

Better late than never

Integrating Selenium after the fact

R. Tyler Croy
tyler@linux.com



I work here

```
ast.rb ~/.viarc
self.fail(Puppet::ParseError, message)
end

Wrap a state in a reusable way so we always throw a parse error.
def parsewrap
  exceptwrap :type => Puppet::ParseError do
    yield
  end
end
```

The version of the evaluate method that should be called, because it correctly handles errors. It is critical to use this method because it can enable you to catch the error where it happens, rather than much higher up the stack.

```
def safeevaluate(*options)
  * We duplicate the evaluate method here because self.evaluate
  * is called so often that it's worth duplicating.
begin
  return self.evaluate(*options)
rescue Puppet::Error => detail
  raise adderrorcontext(detail)
rescue => detail
  error = Puppet::ParseError.new(detail.to_s, nil, nil, detail)
  * We can't use self.fail here because it always expects strings,
  * not exceptions.
  raise adderrorcontext(error, detail)
end
end
```

Initialize the object. Requires a hash as the argument, and takes each of the parameters of the hash and calls the setter method for them. This is probably pretty inefficient and should

Hacking with Ruby



I am not in QA



(so why do I care?)



Quality is important



A cartoon illustration of a white dog with large, expressive eyes and a pink nose. The dog is wearing a red suit jacket with a dark brown bow tie. It is holding a large, ornate gold pocket watch in its right hand. The watch has a large diamond on the top and a thick gold chain. The dog's mouth is wide open in a surprised or excited expression. The background is a simple greyish-blue sky with some green grass at the bottom.

