In this lecture, we will discuss…

- ✧ Rspec Matchers

Rspec Matchers

- RSpec “hangs” **to** and **not_to** methods on **all outcome of expectations**
- to()/not_to() methods take **one parameter** – a **matcher**
- Matcher examples:
  - be_true / be_false
  - eq 3
  - raise_error(SomeError)

Be_predicate – boolean

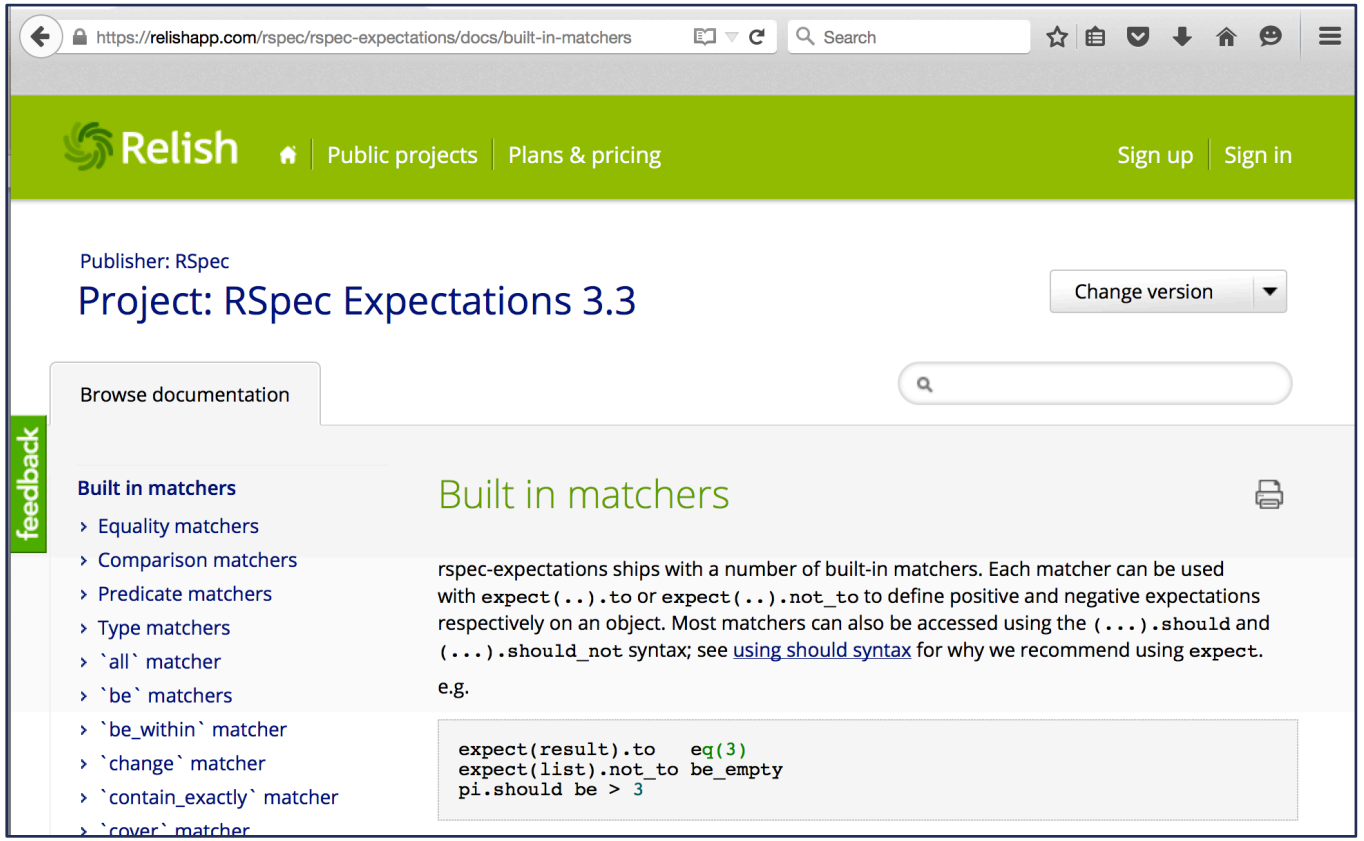
- If the **object** on which the **test is operating** has a **predicate (boolean) method** – you **automatically** get a **be_predicate** matcher
- So, for example **be_nil** is a **valid matcher** since every Ruby object has a :nil? method

```
it "should sum two odd numbers and become even" do
 expect(@calculator.add(3, 3)).to be_even
 expect(@calculator.add(3, 3)).not_to be_odd
end
```

Be_predicate – boolean

```
~/coursera/code-module2/Lecture16-RSpec$ rspec
...

Finished in 0.00169 seconds (files took 0.08464 seconds to load)
3 examples, 0 failures

~/coursera/code-module2/Lecture16-RSpec$ rspec --format documentation

Calculator
  should add 2 numbers correctly
  should subtract 2 numbers correctly
  should sum two odd numbers and become even

Finished in 0.00187 seconds (files took 0.08772 seconds to load)
3 examples, 0 failures
```

More Matchers



Summary

- RSpec has a lot of built-in matchers readily available for simplifying writing tests

What's next?

- Module 3: Introduction to Ruby on Rails