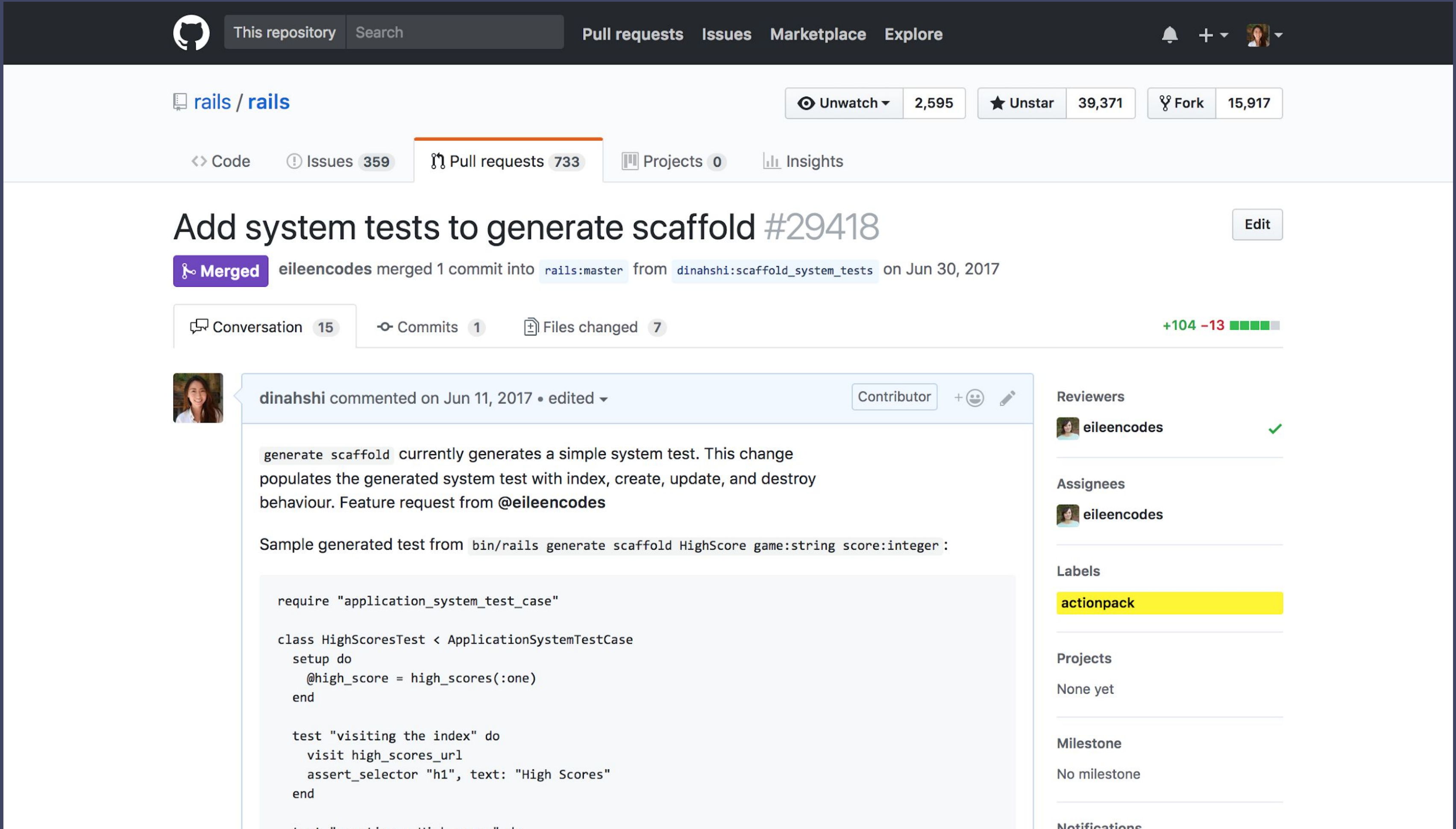
# Your first contribution (and beyond)

