

# Higher level testing

- Testing HTTP APIs
- Browser testing

# Testing HTTP APIs

# What?

- A running HTTP application
- Example - yet another TODO RESTful API written in Ratpack

# GroovyRatpackMainApplicationUnderTest

- Starts when asked for its address
- Implements `AutoCloseable`
- Starts on a random port
- Provides `TestHttpClient`

# How?

- Using an http client
  - Ratpack's TestHttpClient
  - RestAssured
  - groovy-wslite
  - HTTPBuilder
- Using Http protocol
  - Headers
  - Request and response body
  - Status codes

# TestHttpClient

- Configure request headers and body using `requestSpec()`
- Execute requests using `get()`, `post()`, `put()` and `delete()`
- Verify the returned Response
- Might be annotated with `@Delegate`

# Exercises!

# Browser testing with Geb



# Geb

- Drives browsers using Selenium WebDriver
- jQuery-like content selection
- Page Object pattern
- Spock integration via `geb.spock.GebSpec`
- Documentation at [gebish.org](http://gebish.org)

# geb.Browser

- Page navigation using `to()`
- Page verification using `at()`
- `baseUrl` property

# geb.Page

```
import geb.Page

class HomePage extends Page {
    static url = "index.html"

    static at = { title == "Yet another TODO app" }

    static content = {
        addTodoLink { $("a#add") }
    }
}
```

# geb.Module

```
import geb.Page
import geb.Module

class TableRowModule extends Module {
  static content = {
    item { $("tr", 0).text() }
    quantity { $("tr", 1).text() }
  }
}

class TablePage extends Page {
  static content = {
    items { $("table tbody tr")*.module(TableRowModule) }
  }
}
```

**Pages and modules can (should!)  
contain methods**

# Form control value accessing

```
$("#input", name: "firstName").value("Marcin")  
firstName = "Marcin"
```

```
$("#input", name: "firstName").value() == "Marcin"  
firstName == "Marcin"
```

# GebSpec -> Browser -> Page

```
class HomePageSpec extends GebSpec {  
  def "Clicking on a link"() {  
    expect:  
      browser //property of GebSpec  
  
    when:  
      to HomePage //method of Browser  
      addTodoLink.click() //content of HomePage  
  
    then:  
      at AddTodoPage  
  }  
}
```

# Strongly typed style

```
class HomePageSpec extends GebSpec {  
  def "Clicking on a link"() {  
    when:  
      def homePage = to HomePage  
      homePage.addTodoLink.click()  
  
    then:  
      at AddTodoPage  
  }  
}
```



# Exercises!