

Problem:

Consider these code fragments.

- a. `end`
- b. `get :index`
- c. `assert_redirected_to car_path(assigns(:car))`
- d. `assert_template :index`
- e. `assert_template :new`
- f. `assert_not_nil assigns(:cars)`
- g. `get :new`
- h. `test "should get index" do`
- i. `post :create, car:{make:@car.make, model:@car.model, year: @car.year}`
- j. `assert_response :success`

Using the above fragments, create a functional test for the “index” page of a car-themed web app. The test should make sure (1) that the HTTP response does not report an error, (2) that the correct ERB is rendered (index.html.erb), and (3) that the call to `Car.all` in the controller, which sets the `@cars` instance variable, does not fail and return nil. Note that your answer should use only 6 of the above fragments.

Solution:

h

b

j

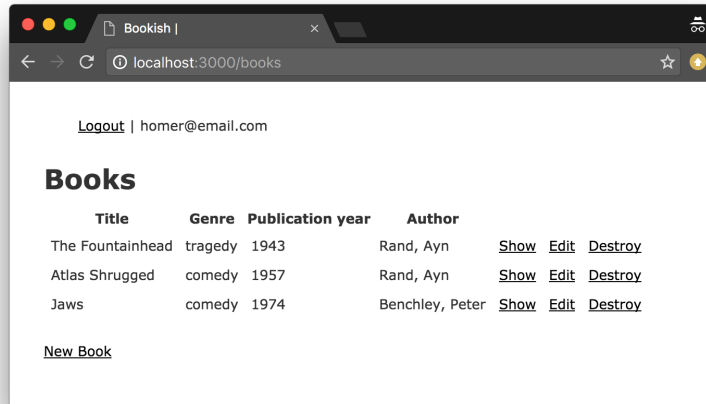
d

F

a

} order may vary

Problem:



Using the fragments below, create a functional test (class and method) for the “index” page of a book-themed web app (illustrated in the above figure). The test should do the following in this order (1) retrieve user fixture “one” and sign in the user, (2) simulate an HTTP request for the index page, (3) check that the HTTP response does not report an error, (4) check that the correct ERB is rendered (index.html.erb), and (5) check that the rendered HTML table contains a cell with everyone’s favorite shark book. Some fragments may be used more than once in your solution. Some fragments may not be used at all.

- a) `user = users(:one)`
- b) `include Devise::Test::IntegrationHelpers` _____
- c) `assert_response :error`
- d) `assert_response :success` _____
- e) `assert_select "h1", "Book"`
- f) `assert_select "td", "Jaws"` _____
- g) `sign_in user`
- h) `class BooksControllerTest < ActionDispatch::IntegrationTest` _____
- i) `assert_template :index`
- j) `assert_template :books` _____
- k) `test "should display books" do` _____
- l) `get books_url` _____
- m) `end` _____

Solution:

h

b

k

a

g

l

d

i

f

m

m

Problem:

Using the fragments below, create a functional test (class and method) for the “index” page of a music-themed web app (which has the typical scaffold layout). The test should do the following in this order (1) retrieve user fixture “two” and sign in the user, (2) simulate an HTTP request for the index page, (3) check that the HTTP response does not report an error, (4) check that the rendered HTML table contains a cell with the famous singer of “Rolling in the Deep” and “Hello”, and (5) check that the correct ERB is rendered (index.html.erb). Some fragments may be used more than once in your solution. Some fragments may not be used at all.

- a) end
- b) include Devise::Test::IntegrationHelpers _____
- c) get albums_url _____
- d) get album_url(@album) _____
- e) assert_select "h1", "Album" _____
- f) assert_select "td", "Adele" _____
- g) assert_template :albums _____
- h) assert_template :index _____
- i) assert_response :error _____
- j) assert_response :success _____
- k) sign_in user _____
- l) album = albums(:two) _____
- m) user = users(:two) _____
- n) test "should display albums" do _____
- o) class AlbumsControllerTest < ActionControllerTest
patch::IntegrationTest _____
- p) class Album < ApplicationRecord _____

Solution:

o

b

n

m

k

c

j

f

h

a

a