April 19, 2018

RailsConf

# Dinah Shi

@dinahshi



# Add system tests to generate scaffold #29418

Merged eileencodes merged 1 commit into `rails:master` from `dinahshi:scaffold_system_tests` on Jun 30, 2017

dinahshi commented on Jun 11, 2017 • edited

`generate scaffold` currently generates a simple system test. This change populates the generated system test with index, create, update, and destroy behaviour. Feature request from @eileencodes

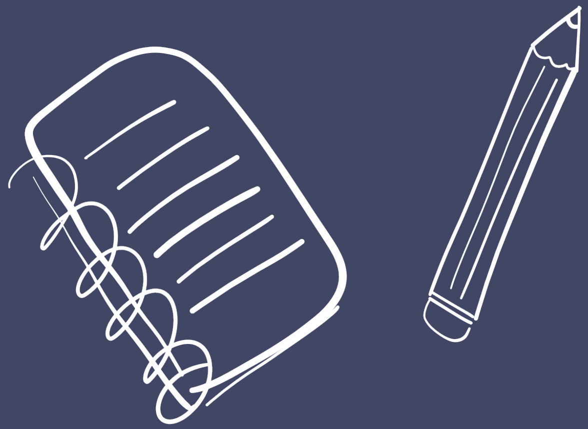
Sample generated test from `bin/rails generate scaffold HighScore game:string score:integer`:

```
require "application_system_test_case"

class HighScoresTest < ApplicationSystemTestCase
  setup do
    @high_score = high_scores(:one)
  end

  test "visiting the index" do
    visit high_scores_url
    assert_selector "h1", text: "High Scores"
  end
```

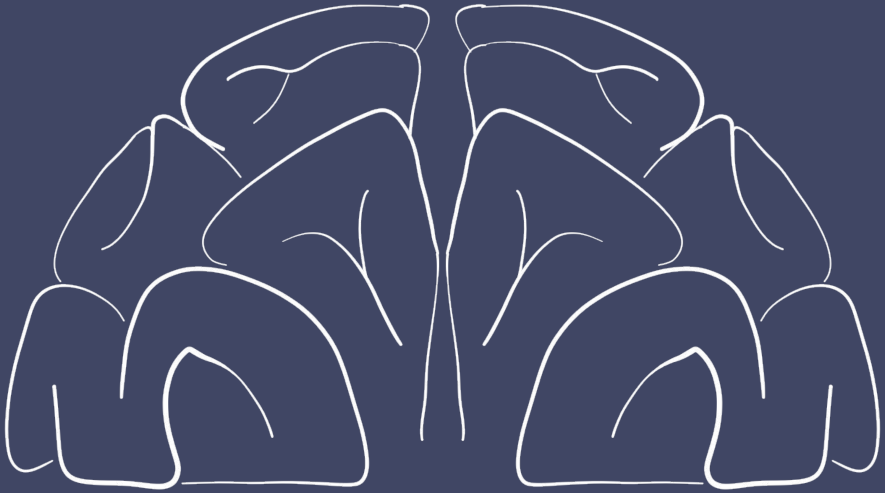
**Article** Rails Guides: Contributing to Ruby on Rails

