

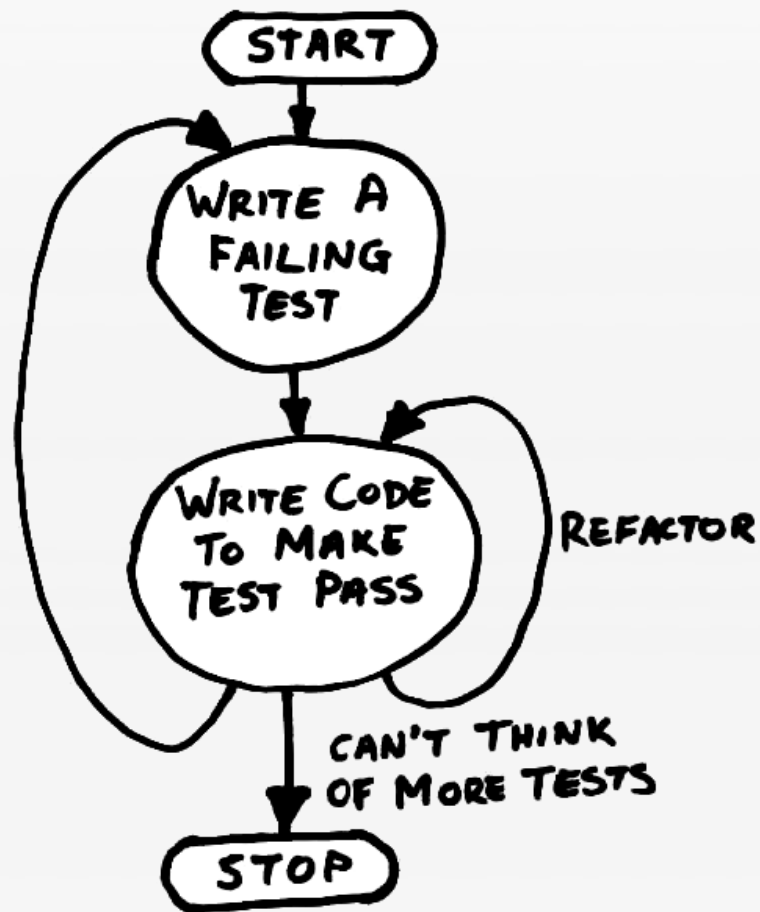


# **Vorlesung 8**

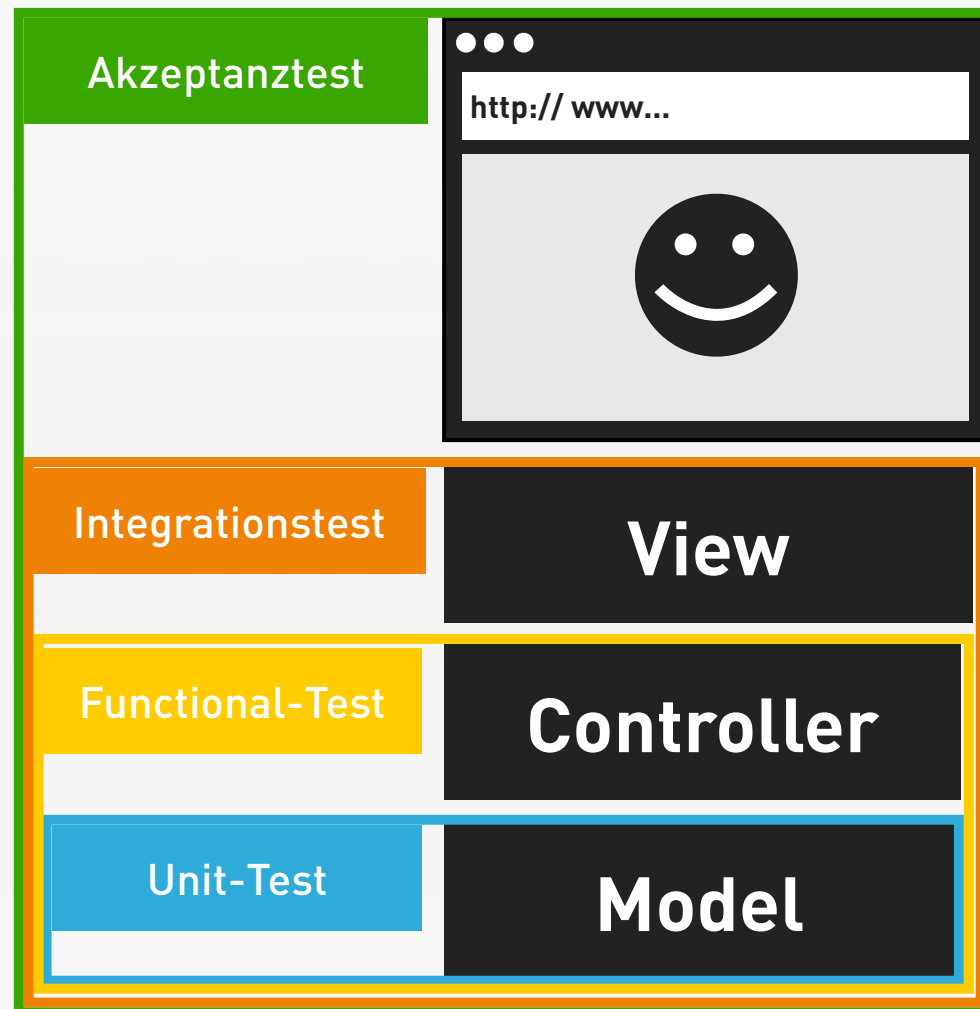
# Warum TDD?

