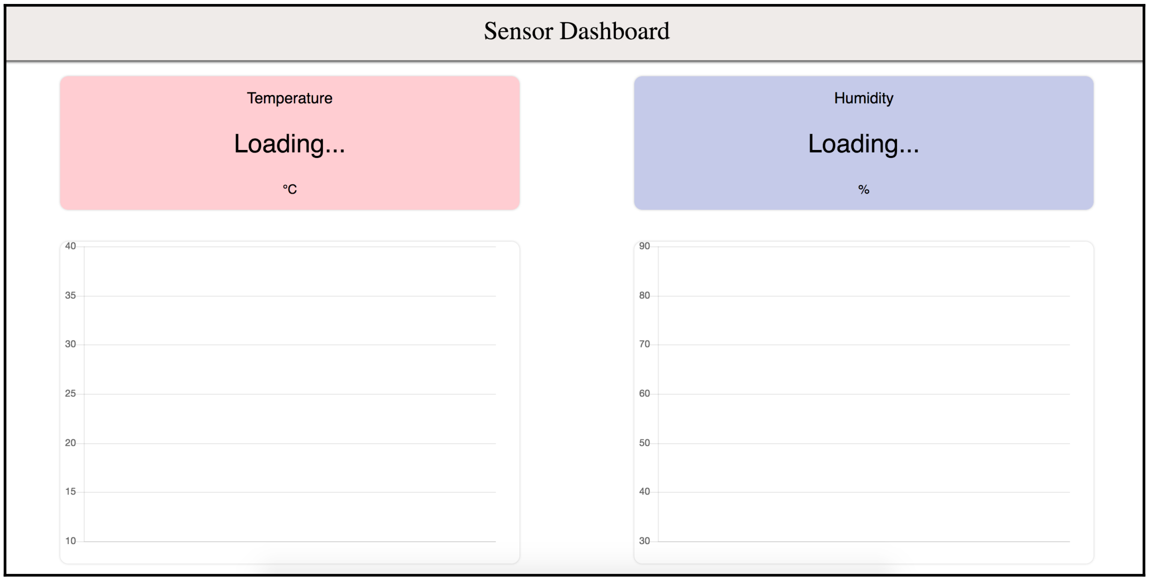


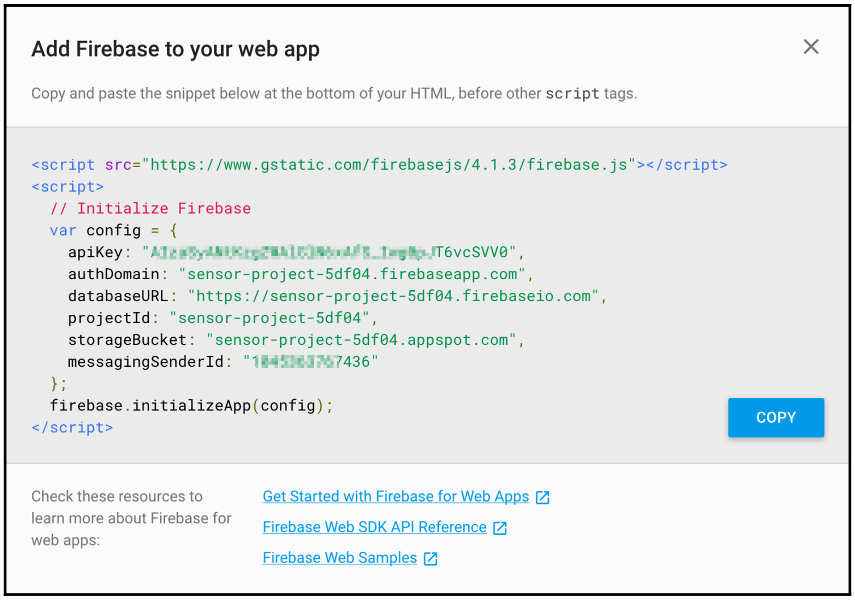
Welcome to Firebase Hosting

You're seeing this because you've successfully setup Firebase Hosting. Now it's time to go build something extraordinary!



```
➜ firebase git:(master) ✗ firebase deploy

=== Deploying to 'sensor-project-5df04'...

i  deploying database, hosting
✔  database: rules ready to deploy.
i  hosting: preparing public directory for upload...
✔  hosting: public folder uploaded successfully
✔  hosting: 2 files uploaded successfully
i  starting release process (may take several minutes)...

✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/sensor-project-5df04/overview
Hosting URL: https://sensor-project-5df04.firebaseapp.com
