



Stefan Wintermeyer

Elixir was created in 2011
by José Valim to be a real
concurrent language.

Rails vs. Hanami vs. Phoenix.

AFRICA'S PREMIERE RUBY CONFERENCE



RubyFuza & Friends 2019

7th & 8th Feb 2019 • Strand Hotel, Cape Town



Stefan Wintermeyer

Rails vs. Phoenix vs. Hanami

by Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.



2019 Sponsors

Stefan Wintermeyer

I am a software firefighter and architect for hire.

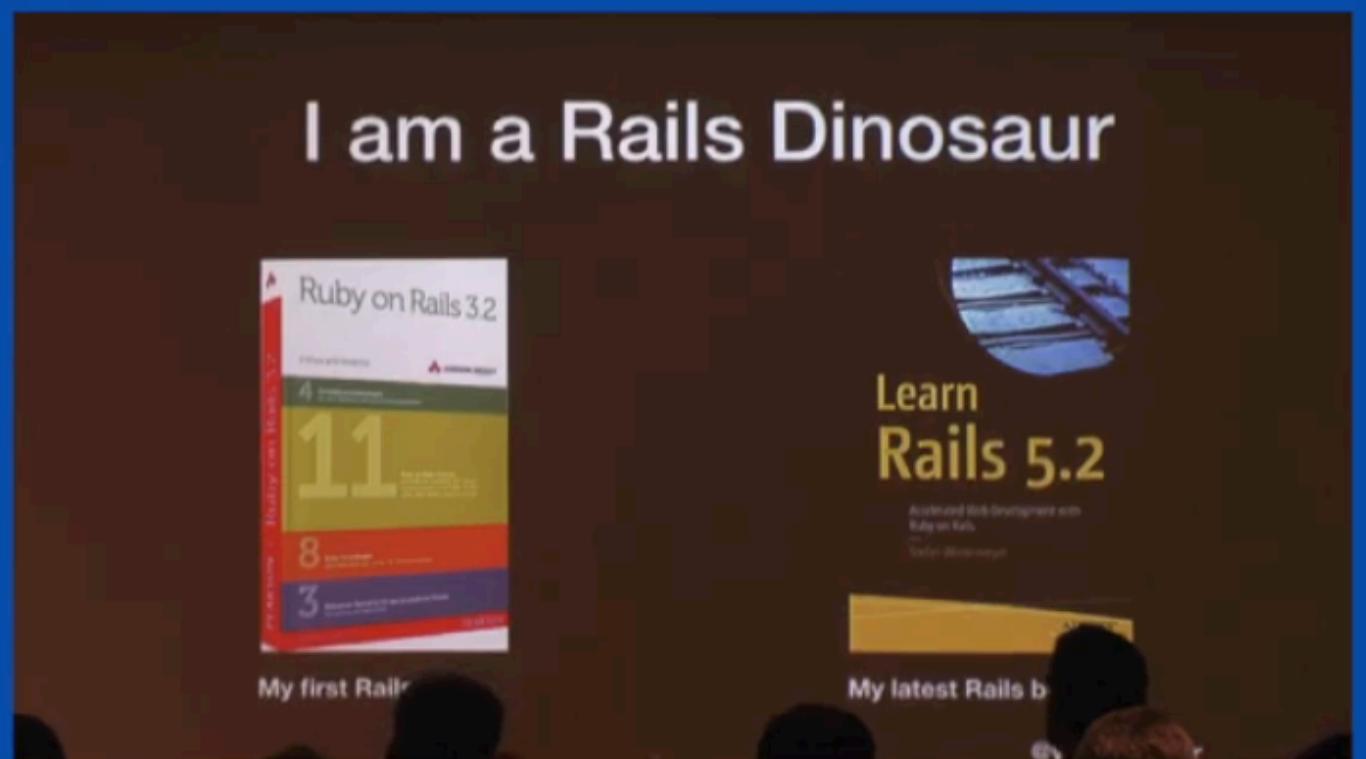
I see a lot of projects which became unmaintainable
over the years.

Rails vs. Hanami vs. Phoenix.