- Wasserfallmodell: Testphase erst nach Umsetzungsphase
- Probleme:
  - Testabdeckung oft mangelhaft
  - Entwickelte Komponenten sind schwierig zu testen
  - Entwicklung neuer Features führt oft zu Regressionsfehlern
- Testgetriebene Entwicklung: Gemeinsame Entwicklung von Tests und Features

# TDD-Zyklus



# Test einer RoR-Anwendung



# RSpec - Expectations

```
require 'spec_helper'

class Calculator
  def plus(n1, n2)
    n1 + n2
  end
end

describe Calculator do
  it 'calculates sums' do
    calc = Calculator.new
    expect(calc.plus(19, 23)).to eq 42
  end
end
```

# RSpec - Mocks

```
require 'spec_helper'

class Calculator
  def plus(n1, n2)
    n1 + n2
  end
end

describe Calculator do
  it 'calculates sums' do
    calc = Calculator.new
    calc.stub(:plus).and_return(42)
    expect(calc.plus(19, 23)).to eq 42
  end
end
```

# User Story 2

- Test anlegen

```
spec/models/tour_spec.rb

require 'rails_helper'

describe Tour do
  it 'needs a category' do
    category = Category.new name: 'Radwandern'
    tour = Tour.new title: 'Test Tour', teaser: 'My Teaser', description: 'My
description', url: 'http://www.google.com/maps/'

    expect(tour).to_not be_valid

    tour.categories << category
    expect(tour).to be_valid
  end
end
```

# User Story 2

- Test ausführen

```
1. mjohann@MBP15: ~/projects/fhms/leeze.ms (zsh)
Category add some examples to (or delete) /Users/mjohann/projects/fhms/leeze.ms/spec/models/category_spec.rb
# Not yet implemented
# ./spec/models/category_spec.rb:4
TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/leeze.ms/spec/models/tour_category_spec.rb
# Not yet implemented
# ./spec/models/tour_category_spec.rb:4

Failures:

  1) Tour needs a category
     Failure/Error: expect(tour).to_not be_valid
       expected #<Tour id: nil, title: "Test Tour", teaser: "My Teaser", description: "My description", url: "http://www.google.com/maps/", created_at: nil, updated_at: nil> not to be valid
       # ./spec/models/tour_spec.rb:8:in `block (2 levels) in <top (required)>'

Finished in 0.34994 seconds (files took 1.34 seconds to load)
11 examples, 1 failure, 2 pending

Failed examples:

rspec ./spec/models/tour_spec.rb:4 # Tour needs a category
x mjohann@MBP15 > ~/projects/fhms/leeze.ms > 2_filter_category •
```



# User Story 2

- Code schreiben

spec/models/tour\_spec.rb

app/models/tour.rb

```
class Tour < ActiveRecord::Base
  has_many :tour_categories
  has_many :categories, through: :tour_categories
  validates :title, :teaser, :description, :url, presence: true
  validates :url, format: %r|^http(s?)://www.google.com/maps/ |
  ! validates :categories, presence: true
end
```

# User Story 2

- Test ausführen

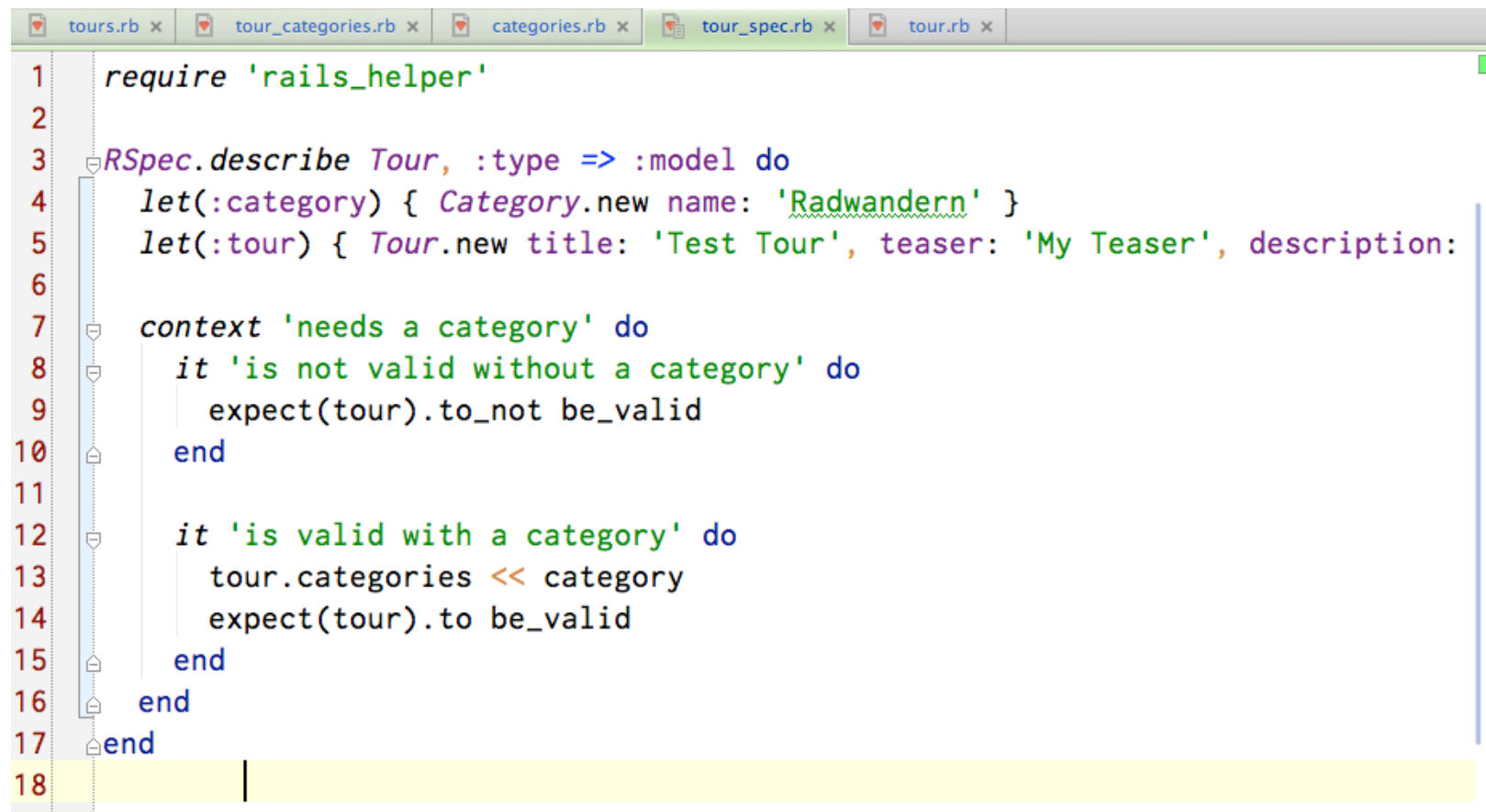
```
1. mjohann@MBP15: ~/projects/fhms/leeze.ms (zsh)
mjohann@MBP15 > ~/projects/fhms/leeze.ms 2_filter_category • bundle exec
rspec spec
**.....

Pending:
  Category add some examples to (or delete) /Users/mjohann/projects/fhms/leeze.m
s/spec/models/category_spec.rb
    # Not yet implemented
    # ./spec/models/category_spec.rb:4
  TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/lee
ze.ms/spec/models/tour_category_spec.rb
    # Not yet implemented
    # ./spec/models/tour_category_spec.rb:4

Finished in 0.27392 seconds (files took 1.27 seconds to load)
11 examples, 0 failures, 2 pending
mjohann@MBP15 > ~/projects/fhms/leeze.ms 2_filter_category •
```