```




Add Firebase to your web app

Copy and paste the snippet below at the bottom of your HTML, before other `script` tags.

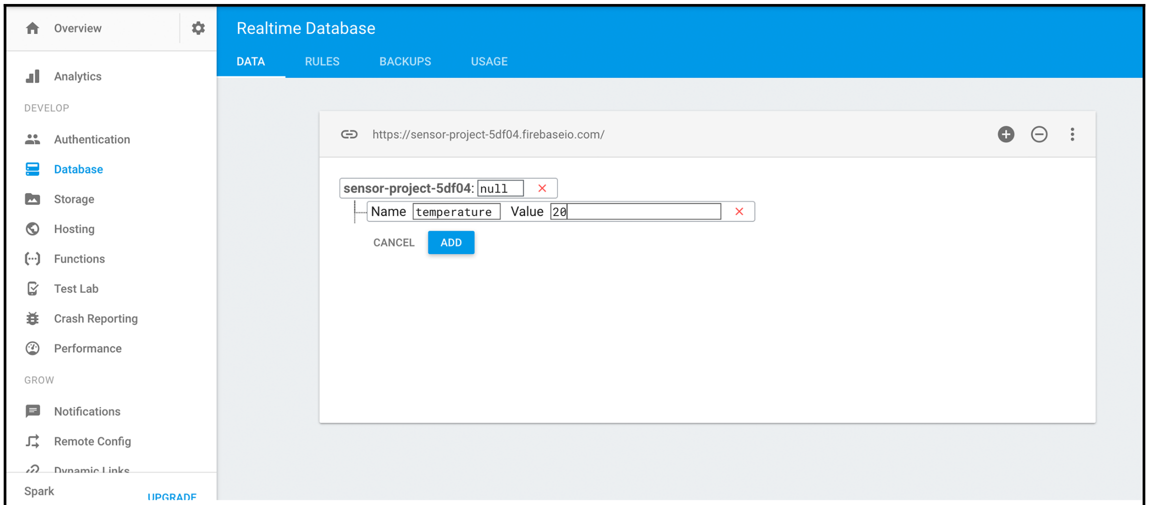
```
<script src="https://www.gstatic.com/firebasejs/4.1.3/firebase.js"></script>
<script>
  // Initialize Firebase
  var config = {
    apiKey: "[redacted]T6vcSVV0",
    authDomain: "sensor-project-5df04.firebaseapp.com",
    databaseURL: "https://sensor-project-5df04.firebaseio.com",
    projectId: "sensor-project-5df04",
    storageBucket: "sensor-project-5df04.appspot.com",
    messagingSenderId: "1[redacted]7436"
  };
  firebase.initializeApp(config);
</script>
```

Check these resources to learn more about Firebase for web apps:

Get Started with Firebase for Web Apps

Firebase Web SDK API Reference

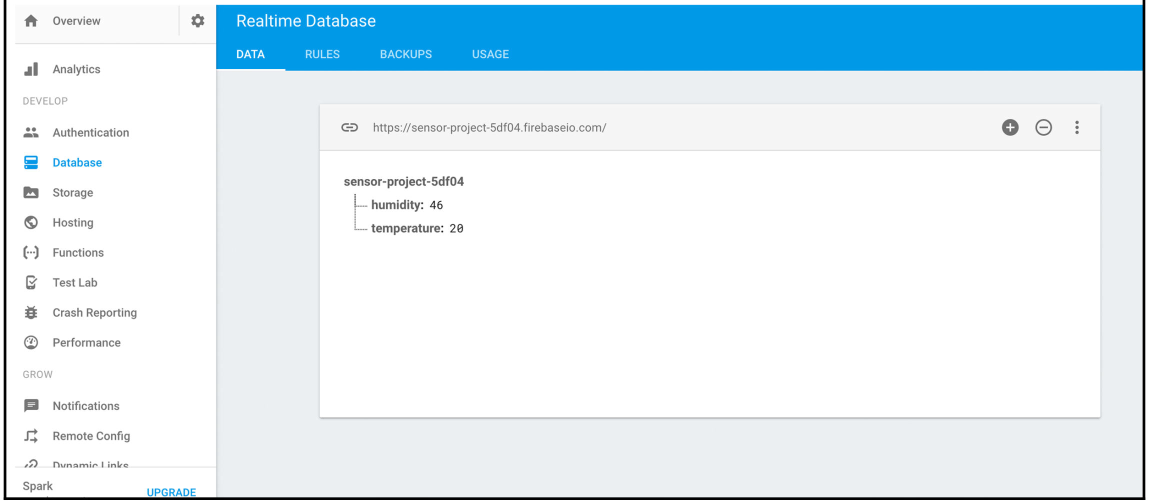
Firebase Web Samples



https://sensor-project-5df04.firebaseio.com/

sensor-project-5df04: `null`

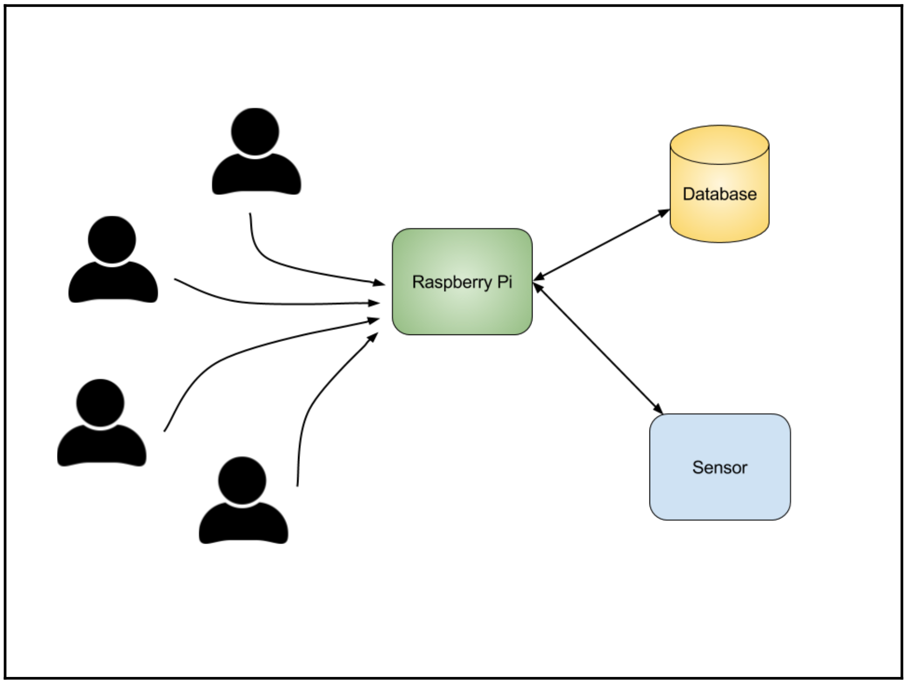