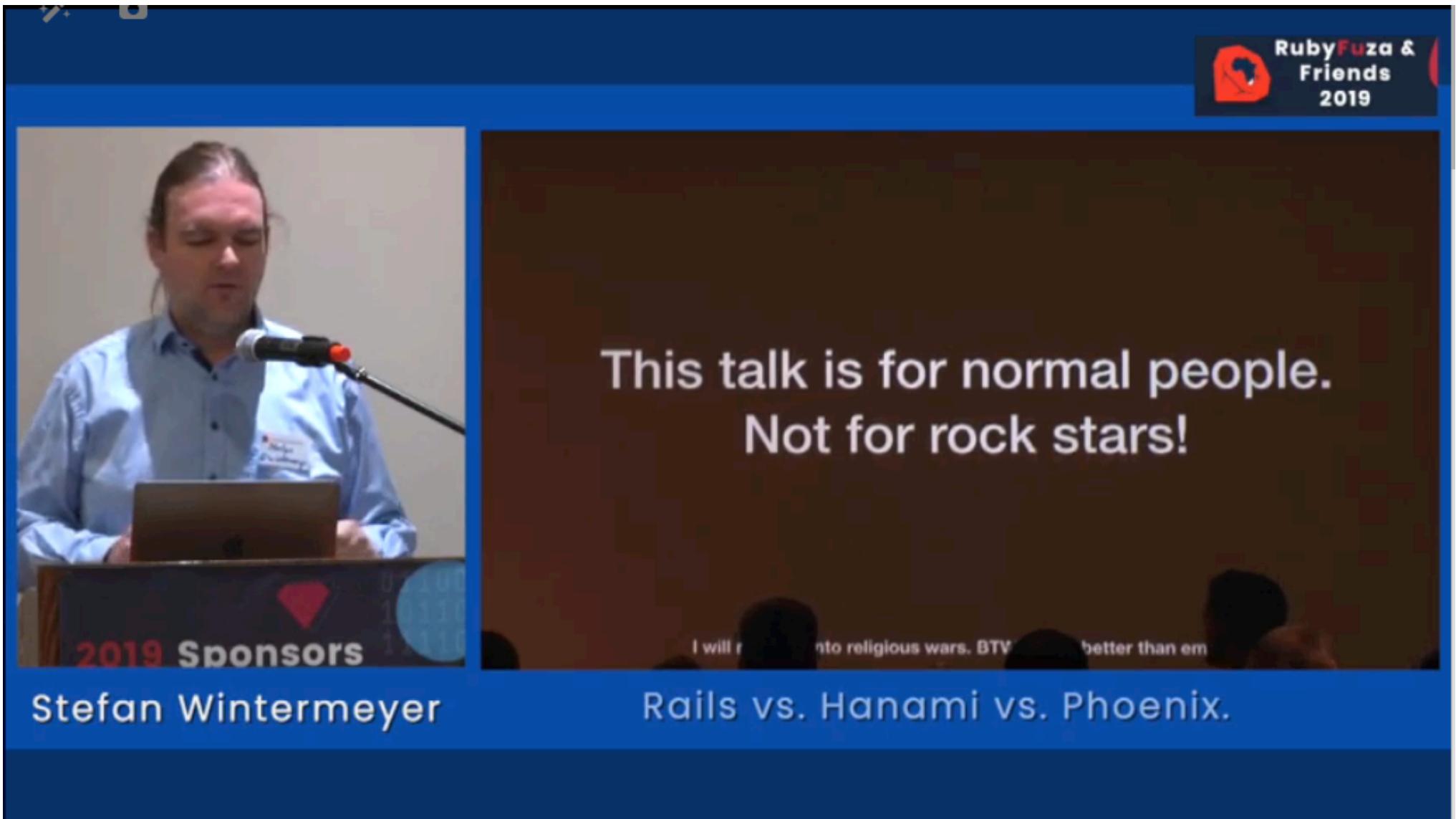


2019 Sponsors

Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



RubyFuza & Friends 2019

2019 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.

<https://rubyonrails.org> <https://phoenixframework.org> <http://hanamirb.org>

Rails **Phoenix** **Hanami**

Frameworks to develop web applications.



