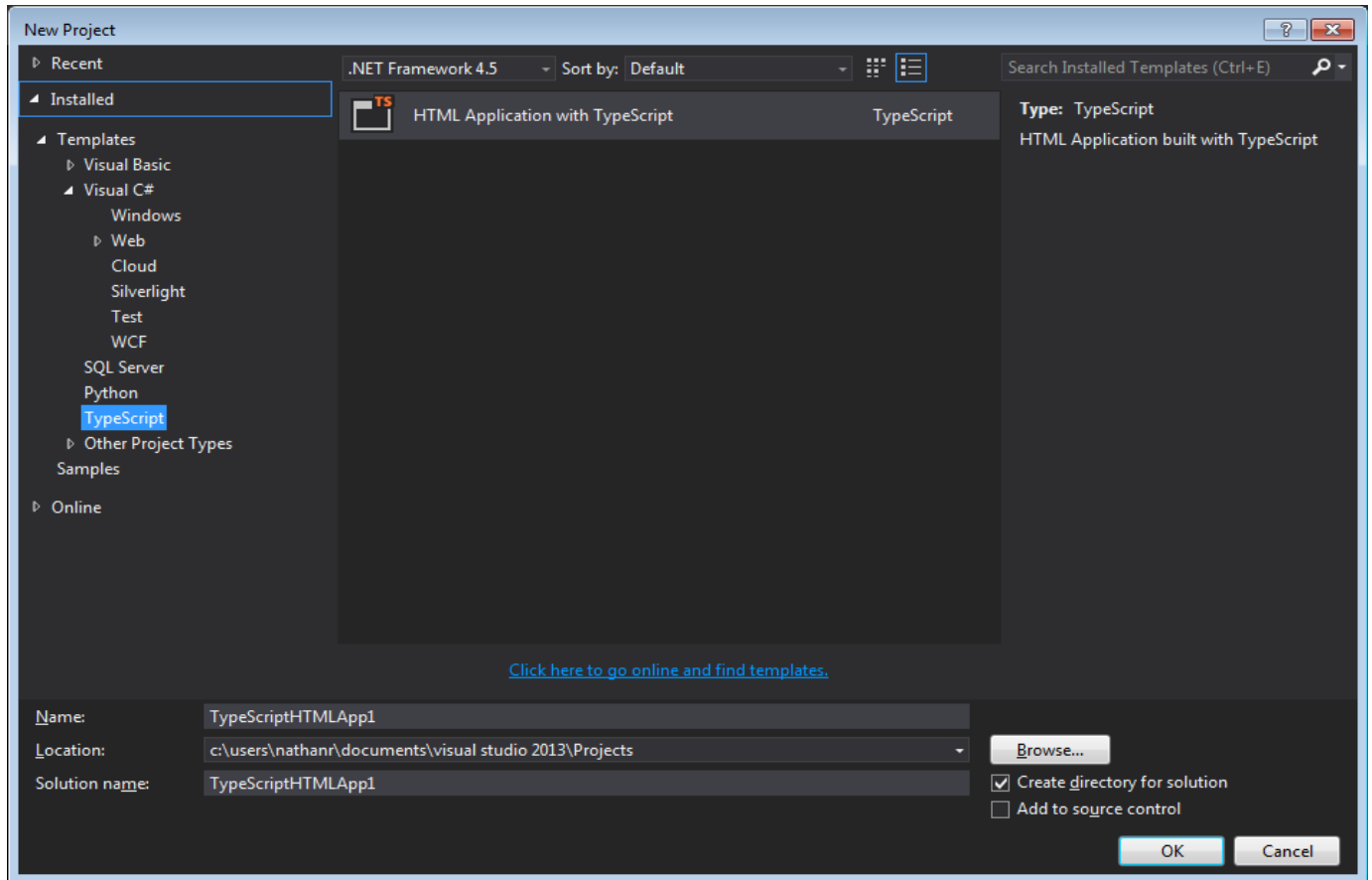
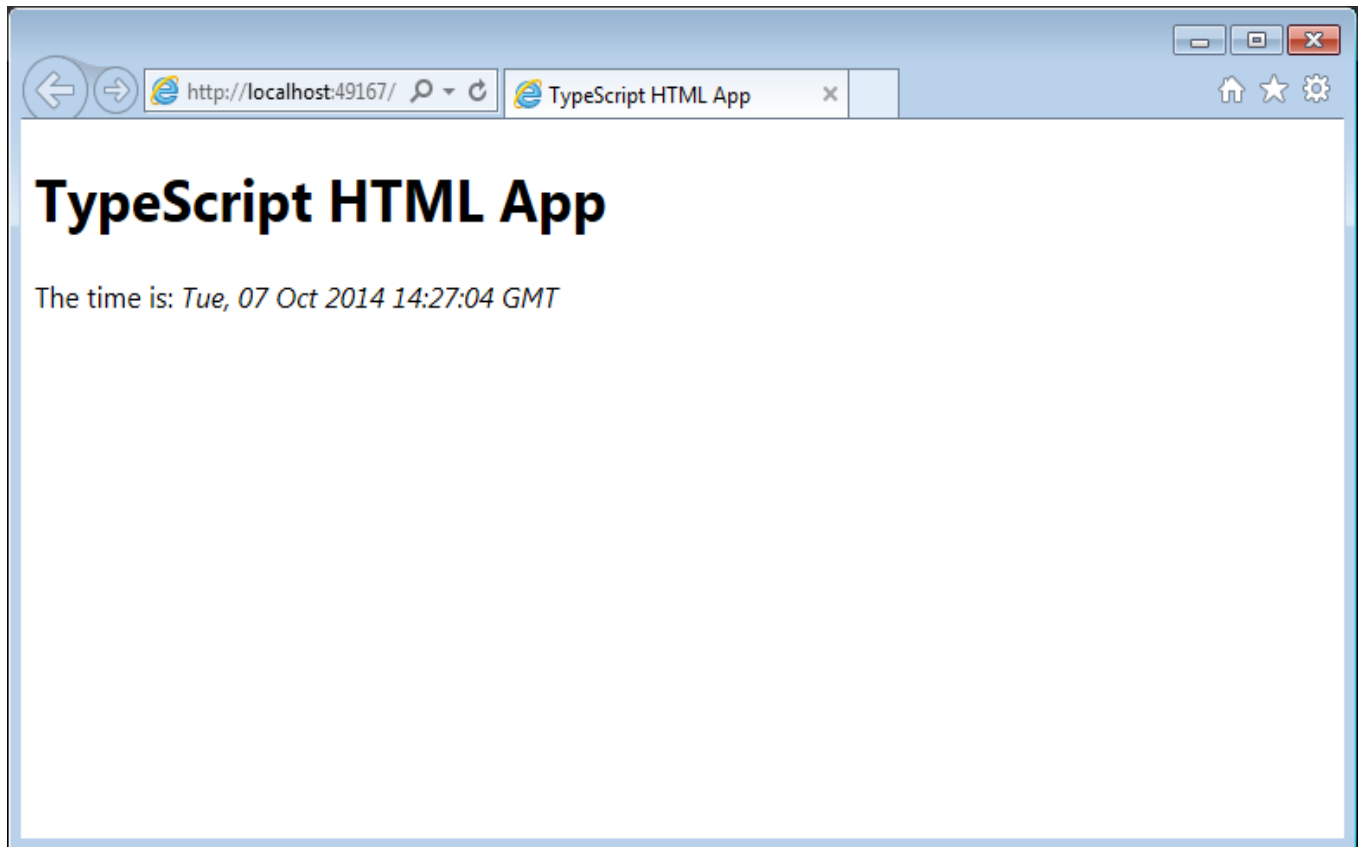


Chapter 1: TypeScript – Tools and Framework Options

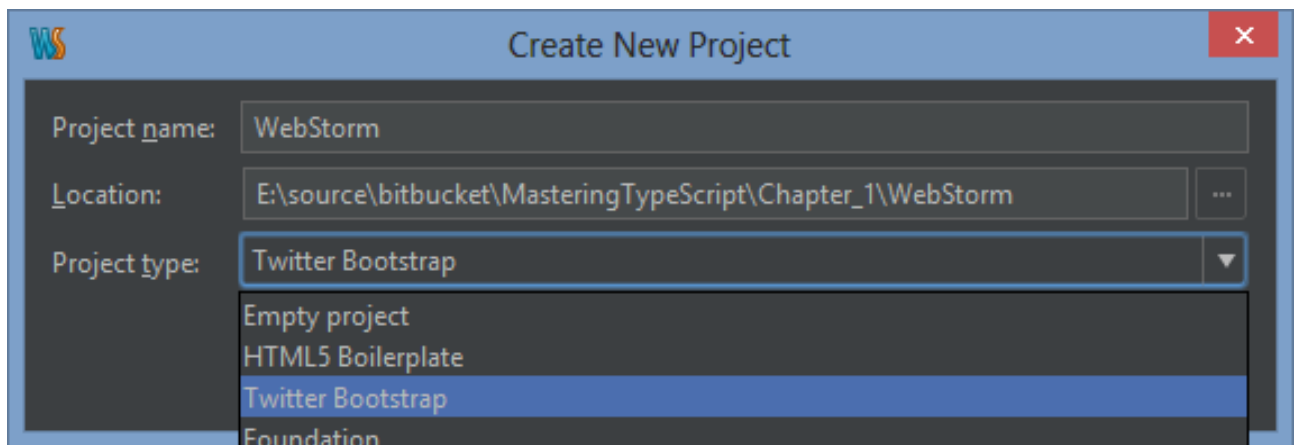
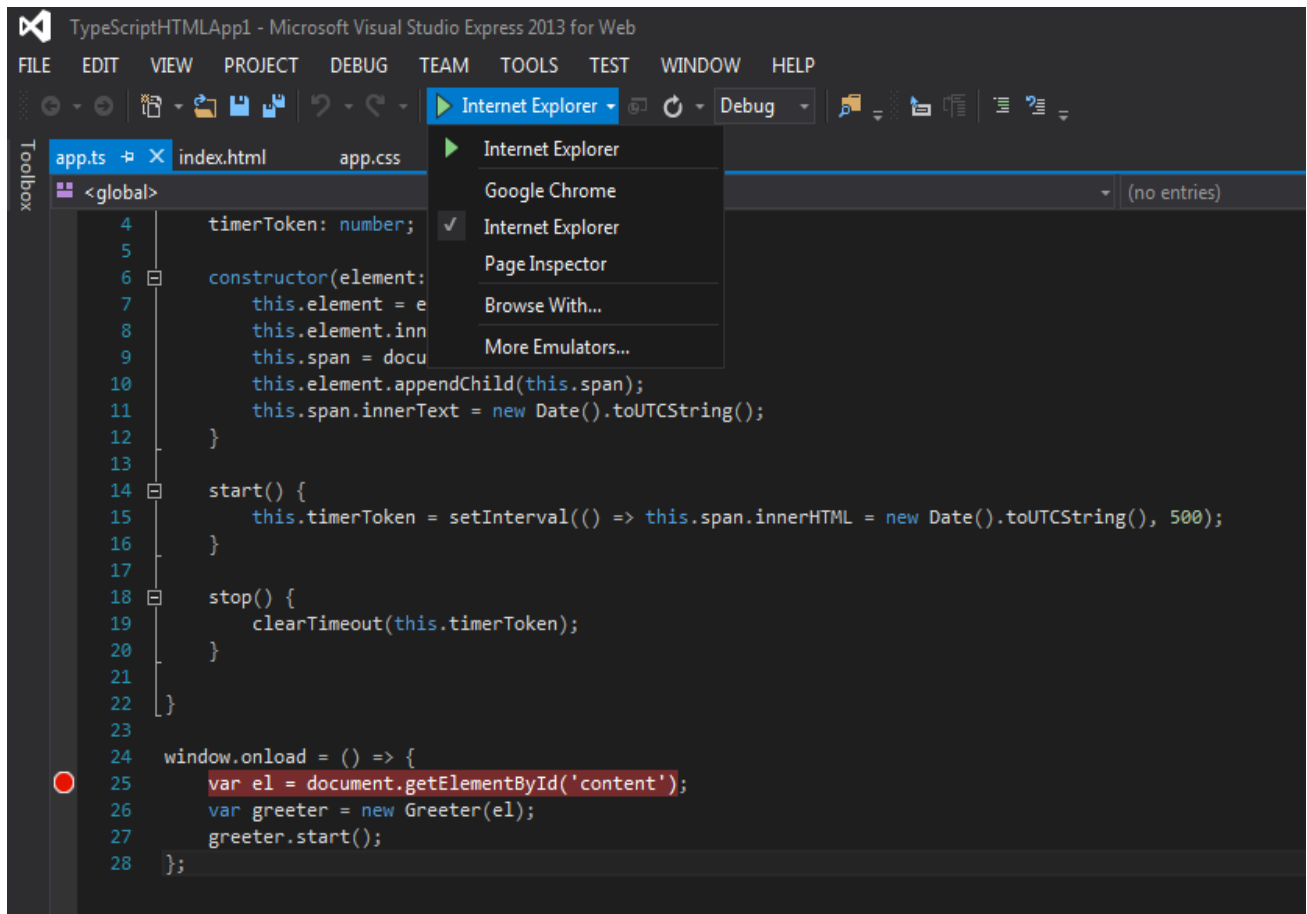


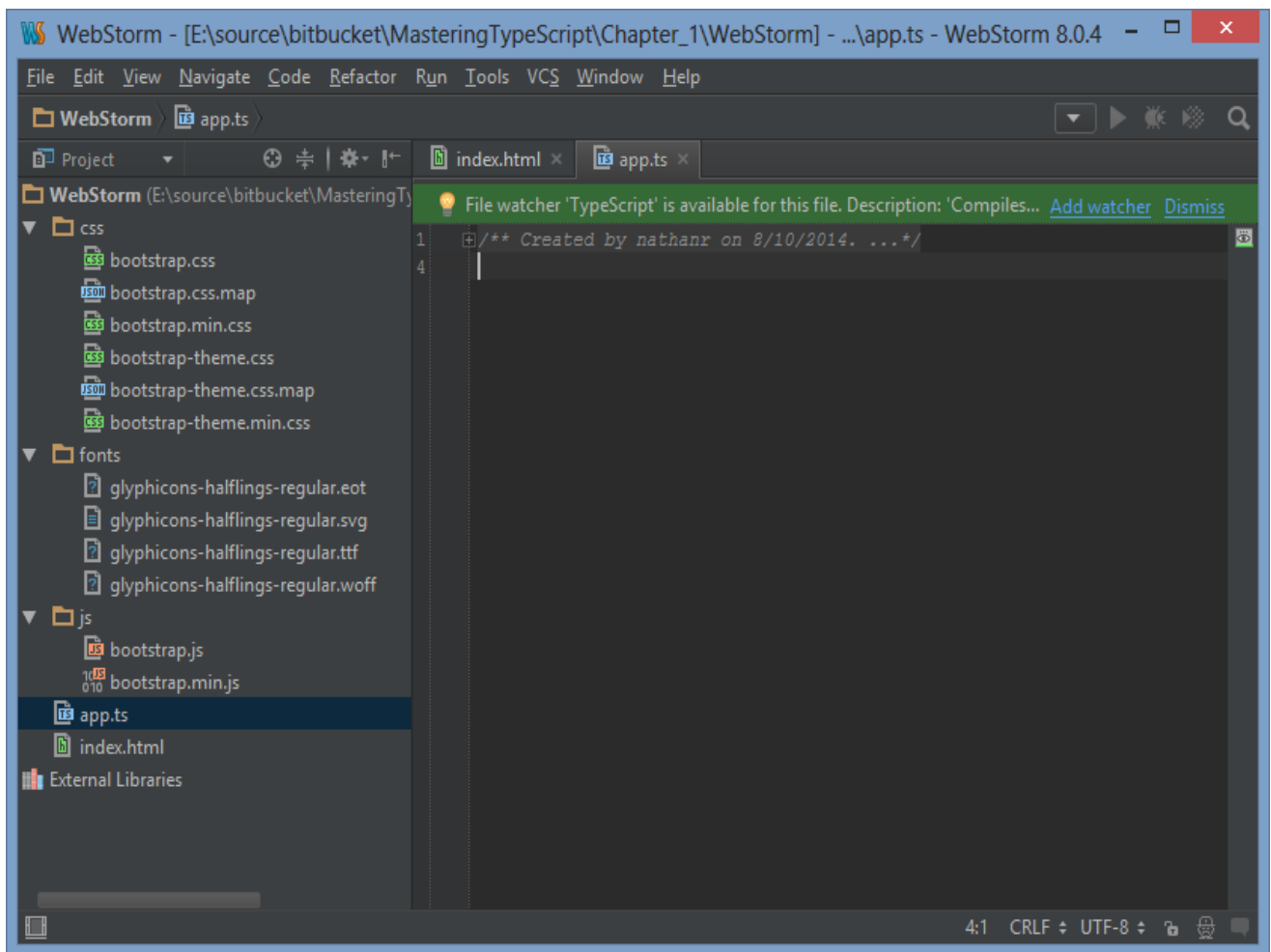
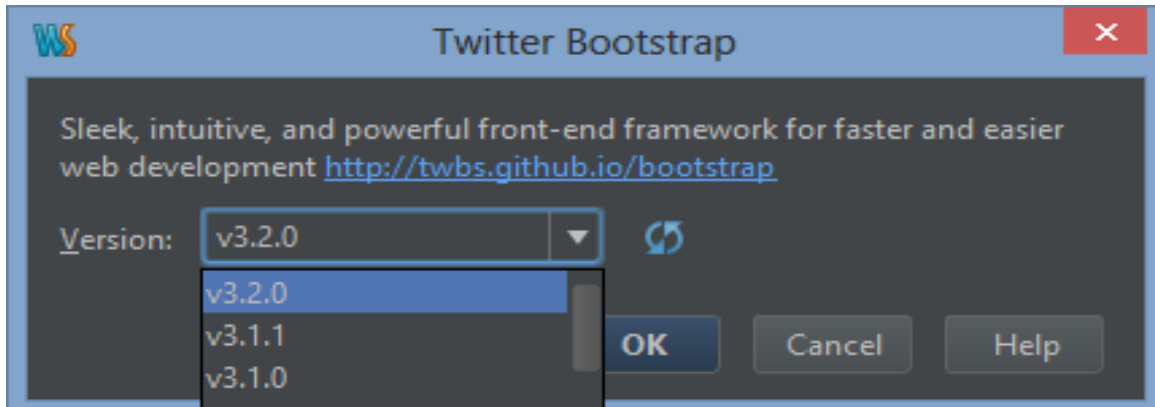