Name `temperature` Value `20`

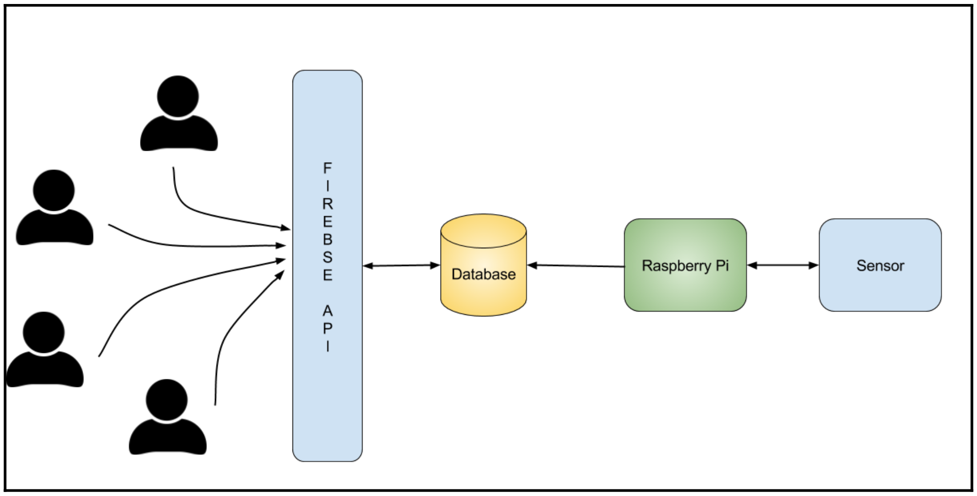


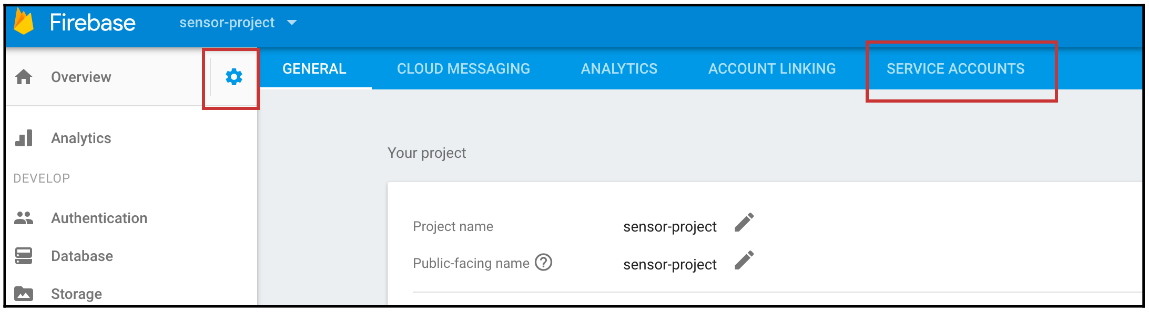
https://sensor-project-5df04.firebaseio.com/

```
sensor-project-5df04
  humidity: 46
  temperature: 20
```