# User Story 2

- Test refactor



The screenshot shows a code editor with five tabs: tours.rb, tour\_categories.rb, categories.rb, tour\_spec.rb (active), and tour.rb. The active tab displays the following RSpec test code:

```
1  require 'rails_helper'
2
3  RSpec.describe Tour, :type => :model do
4    let(:category) { Category.new name: 'Radwandern' }
5    let(:tour) { Tour.new title: 'Test Tour', teaser: 'My Teaser', description:
6
7    context 'needs a category' do
8      it 'is not valid without a category' do
9        expect(tour).to_not be_valid
10      end
11
12      it 'is valid with a category' do
13        tour.categories << category
14        expect(tour).to be_valid
15      end
16    end
17  end
18
```

# Taskboard

Inbox	In Progress	In Review	Complete
<div>Ein Anwendung soll über ein einfaches Styling verfügen</div>	<div>Ein Besucher soll Touren nach Kategorie filtern können, ...</div>	<div>Ein Besucher soll Touren anlegen können, ...</div>	

# User Story 2

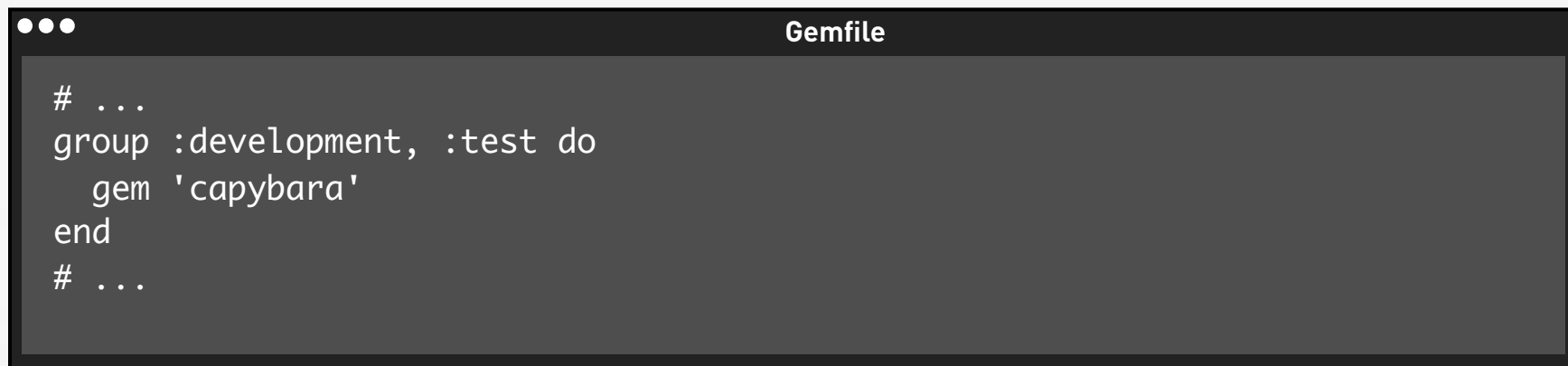
*Ein Besucher soll Touren nach Kategorie filtern können, um schneller die passende Tour zu finden.*

**Integration-Test:**

Ein Besucher soll alle Touren auf der Startseite sehen

# Voraussetzungen: Capybara

- Gem 'capybara' ins Gemfile aufnehmen:



```
Gemfile
# ...
group :development, :test do
  gem 'capybara'
end
# ...
```

\$ bundle install

# Capybara

- Capybara bietet eine domänenspezifische Sprache (DSL) zur Steuerung eines virtuellen Browsers
- RSpec startet zum Testen einen Testserver (~ rails server)
- Capybara greift auf diesen Testserver zu

# Capybara

- Steuerbefehle:

- Seite '/tours' aufrufen:  
`visit '/tours'`
- Link klicken:  
`click_link 'New tour'`
- Formular ausfüllen und abschicken:  
`fill_in 'Name', with: 'Münsterland Giro'`  
`click_button 'Create tour'`

- Expectations:

- Seite soll bestimmten Inhalt haben:  
`page.should have_content 'Created!'`



# 1. Test schreiben

```
tours_spec.rb x
1  require 'rails_helper'
2
3  describe 'Tours' do
4    context 'start page' do
5      let!(:category) { Category.new name: 'Radwandern' }
6      let!(:tour) { Tour.new title: 'Test Tour', teaser: 'My Teaser', descriptio
7
8      before(:each) do
9        tour.categories << category
10       tour.save
11     end
12
13     it 'shows all tours on the start page' do
14       visit root_path
15
16       expect(page).to have_content tour.title
17     end
18   end
19 end
20
```

