

[Courseware \(/courses/BerkeleyX/CS169.1x/2013\\_Spring/courseware\)](/courses/BerkeleyX/CS169.1x/2013_Spring/courseware)[Course Info \(/courses/BerkeleyX/CS169.1x/2013\\_Spring/info\)](/courses/BerkeleyX/CS169.1x/2013_Spring/info)[Syllabus \(/courses/BerkeleyX/CS169.1x/2013\\_Spring/syllabus/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/syllabus/)[Textbook & VM \(/courses/BerkeleyX/CS169.1x/2013\\_Spring/textbook\\_vm/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/textbook_vm/)[Tutorials & Resources \(/courses/BerkeleyX/CS169.1x/2013\\_Spring/tutorials\\_resources/\)](/courses/BerkeleyX/CS169.1x/2013_Spring/tutorials_resources/)[Discussion \(/courses/BerkeleyX/CS169.1x/2013\\_Spring/discussion/forum\)](/courses/BerkeleyX/CS169.1x/2013_Spring/discussion/forum)[Wiki \(/courses/BerkeleyX/CS169.1x/2013\\_Spring/course\\_wiki\)](/courses/BerkeleyX/CS169.1x/2013_Spring/course_wiki)[Progress \(/courses/BerkeleyX/CS169.1x/2013\\_Spring/progress\)](/courses/BerkeleyX/CS169.1x/2013_Spring/progress)

## ADDING DIRECTOR INFO TO ROTTENPOTATOES

Your new feature will be a "search for movies by director" feature. It will provide a link on each movie-details pages that, when clicked, displays a list of movies by the director on the page. To create this feature, follow the steps below.

### 1. Setup Cucumber and RSpec.

- Make sure your Gemfile contains these changes from Section 4.4 of ELLS (<http://pastebin.com/HnHbnaZD>), as well as the line

```
gem 'rspec-rails'
```

within the group `:test, :development` block.

- Run `bundle install --without production` to make sure the gems are properly installed.
- If necessary, set up the Cucumber directories (under `features/`) and the RSpec directories (under `spec/`), allowing overwrite of any existing files, by running the commands

```
rails generate cucumber:install capybara
```

```
rails generate cucumber_rails_training_wheels:install
```

```
rails generate rspec:install
```

- Double-check that everything was installed by running the tasks

```
rake spec
```

and

```
rake cucumber
```

. Since presumably you have no features or specs yet, both tasks should execute correctly and report that there are zero tests to run. Depending on your version of RSpec, it may also display a message stating that it was not able to find any `'_spec.rb'` files.

- ### 2. Create and apply a migration that adds the `director` field to the `movies` table. The `director` field should be a string containing the name of the movie's director.

*Hint:* Use the `add_column` method of `ActiveRecord::Migration` to do this. Remember that once the migration is applied, you also have to run

```
rake db:test:prepare
```

to load the new post-migration schema into the test database!

### 3. Create controller and model specs.

- We've provided three Cucumber scenarios (<http://pastebin.com/L6FYWyV7>) as a high level description of the happy path of "search for movies by director":
  - The first scenario tests the functionality for adding director info to an existing movie. Implementing this doesn't require creating any new views or controller actions, but it does require (1) modifying existing views, (2) creating a new step definition, and (3) possibly adding a line or two to `features/support/paths.rb`.
  - The second scenario tests the functionality for clicking a "Find Movies With Same Director" link on the movie-details page for a movie that has director info in the database. For this scenario, you'll have to modify the existing "Show Movie" view, and you'll have to add a route, view and controller method for "find movies with same director".
  - The third scenario handles the sad path when the current movie has no director info but we try to follow "Find Movies With Same Director" anyway.
- Going one Cucumber step at a time, use RSpec to create the appropriate controller and model specs to drive the creation of the new controller and model methods. You will need to write tests to drive the creation of
  - a RESTful route for "find similar movies" (*Hint*: Use the 'match' syntax for routes as suggested in "Non-Resource-Based Routes" at the end of section 3.9 of ELLS),
  - a controller method to receive the click on "Find Movies With Same Director", and grab the id (for example) of the current movie (i.e. the one we're trying to find movies similar to), and
  - a model method in the `Movie` model to find movies whose director matches that of the current movie.
- It's up to you to decide whether you want to handle the sad path of "no director" in the controller method or in the model method, but you must provide a test for whichever one you do. Remember to include the line

```
require 'spec_helper'
```

at the top of your `_spec.rb` files.

Once you have finished creating your feature, use SimpleCov to generate a report of your code coverage. To do this, first make sure that the line

```
gem 'simplecov'
```

exists in your Gemfile. Add the following code:

```
require 'simplecov'  
SimpleCov.start 'rails'
```

to the **top** of "`spec/spec_helper.rb`" and "`features/support/env.rb`". Now when you run

```
rake spec
```

or

```
rake cucumber
```

SimpleCov will generate a report in a directory named "`coverage/`". Since both RSpec and Cucumber are so widely used, SimpleCov can intelligently merge the results, so running the tests for RSpec will not overwrite the coverage results from SimpleCov and vice versa. See the screencast (<http://vimeo.com/34754907>) for step-by-step instructions on setting up SimpleCov.

Show Discussion

New Post



---

[Find Courses \(/courses\)](/courses)   [About \(/about\)](/about)   [Blog \(http://blog.edx.org/\)](http://blog.edx.org/)   [Jobs \(/jobs\)](/jobs)   [Contact \(/contact\)](/contact)



(<http://youtube.com/user/edxonline>)



(<https://plus.google.com/108235383044095082735>)



(<http://www.facebook.com/EdxOnline>)



(<https://twitter.com/edXOnline>)