Chapter 12: Using Firebase APIs to Update Our Application

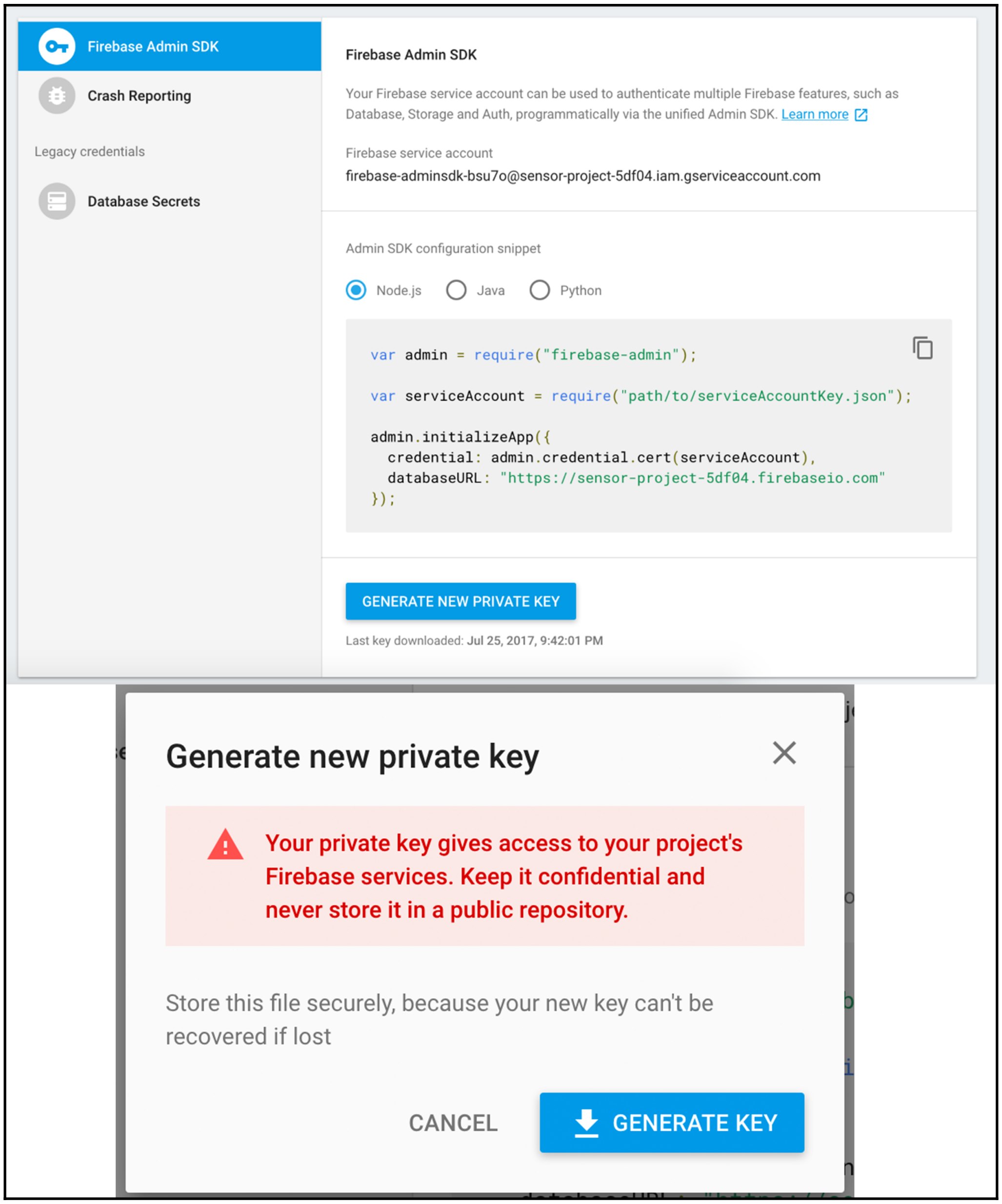






Project name sensor-project

Public-facing name sensor-project



Firebase Admin SDK

Your Firebase service account can be used to authenticate multiple Firebase features, such as Database, Storage and Auth, programmatically via the unified Admin SDK. Learn more

Firebase service account
firebase-adminsdk-bsu7o@sensor-project-5df04.iam.gserviceaccount.com

Admin SDK configuration snippet

[✓ Node.js] Java Python

```
var admin = require("firebase-admin");

var serviceAccount = require("path/to/serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
  databaseURL: "https://sensor-project-5df04.firebaseio.com"
});
```