TypeScript HTML App

The time is: *Tue, 07 Oct 2014 14:27:04 GMT*

```
TypeScriptHTMLApp1
app.ts
<global> (no entries)
4   timerToken: number;
5
6   constructor(element: HTMLElement) {
7       this.element = element;
8       this.element.innerHTML += "The time is: ";
9       this.span = document.createElement('span');
10      this.element.appendChild(this.span);
11      this.span.innerText = new Date().toUTCString();
12  }
13
14  start() {
15      this.timerToken = setInterval(() => this.span.innerHTML = new Date().toUTCString(), 500);
16  }
17
18  stop() {
19      clearInterval(this.timerToken);
20  }
21
22  }
23
24  window.onload = () => {
25      var el = document.getElementById('content');
26      var greeter = new Greeter(el);
27      greeter.start();
28  };
100 %
```





New Watcher [X]

Name: TypeScript

Description: Compiles .ts files into .js files

Options

Output Filters... Show console: Error ▼

Immediate file synchronization Trigger watcher regardless of syntax errors

Compile main files only

Watcher Settings

File type: TypeScript files ▼

Scope: Project Files ▼ ...

Program: gram Files (x86)\Microsoft SDKs\TypeScript\1.0\tsc.exe ...

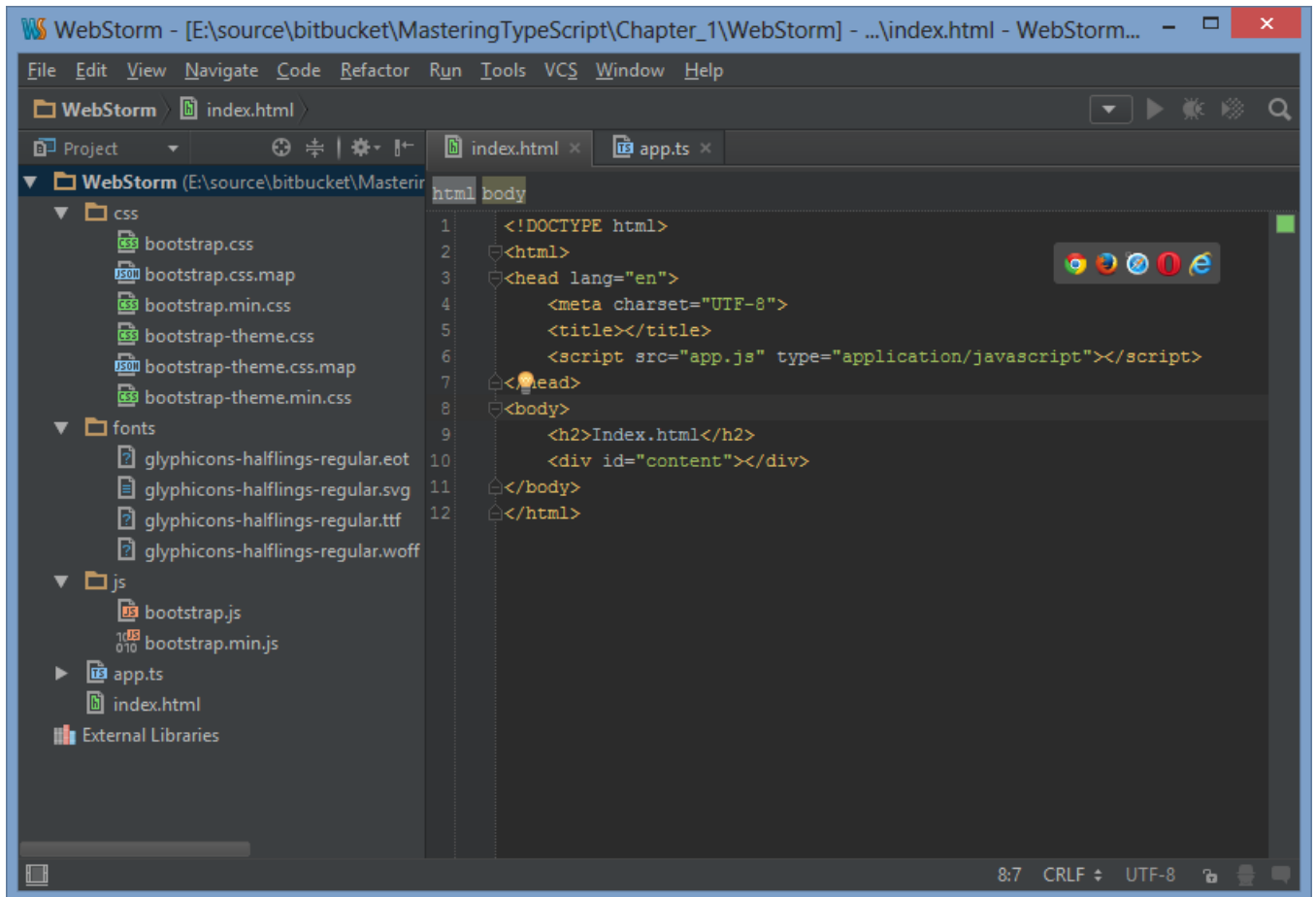
Arguments: --sourcemap \$FileName\$

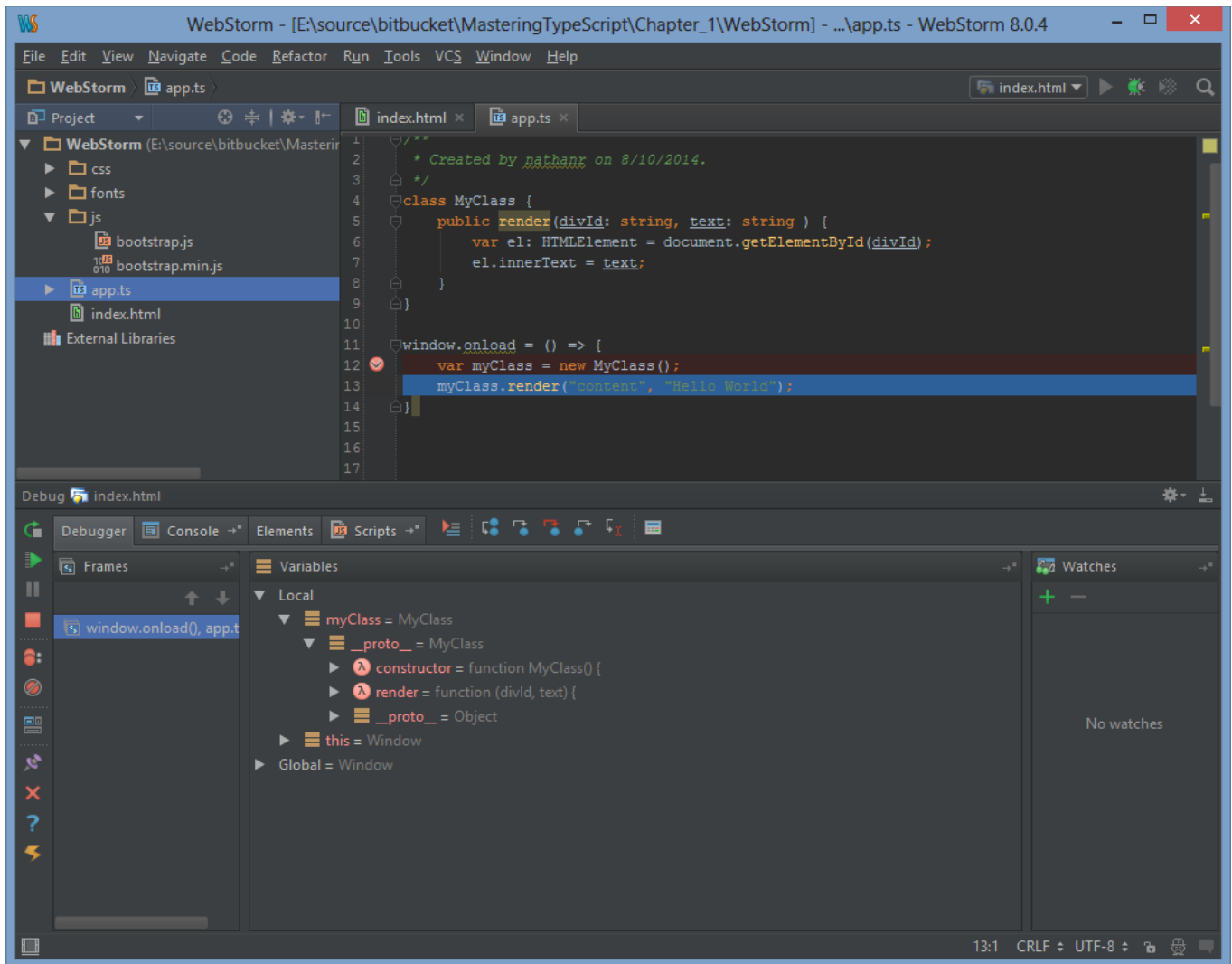
Working directory: \$FileDir\$...

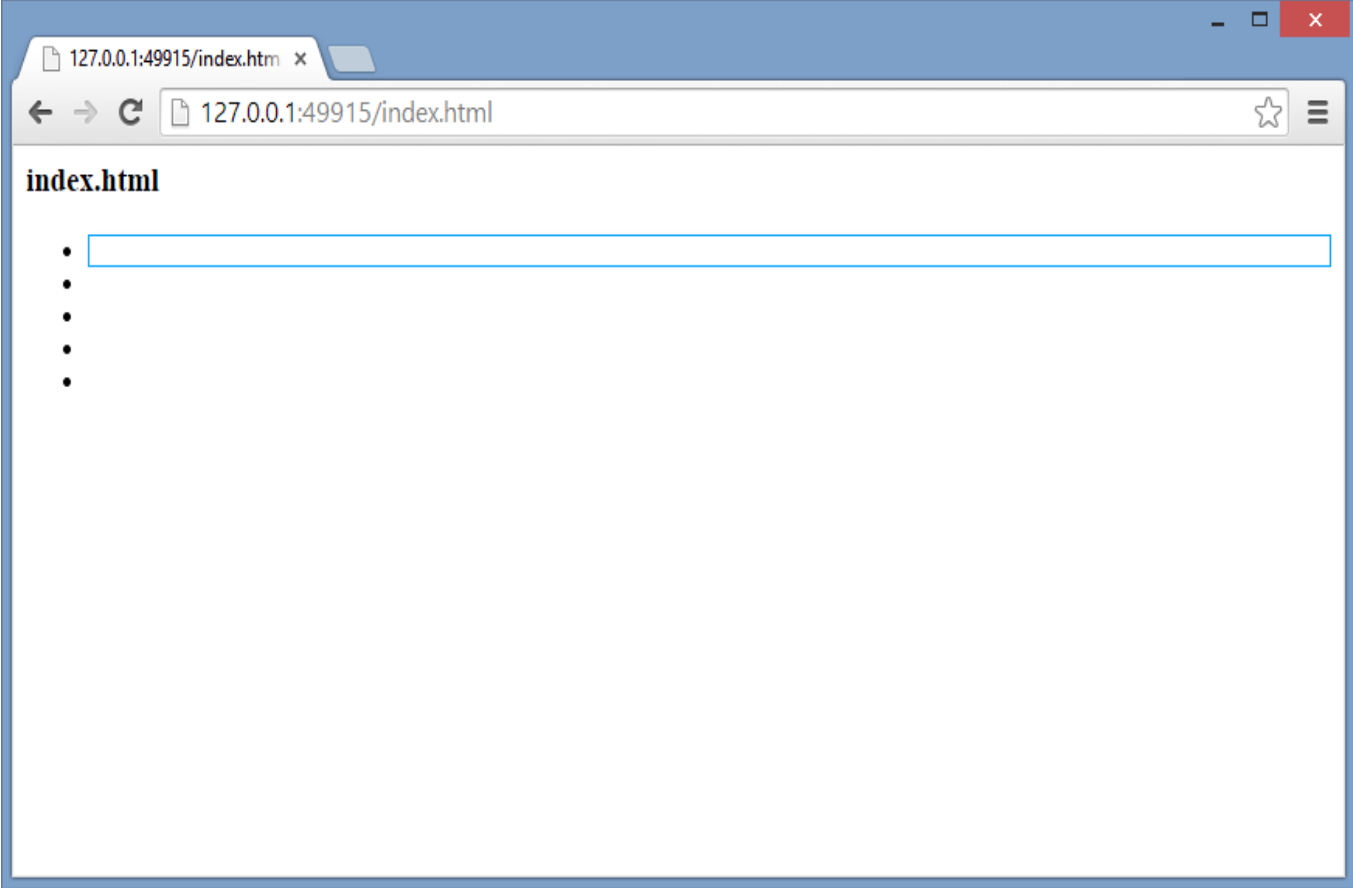
Environment variables: ...