# 2. Tests ausführen

```
1. mjohann@MBP15: ~/projects/fhms/leeze.ms (zsh)
# Not yet implemented
# ./spec/models/category_spec.rb:4
TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/leeze.ms/spec/models/tour_category_spec.rb
# Not yet implemented
# ./spec/models/tour_category_spec.rb:4

Failures:

  1) Tours start page shows all tours on the start page
     Failure/Error: visit root_path
     NameError:
       undefined local variable or method `root_path' for #<RSpec::ExampleGroups::Tours::StartPage:0x007f82e40ff398>
     # ./spec/features/tours_spec.rb:14:in `block (3 levels) in <top (required)>'

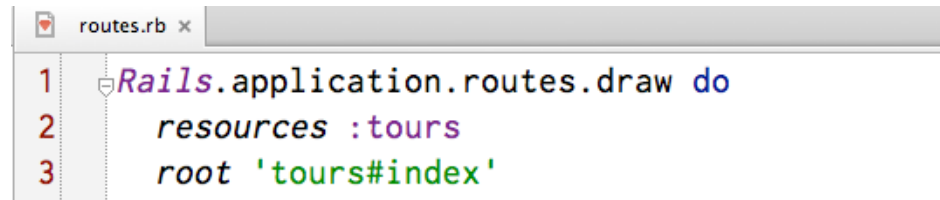
Finished in 0.27407 seconds (files took 1.4 seconds to load)
13 examples, 1 failure, 2 pending

Failed examples:

rspec ./spec/features/tours_spec.rb:13 # Tours start page shows all tours on the start page
x mjohann@MBP15 > ~/projects/fhms/leeze.ms > 2_filter_category •+
```

# 3. Code schreiben

- Root-Route eintragen:



```
routes.rb x
1  Rails.application.routes.draw do
2    resources :tours
3    root 'tours#index'
```

# 4. Test ausführen

```
1. mjohann@MBP15: ~/projects/fhms/leeze.ms (zsh)
✖ mjohann@MBP15 > ~/projects/fhms/leeze.ms > 2_filter_category •+ bundle ex
ec rspec spec
.**.....

Pending:
  Category add some examples to (or delete) /Users/mjohann/projects/fhms/leeze.m
s/spec/models/category_spec.rb
    # Not yet implemented
    # ./spec/models/category_spec.rb:4
  TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/lee
ze.ms/spec/models/tour_category_spec.rb
    # Not yet implemented
    # ./spec/models/tour_category_spec.rb:4

Finished in 0.27814 seconds (files took 1.43 seconds to load)
13 examples, 0 failures, 2 pending
mjohann@MBP15 > ~/projects/fhms/leeze.ms > 2_filter_category •+
```

# 5. Refactoring

- Wir haben zwei Tests, in denen eine Tour und eine Kategorie erstellt werden  
--> nicht DRY!
- Auch Tests sollten refactored werden!
- Einführung von Factories

# factory\_girl

- factory\_girl erlaubt das Erstellen von Factories (Fabriken) für Models
- Eine Factory erzeugt eine Instanz des Models mit sinnvollen Standardwerten
- Factories werden in spec/factories/ erstellt

# factory\_girl

## Factory definieren:

```
FactoryGirl.define do
  factory :category do
    name "My Category"
  end
end
```

## Factory nutzen:

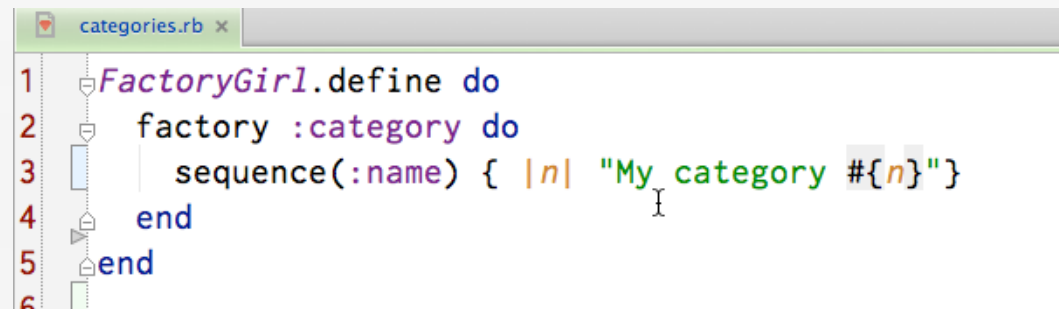
```
FactoryGirl.build(:category)
# ~> Category.new name: "My Category"
```

```
FactoryGirl.create(:category)
# ~> Category.create name: "My Category"
```

```
FactoryGirl.create(:category, name: "Some other name")
# ~> Category.new name: "Some other name"
```

# 5. Refactoring

Factories definieren:

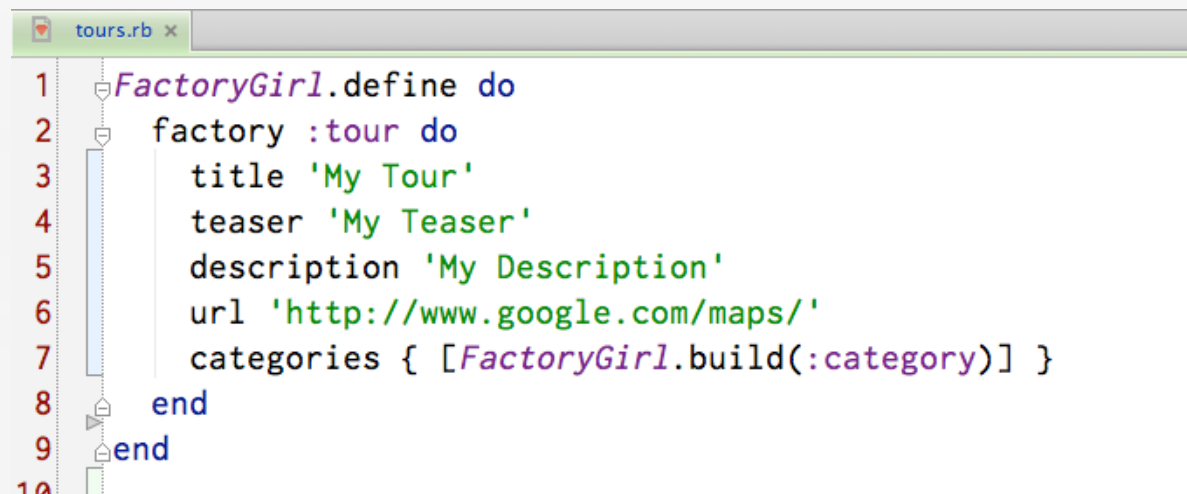


```
categories.rb x
1  FactoryGirl.define do
2    factory :category do
3      sequence(:name) { |n| "My category #{n}" }
4    end
5  end
6
```



