

Time	~ 4 - 5 hours
Learning Goals	<ul style="list-style-type: none"> • Fix a Rails application so that the existing tests pass • Implement pending tests

Objectives

In this challenge, git fork from https://github.com/NextAcademy/url_shortener and clone the repo to your machine. Remember to read the README.md file. Do the following:

1. Fix the Rails application by writing relevant code at the right places so that the existing tests pass.
2. Begin with model unit tests, and then the controller unit tests. Implement the tests marked as pending.
3. Create three view files with the following (note that the setup is slightly different from what you have done in Week 4):
 - **index.html.erb**: build a table that will display the long URL and shortened URL. Click on a link that will render a form.
 - **new.html.erb**: build a form that takes in a long URL and shortens it. The shortened URL will then be displayed underneath the form.
 - **show.html.erb**: from the index page, when you click on a long URL, the user will be brought to another page that will display that specific long URL and its corresponding short URL.

After creating the necessary views, write and perform feature tests using [Capybara](#). Capybara is a testing framework that allows one to perform tests on a web app. It simulates user interactions on your app, for example, filling in a form, navigating the app, clicking a button, etc. [This RailsCast](#) will help you get started. You will also find a comprehensive documentation for Capybara [here](#).

Rules

You are not allowed to modify existing tests, except when removing **pending** to complete the implementation.

Extra Reading

- [Thoughtbot: Write Reliable, Asynchronous Integration Tests with Capybara](#)
- [Rails Cast \(Episode275\): How I Test](#)

- [Difference between view specs and feature specs](#)ô