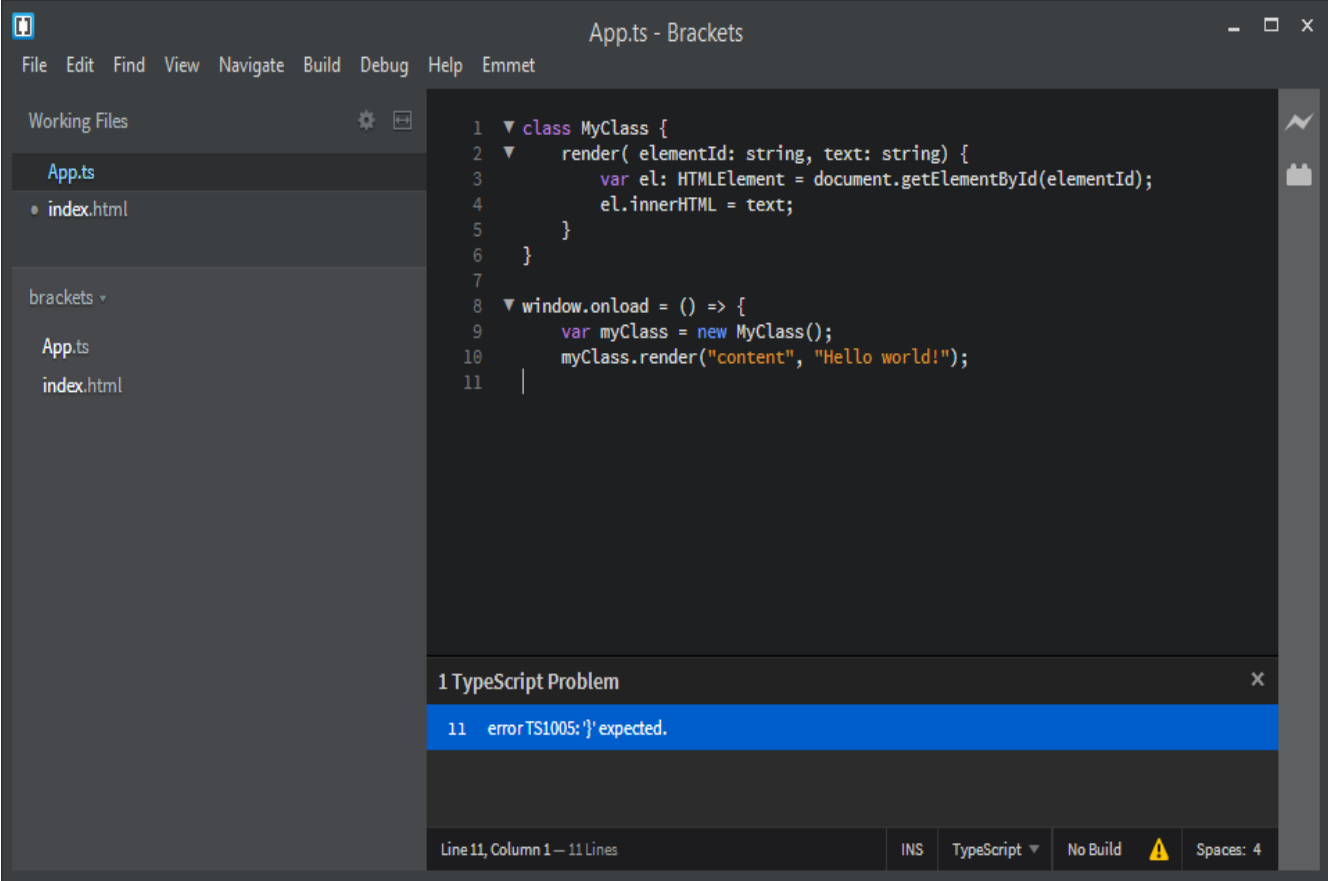
Output paths to refresh: %WithoutExtension\$.js:\$FileNameWithoutExtension\$.js.map

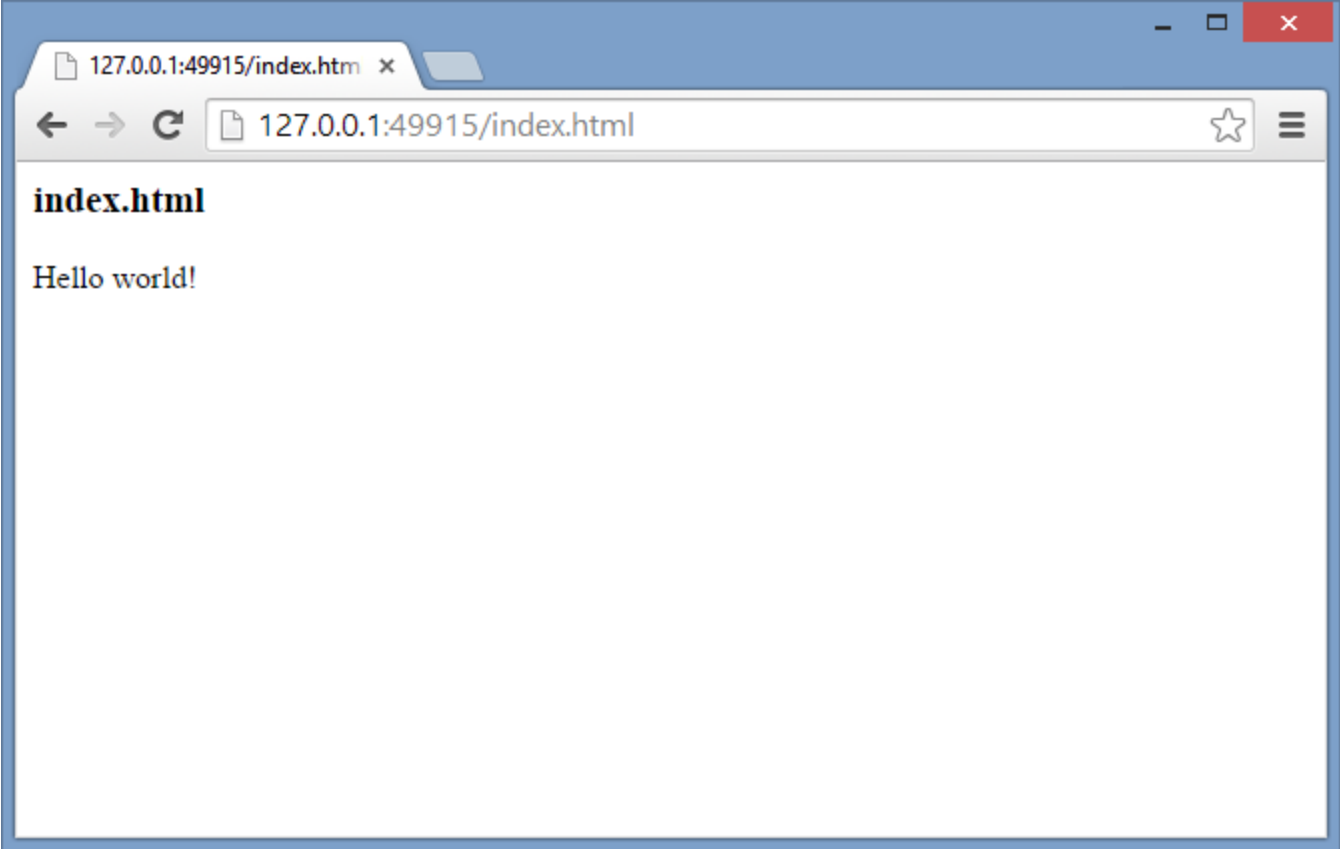
Create output file from stdout

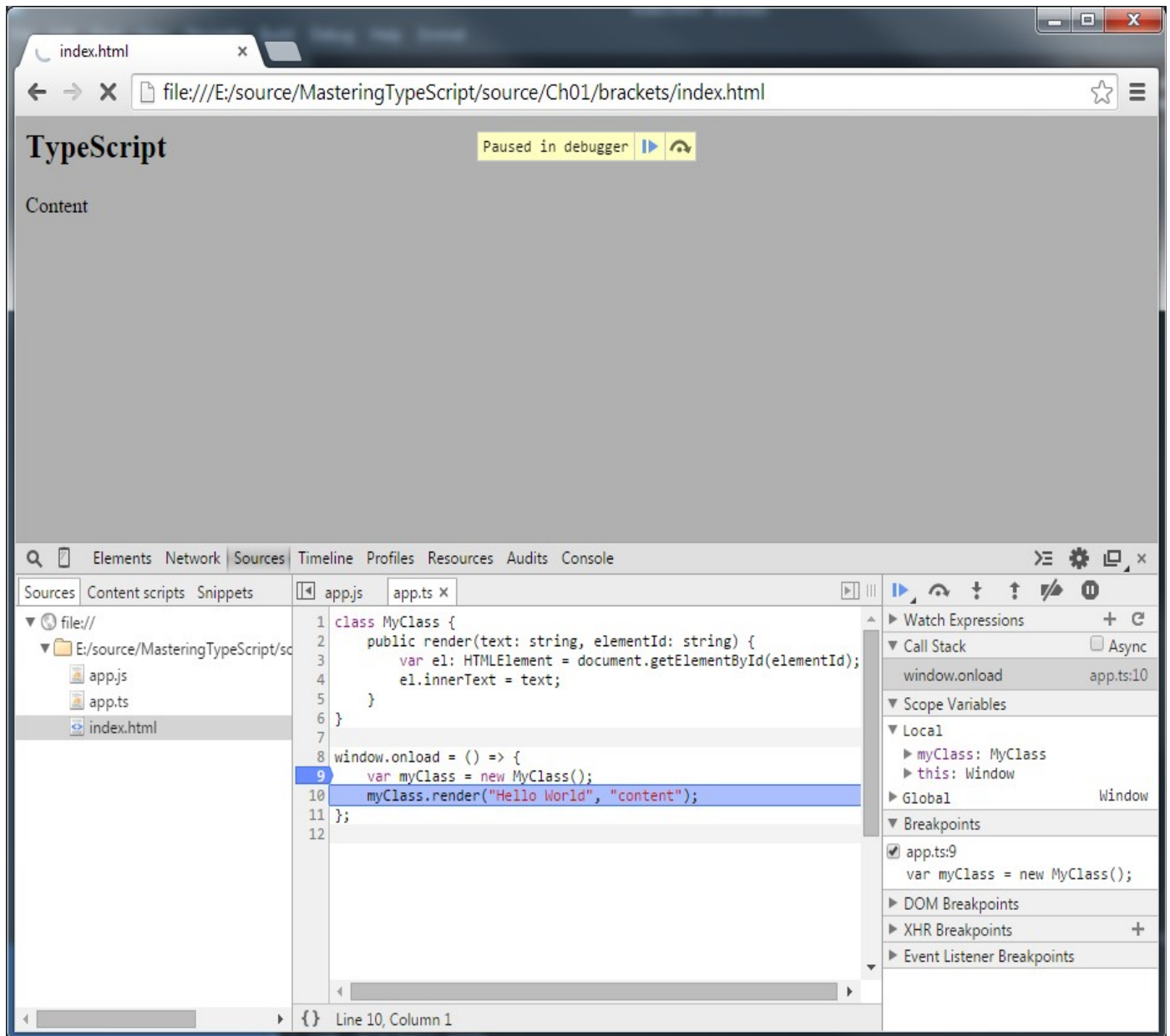




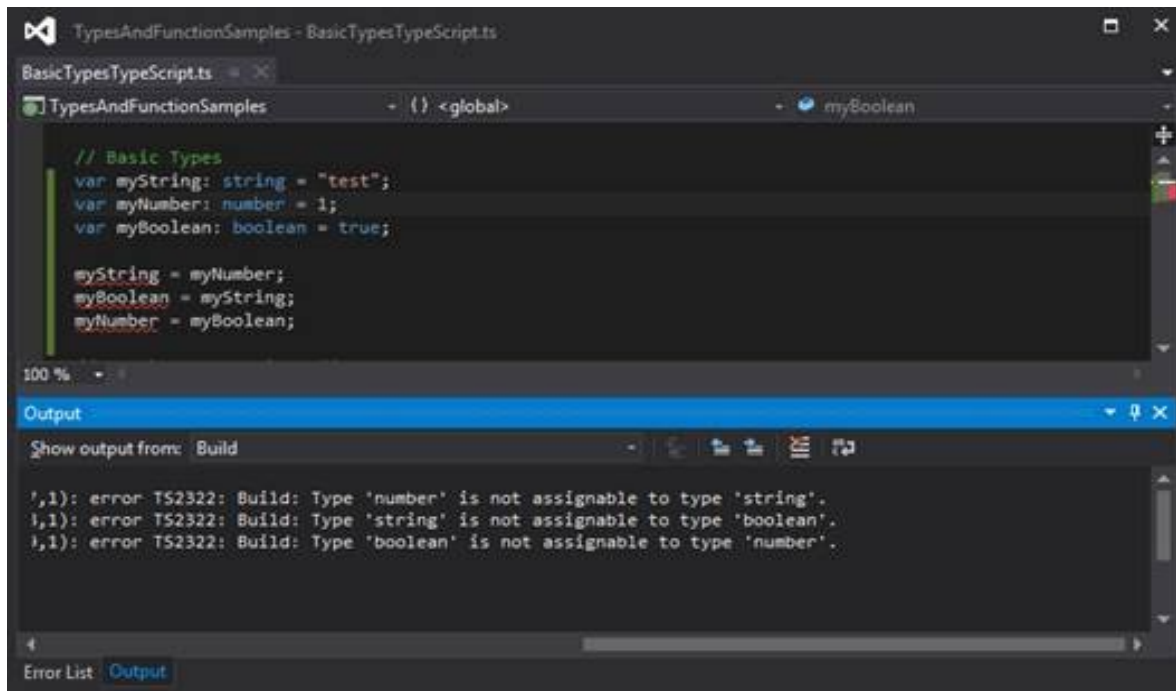








Chapter 2: Types, Variables and Function Techniques



The screenshot shows a code editor window titled "TypesAndFunctionSamples - BasicTypesTypeScript.ts". The code defines three variables: `myString` of type `string`, `myNumber` of type `number`, and `myBoolean` of type `boolean`. It then attempts to assign `myNumber` to `myString`, `myString` to `myBoolean`, and `myBoolean` to `myNumber`. The output window shows three TypeScript errors (TS2322) indicating that these assignments are not possible because the types do not match.

```
// Basic Types
var myString: string = "test";
var myNumber: number = 1;
var myBoolean: boolean = true;

myString = myNumber;
myBoolean = myString;
myNumber = myBoolean;
```