# 5. Refactoring

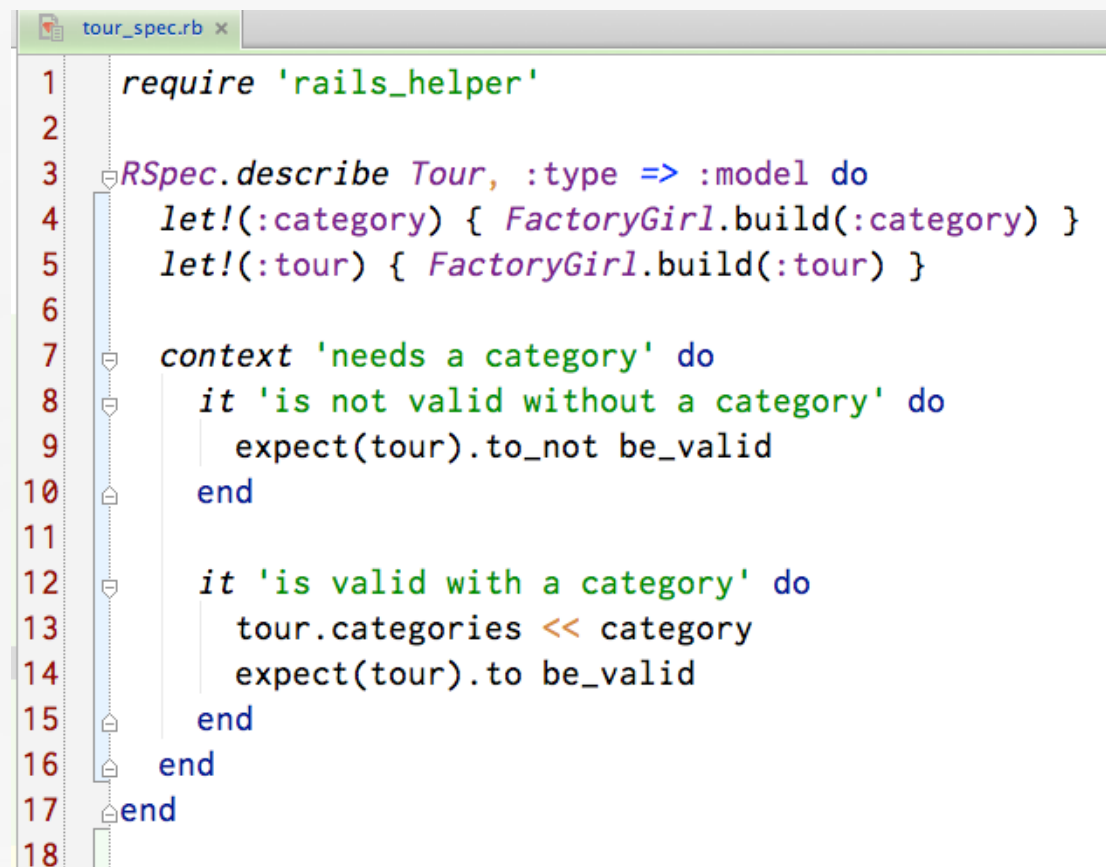
Factories definieren:



```
tours.rb x
1  FactoryGirl.define do
2    factory :tour do
3      title 'My Tour'
4      teaser 'My Teaser'
5      description 'My Description'
6      url 'http://www.google.com/maps/'
7      categories { [FactoryGirl.build(:category)] }
8    end
9  end
10
```

# 5. Refactoring

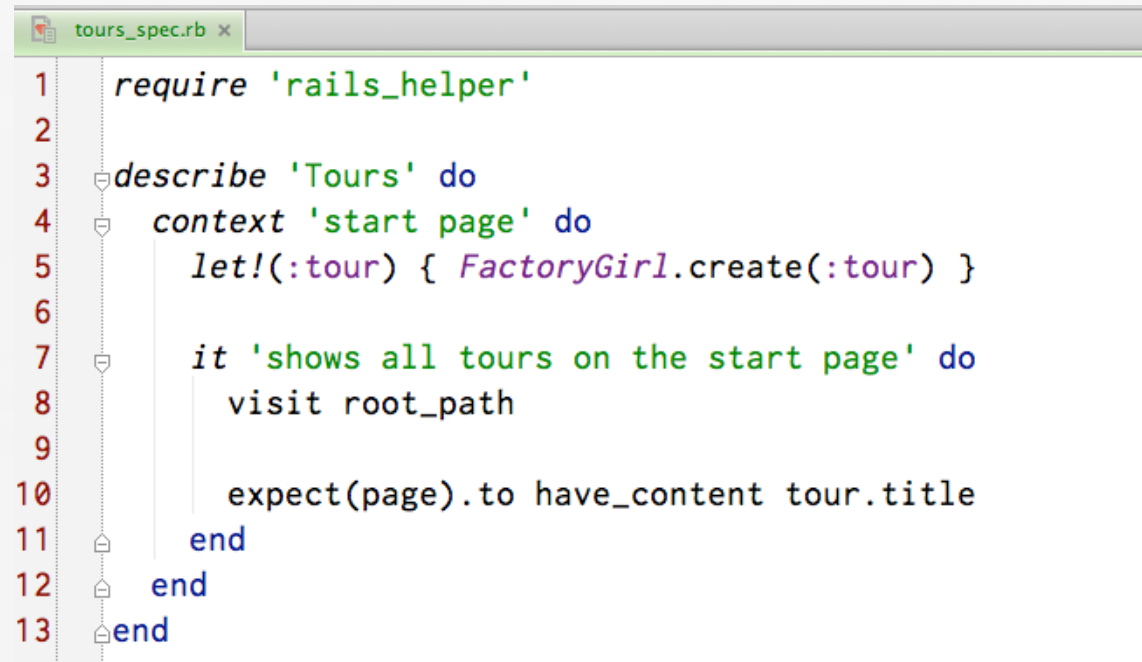
## Factories einbauen:



```
1  require 'rails_helper'
2
3  RSpec.describe Tour, :type => :model do
4    let!(:category) { FactoryGirl.build(:category) }
5    let!(:tour) { FactoryGirl.build(:tour) }
6
7    context 'needs a category' do
8      it 'is not valid without a category' do
9        expect(tour).to_not be_valid
10      end
11
12      it 'is valid with a category' do
13        tour.categories << category
14        expect(tour).to be_valid
15      end
16    end
17  end
18
```