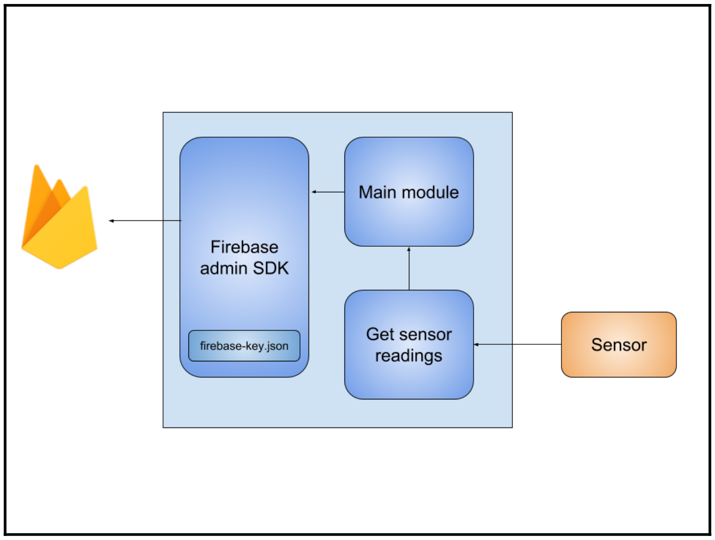
GENERATE NEW PRIVATE KEY

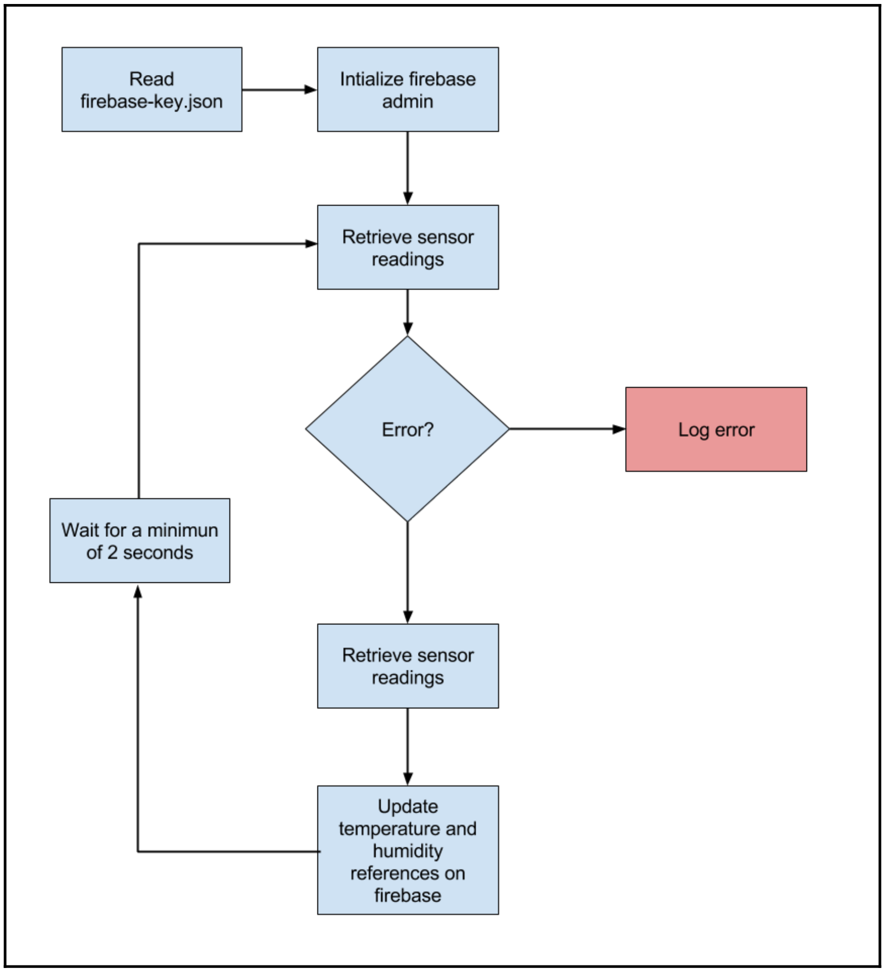
Last key downloaded: Jul 25, 2017, 9:42:01 PM