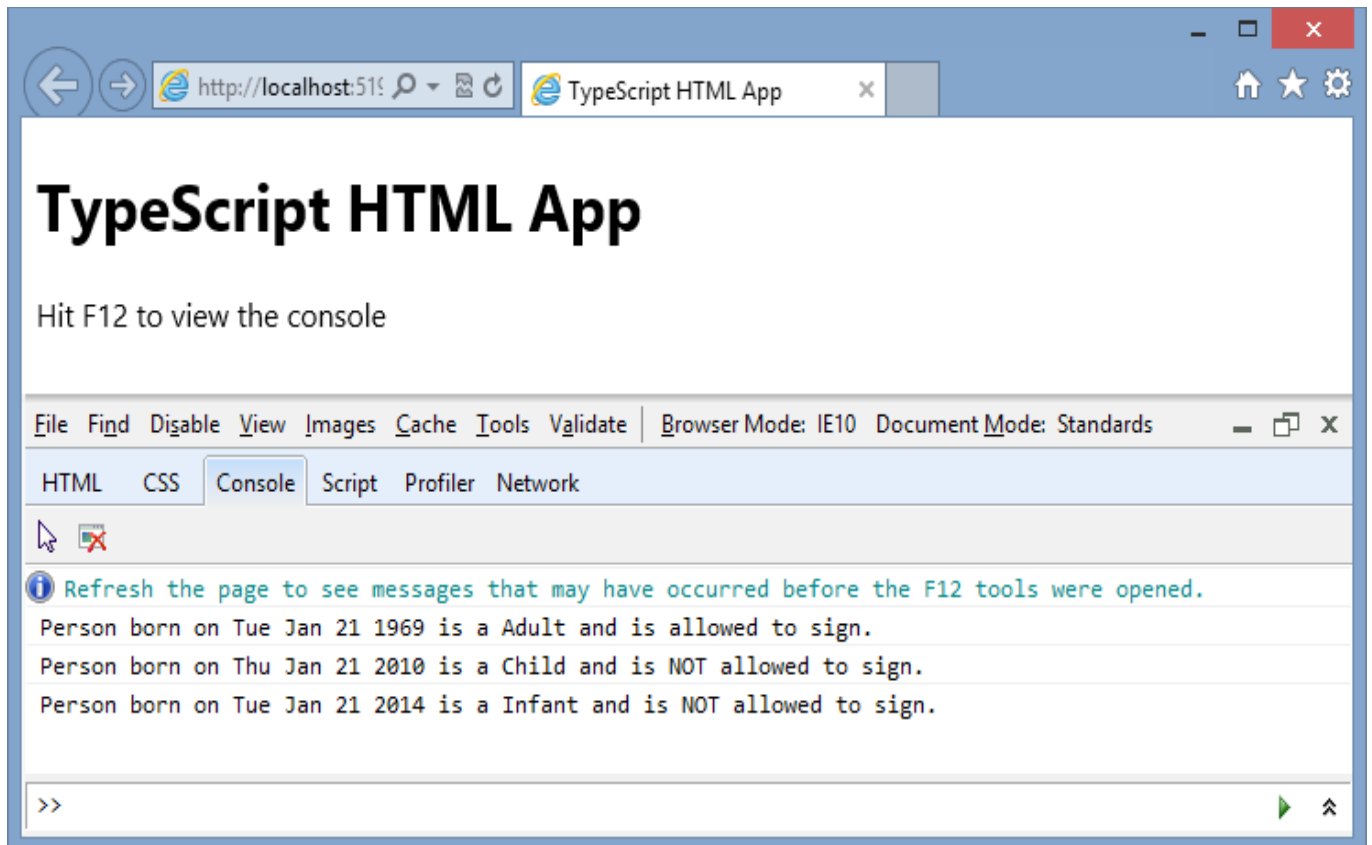
Output

```
1,1): error TS2322: Build: Type 'number' is not assignable to type 'string'.
1,1): error TS2322: Build: Type 'string' is not assignable to type 'boolean'.
1,1): error TS2322: Build: Type 'boolean' is not assignable to type 'number'.
```

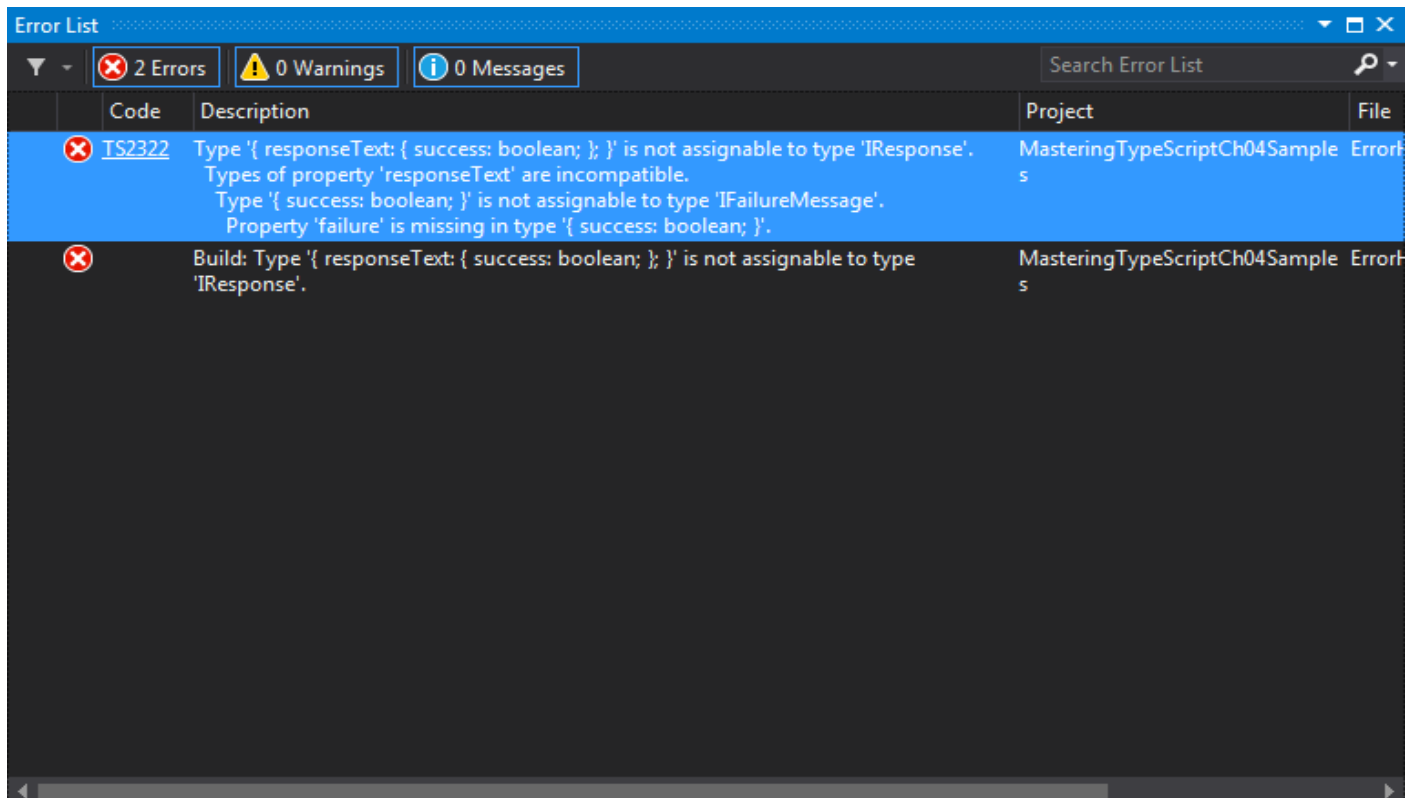
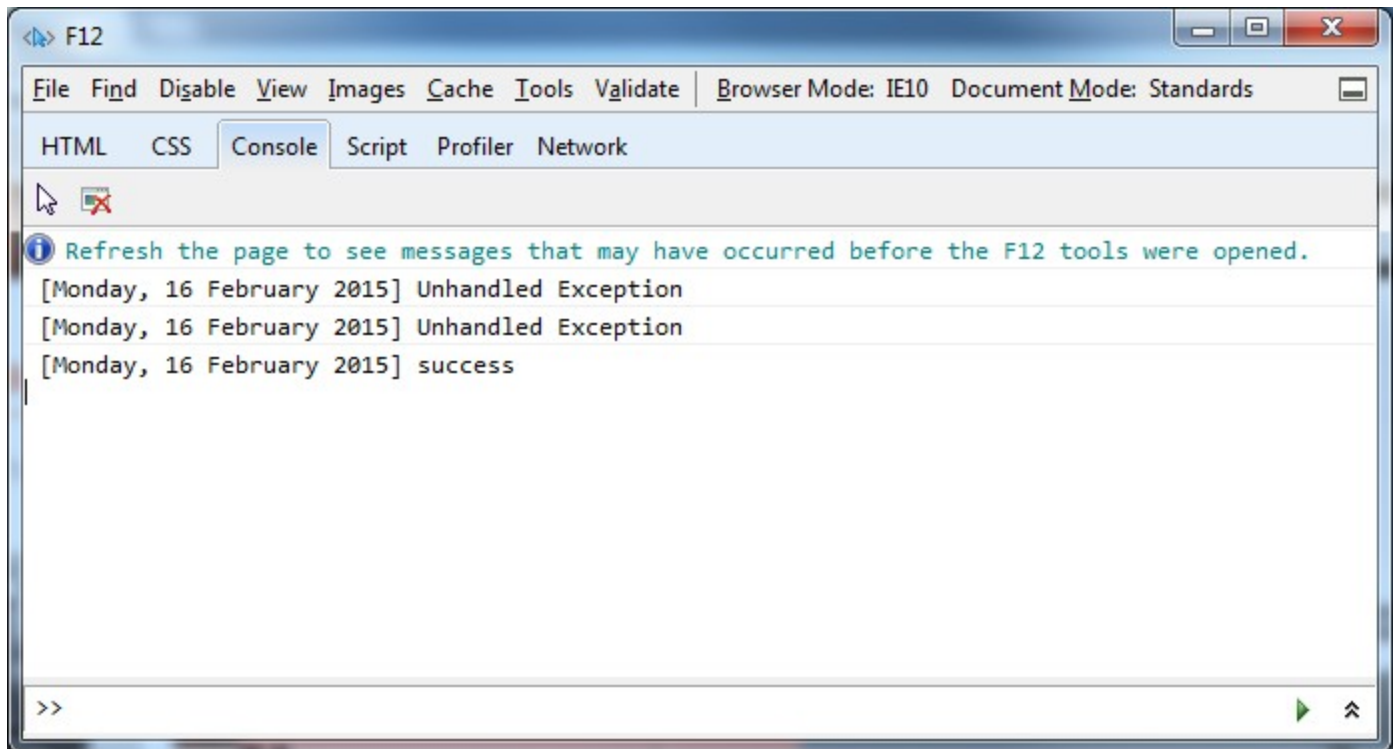
Chapter 3: Interfaces, Classes and Generics

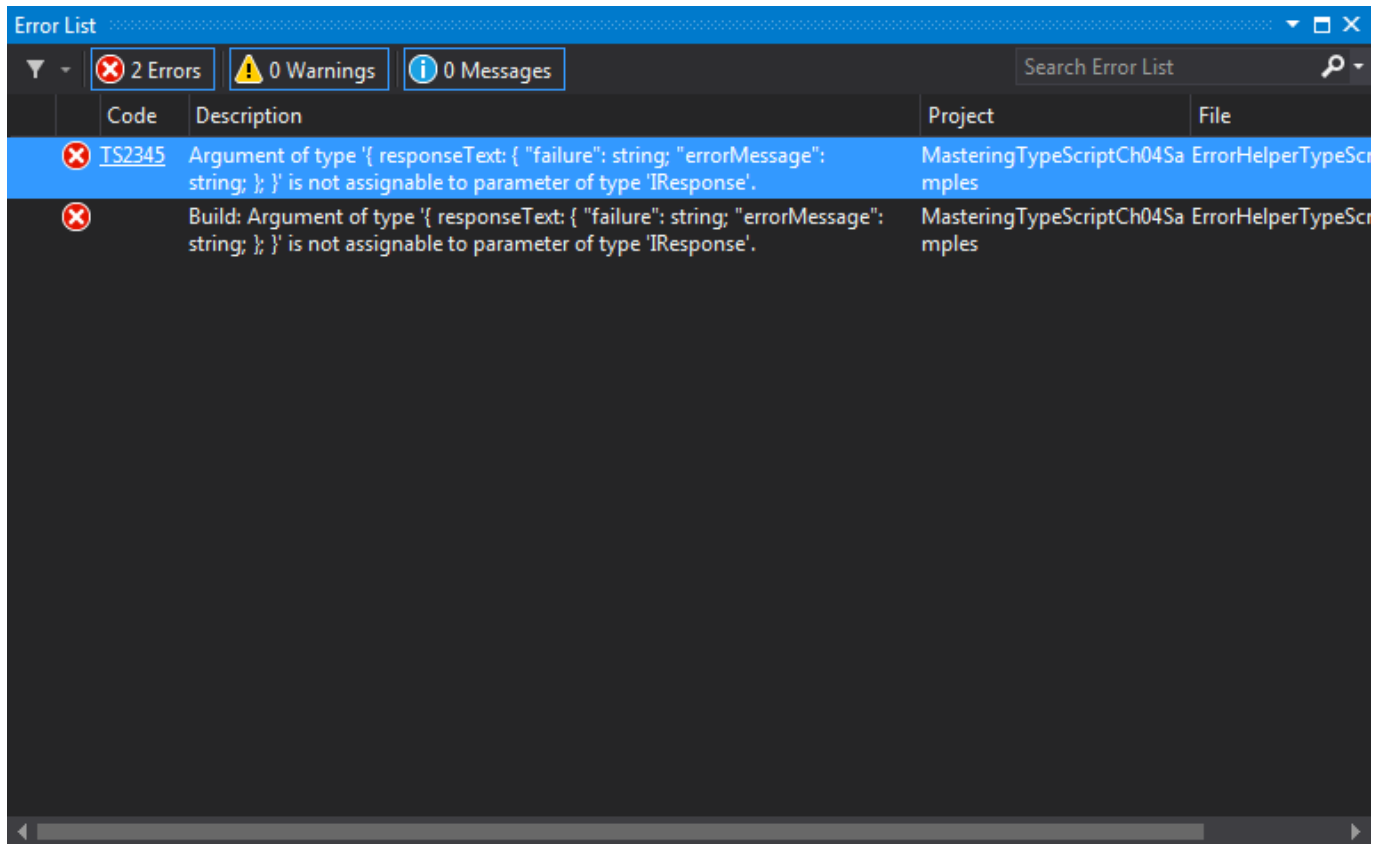
The screenshot shows a web browser window with the following elements:

- Address Bar:** `http://localhost:515/` and the page title "TypeScript HTML App".
- Main Content:** A large heading "TypeScript HTML App" and a subtext "Hit F12 to view the console."
- Browser Menu:** File, Find, Disable, View, Images, Cache, Tools, Validate, Browser Mode: IE10, Document Mode: Standards.
- Developer Tools:** A panel is open with tabs for HTML, CSS, Console, Script, Profiler, and Network. The "Console" tab is active.
- Console Log:**
 - Information icon: Refresh the page to see messages that may have occurred before the F12 tools were opened.
 - Log entry 1: A person with a birth date of :Wed Oct 29 2014 is categorised as : Infant and can sign : false
 - Log entry 2: A person with a birth date of :Sun Oct 29 2000 is categorised as : Child and can sign : false
 - Log entry 3: A person with a birth date of :Sun Oct 29 1950 is categorised as : Adult and can sign : true
- Bottom Bar:** A status bar with a double right arrow ">>" and a play button icon.