# 5. Refactoring

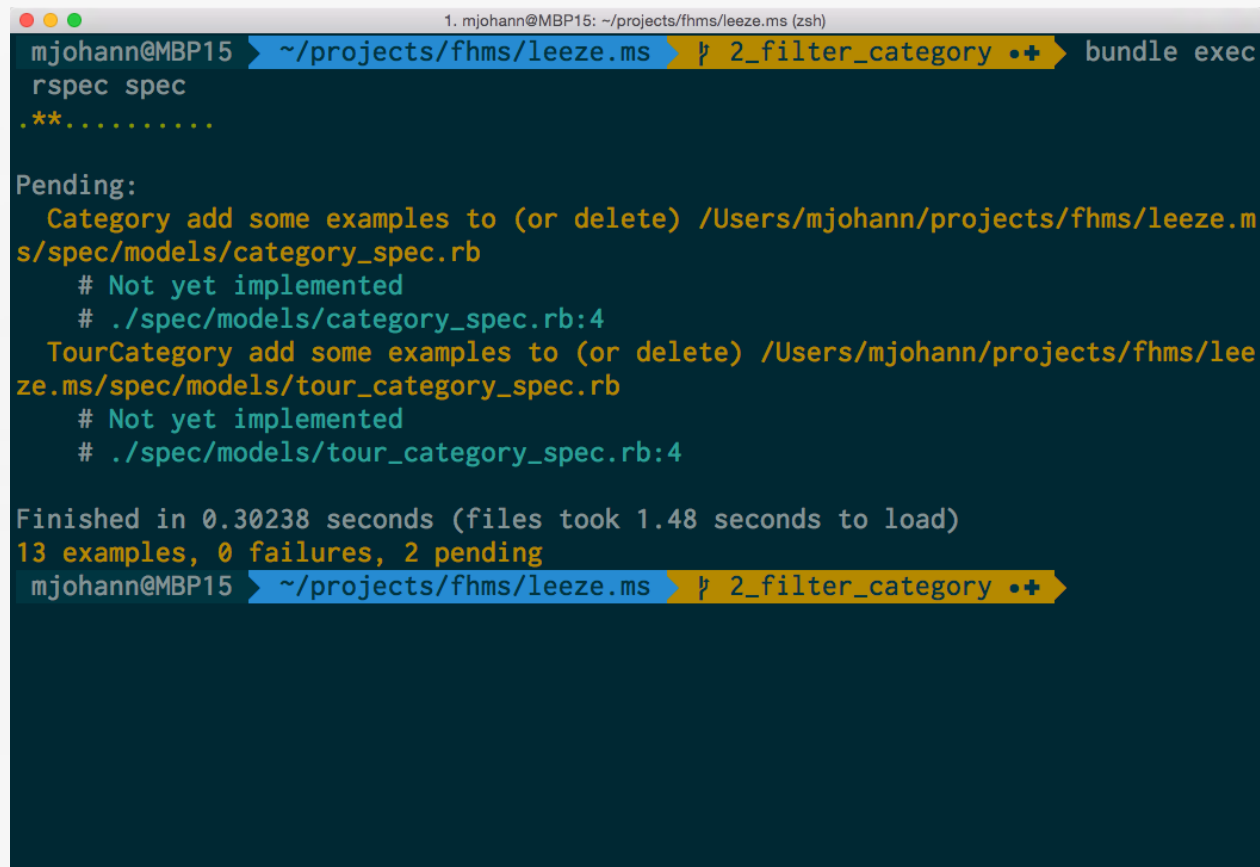
Factories einbauen:



```
tours_spec.rb x
1  require 'rails_helper'
2
3  describe 'Tours' do
4    context 'start page' do
5      let!(:tour) { FactoryGirl.create(:tour) }
6
7      it 'shows all tours on the start page' do
8        visit root_path
9
10       expect(page).to have_content tour.title
11     end
12   end
13 end
```

# 5. Refactoring

Tests ausführen:



```
1. mjohann@MBP15: ~/projects/fhms/leeze.ms (zsh)
mjohann@MBP15 > ~/projects/fhms/leeze.ms > 2_filter_category •+ bundle exec
rspec spec
.**.....

Pending:
  Category add some examples to (or delete) /Users/mjohann/projects/fhms/leeze.m
s/spec/models/category_spec.rb
    # Not yet implemented
    # ./spec/models/category_spec.rb:4
  TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/lee
ze.ms/spec/models/tour_category_spec.rb
    # Not yet implemented
    # ./spec/models/tour_category_spec.rb:4

Finished in 0.30238 seconds (files took 1.48 seconds to load)
13 examples, 0 failures, 2 pending
mjohann@MBP15 > ~/projects/fhms/leeze.ms > 2_filter_category •+
```

# User Story 2

Ein Besucher soll Touren nach Kategorie filtern können, um schneller die passende Tour zu finden.

