RSpec Examples (with Factory Girl) Cheat Sheet

by dwapi via cheatography.com/43665/cs/13097/

Controllers

```
require 'rails_helper'
describe Articl esC ont roller do
    bef ore (:each) do
        @ar ticles = Articl e.c reate([
                {
                    title: Faker: :Lo rem.se nte -
nce (ra nd( 3)+2),
                    body: Faker: :Lo rem.pa rag -
rap h(r and (30 )+30),
                }
            ])
   end
  des cribe 'GET search' do

       it 'renders search' do
           get :search
            exp ect (re spo nse.st atu s).to
eq(200)
            exp ect (re spo nse).to render _te -
mpl ate (:i ndex)
       end
       it 'assigns instance variables' do
           get :search, :params => {:query =>
@artic les.fi rst.title}

            exp ect (re spo nse.st atu s).to
eq(200)
            exp ect (as sig ns( :qu ery )).to
eq(@ar tic les.fi rst.title)
            exp ect (as sig ns( :sh ow_ sea rch -
_re sul ts)).to be(true)
       end
       it 'returns search results' do
            sea rch _result = Articl e.c reate({
                title: "Find Me",
                body: "One Result ",
                pub lished: true,
                aut hor_id: @author.id
           })
           get :search, :params => {:query => 'One
Result'}
            exp ect (re spo nse.st atu s).to
eq(200)
```

Controllers (cont)

```
>      expect(assigns(:articles).length).to eq(1)
     expect(assigns(:articles).first.title).to eq("Find Me")
     #expect(response).to redirect_to(location)
   end
 end
end
```

Models

```
RSpec.describe MyModel, type: :model do

    let (:s tring) { "foo bar bazz" }
   it 'checks something' do

            exp ect (my _mo del).to be_valid
            expect  (actua l).to eq(exp ected)
            exp ect (ac tua l).to be_nil
            exp ect (ac tua l).to be true
            exp ect (ac tua l).to be > expected
            exp ect (ac tua l).to be >= expected
            exp ect (ac tua l).to be <= expected
            exp ect (ac tua l).to be < expected
            exp ect (ac tua l).to be_bet wee n(min,
max).i ncl usive
            exp ect (ac tua l).to be_bet wee n(min,
max).e xcl usive
            exp ect (ac tua l).to match( /ex pre -
ssion/)
            exp ect (ac tua l).to be_wit hin (de -
lta ).of( expected)
            exp ect (ac tua l).to start_with
expected
            exp ect (ac tua l).to end_with expected
            exp ect (ac tua l).to start_ wit h("f -
oo").and end_wi th( " baz z")
            exp ect (re sul t).n ot_to be_empty
            exp ect (ac tua l).to be_ins tan ce_of(
expected)
            exp ect (ac tua l).to be_kin d_of(
expected)
            exp ect (ac tua l).to respon d_to(
expected)
            exp ect  (a r r).to be_an(  Array)
            expect { ... }.to raise_ error
```


Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Models (cont)

```
>      expect { ... }.to raise_error(ErrorClass)
      expect { ... }.to raise_error("message")
      expect { ... }.to raise_error(ErrorClass, "message")
      expect { ... }.to throw_symbol
      expect { ... }.to throw_symbol(:symbol)
      expect { ... }.to throw_symbol(:symbol, 'value')

  end

end
```

Using not_to instead of to reverses the condition.

Factory Girl

```
#Defining factories
Factor yGi rl.d efine do
    factory :user do
        fir st_name 'John'
        las t_name 'Doe'
        bir thdate { 21.yea rs.ago }
        admin false
        seq uen ce( :us ername) { |n| " use r#{ n}"
}
   end
end
# Using
Factor yGi rl.b ui ld( :user, name: 'John')   # not
saved
create (:user)     # saved
attrib ute s_f or( :user)     # hash
```


Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com