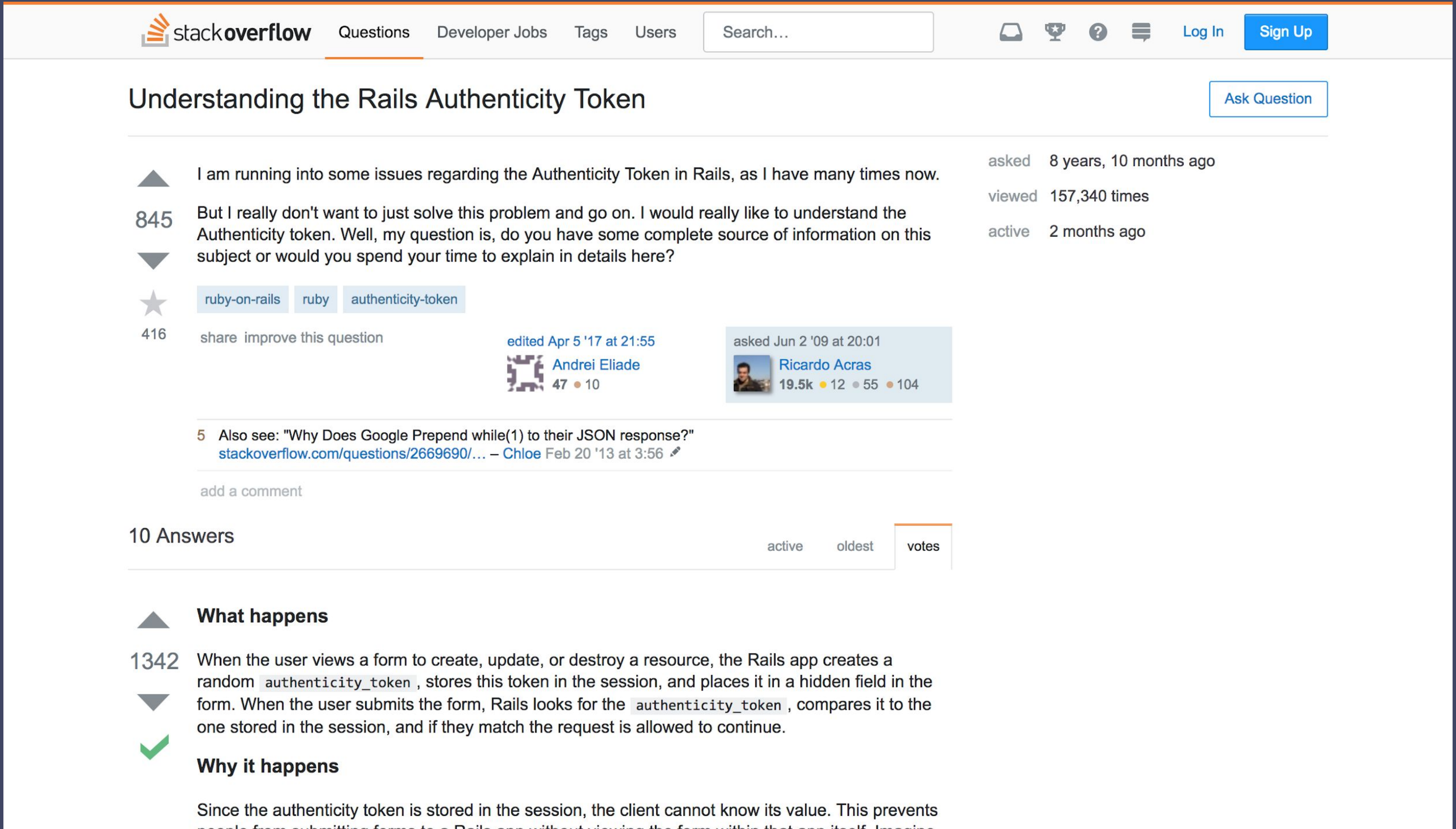
# What is a contribution?

# Blog



# Sharing Knowledge





# Understanding the Rails Authenticity Token

845

I am running into some issues regarding the Authenticity Token in Rails, as I have many times now.

But I really don't want to just solve this problem and go on. I would really like to understand the Authenticity token. Well, my question is, do you have some complete source of information on this subject or would you spend your time to explain in details here?