# 1. Test schreiben

```
tours_spec.rb x
1  require 'rails_helper'
2
3  describe 'Tours' do
4    context 'start page' do ... end
13
14   context 'filter' do
15     let!(:tour1) { FactoryGirl.create(:tour) }
16     let!(:category1) { tour1.categories.first }
17     let!(:tour2) { FactoryGirl.create(:tour) }
18     let!(:category2) { tour2.categories.first }
19
20     it 'allows to select categories' do
21       visit root_path
22
23       click_link category1.name
24       expect(page).to have_content tour1.title
25       expect(page).to_not have_content tour2.title
26     end
27   end
28
29 end
30
```

# 2. Tests ausführen

```
1. mjohann@MBP15: ~/projects/fhms/leeze.ms (zsh)
.F**F.....

Pending:
  Category add some examples to (or delete) /Users/mjohann/projects/fhms/leeze.ms/spec/models/category_spec.rb
    # Not yet implemented
    # ./spec/models/category_spec.rb:4
  TourCategory add some examples to (or delete) /Users/mjohann/projects/fhms/leeze.ms/spec/models/tour_category_spec.rb
    # Not yet implemented
    # ./spec/models/tour_category_spec.rb:4

Failures:

  1) Tours filter allows to select categories
     Failure/Error: click_link category1.name
     Capybara::ElementNotFound:
       Unable to find link "My category 2"
     # ./spec/features/tours_spec.rb:23:in `block (3 levels) in <top (required)>'

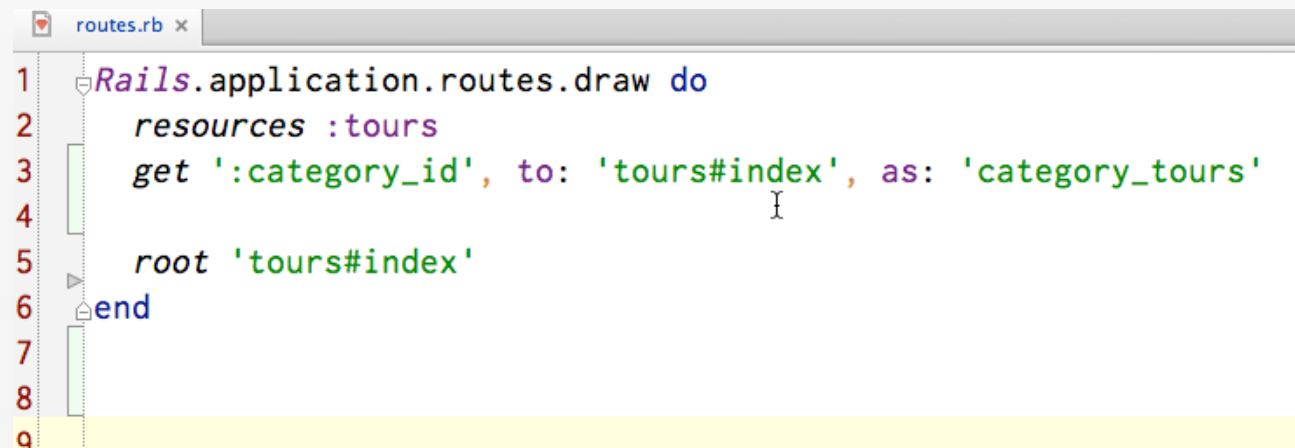
  2) Tour needs a category is not valid without a category
     Failure/Error: expect(tour).to_not be_valid
       expected #<Tour id: nil, title: "My Tour", teaser: "My Teaser", description: "My Description", url: "http://www.google.com/maps/", created_at: nil, updated_at: nil> not to be valid
     # ./spec/models/tour_spec.rb:9:in `block (3 levels) in <top (required)>'

Finished in 0.08356 seconds (files took 1.35 seconds to load)
14 examples, 2 failures, 2 pending

Failed examples:

rspec ./spec/features/tours_spec.rb:20 # Tours filter allows to select categories
rspec ./spec/models/tour_spec.rb:8 # Tour needs a category is not valid without a category
x mjohann@MBP15 ~/projects/fhms/leeze.ms 2_filter_category ↵
```

# 3. Code schreiben



```
routes.rb x
1 Rails.application.routes.draw do
2   resources :tours
3   get ':category_id', to: 'tours#index', as: 'category_tours'
4
5   root 'tours#index'
6 end
7
8
9
```



# 3. Code schreiben

```
1. mjohann@MBP15: ~/projects/fhms/leeze.ms (zsh)
mjohann@MBP15 ~/projects/fhms/leeze.ms ▶ 2_filter_category •+ ▶ bundle exec rake routes
Prefix Verb  URI Pattern          Controller#Action
  tours GET    /tours(.:format)     tours#index
      POST /tours(.:format)     tours#create
  new_tour GET    /tours/new(.:format) tours#new
  edit_tour GET    /tours/:id/edit(.:format) tours#edit
    tour GET    /tours/:id(.:format)  tours#show
      PATCH /tours/:id(.:format)  tours#update
      PUT    /tours/:id(.:format)  tours#update
      DELETE /tours/:id(.:format)  tours#destroy
category_tours GET    /:category_id(.:format) tours#index
  root GET    /                     tours#index
mjohann@MBP15 ~/projects/fhms/leeze.ms ▶ 2_filter_category •+ ▶
```

# 3. Code schreiben

```
tours_controller.rb x
1 class ToursController < ApplicationController
2   before_action :set_tour, only: [:show, :edit, :update, :destroy]
3
4   # GET /tours
5   # GET /tours.json
6   def index
7     @tours = if params[:category_id]
8               Category.find(params[:category_id]).tours
9             else
10              Tour.all
11            end
12   end
13
```

# 3. Code schreiben



The screenshot shows a code editor with a tab labeled "tours/index.html.erb". The code is written in Ruby ERB syntax and is numbered 1 through 10 on the left margin. The code defines an HTML structure for listing tours, including a title, a list of categories, and a link to create a new tour.

```
1 <h1>Listing tours</h1>
2
3 <ul>
4   <% Category.all do |category| %>
5     <li><%= link_to category.name, category_tours_path(category) %></li>
6   <% end %>
7 </ul>
8 <%= link_to 'New Tour', new_tour_path %>
9
10
```

