

---

# Contents

Acknowledgements .....	xiv
1. Introduction .....	1
1.1. Why test? .....	1
1.2. Why should I change my code? .....	2
1.3. What is this book not about? .....	3
1.4. How the book is structured .....	4
1.5. Coding conventions .....	4
1.6. Example code .....	6
2. Building blocks .....	8
2.1. What is an automated test suite? .....	8
2.2. Testing frameworks .....	9
2.3. Hello, world! .....	9
2.4. Assertions .....	11
2.4.1. The Node assert module .....	12
2.4.2. Jasmine matchers .....	13
2.4.3. Chai assertion chains .....	13
2.4.4. QUnit assertions .....	14
2.4.5. jstest assertions and matchers .....	15
2.4.6. Other types of assertions .....	16
2.5. Organising tests .....	17
2.5.1. Setting up with before() blocks .....	18
2.5.2. Wrapping up with after() blocks .....	19
2.5.3. Contexts with describe() blocks .....	20
2.6. Stubs, mocks and spies .....	22
2.6.1. Enter stubs .....	24
2.6.2. Pattern-matching arguments .....	25
2.6.3. Testing interactions with mocks .....	27
2.6.4. Checking your wiring with spies .....	28
2.7. Asynchronous tests .....	28
2.7.1. The error-first callback convention .....	30
2.7.2. Multi-step async tests .....	31
2.7.3. Sequencing functions with Async .....	33
2.7.4. Generating steps with continuables .....	34
2.7.5. Async stories with jstest .....	36
3. Events and streams .....	37
3.1. Event emitters .....	37
3.1.1. Black-box testing .....	38
3.1.2. Mock-based testing .....	40
3.1.3. Cutting out the listener .....	41
3.1.4. Spy-based testing .....	44
3.2. Event listeners .....	44
3.2.1. Decoupling listeners from actions .....	45
3.2.2. Decoupling and framework patterns .....	47
3.3. Pitfalls of event testing .....	48
3.3.1. Events as a public interface .....	50
3.3.2. Selective integration testing .....	50
3.3.3. Changing your API design .....	51
3.4. Streams .....	54
3.4.1. Building a data source .....	55
3.4.2. Checking a readable stream's output .....	56

3.4.3. Testing a transform stream .....	57
3.4.4. Making your own transforms .....	58
3.4.5. Decoupled streams .....	60
4. Browser applications .....	64
4.1. Form validation .....	64
4.1.1. Extracting user interface logic .....	66
4.1.2. Extracting markup for tests .....	66
4.1.3. Separating business logic from the user interface .....	69
4.1.4. Testing user interface logic .....	71
4.1.5. Simulating user input .....	74
4.2. Modular interfaces .....	75
4.2.1. Directly communicating components .....	77
4.2.2. Dependency injection .....	80
4.2.3. Introducing a model .....	81
4.2.4. Building and testing the model .....	87
4.2.5. Using frameworks .....	90
4.3. Talking to the server .....	90
4.3.1. jQuery.ajax() with callbacks .....	92
4.3.2. jQuery.ajax() with promises .....	96
4.3.3. jQuery.post() shorthand .....	99
4.3.4. Using a fake server .....	99
4.3.5. WebSocket and other protocols .....	102
4.4. Storing data .....	103
4.4.1. Black-box testing .....	104
4.4.2. Testing persistence contracts .....	106
4.4.3. Mocking database interactions .....	107
4.5. location, pushState and routers .....	109
4.5.1. Testing location changes .....	110
4.5.2. Testing pushState and routers .....	112
4.6. Using the timer functions .....	114
4.7. Working with build tools .....	117
4.7.1. Precompiling templates .....	117
4.7.2. CommonJS modules and Browserify .....	118
4.7.3. Minification and other build tasks .....	121
4.8. Running browser tests .....	122
4.8.1. Running tests on PhantomJS .....	122
4.8.2. TestSwarm: cross-browser CI .....	124
4.8.3. Local cross-browser testing: Buster, Karma, Testem .....	125
4.9. Limitations of browser testing .....	128
5. Server-side applications .....	129
5.1. A basic Node server .....	129
5.1.1. Black-box server testing .....	130
5.1.2. Mock-based server testing .....	133
5.1.3. Mocking and pattern-matching .....	134
5.1.4. Limitations of mock-based server testing .....	137
5.2. Dealing with databases .....	140
5.2.1. Designing an HTTP API .....	140
5.2.2. Creating a database schema .....	142
5.2.3. Building the application .....	143
5.2.4. Monolithic request handlers .....	146
5.2.5. Working with time .....	150
5.3. Extracting services .....	156

5.3.1. Encapsulating the database .....	159
5.3.2. Dependency injection .....	162
5.3.3. Wiring everything together .....	167
5.4. Authentication .....	172
5.4.1. Passwords .....	173
5.4.2. Validation .....	179
5.4.3. Persistence .....	180
5.4.4. Signing up and logging in .....	184
5.4.5. Testing via the browser .....	189
5.4.6. Simulating a browser .....	193
5.5. Client-side JavaScript .....	198
5.6. Real-time updates .....	203
5.6.1. Testing with real connections .....	207
5.6.2. Unit testing with fake sockets .....	209
5.6.3. Full-stack testing .....	210
5.6.4. Simulating the client .....	212
6. Command-line interfaces .....	215
6.1. A black-box program .....	215
6.1.1. Command-line arguments .....	217
6.1.2. Environment variables .....	219
6.1.3. Configuration files .....	221
6.1.4. Standard input .....	222
6.2. Breaking open the black box .....	226
6.2.1. Talking to the internet .....	226
6.2.2. Wrapping external dependencies .....	227
6.2.3. Presenting information .....	230
6.2.4. Modelling the CLI .....	231
6.2.5. Command-line arguments .....	234
6.2.6. Standard input .....	236
A. JavaScript functions .....	239
A.1. Defining a function .....	239
A.2. Function calls .....	240
A.2.1. As a method .....	240
A.2.2. As a function .....	240
A.2.3. As an application .....	242
A.2.4. As a constructor .....	242
A.2.5. Getters and setters .....	243
A.3. Functions and mocking .....	245
A.3.1. Mocking and bound methods .....	246
B. The ‘with’ statement .....	248
B.1. The global variable problem .....	248
B.2. What does with do? .....	249
B.3. Caveats .....	249
C. Currying .....	251
C.1. Continuals .....	251
C.2. Converting functions into continuals .....	252