5 Also see: "Why Does Google Prepend while(1) to their JSON response?" stackoverflow.com/questions/2669690/… – Chloe Feb 20 '13 at 3:56

## 10 Answers

1342

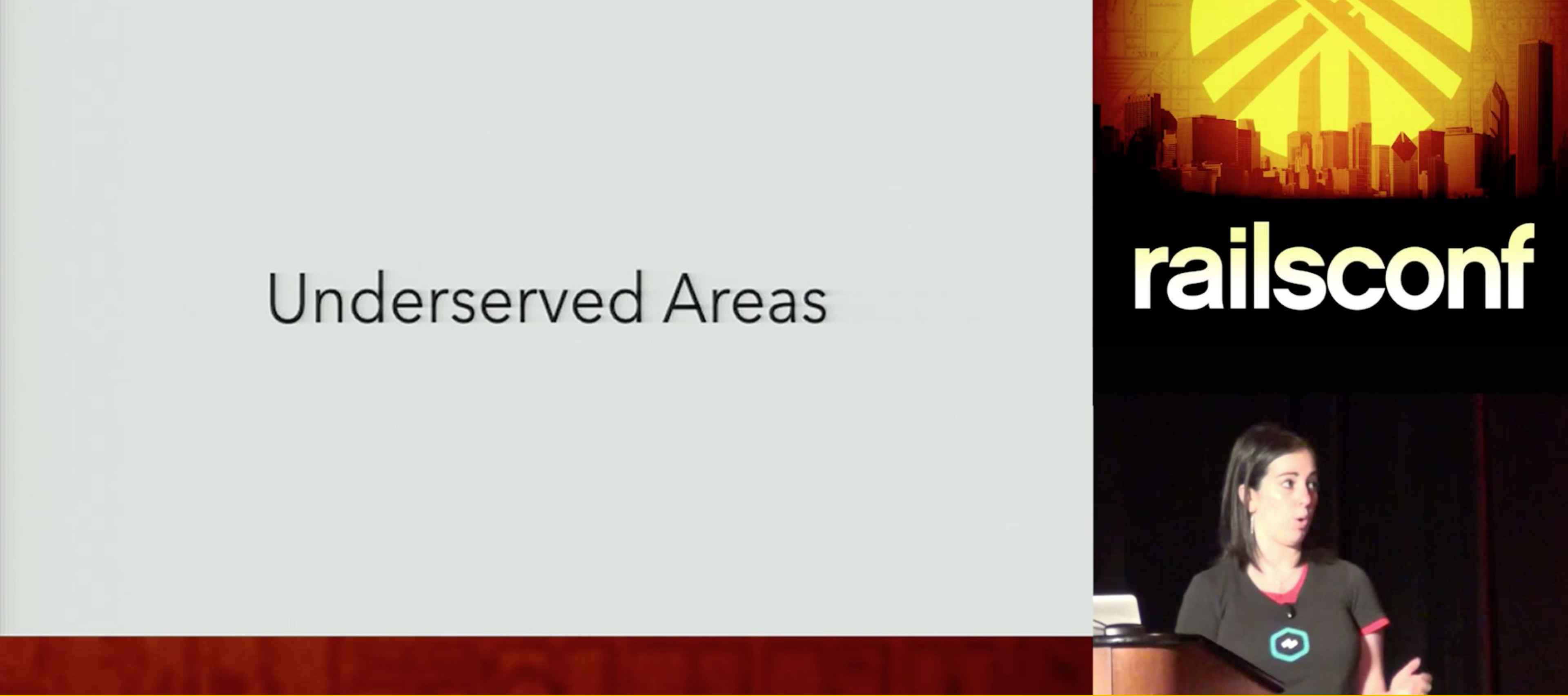
### What happens

When the user views a form to create, update, or destroy a resource, the Rails app creates a random `authenticity_token`, stores this token in the session, and places it in a hidden field in the form. When the user submits the form, Rails looks for the `authenticity_token`, compares it to the one stored in the session, and if they match the request is allowed to continue.