# 4. Tests ausführen

```
thomas@t420: ~/Coding/Ruby/we2/leeze.ms
thomas@t420 ~/Coding/Ruby/we2/leeze.ms (git)-[2_filter_by_categories] % rspec spec
.....*.....*

Pending:
  TourCategory add some examples to (or delete) /home/thomas/Coding/Ruby/we2/leeze.ms/spec/models/tour_category_spec.rb
    # No reason given
    # ./spec/models/tour_category_spec.rb:4
  Category add some examples to (or delete) /home/thomas/Coding/Ruby/we2/leeze.ms/spec/models/category_spec.rb
    # No reason given
    # ./spec/models/category_spec.rb:4

Finished in 0.59092 seconds
13 examples, 0 failures, 2 pending

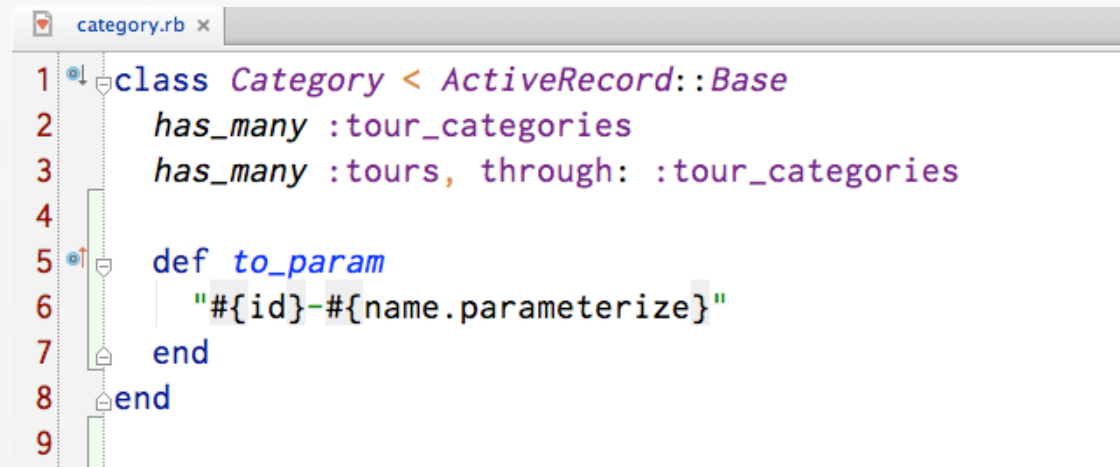
Randomized with seed 40362

thomas@t420 ~/Coding/Ruby/we2/leeze.ms (git)-[2_filter_by_categories] % █
```

# 5. Refactoring

- Problem: Die generierten URLs sind hässlich
  - `http://localhost:3000/1`
- Lösung: Name der Kategorie an die ID anhängen
  - `http://localhost:3000/1-radtour`

# 5. Refactoring



The screenshot shows a code editor window titled 'category.rb'. The code defines a class `Category` that inherits from `ActiveRecord::Base`. It includes two `has_many` associations: `tour_categories` and `tours` (via a `through` association). A `to_param` method is defined to return a string in the format `"#{id}-#{name.parameterize}"`. The code is formatted with syntax highlighting and line numbers.

```
1 class Category < ActiveRecord::Base
2   has_many :tour_categories
3   has_many :tours, through: :tour_categories
4
5   def to_param
6     "#{id}-#{name.parameterize}"
7   end
8 end
9
```

# 5. Refactoring

- Tests ausführen:

```
thomas@t420: ~/Coding/Ruby/we2/leeze.ms
thomas@t420 ~/Coding/Ruby/we2/leeze.ms (git)-[2_filter_by_categories] % rspec spec
.....*.....*
Pending:
  TourCategory add some examples to (or delete) /home/thomas/Coding/Ruby/we2/leeze.ms/spec/models/tour_category_spec.rb
    # No reason given
    # ./spec/models/tour_category_spec.rb:4
  Category add some examples to (or delete) /home/thomas/Coding/Ruby/we2/leeze.ms/spec/models/category_spec.rb
    # No reason given
    # ./spec/models/category_spec.rb:4
Finished in 0.59092 seconds
13 examples, 0 failures, 2 pending
Randomized with seed 40362
thomas@t420 ~/Coding/Ruby/we2/leeze.ms (git)-[2_filter_by_categories] %
```

# Taskboard





# Taskboard

Inbox	In Progress	In Review	Complete
<p>Ein Anwendung soll über ein einfaches Styling verfügen</p>	<p>Ein Anwendung soll über ein einfaches Styling verfügen</p>	<p>Ein Besucher soll Touren anlegen können, ...</p> <p>Ein Besucher soll Touren nach Kategorie filtern können, ...</p>	

# User Story 3

*Die Anwendung soll über ein  
einfaches Styling verfügen*

# Twitter Bootstrap

Benötigte Gems einbauen:

A screenshot of a code editor window titled "Gemfile". The window has a dark background and shows the following Ruby code:

```
# ...
group :asset do
  gem 'therubyracer'
  gem 'twitter-bootstrap-rails'
  gem 'less-rails'
end
# ...
```

\$ bundle install

# Twitter Bootstrap

- Nötige Assets in die Anwendung kopieren:  
`$ rails generate bootstrap:install`
- Application Layout durch Bootstrap ersetzen:  
`$ rails generate bootstrap:layout  
application fluid`

# Twitter Bootstrap

Tabelle auf der Übersichtsseite anzeigen:

```
app/views/tours/index.html.erb

<table>
  <tr>
    <th>Title</th>
    <th></th>
    <th></th>
    <th></th>
  </tr>
  <% @tours.each do |tour| %>
    <tr>
      <td><%= tour.title %></td>
      <td><%= link_to 'Show', tour %></td>
      <td><%= link_to 'Edit', edit_tour_path(tour) %></td>
      <td><%= link_to 'Destroy', tour, method: :delete, data: { confirm: 'Are you
sure?' } %></td>
    </tr>
  <% end %>
</table>

<br />
<%= link_to 'New Tour', new_tour_path %>
```

# Twitter Bootstrap