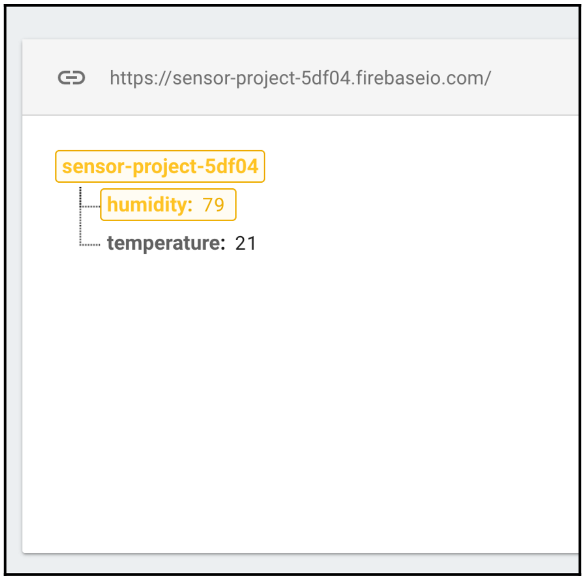
Generate new private key

Your private key gives access to your project's Firebase services. Keep it confidential and never store it in a public repository.

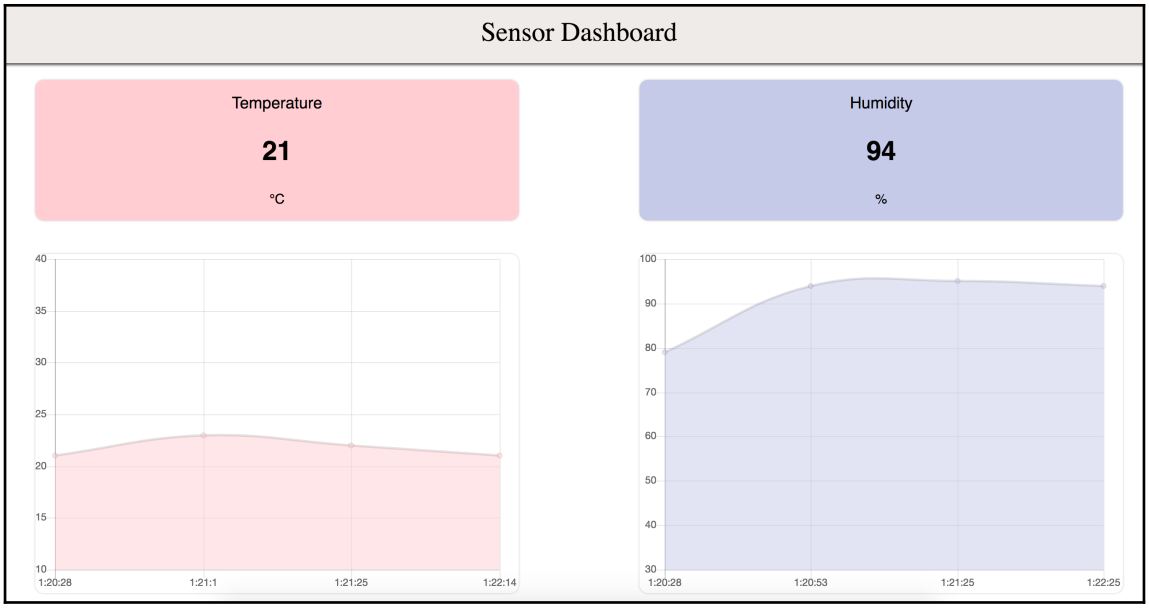
Store this file securely, because your new key can't be recovered if lost


Firebase admin SDK
firebase-key.json
Main module
Get sensor readings
Sensor


Read firebase-key.json
Intialize firebase admin
Retrieve sensor readings
Error?
Log error
Wait for a minimun of 2 seconds
Retrieve sensor readings
Update temperature and humidity references on firebase


https://sensor-project-5df04.firebaseio.com/

```
sensor-project-5df04
├ humidity: 79
└ temperature: 21
```


Sensor Dashboard

Temperature

21

°C

Humidity

94

%