

contents

<i>preface</i>	xiii
<i>acknowledgments</i>	xv
<i>about this book</i>	xviii

PART 1 BASICS OF THE IoT AND THE WoT 1

1 *From the Internet of Things to the Web of Things* 3

1.1 Defining the Internet of Things 4

1.2 Enter the Web of Things 6

Web of Things scenario: connected hotel 6

Comparing IoT and WoT 8 • *The Internet of Things—a brief history* 12

1.3 Use cases—why connected objects? 14

Wireless sensor networks and distributed

sensing 14 • *Wearables and quantified self* 16 • *Smart homes and buildings* 17 • *Smart cities and energy grids* 18 • *Smart manufacturing and Industry 4.0* 19 • *Smart logistics and supply chains* 19 • *Marketing 2.0* 21

1.4	The Web of Things—a supercharged Internet of Things	22
	<i>Easier to program</i>	23
	<i>Open and extensible standards</i>	24
	<i>Fast and easy to deploy, maintain, and integrate</i>	24
	<i>Loose coupling between elements</i>	25
	<i>Widely used security and privacy mechanisms</i>	25
	<i>WoT—the shortcomings</i>	26
1.5	Summary	27

2 *Hello, World Wide Web of Things* 29

2.1	Meet a Web of Things device	30
	<i>The suspect: Raspberry Pi</i>	31
2.2	Exercise 1—Browse a device on the Web of Things	32
	<i>Part 1—The web as user interface</i>	32
	<i>Part 2—The web as an API</i>	36
	<i>So what?</i>	41
2.3	Exercise 2—Polling data from a WoT sensor	42
	<i>Part 1—Polling the current sensor value</i>	42
	<i>Part 2—Polling and graphing sensor values</i>	43
	<i>Part 3—Real-time data updates</i>	44
	<i>So what?</i>	45
2.4	Exercise 3—Act on the real world	46
	<i>Part 1—Use a form to update text to display</i>	46
	<i>Create your own form to control devices</i>	48
	<i>So what?</i>	50
2.5	Exercise 4—Tell the world about your device	50
	<i>So what?</i>	54
2.6	Exercise 5—Create your first physical mashup	54
	<i>So what?</i>	57
2.7	Summary	57

3 *Node.js for the Web of Things* 59

3.1	The rise of JavaScript: from clients to servers to things!	60
	<i>Pushing JavaScript to things</i>	61
3.2	Introduction to Node.js	62
	<i>Installing Node.js on your machine</i>	63
	<i>Your first web server in Node.js</i>	63
	<i>Returning sensor data as JSON</i>	65
3.3	Modularity in Node.js	66
	<i>npm—the Node package manager</i>	67
	<i>Clean dependencies with package.json and npm</i>	68
	<i>Your first Node module</i>	69

3.4	Understanding the Node.js event loop	71
	<i>Multithreaded web servers</i>	71
	<i>Single-threaded, non-blocking web servers</i>	71
3.5	Getting started with asynchronous programming	73
	<i>Anonymous callbacks</i>	74
	<i>Named callbacks</i>	77
	<i>Control flow libraries</i>	78
3.6	Summary and beyond the book	81

4 Getting started with embedded systems 83

4.1	The world of embedded devices	84
	<i>Devices for hobbyists vs. industrial devices</i>	84
	<i>Real-time operating systems vs. Linux</i>	85
	<i>Summary and beyond the Pi</i>	87
4.2	Set up your first WoT device—Raspberry Pi	88
	<i>Meet the Raspberry Pi</i>	88
	<i>Choosing your Pi</i>	89
	<i>Shopping list</i>	90
	<i>Setting up your Raspberry Pi</i>	91
	<i>Connecting to your device</i>	96
4.3	Installing Node.js on the Raspberry Pi	97
	<i>Using Git and GitHub on the Pi</i>	99
	<i>So what?</i>	99
4.4	Connecting sensors and actuators to your Pi	99
	<i>Understanding GPIO ports</i>	100
	<i>Working with breadboards and electronic components</i>	100
	<i>Accessing GPIOs from Node.js</i>	102
	<i>Beyond the book</i>	107
4.5	Summary	108

5 Building networks of Things 109

5.1	Connecting Things	111
	<i>Network topologies</i>	111
	<i>Network classification models</i>	113
5.2	Networking protocols for Things	115
	<i>Spatial considerations</i>	115
	<i>Internet protocols and the IoT</i>	116
	<i>IoT personal area networks</i>	120
	<i>IoT wide area networks</i>	124
	<i>So, which one should I choose?</i>	127
5.3	Application protocols for Things	130
	<i>ZigBee and Bluetooth application stacks</i>	130
	<i>Apple HomeKit and Google Weave</i>	132
	<i>Message Queuing Telemetry Transport</i>	132
	<i>Constrained Application Protocol</i>	135
	<i>So, which one should I use?</i>	135