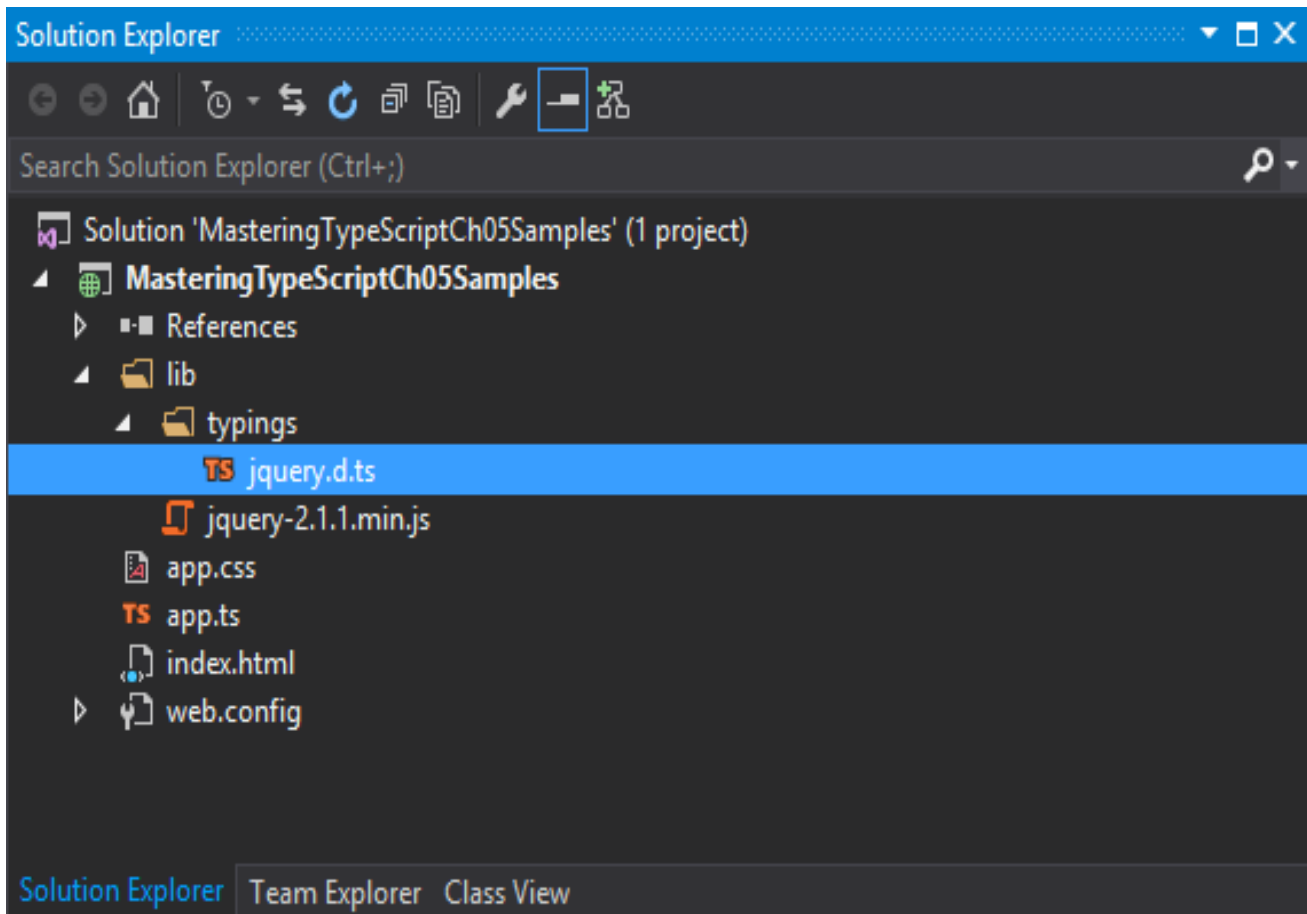


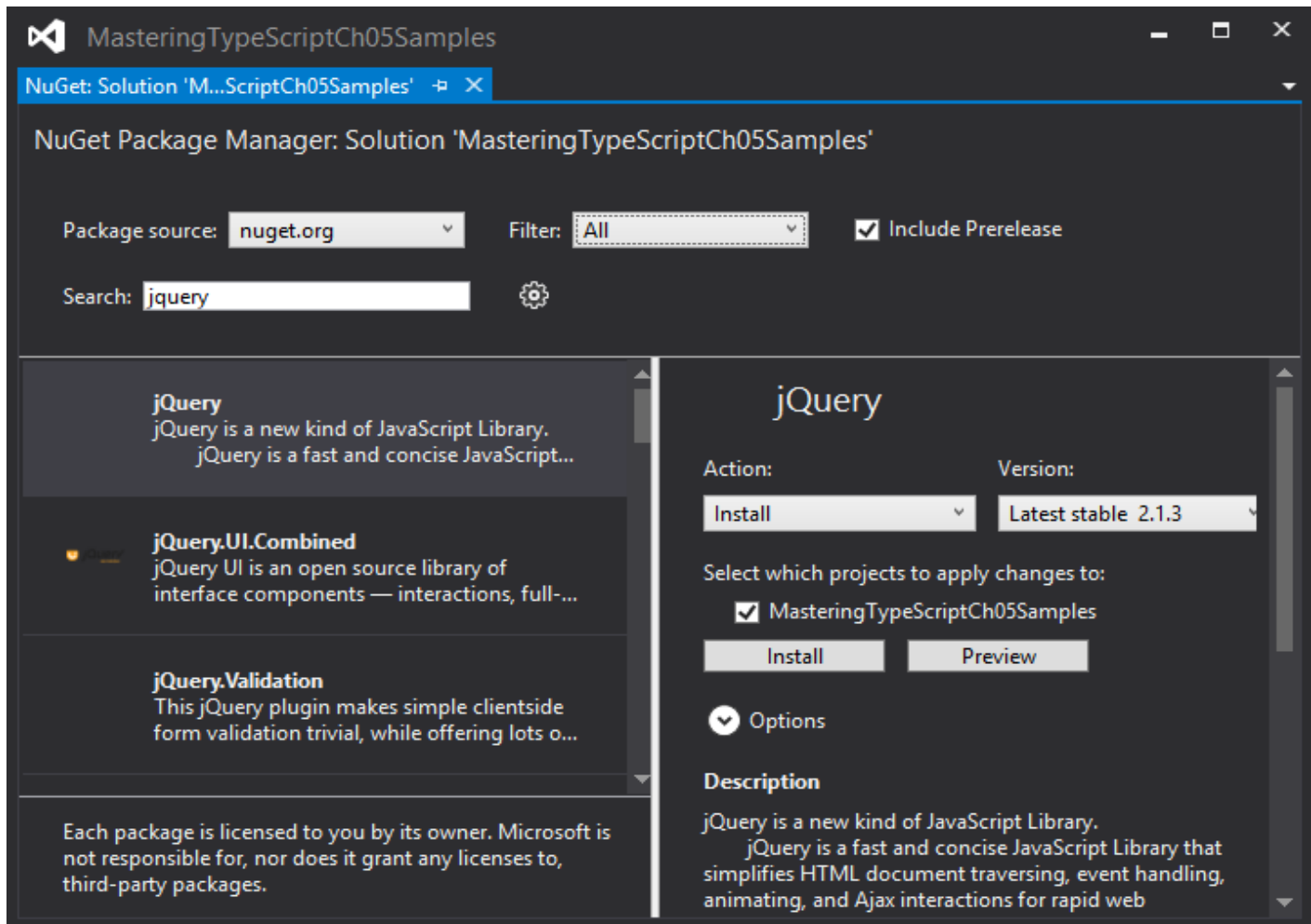
Chapter 4: Writing and Using Declaration Files





Chapter 5: Third Party Libraries





```
Ext.application(  
  <Ext.app.IApplication> {  
    // this JavaScript block is strongly  
    // type to be of Ext.app.IApplication  
    a  
  }  
);
```

- alternateClassName
- appFolder
- application
- appProperty
- autoCreateViewport
- callOverridden
- callParent
- callSuper
- clearListeners

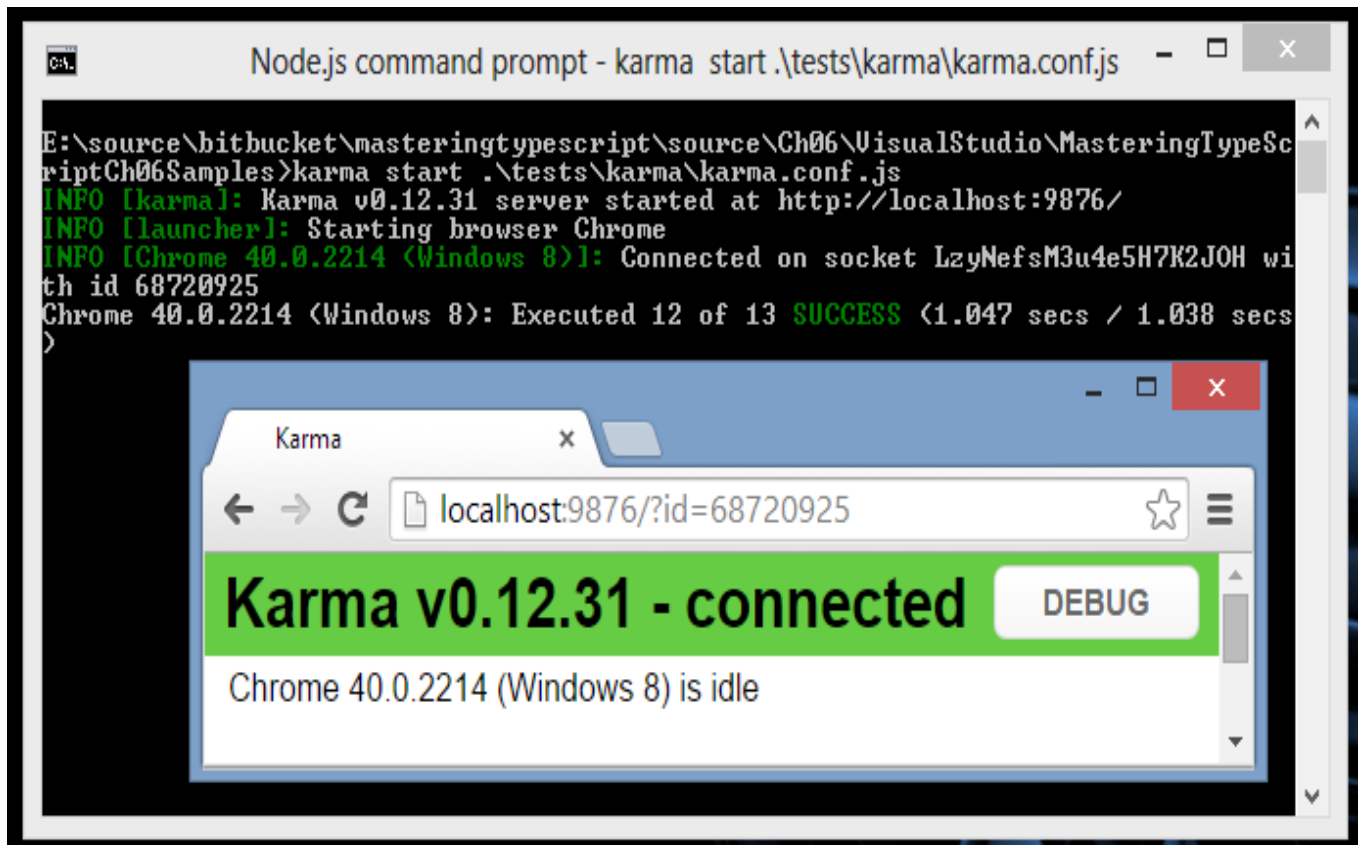
(property) Ext.app.IApplication.appFolder: string
[Config Option] (String)

Chapter 6: Test Driven Development

The screenshot shows a web browser window with the URL `http://localhost:64227`. The page displays the Jasmine 2.0.0 test runner interface. At the top right, it indicates the test suite "finished in 0.004s". A red banner shows "1 spec, 1 failure" and a "raise exceptions" button. Below this, a "Spec List | Failures" section is visible. A specific failure is highlighted: "tests/01_SimpleJasmineTests.ts should fail". The error message is "Expected undefined to be defined." followed by a stack trace:

```
Error: Expected undefined to be defined.  
  at stack (http://localhost:64227/Scripts/jasmine/jasmine.js:1293:11)  
  at buildExpectationResult (http://localhost:64227/Scripts/jasmine/jasmine.js:1265:5)  
  at expectationResultFactory (http://localhost:64227/Scripts/jasmine/jasmine.js:484:11)  
  at addExpectationResult (http://localhost:64227/Scripts/jasmine/jasmine.js:260:5)  
  at addExpectationResult (http://localhost:64227/Scripts/jasmine/jasmine.js:442:9)  
  at Anonymous function (http://localhost:64227/Scripts/jasmine/jasmine.js:1209:7)  
  at Anonymous function (http://localhost:64227/tests/01_SimpleJasmineTests.js:4:9)  
  at attemptSync (http://localhost:64227/Scripts/jasmine/jasmine.js:1510:9)  
  at run (http://localhost:64227/Scripts/jasmine/jasmine.js:1498:9)  
  at execute (http://localhost:64227/Scripts/jasmine/jasmine.js:1485:5)
```

```
Node.js command prompt - testem
TEST'EM 'SCRIPTS!
Open the URL below in a browser to connect.
http://localhost:7357/
-----+
  IE 10.0      | Chrome 40.0   | Firefox 35.0
  12/12 v     | 12/12 v     | 12/12 v
-----+-----
v 12 tests complete.
[Press ENTER to run tests; q to quit; p to pause]
```



http://localhost:64... Jasmine Spec Runner

Jasmine 2.0.0 finished in 0.008s

.....

5 specs, 0 failures raise exceptions

Player

- should be able to play a Song
- when song has been paused
 - should indicate that the song is currently paused
 - should be possible to resume
- tells the current song if the user has made it a favorite
- #resume
 - should throw an exception if song is already playing

Browser window showing Jasmine Spec Runner results. The page title is "Jasmine 2.0.0" and it indicates "finished in 0.006s". The summary shows "5 specs, 0 failures" and a "raise exceptions" checkbox. The test suite is "Player" and includes the following specs:

- should be able to play a Song
- when song has been paused
 - should indicate that the song is currently paused
 - should be possible to resume
- tells the current song if the user has made it a favorite

The browser's developer console is open, showing the following log entries:

```
##teamcity[progressStart 'Running Jasmine Tests']  
##teamcity[testSuiteStarted name=' Player' timestamp='2015-03-03T03:36:55.004']  
##teamcity[testStarted name=' should be able to play a Song' captureStandardOutput='true' timestamp=  
##teamcity[testFinished name=' should be able to play a Song' timestamp='2015-03-03T03:36:55.005']  
##teamcity[testSuiteStarted name=' when song has been paused' timestamp='2015-03-03T03:36:55.005']  
##teamcity[testStarted name=' should indicate that the song is currently paused' captureStandardOutp  
##teamcity[testFinished name=' should indicate that the song is currently paused' timestamp='2015-03-
```

http://localhost:64: Jasmine Spec Runner

```
##teamcity[testFinished name=' should be possible to resume' timestamp='2015-03-03T03:41:40.423']
##teamcity[testSuiteFinished name=' when song has been paused' timestamp='2015-03-03T03:41:40.423']
##teamcity[testStarted name=' tells the current song if the user has made it a favorite' captureStandardOutput='true'
timestamp='2015-03-03T03:41:40.423']
##teamcity[testFinished name=' tells the current song if the user has made it a favorite' timestamp='2015-03-
03T03:41:40.423']
##teamcity[testSuiteStarted name=' #resume' timestamp='2015-03-03T03:41:40.423']
##teamcity[testStarted name=' should throw an exception if song is already playing' captureStandardOutput='true'
timestamp='2015-03-03T03:41:40.423']
##teamcity[testFinished name=' should throw an exception if song is already playing' timestamp='2015-03-
03T03:41:40.427']
##teamcity[testSuiteFinished name=' #resume' timestamp='2015-03-03T03:41:40.427']
##teamcity[testSuiteFinished name=' Player' timestamp='2015-03-03T03:41:40.427']
##teamcity[progressFinish 'Running Jasmine Tests']
```

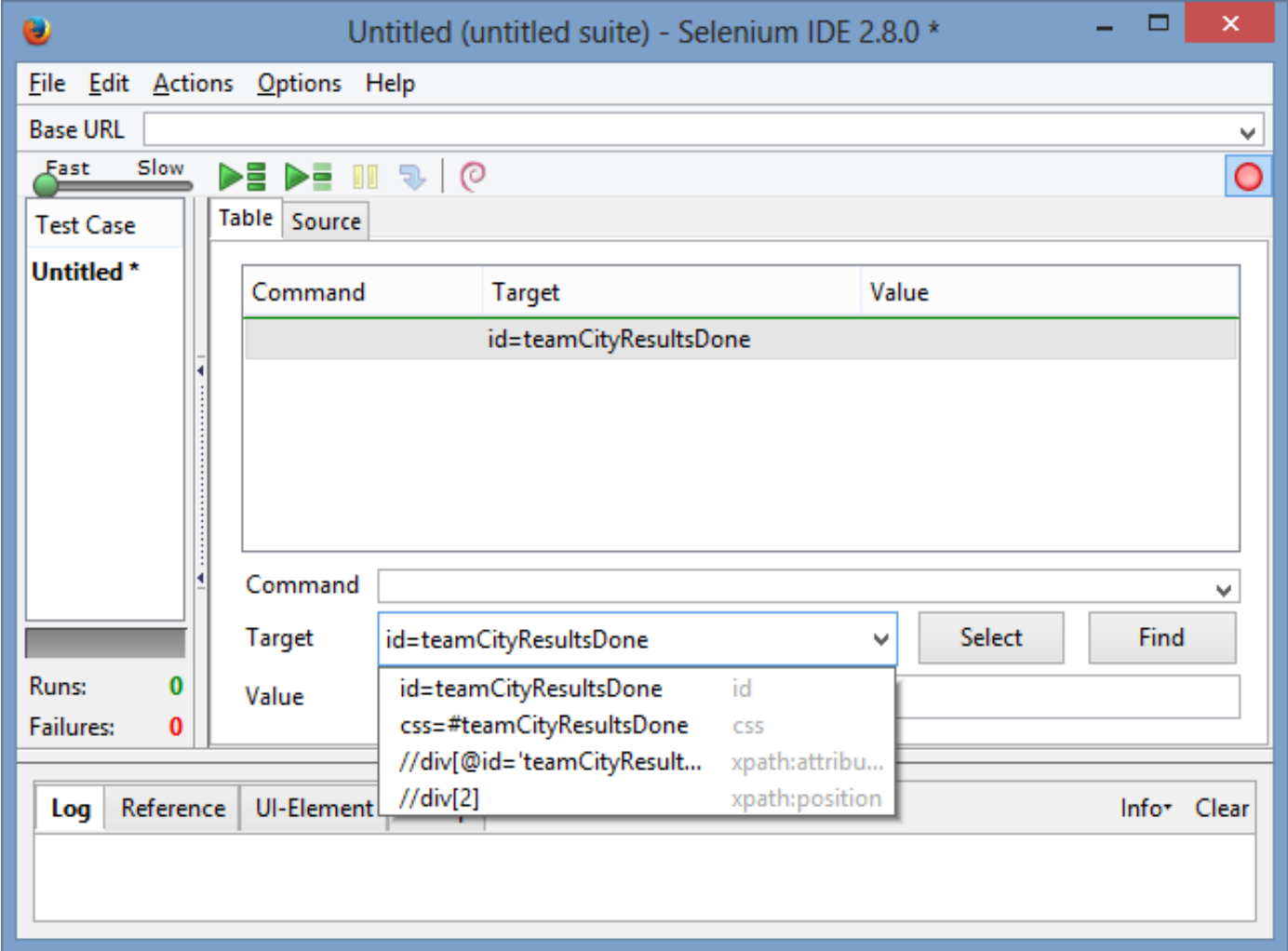
Jasmine 2.0.0 finished in 0.006s

.....