### Why it happens

Since the authenticity token is stored in the session, the client cannot know its value. This prevents people from submitting forms to a Rails app without viewing the form within that app itself. Imagine

# Events



Underserved Areas

**Video** Building an OSS-Centric Company (and Why You Want To) by Leah Silber - RailsConf 2014

**Article** GitHub Open Source Guides: How to Contribute > What it means to contribute

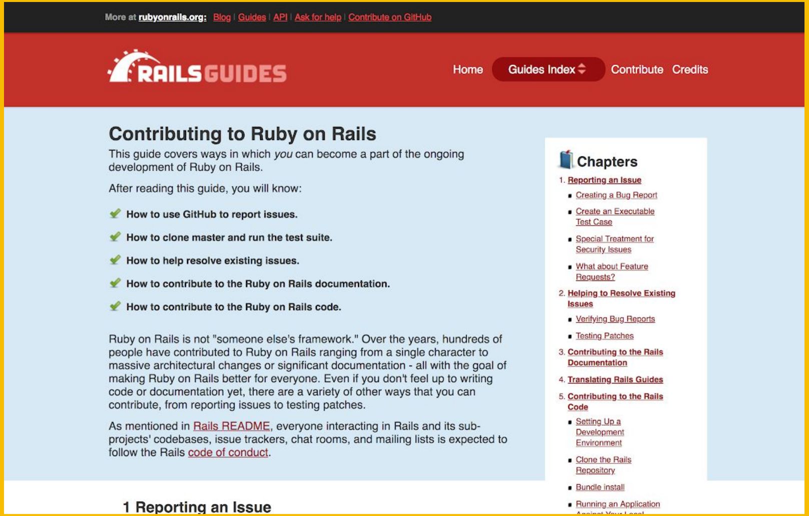
# Your first patch

# Documentation

Lower barrier to entry

Learn the flow of open source

You are the best person for the job


More at rubyonrails.org: Blog | Guides | API | Ask for help | Contribute on GitHub
RAILS GUIDES
Home
Guides Index
Contribute
Credits

# Contributing to Ruby on Rails

This guide covers ways in which *you* can become a part of the ongoing development of Ruby on Rails.

After reading this guide, you will know:

- ✓ How to use GitHub to report issues.
- ✓ How to clone master and run the test suite.
- ✓ How to help resolve existing issues.
- ✓ How to contribute to the Ruby on Rails documentation.
- ✓ How to contribute to the Ruby on Rails code.

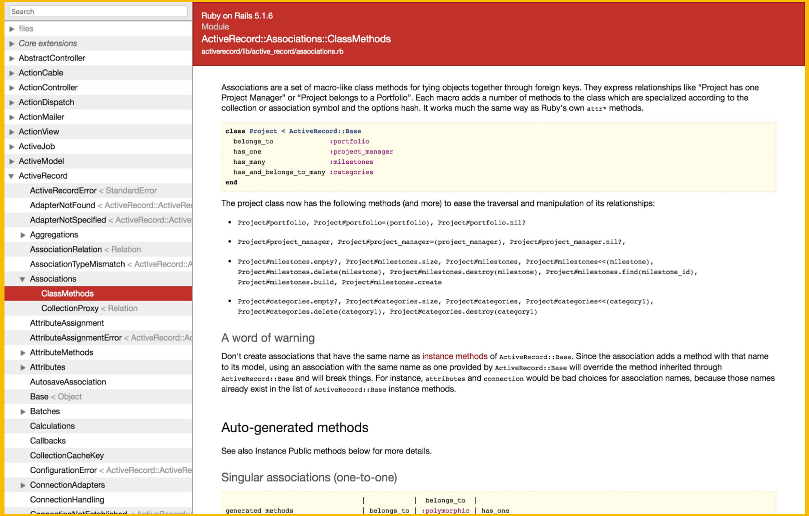
Ruby on Rails is not "someone else's framework." Over the years, hundreds of people have contributed to Ruby on Rails ranging from a single character to massive architectural changes or significant documentation - all with the goal of making Ruby on Rails better for everyone. Even if you don't feel up to writing code or documentation yet, there are a variety of other ways that you can contribute, from reporting issues to testing patches.