Challenges

Browser test suite

Browser testing infrastructure



Unit Test-Driven Team



Lack of experience



Directionless





BAD



IDEA



GOLD

IDEA



BAD



IDEA



GOLD

IDEA



BAD



IDEA





BAD

IDEA





GOLD

IDEA

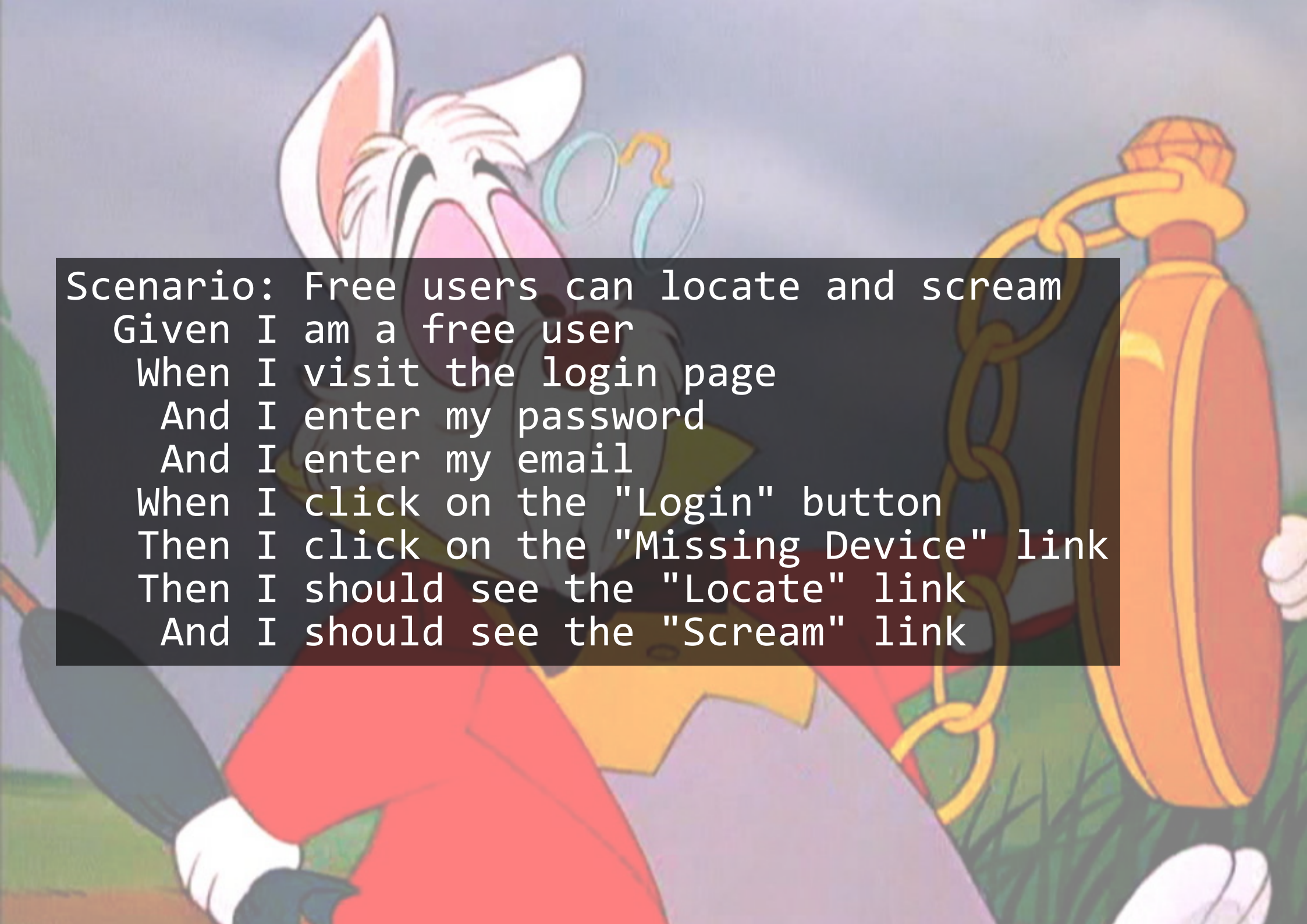


BAD



IDEA






Scenario: Free users can locate and scream
Given I am a free user
When I visit the login page
And I enter my password
And I enter my email
When I click on the "Login" button
Then I click on the "Missing Device" link
Then I should see the "Locate" link
And I should see the "Scream" link

GOLD

IDEA





Scenario: Free users should have locate/scream capability

Users that are either in trial or in the free account state should always be able to locate and scream from the non-mobile version of the web site