5.4	The Web of Things architecture	136
	<i>Layer 1: Access</i>	136
	<i>Layer 2: Find</i>	137
	<i>Layer 3: Share</i>	138
	<i>Layer 4: Compose</i>	138
	<i>Why does the WoT matter?</i>	138
	<i>Beyond the book</i>	139
5.5	Summary	140

PART 2 BUILDING THE WOT..... 141

6 Access: *Web APIs for Things* 143

6.1	Devices, resources, and web Things	144
	<i>Representational State Transfer</i>	144
	<i>Why do we need a uniform interface?</i>	146
	<i>Principle 1: addressable resources</i>	147
	<i>Principle 2: manipulation of resources through representations</i>	151
	<i>Principle 3: self-descriptive messages</i>	154
	<i>Principle 4: Hypermedia as the Engine of Application State</i>	160
	<i>Summary—web Things design process</i>	163
6.2	Beyond REST: the real-time Web of Things	163
	<i>The WoT needs events!</i>	164
	<i>Publish/subscribe</i>	165
	<i>Webhooks—HTTP callbacks</i>	166
	<i>Comet—hacking HTTP for a real-time web</i>	167
	<i>WebSockets</i>	168
	<i>The future: from HTTP/1.1 to HTTP/2</i>	172
6.3	Summary	173

7 Implementing web Things 175

7.1	Connecting devices to the web	176
7.2	Direct integration pattern—REST on devices	177
	<i>Creating a WoT server</i>	178
	<i>Resource design</i>	180
	<i>Representation design</i>	186
	<i>Interface design</i>	189
	<i>Pub/sub interface via WebSockets</i>	191
	<i>Summary—direct integration pattern</i>	194
7.3	Gateway integration pattern—CoAP example	194
	<i>Running a CoAP server</i>	195
	<i>Proxying CoAP via a gateway</i>	196
	<i>Summary—gateway integration pattern</i>	198
7.4	Cloud integration pattern—MQTT over EVRYTHNG	199
	<i>Set up your EVRYTHNG account</i>	201
	<i>Create your MQTT client application</i>	204
	<i>Use actions to control the power</i>	

7.5	Summary	212
8	Find: Describe and discover web Things	214
8.1	The findability problem	215
8.2	Discovering Things	217
<i>Network discovery</i>	217	▪ <i>Resource discovery on the web</i> 220
8.3	Describing web Things	223
<i>Introducing the Web Thing Model</i>	225	▪ <i>Metadata</i> 227
<i>Properties</i>	227	▪ <i>Actions</i> 229
<i>Things</i>	231	
<i>Implementing the Web Thing Model on the Pi</i>	232	
<i>Summary—the Web Thing Model</i>	238	
8.4	The Semantic Web of Things	239
<i>Linked data and RDFa</i>	239	▪ <i>Agreed-upon semantics: Schema.org</i> 243
<i>JSON-LD</i>	244	▪ <i>Beyond the book</i> 246
8.5	Summary	247
9	Share: Securing and sharing web Things	248
9.1	Securing Things	250
<i>Encryption 101</i>	252	▪ <i>Web security with TLS: the S of HTTPS!</i> 253
<i>Enabling HTTPS and WSS with TLS on your Pi</i>	255	
9.2	Authentication and access control	260
<i>Access control with REST and API tokens</i>	260	▪ <i>OAuth: a web authorization framework</i> 263
9.3	The Social Web of Things	265
<i>A Social Web of Things authentication proxy</i>	266	
<i>Implementing a Social WoT authentication proxy</i>	269	
9.4	Beyond the book	276
9.5	Summary	278
10	Compose: Physical mashups	279
10.1	Building a simple app—automated UI generation	280
<i>A universal user interface for web Things</i>	281	
10.2	Physical mashups	288
<i>Boxes and wires mashups for the Physical Web: Node-RED</i>	289	

10.3	Using wizards for physical mashups: IFTTT	295
	<i>Pushing intruder alert tweets to a Google spreadsheet</i>	296
	<i>Sending requests to a Thing with the Maker Channel</i>	298
	<i>Pushing intruder alert tweets to a Google spreadsheet</i>	299
10.4	Beyond the book	300
	<i>From simple mashups to big data mashups</i>	300
	<i>A better user experience</i>	301
10.5	Summary	302
appendix	<i>Arduino, BeagleBone, Intel Edison, and the WoT</i>	303

index 309