As mentioned in Rails README, everyone interacting in Rails and its sub-projects' codebases, issue trackers, chat rooms, and mailing lists is expected to follow the Rails code of conduct.

## Chapters

1. Reporting an Issue
   - Creating a Bug Report
   - Create an Executable Test Case
   - Special Treatment for Security Issues
   - What about Feature Requests?
2. Helping to Resolve Existing Issues
   - Verifying Bug Reports
   - Testing Patches
3. Contributing to the Rails Documentation
4. Translating Rails Guides
5. Contributing to the Rails Code
   - Setting Up a Development Environment
   - Clone the Rails Repository
   - Bundle install
   - Running an Application

## 1 Reporting an Issue



Ruby on Rails 5.1.6

Module

ActiveRecord::Associations::ClassMethods

activerecord/lib/active_record/associations.rb

Associations are a set of macro-like class methods for tying objects together through foreign keys. They express relationships like "Project has one Project Manager" or "Project belongs to a Portfolio". Each macro adds a number of methods to the class which are specialized according to the collection or association symbol and the options hash. It works much the same way as Ruby's own `attr*` methods.

```
class Project < ActiveRecord::Base
  belongs_to              :portfolio
  has_one                 :project_manager
  has_many                :milestones
  has_and_belongs_to_many :categories
end
```

The project class now has the following methods (and more) to ease the traversal and manipulation of its relationships:

- `Project#portfolio, Project#portfolio=(portfolio), Project#portfolio.nil?`
- `Project#project_manager, Project#project_manager=(project_manager), Project#project_manager.nil?,`
- `Project#milestones.empty?, Project#milestones.size, Project#milestones, Project#milestones<<(milestone), Project#milestones.delete(milestone), Project#milestones.destroy(milestone), Project#milestones.find(milestone_id), Project#milestones.build, Project#milestones.create`
- `Project#categories.empty?, Project#categories.size, Project#categories, Project#categories<<(category1), Project#categories.delete(category1), Project#categories.destroy(category1)`

### A word of warning

Don't create associations that have the same name as instance methods of `ActiveRecord::Base`. Since the association adds a method with that name to its model, using an association with the same name as one provided by `ActiveRecord::Base` will override the method inherited through `ActiveRecord::Base` and will break things. For instance, `attributes` and `connection` would be bad choices for association names, because those names already exist in the list of `ActiveRecord::Base` instance methods.

## Auto-generated methods

See also Instance Public methods below for more details.

### Singular associations (one-to-one)

```
                                  |            |  belongs_to  |
generated methods                 | belongs_to | :polymorphic | has_one
```