5 specs, 0 failures raise exceptions

Player

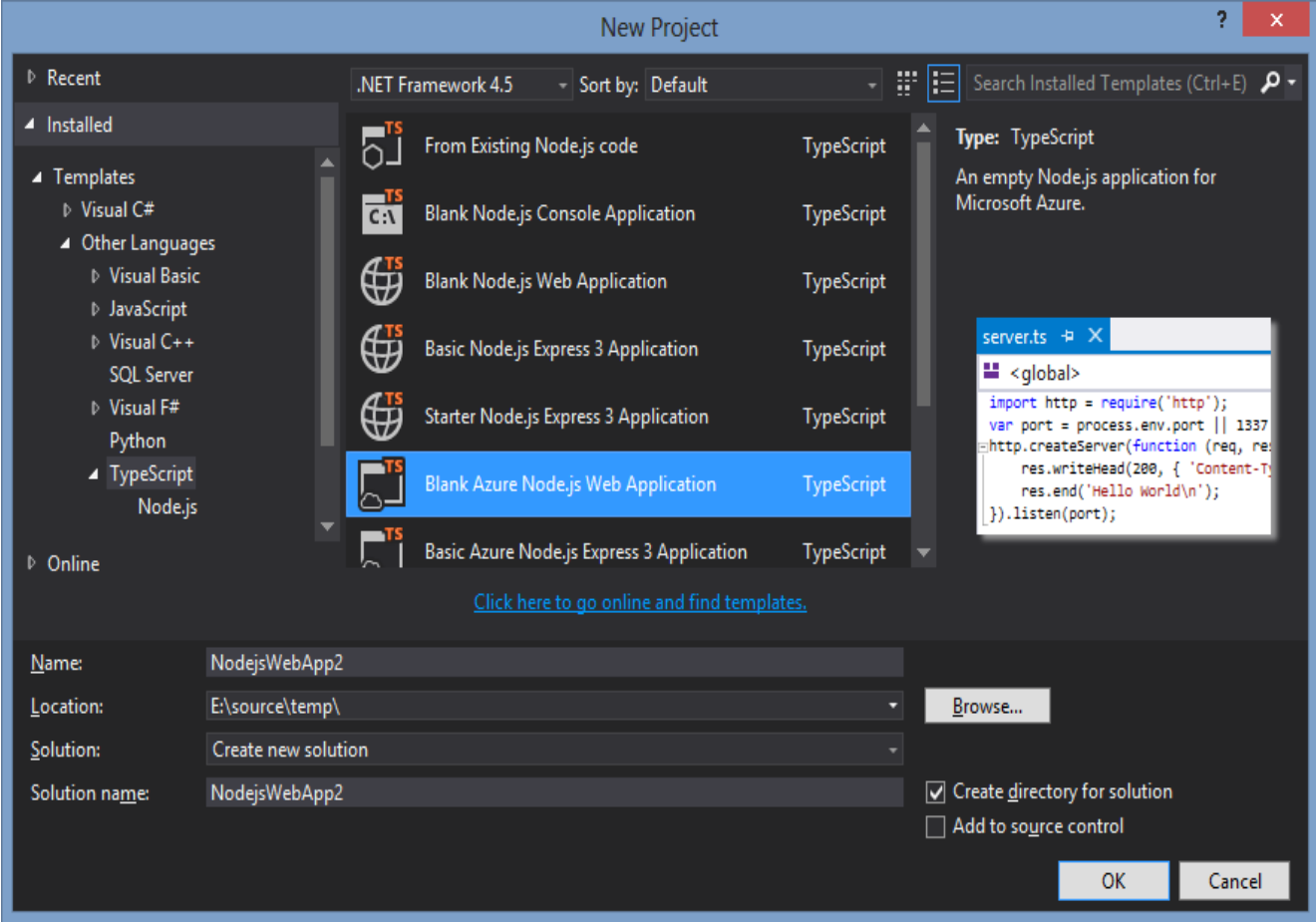
- should be able to play a Song
- when song has been paused
 - should indicate that the song is currently paused
 - should be possible to resume
- tells the current song if the user has made it a favorite
- #resume
 - should throw an exception if song is already playing

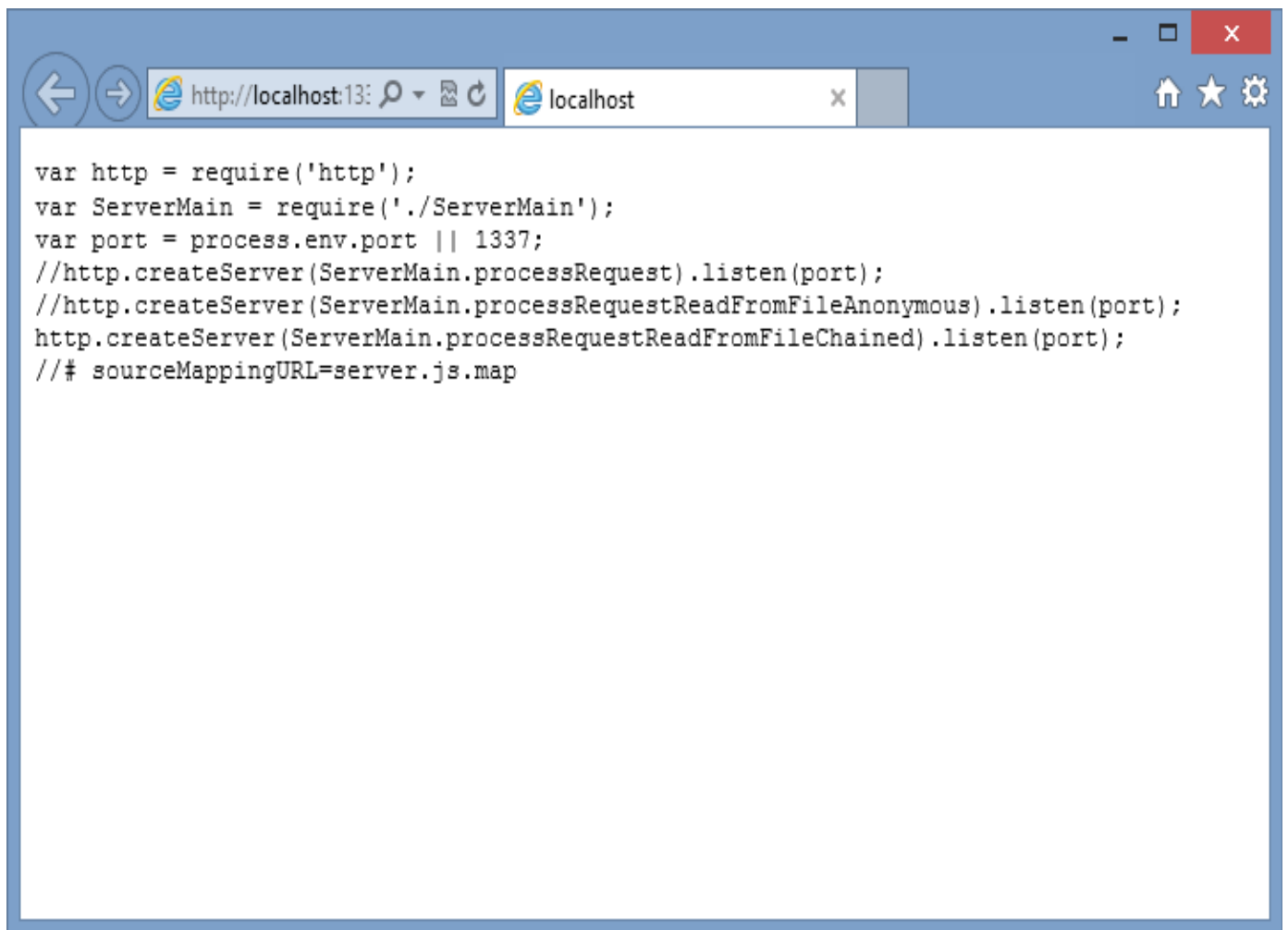


```
λ protractor .\tests\protractor\protractor.conf.js
Using the selenium server at http://localhost:4444/wd/hub
##teamcity[progressStart 'Running Jasmine Tests']
##teamcity[testSuiteStarted name=' Player' timestamp='2014-12-06T08:54:27.827']
##teamcity[testStarted name=' should be able to play a Song' captureStandardOutput='true' timestamp='2014-12-06T08:54:27.829']
##teamcity[testFinished name=' should be able to play a Song' timestamp='2014-12-06T08:54:27.832']
##teamcity[testSuiteStarted name=' when song has been paused' timestamp='2014-12-06T08:54:27.832']
##teamcity[testStarted name=' should indicate that the song is currently paused' captureStandardOutput='true' timestamp='2014-12-06T08:54:27.833']
##teamcity[testFinished name=' should indicate that the song is currently paused' timestamp='2014-12-06T08:54:27.834']
##teamcity[testStarted name=' should be possible to resume' captureStandardOutput='true' timestamp='2014-12-06T08:54:27.834']
##teamcity[testFinished name=' should be possible to resume' timestamp='2014-12-06T08:54:27.835']
##teamcity[testSuiteFinished name=' when song has been paused' timestamp='2014-12-06T08:54:27.835']
##teamcity[testStarted name=' tells the current song if the user has made it a favorite' captureStandardOutput='true' timestamp='2014-12-06T08:54:27.836']
##teamcity[testFinished name=' tells the current song if the user has made it a favorite' timestamp='2014-12-06T08:54:27.838']
##teamcity[testSuiteStarted name=' #resume' timestamp='2014-12-06T08:54:27.839']
##teamcity[testStarted name=' should throw an exception if song is already playing' captureStandardOutput='true' timestamp='2014-12-06T08:54:27.840']
##teamcity[testFinished name=' should throw an exception if song is already playing' timestamp='2014-12-06T08:54:27.842']
##teamcity[testSuiteFinished name=' #resume' timestamp='2014-12-06T08:54:27.843']
##teamcity[testSuiteFinished name=' Player' timestamp='2014-12-06T08:54:27.844']
##teamcity[progressFinish 'Running Jasmine Tests']
.

Finished in 4.406 seconds
1 test, 1 assertion, 0 failures
```

Chapter 7: Modularization





The image shows a browser window with a blue header bar. The address bar contains the URL `http://localhost:1337` and a search icon. The page content is a JavaScript code snippet for an HTTP server. The code defines a `ServerMain` module and uses `http.createServer` to listen on port 1337. It includes comments for different request processing methods: `processRequest`, `processRequestReadFromFileAnonymous`, and `processRequestReadFromFileChained`. A comment at the end indicates the source mapping URL: `sourceMappingURL=server.js.map`.

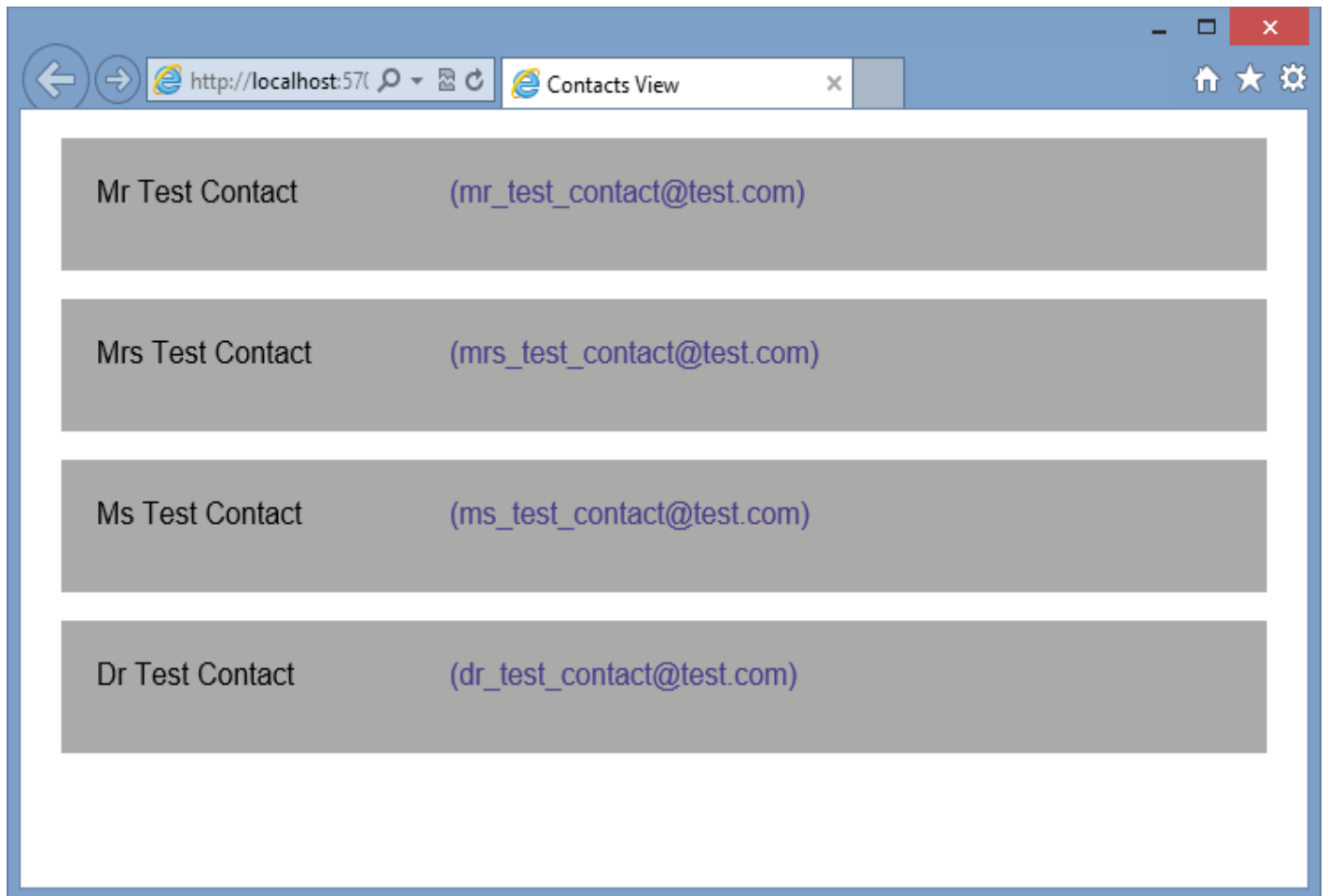
```
var http = require('http');
var ServerMain = require('./ServerMain');
var port = process.env.port || 1337;
//http.createServer(ServerMain.processRequest).listen(port);
//http.createServer(ServerMain.processRequestReadFromFileAnonymous).listen(port);
http.createServer(ServerMain.processRequestReadFromFileChained).listen(port);
//# sourceMappingURL=server.js.map
```