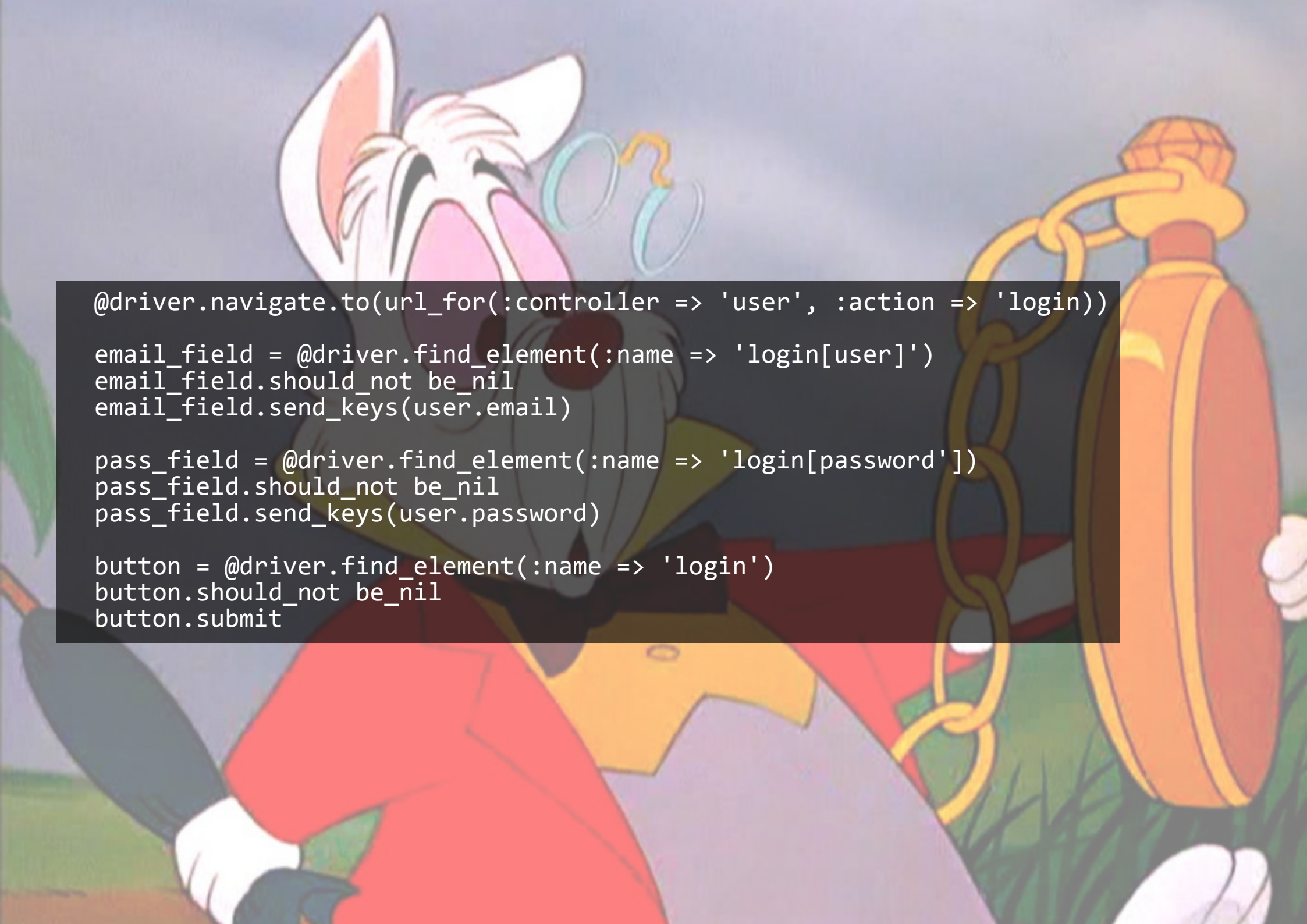
Given I am a logged in free user
When I visit the "Missing Device" page
Then I should be able to locate my phone
And I should be able to scream my phone

BAD



IDEA





```
@driver.navigate.to(url_for(:controller => 'user', :action => 'login'))

email_field = @driver.find_element(:name => 'login[user]')
email_field.should_not be_nil
email_field.send_keys(user.email)


pass_field = @driver.find_element(:name => 'login[password]')
pass_field.should_not be_nil
pass_field.send_keys(user.password)

button = @driver.find_element(:name => 'login')
button.should_not be_nil
button.submit
```