# Finding an issue

Browse the API docs

Check coverage for newer features

Watch pull requests

# First code patch

# good-first-patch

# Finding an issue

Watch the repository

CodeTriage

Verify bugs, clarify repro steps, create test scripts

# Build trust, establish rapport.

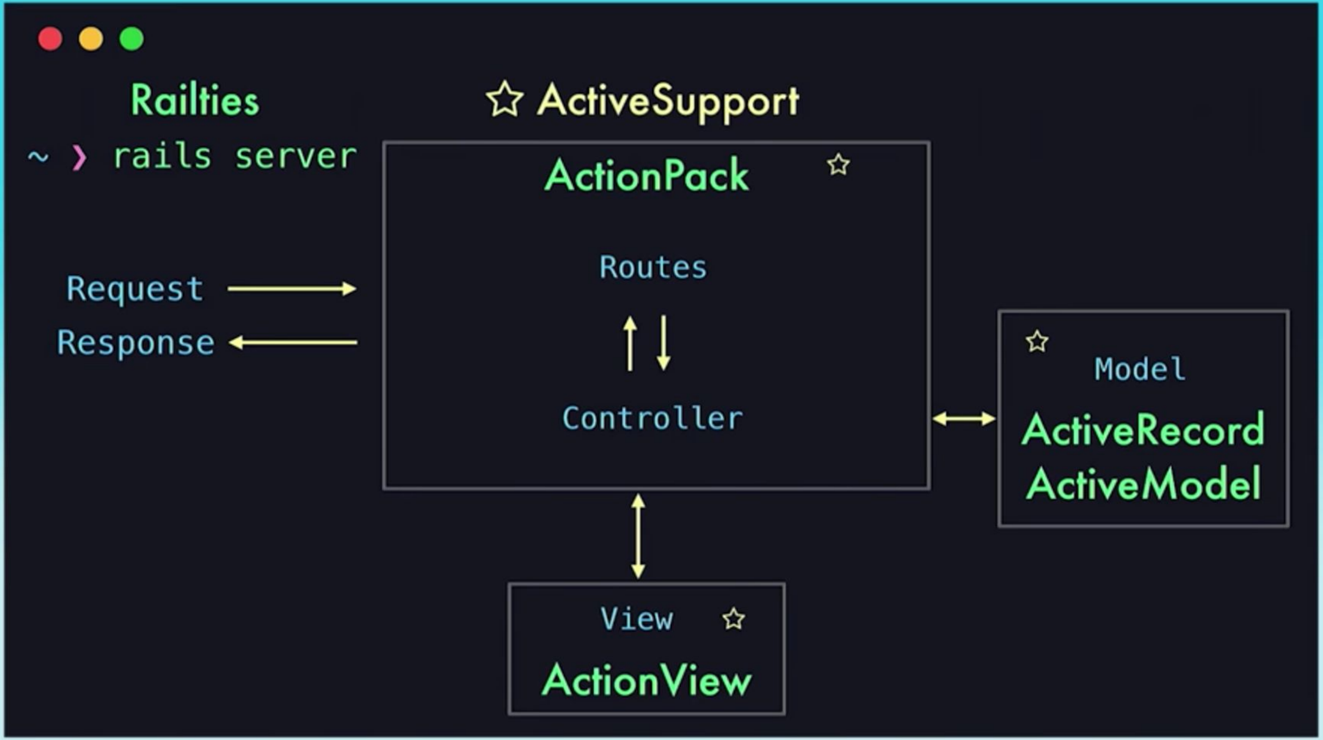
# rapport

*noun*

A close and harmonious relationship in which the people or groups concerned understand each other's feelings or ideas and communicate well.