Network - http://localhost:57013/tscode/tests/SpecRunner.html

Inspector Console Debugger Style Editor Performance Network

✓	Method	File	Domain	Type	Size	0 ms	160 ms
▲ 304	GET	SpecRunner.html	localhost:57013	html	0.38 kB	→ 4 ms	
▲ 304	GET	jasmine.css	localhost:57013	css	19.17 kB	→ 5 ms	
▲ 304	GET	require.js	localhost:57013	js	81.13 kB	→ 3 ms	
▲ 304	GET	TestConfig.js	localhost:57013	js	1.39 kB	→ 4 ms	
■ 404	GET	jquery-2.1.1.1.js	localhost:57013	html	4.90 kB		→ 8 ms

All HTML CSS JS XHR Fonts Images Media Flash Other 5 requests, 106.99 kB, 0.16 s Cle

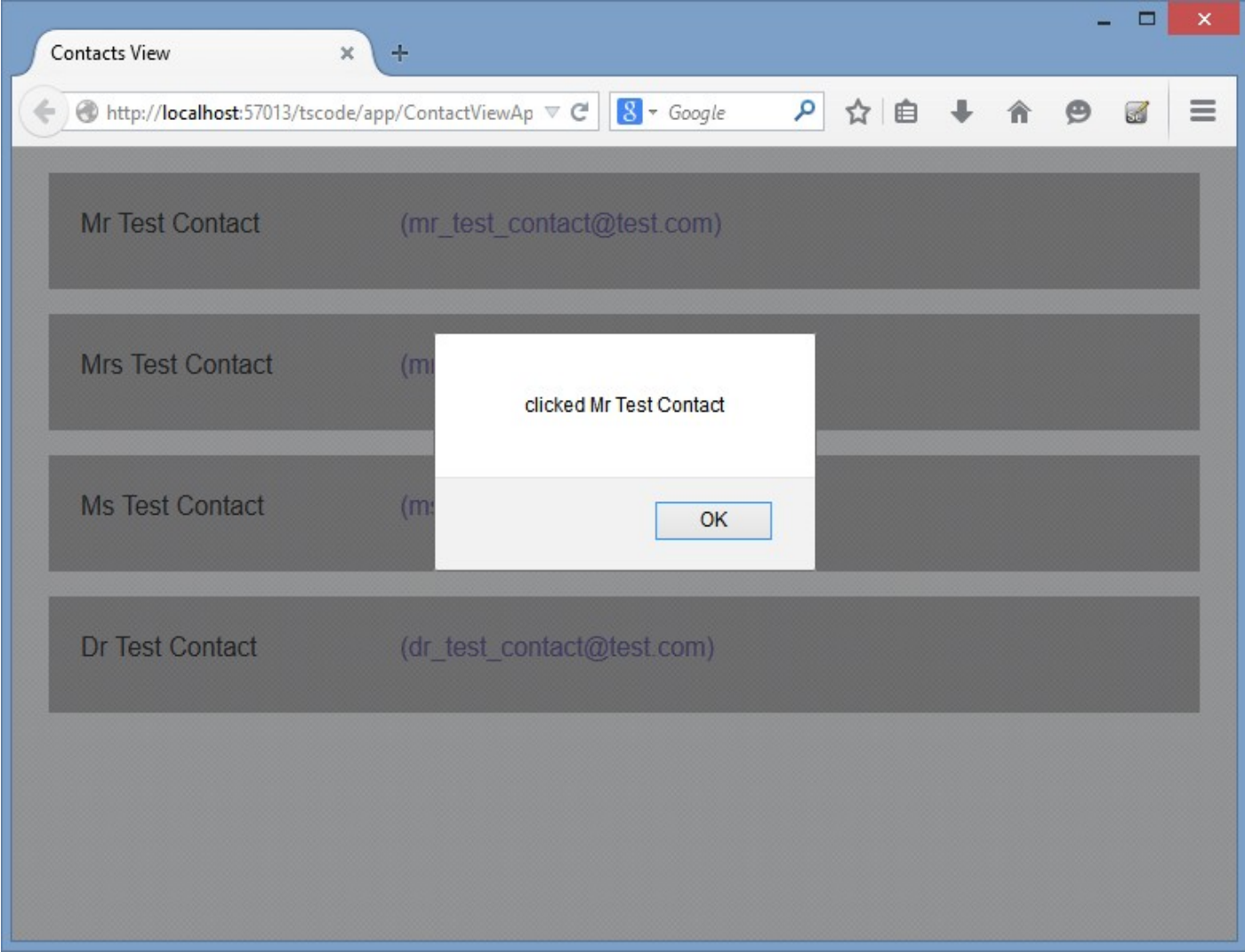


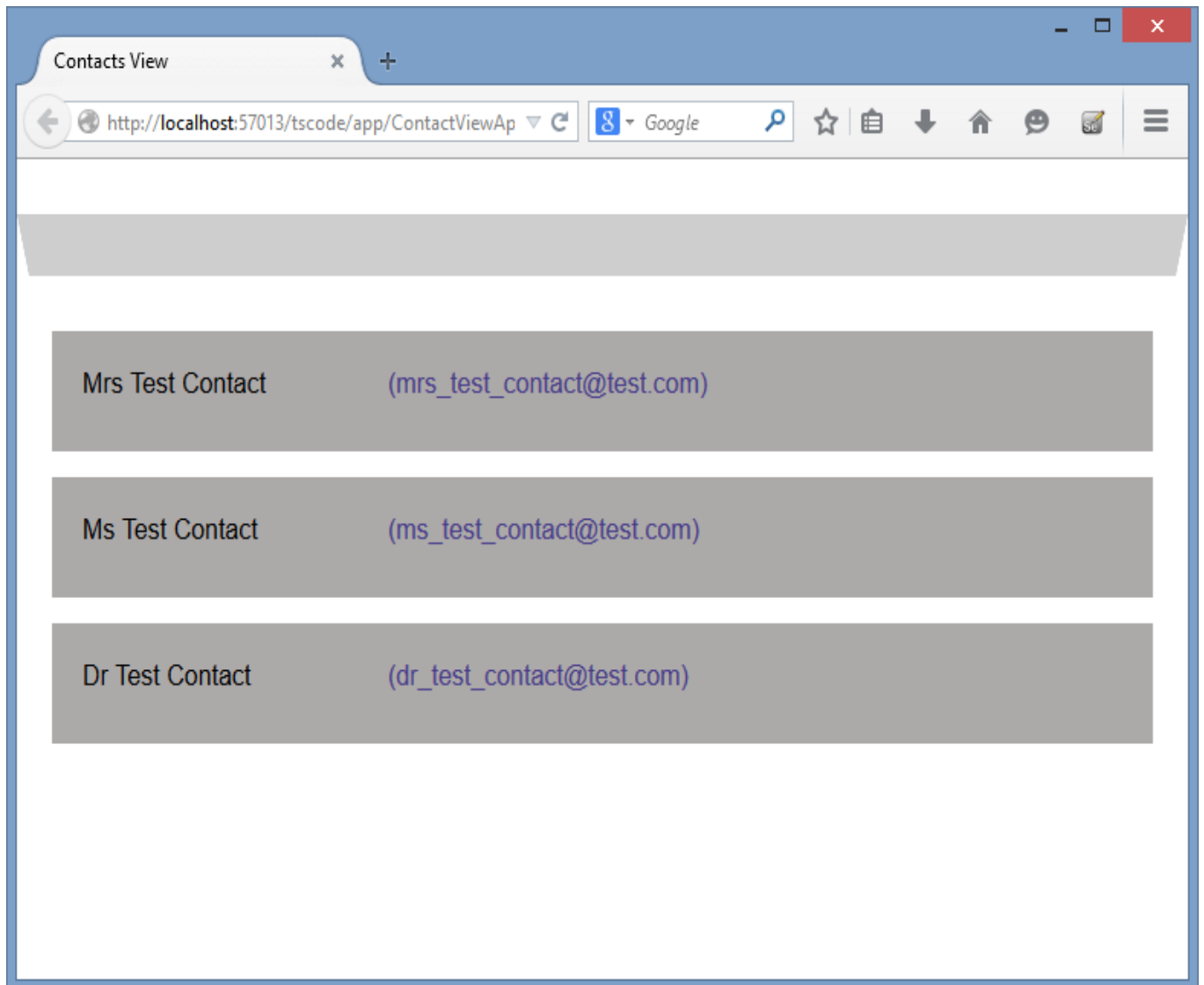
Mr Test Contact (mr_test_contact@test.com)

Mrs Test Contact (mrs_test_contact@test.com)

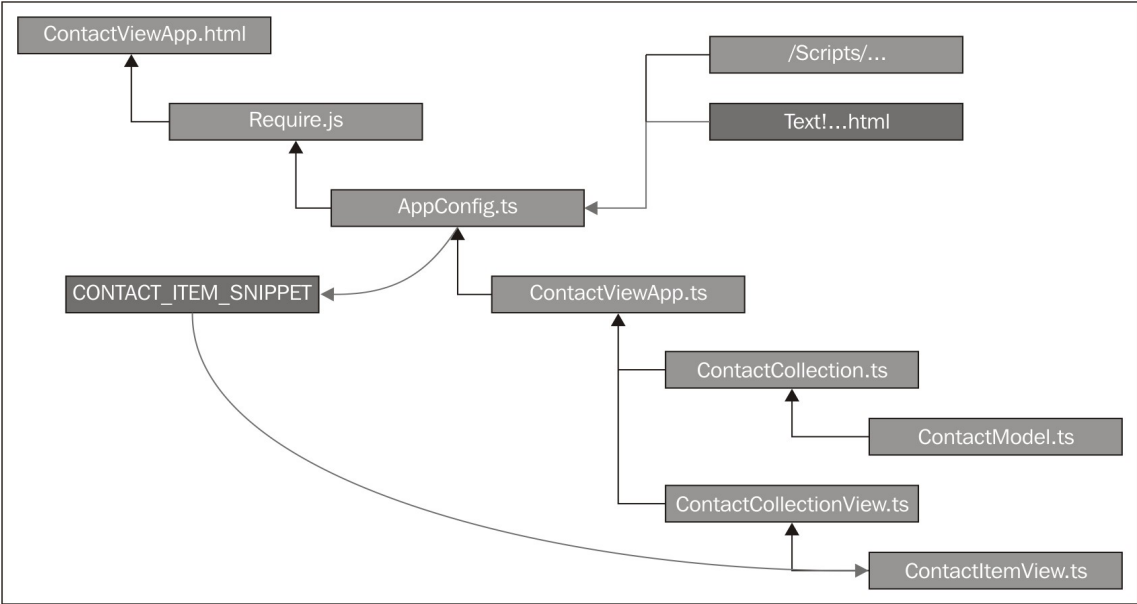
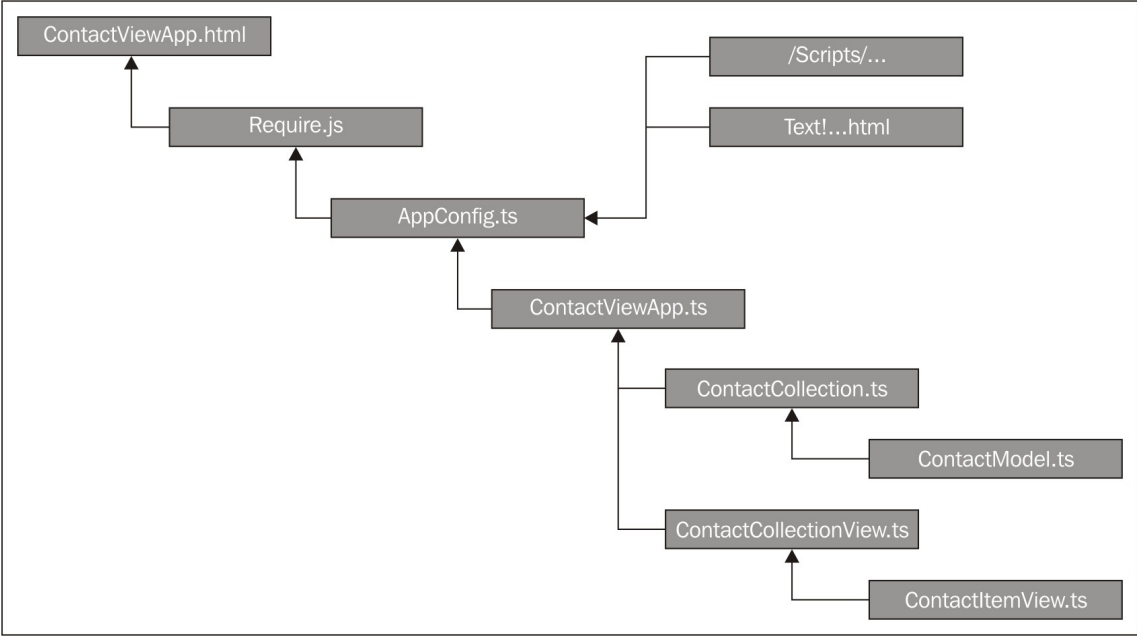
Ms Test Contact (ms_test_contact@test.com)