GOLD

IDEA



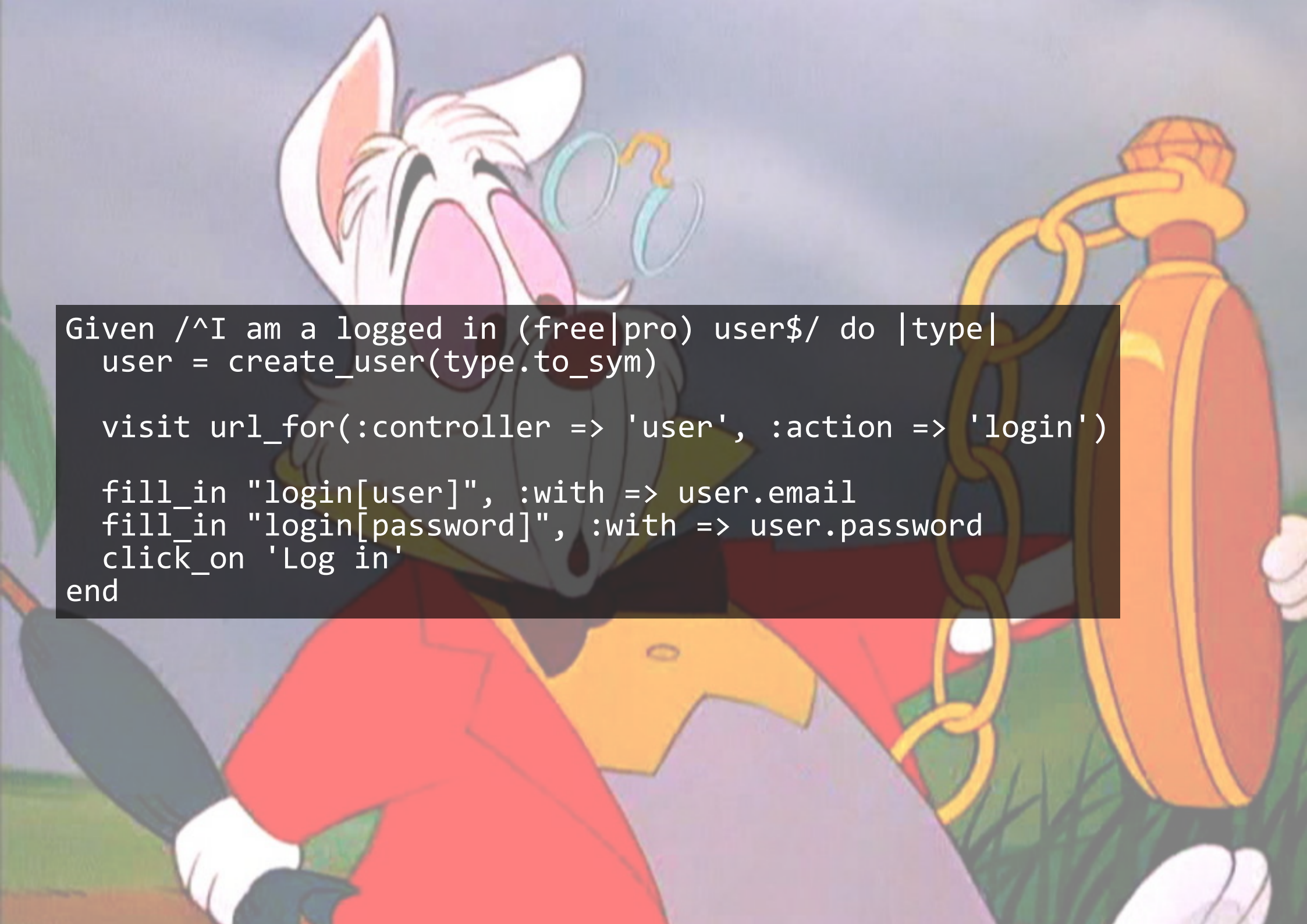


```
visit_url_for(:controller => 'user', :action => 'login')  
  
fill_in "login[user]", :with => user.email  
fill_in "login[password]", :with => user.password  
click_on 'Log in'
```

GOLD

IDEA




A cartoon illustration of a grey rabbit wearing a red suit and a yellow tie, looking up with its mouth open. To the right, there is a large, ornate treasure chest with a yellow handle and a red body. A yellow chain is attached to the chest. The background is a light blue sky with green grass at the bottom.

```
Given /^I am a logged in (free|pro) user$/ do |type|
  user = create_user(type.to_sym)

  visit url_for(:controller => 'user', :action => 'login')

  fill_in "login[user]", :with => user.email
  fill_in "login[password]", :with => user.password
  click_on 'Log in'
end
```


```
Given /^I am a logged in (free|pro) user$/ do |type|  
  user = create_user(type.to_sym)  
  
  login_with(user)  
end
```

BAD



IDEA





```
Given /^I am a logged in free user$/ do
  Given %{I am a free user}
  Given %{I am logged in to the site}
end
```

BAD



IDEA



GOLD

IDEA



BAD



IDEA



GOLD

IDEA



BAD



IDEA



GOLD

IDEA



BAD



IDEA



GOLD

IDEA





GOLD

IDEA







Questions?



Thanks