



> RubyThursday

# Ruby Snack #5

Nested Forms

Hosted By: Melissa Wahnish

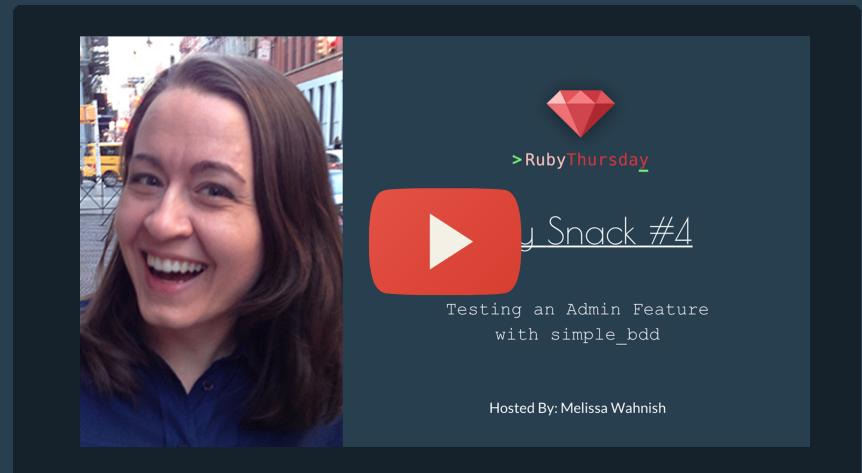
# Want To Code Along?



1. Rails app created
2. RSpec installed and set up
3. Simple BDD installed and set up



A video thumbnail featuring a woman with long brown hair smiling. The thumbnail has a dark blue header with a red play button icon on the left. In the center, there is a red diamond logo with the text "RubyThursday" below it. Below the play button, the text "GEMS DEMO: rspec-rails" is displayed in white. At the bottom, the text "Host: Melissa Wahnish" is shown in white.



A video thumbnail featuring the same woman smiling. The layout is identical to the first thumbnail, with the red play button, red diamond logo, and "RubyThursday" text. Below the play button, the text "Ruby Snack #4" is displayed in white. At the bottom, the text "Testing an Admin Feature with simple\_bdd" is shown in white, followed by "Hosted By: Melissa Wahnish" at the very bottom.

# Nested Forms



Nested Forms allow you to create a parent and child objects in the same form. This makes the user experience better by reducing the number of forms and takes advantage of Active Record Associations.

[http://guides.rubyonrails.org/form\\_helpers.html#building-complex-forms](http://guides.rubyonrails.org/form_helpers.html#building-complex-forms)  
[http://guides.rubyonrails.org/association\\_basics.html](http://guides.rubyonrails.org/association_basics.html)

# Starship Spec:

/spec/features/starship\_spec.rb



```
require 'rails_helper'

feature "Starship with Crew Members" do
  #communicate purpose upfront with Simple BDD
  scenario "visitor can create starship with crew members" do
    Given "visitor can view new starship form"
    When "visitor enters information for starship and crew members"
    Then "visitor can see starship listing"
  end
  #define methods from scenario
  def visitor_can_view_new_starship_form
    visit new_starship_path
    expect(page).to have_content("New Starship")
  end
  def visitor_enters_information_for_starship_and_crew_members
    enter_starship_info
    enter_crew_members_info
    click_button 'Create Starship'
  end
end
```

# Starship Spec (cont.) : /spec/features/starship\_spec.rb



```
def visitor_can_see_starship_listing
  expect(page).to have_content("USS Enterprise (NCC-1701-A)")
  expect(page).to have_content("James T. Kirk")
end

# Define additional methods
def enter_starship_info
  fill_in("starship[name]", with: "USS Enterprise (NCC-1701-A)")
end

def enter_crew_members_info
  fill_in("starship[crew_members_attributes][0][name]", with: "James T. Kirk")
  select('Command', :from => "starship[crew_members_attributes][0][division]")
  fill_in("starship[crew_members_attributes][1][name]", with: "Spock")
  select('Science', :from => "starship[crew_members_attributes][1][division]")
  fill_in("starship[crew_members_attributes][2][name]", with: "Leonard McCoy")
  select('Engineering', :from => "starship[crew_members_attributes][0][division]")
end

end
```

Run the Spec:



```
$ rspec spec/features/starship_spec.rb
```

# Generate Starship model:



```
$ rails g scaffold Starship name:string
```

```
$ rake db:migrate
```

# Generate Crew Member model:



```
$ rails g model CrewMember name:string division:string starship_id:integer
```

```
$ rake db:migrate
```



## Models:

```
/app/models/starship.rb
```

```
class Starship < ActiveRecord::Base  
  has_many :crew_members  
  accepts_nested_attributes_for :crew_members,  
    reject_if: lambda { |attributes| attributes["name"].blank? }  
end
```

```
/app/models/crew_member.rb
```

```
class CrewMember < ActiveRecord::Base  
  belongs_to :starship  
end
```

# New Starship HTML: /app/views/starships/\_form.html.erb



```
Crew Members:  
  
<ul>  
  
  <%= f.fields_for :crew_members do |cm| %>  
  
    <li>  
  
      <%= cm.label :name %>  
  
      <%= cm.text_field :name %>  
  
      <%= cm.label :division %>  
  
      <%= cm.select(:division, ["Command", "Engineering", "Science"]) %>  
  
    </li>  
  
  <% end %>  
  
</ul>
```

# Starship Controller:

/app/controllers/starships\_controller.rb



```
def new
  @starship = Starship.new
  3.times { @starship.crew_members.build}
end
# ...
def starship_params
  params.require(:starship).permit(:name,
  crew_members_attributes: [:starship_id, :name, :division])
end
```

## Show Starship HTML: /app/views/starships/show.html.erb

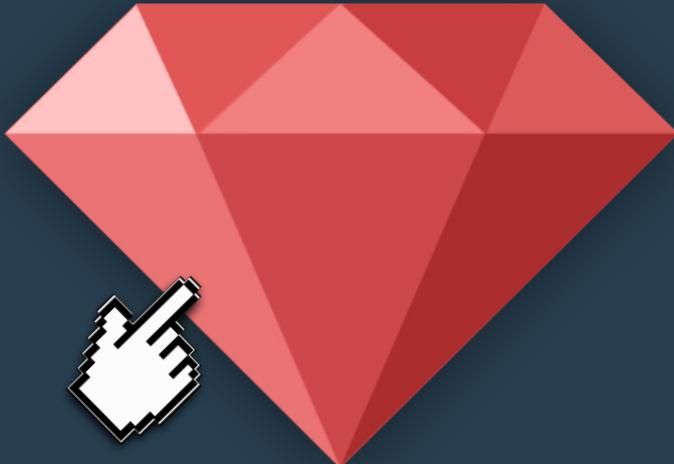


```
<strong>Crew Members:</strong>  
  
<% @starship.crew_members.each do |crew_member| %>  
  
  <%= crew_member.name %>  
  
<% end %>
```

Want More Free Ruby Snacks?



Subscribe



> RubyThursday