# Finding an issue

Check the rails-core-mailing-list for feature requests



Railties

```
~ ❯ rails server
```

☆ ActiveSupport

ActionPack ☆

Routes

Controller

Request

Response

☆ Model

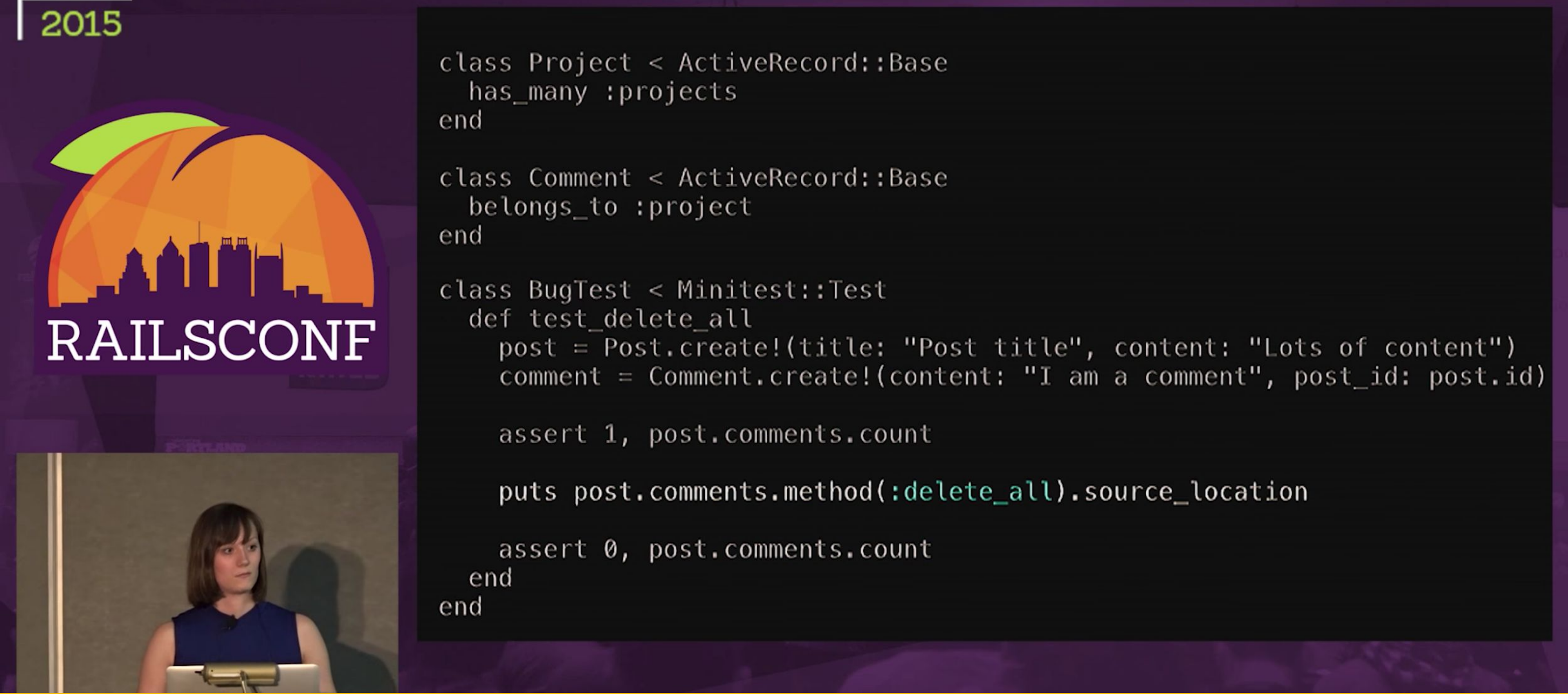
ActiveRecord

ActiveModel

View ☆

ActionView

**Video** Perusing the Rails Source Code: A Beginners Guide by Alex Kitchens - RailsConf 2017



```
class Project < ActiveRecord::Base
  has_many :projects
end

class Comment < ActiveRecord::Base
  belongs_to :project
end

class BugTest < Minitest::Test
  def test_delete_all
    post = Post.create!(title: "Post title", content: "Lots of content")
    comment = Comment.create!(content: "I am a comment", post_id: post.id)

    assert 1, post.comments.count

    puts post.comments.method(:delete_all).source_location

    assert 0, post.comments.count
  end
end
```

**Video** Breaking Down the Barrier: Demystifying Contributing to Rails - RailsConf 2015

# Need more help?

Sometimes it’s nice to talk to a real person.
Here’s where you’ll find em:

- Git history
- Regular contributor
- Core team

# Submitting a PR

- Tone is *so* important
- Working in a distributed team
- Patience is a virtue
- Your PR might never get accepted, and that’s okay!

# How I found my first issue

# Mentorship

# Pros

- Changes more likely to be merged
- Go to person for questions
- Guide through first PR
- Great first patch experience

# Cons

- Mentorships are hard
- Does not scale

# Before reaching out…

Show that you have made an effort

Come with a bug/feature

Balance request with the amount of trust you’ve built up

**Video** Incognito Mentorship by Katrina Owen - Codeland NYC 2017

# Keep in mind

1. It doesn’t have to be a core team member
2. You don’t have to call it “mentorship”

# The next patch

# Rails Maintainers

Issues team: can commit documentation

Committers team: can commit code

Core team: can release gems, set the vision for future of Rails

# You are ready.