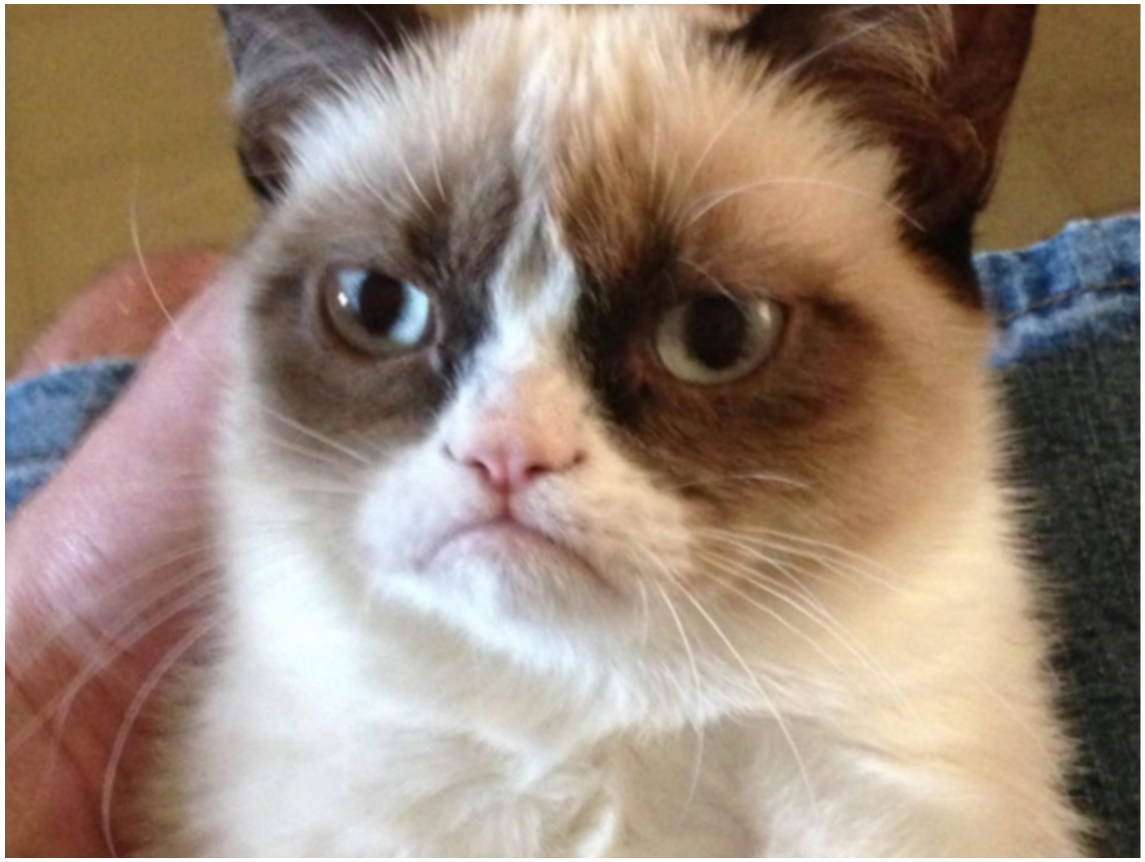
Feature Testing

With Capybara and Sinatra

TaskManager is great…

…but testing was



meh.

Let’s fix that with Capybara.

What is Capybara?

An aquatic herbivore that lives in South America.

It is either the largest or second-largest rodent in the world.

It’s also a test framework.

* Capybara was developed by Jonas Nicklas, who works in Elabs in Sweden.
* It allows you to test any rack-based app and uses Rack::Test behind the scenes.
* It uses a web driver called Selenium, but supports other drivers such as webkit. A web driver is a headless browser (no graphical interface)

Capybara is used for feature tests.

What is a feature test?

There are several types of tests.

Unit Tests

* They are used to test a particular class or method independently in your code.
* This is the test that you used when you built the Unicorn, the Centaur and the Medusa.
* This test is very common when testing the models in your application.

Integration Tests

* They are used to test how different parts of your code work together.
* It could include several methods within one or various classes.
* We used this type of test when we verified that Medusa was indeed converting different people into stone and adding them to her statue collection

Feature Tests

* used to test how a particular functionality is implemented from the perspective of the user
* In the case of web applications, this type of test mimics the behavior of the user clicking around and opening pages
* have to know little about your code implementation (user does not know about your code!)
* based on user stories.

What is a User Story?

* A user story is a tool used to communicate user needs to software developers.
* It is used in Agile Development, and it was first introduced in 1998 by proponents of Extreme Programming.
* It describes what a user needs to do in order to fulfill job function.

Example

As a user,
When I visit the home page,
And I introduce the title and description of a task,
And I click the submit button,
Then my task is saved.

As a user,
When I visit the new page,
And I introduce the title and description of a task,
And I click the submit button,
Then my task is saved.

As a [user],
When I [perform an action],
And I [perform an action],
And I [perform an action],
Then [expected result].

Workshop

1. Adding a task to TaskManager.
2. Signing up for a new account.
3. Logging into an account.
4. Viewing only the tasks associated with my account

As a [user],
When I [perform an action],
And I [perform an action],
And I [perform an action],
Then [expected result].

Hello, Capybara

Capybara helps you query and interact with the DOM

`visit path`

* Use the `visit` method to load a page
* It uses Rack::Test to ‘virtually’ load the page, not a real browser

```
assert page.has_content?
```

```
visit ‘/‘
assert page.has_content?(‘Hello, World!’)
```

`has_content?` **sucks**

```
assert page.has_css?
```

```
visit ‘/‘
assert page.has_css?(‘#hello’)
```

`has_css?` **is better**

```
within(identifier) do
  # scoped queries
end
```

Coding Time

- **Add/run tests to verify:**
  - **The front page has the CSS "#greeting" and the content 'Welcome to the TaskManager!' is within that CSS. You'll need to implement the CSS.**

Interactions

```
fill_in(identifier,
        with: “content”)
```

```
click_link(identifier)
click_button(identifier)
click_link_or_button(identifier)
```

```
assert_equal '/', current_path
```

save_and_open_page

Workshop

- **Write feature tests to exercise:**
  - **The process of creating a task**
  - **The process of editing a task**
  - **The process of deleting a task**

Documentation:
https://github.com/jnicklas/capybara

Recap

Old Slides

- Clone the App: https://github.com/rwarbelow/1410-example-ideabox/
- Switch to the branch test-with-capybara
  (git checkout -b capybara origin/test-with-capybara)
- Run `bundle` and `shotgun`
- Open http://localhost:9393 in your browser
- Look at test/features/feature_test_helper.rb
- Add a test/features/front_page_test.rb