

HELLO!

CAPYBARA = GREAT

Allow work to happen in the background

Get a callback when it's done

**TRANSACTIONS FOR
TEST TEARDOWN =
GREAT**

insert fixture data

START TRANSACTION

- before hooks
- run your test
- after hooks

ROLLBACK

**DATABASE
TRANSACTIONS ARE
FAST**

**CAPYBARA +
TRANSACTIONS
= NOT SO GREAT**

THREADS

Thread #1
(test driver + browser)

insert fixture data

START TRANSACTION

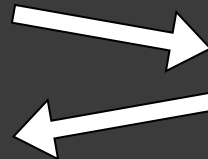
test runs

ROLLBACK

Thread #2
(Rails server)

starts Rails server

DB queries



SHARE A DATABASE CONNECTION BETWEEN THREADS

```
class ActiveRecord::Base
  attr_accessor :shared_connection
  @@shared_connection = nil

  def self.connection
    @@shared_connection || retrieve_connection
  end
end
ActiveRecord::Base.shared_connection =
  ActiveRecord::Base.connection
```

GOOD START, BUT...

```
Mysql2::Error: This connection is  
in use by: #<Thread:0x000000bb400b8>
```

**WHY EVERYTHING IS
BROKEN**

Thread #1
(test driver + browser)

insert fixture data

START TRANSACTION

test runs

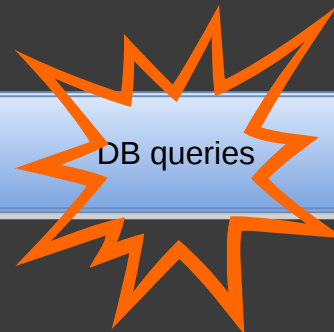
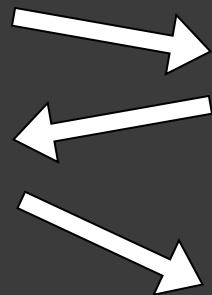
ROLLBACK

Thread #2
(Rails server)

starts Rails server

DB queries

DB queries



TEMPORARY SOLUTION

```
after(:each) do  
  sleep 2  
end
```

BETTER SOLUTION

Wait for pending AJAX calls before finishing the test

BETTER SOLUTION

```
def wait_for_ajax
  Timeout.timeout(Capybara.default_wait_time) do
    loop until finished_all_ajax_requests?
  end
end

def finished_all_ajax_requests?
  page.evaluate_script('jQuery.active').zero?
end
```

PROBLEMS

1. Post-jQuery world
2. JavaScript is terrible

BEST SOLUTION

Don't build it yourself

github.com/iangreenleaf/transactional_capybara

INSTALL IT

```
group :test do  
  gem 'transactional_capybara'  
end
```

WITH RSPEC

```
# rails_helper.rb  
require "transactional_capybara/rspec"
```

That's it.

WITH OTHER TEST FRAMEWORKS

```
# test_helper.rb
TransactionalCapybara.share_connection

after :each do
  TransactionalCapybara::AjaxHelpers.wait_for_ajax(page)
end
```

PATCHES WELCOME!

Currently supports:

- ActiveRecord
- jQuery
- Angular

**DATABASE
TRANSACTIONS ♡
CAPYBARA**

THANKS!

SAY HI

@iangreenleaf

FOLLOW MY BLOG

technotes.iangreenleaf.com

HIRE ME

ian@iangreenleaf.com