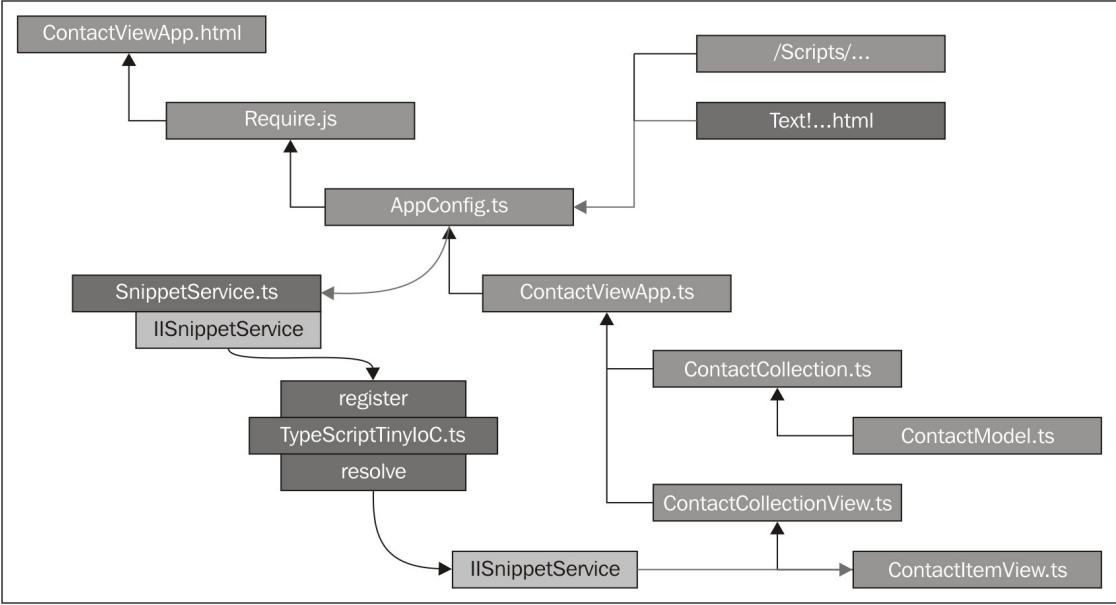
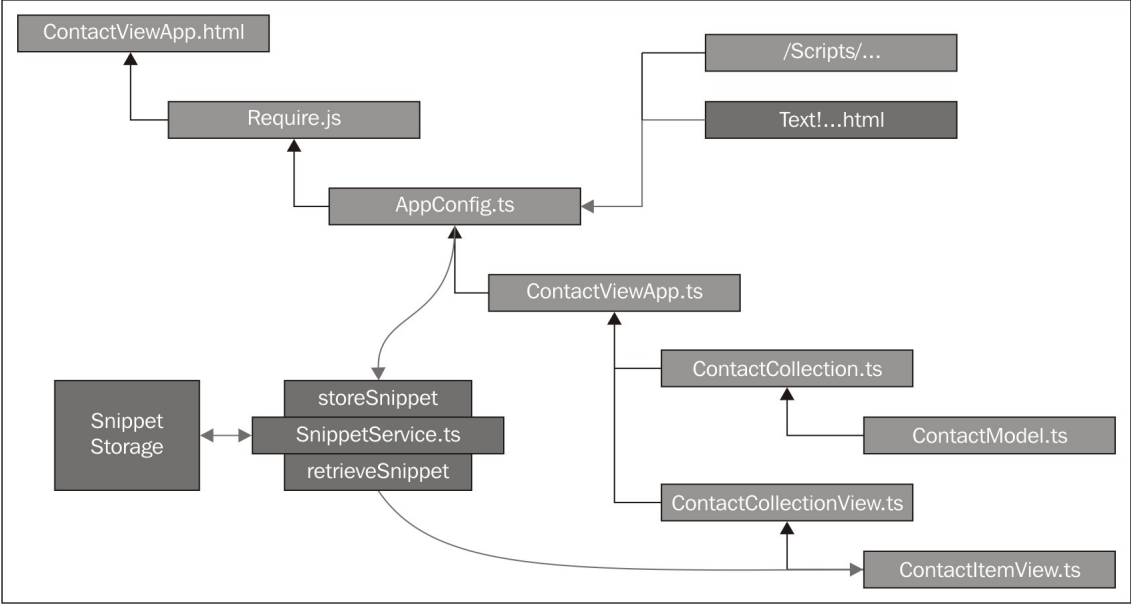
Dr Test Contact (dr_test_contact@test.com)





Chapter 8: Object-oriented Programming with TypeScript





AMD SpecRunner

http://localhost:61250

Jasmine 2.1.2 finished in 0.057s

.....XX..X

Ran 30 of 31 specs - run all raise exceptions

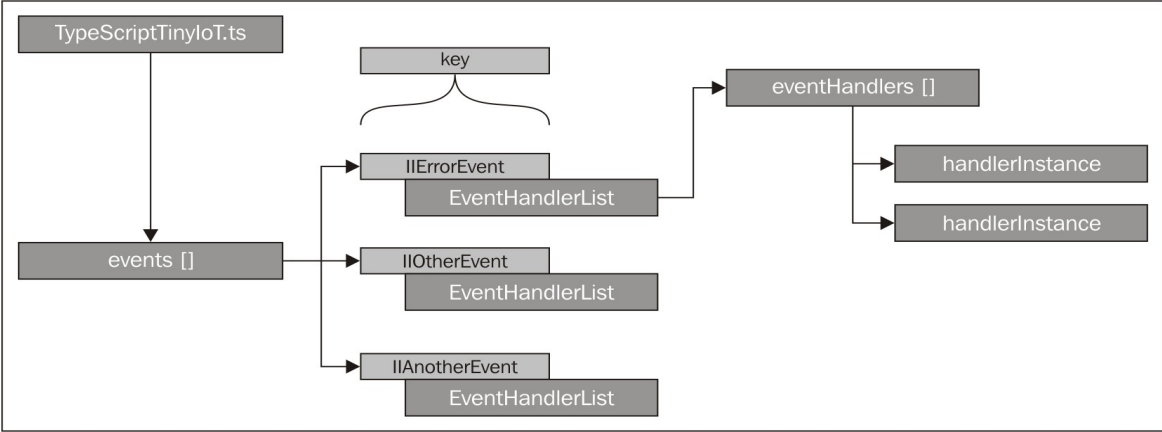
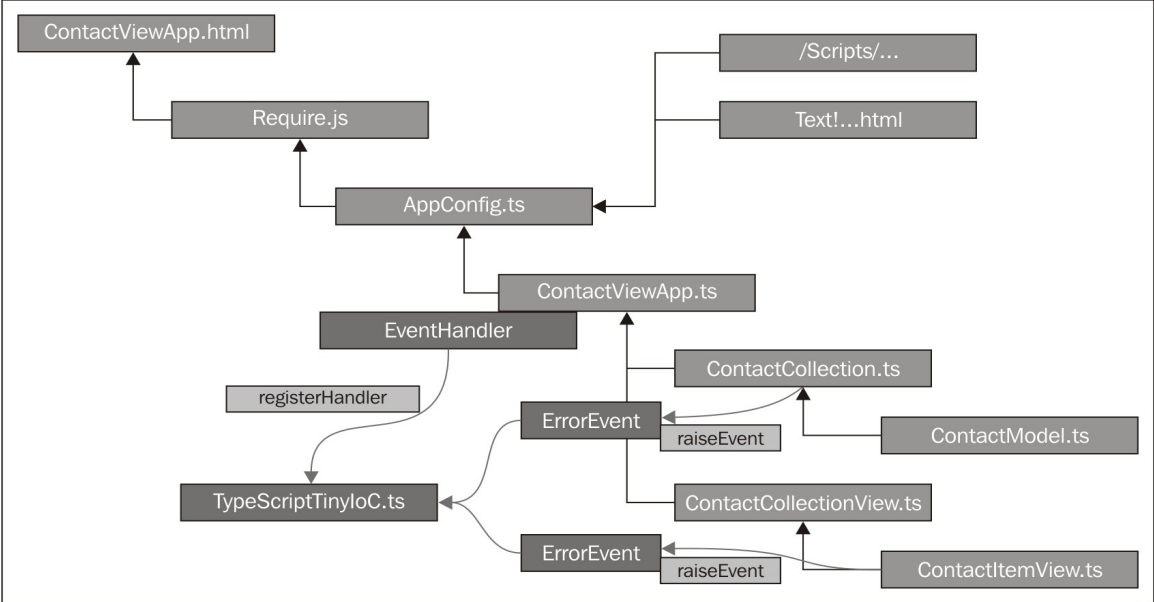
30 specs, 3 failures

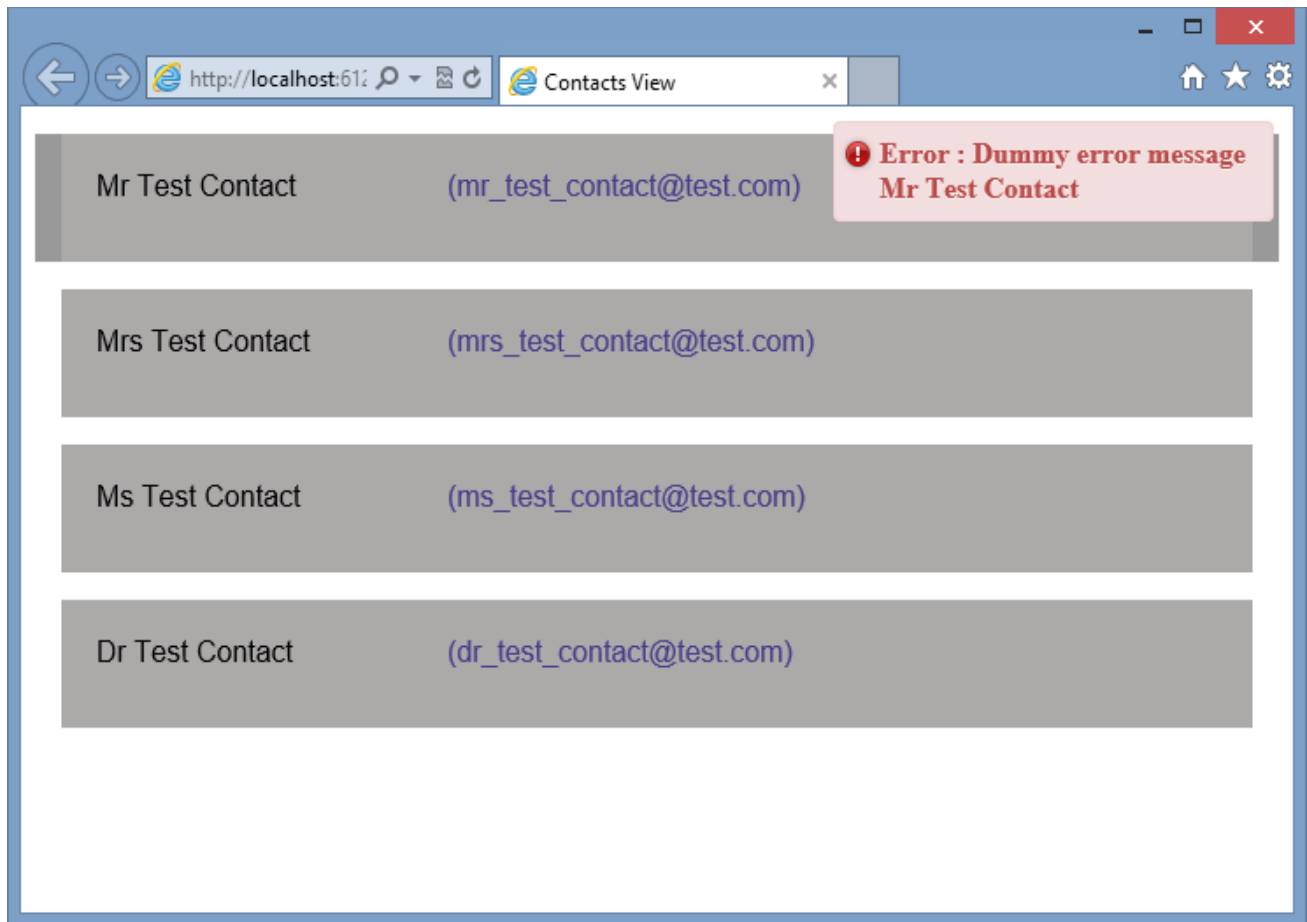
Spec List | Failures

/tscode/tests/views/ContactItemViewTests should generate html from template and model

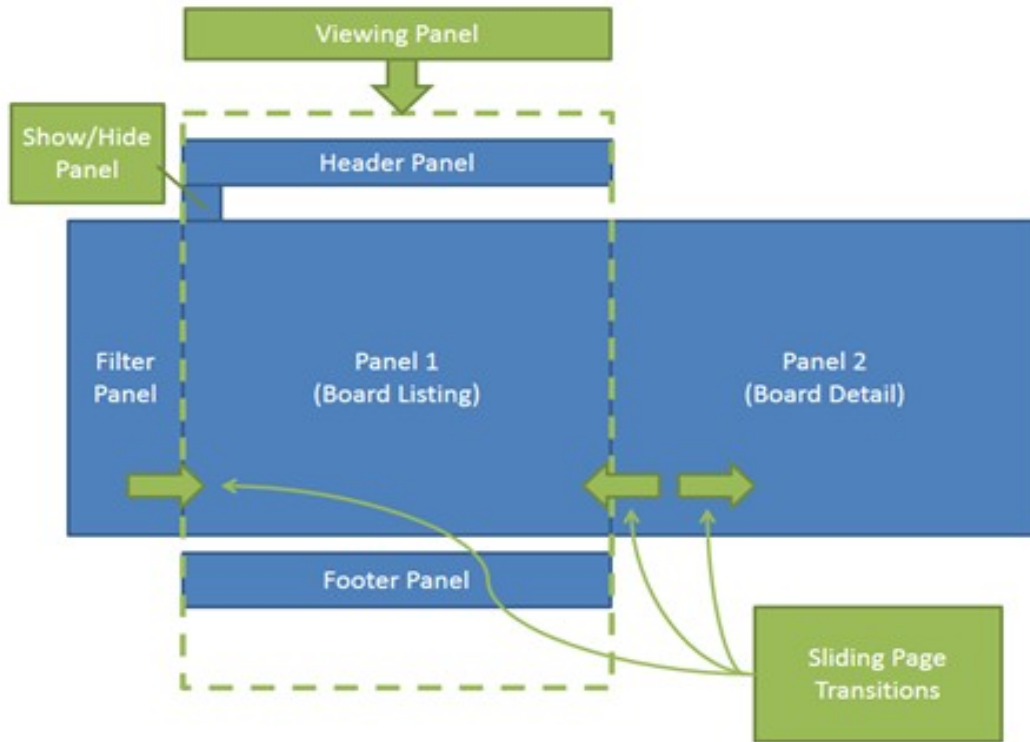
ReferenceError: 'CONTACT_ITEM_SNIPPET' is undefined

```
ReferenceError: 'CONTACT_ITEM_SNIPPET' is undefined
  at ContactItemView (http://localhost:61250/tscode/app/views/ContactItemView.js:22:13)
  at Anonymous function (http://localhost:61250/tscode/tests/views/ContactItemViewTests.js:15:13)
  at attemptSync (http://localhost:61250/Scripts/jasmine/jasmine.js:1760:9)
  at run (http://localhost:61250/Scripts/jasmine/jasmine.js:1748:9)
  at execute (http://localhost:61250/Scripts/jasmine/jasmine.js:1734:5)
  at queueRunnerFactory (http://localhost:61250/Scripts/jasmine/jasmine.js:569:7)
  at execute (http://localhost:61250/Scripts/jasmine/jasmine.js:318:5)
  at fn (http://localhost:61250/Scripts/jasmine/jasmine.js:2073:37)
  at attemptAsync (http://localhost:61250/Scripts/jasmine/jasmine.js:1790:9)
  at run (http://localhost:61250/Scripts/jasmine/jasmine.js:1746:9)
```





Chapter 9: Let's Get Our Hands Dirty



The screenshot displays a web browser window with the URL `http://localhost:5000/BoardSales`. The application layout includes:

- Navbar:** A dark navigation bar with links for 'Board Sales', 'Home', and two 'Link' items.
- Filter:** An accordion menu on the left with options: 'Manufacturer', 'Board Type', and 'All'.
- Board List:** A carousel panel showing three board listings:
 - Radical Quad:** Radical Wave Board. Sizes: 68 L, 74 L, 82 L.
 - Thruster Quad:** Allround Wave Board. Sizes: 73 L, 83 L, 93 L.
 - Freestyle Wave:** Wave Board | Small Allround. Sizes: 77 L, 85 L, 93 L.
- Footer:** A light gray area at the bottom of the page.

Annotations with arrows point to the following elements:

- Show hide panel button:** A small arrow icon in the top left of the board list area.
- Navbar:** The dark navigation bar at the top.
- Accordion:** The filter menu on the left.
- Carousel Panel 1:** The main content area displaying the board listings.
- Footer:** The bottom section of the page.

back button


Navbar

Board Sales Home Link Link

Board Detail

Hardcore Wave Ltd V5

Side Shore Wave Pro



After 4 years of riding waves all over the world with the Hardcore wave shapes V4 we have been able to test lots of different new shapes or prototype boards, used with different riding styles. Today we have been able to finalize 4 new magic shapes. The HardcoreWave V5 are the most versatile Hardcore boards ever built since their introduction in our collection. 4 Shapes with 10lt of difference in between 98 all the way down to 68 all equipped with 4 Fins slot box system for maximum drive through the top and bottom turns.

Available Sizes:

Volume	Width	Height	Sail Min	Sail Max
68 L	227	53.5	3.2	4.7
78 L	227	55.5	3.7	5.3
88 L	228	58.5	4.2	5.6

Footer

Carousel Panel 2

Footer