Stefan Wintermeyer

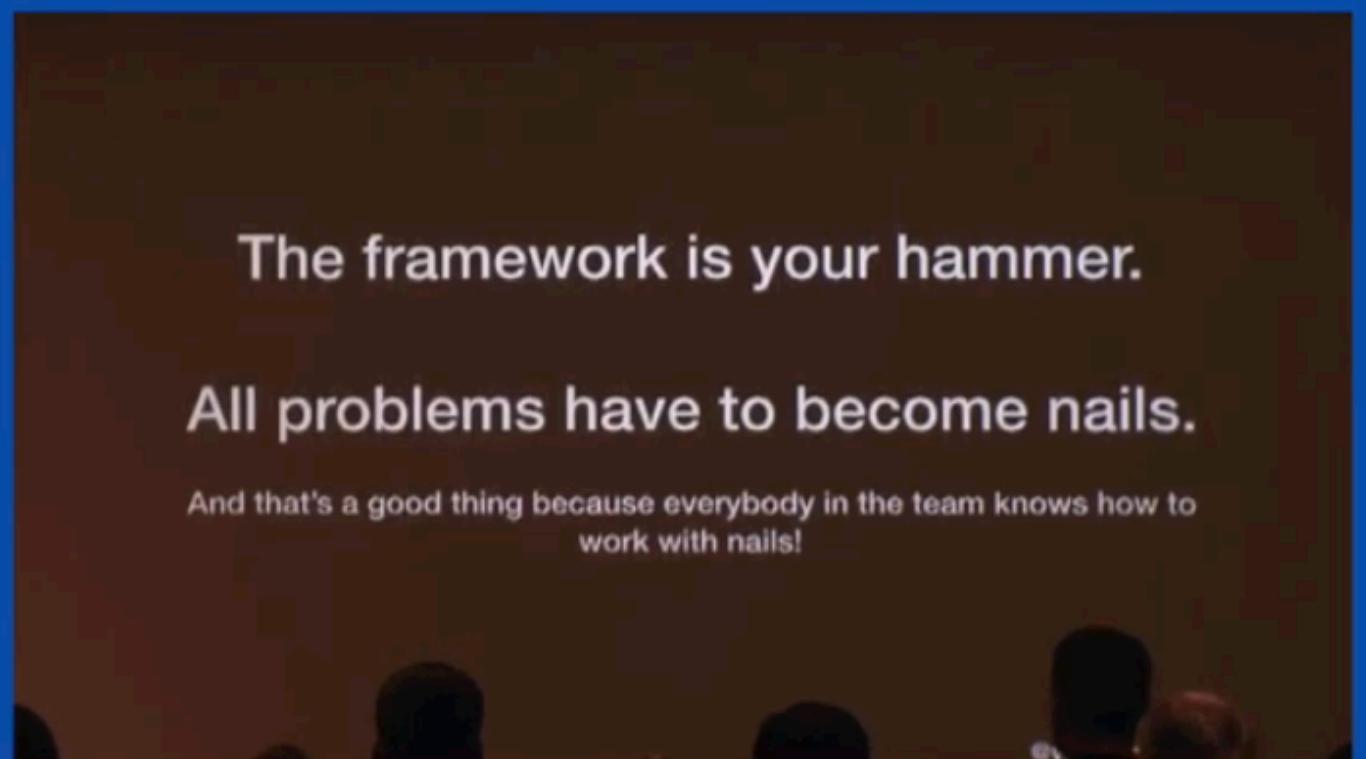
Why use a Framework?

- Structure and order within the project
- No need to reinvent the wheel every day.
Let others do the heavy lifting.
- Documentation
- Easier on boarding of new team members

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer





Stefan Wintermeyer

	Inventor/Leader	Sponsor
<i>Ruby</i>	Yukihiro Matsumoto	Heroku
<i>Elixir</i>	José Valim	Plataformatec
Ruby on Rails	David Heinemeier Hansson	Basecamp
Phoenix Framework	Chris McCord	DockYard
Hanami	Luca Guidi	-

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

	Version	Command
Ruby	2.5.3p105	<code>ruby -v</code>
Elixir	1.8.0	<code>elixir -v</code>
Ruby on Rails	5.2.2	<code>rails -v</code>
Phoenix Framework	1.4.0	<code>mix phx.new --version</code>
Hanami	1.3.1	<code>hanami -v</code>

Rails vs. Hanami vs. Phoenix.



Ruby vs. Elixir

2019 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

Elixir was created in 2011
by José Valim to be a real
concurrent language.

Rails vs. Hanami vs. Phoenix.



2019 Sponsors

Stefan Wintermeyer

Don't let the Elixir syntax
fool you into believing it's
easy to learn.

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

A Ruby programmer needs
quite some time to learn Elixir.

Functional programming is a
totally different ball game.

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

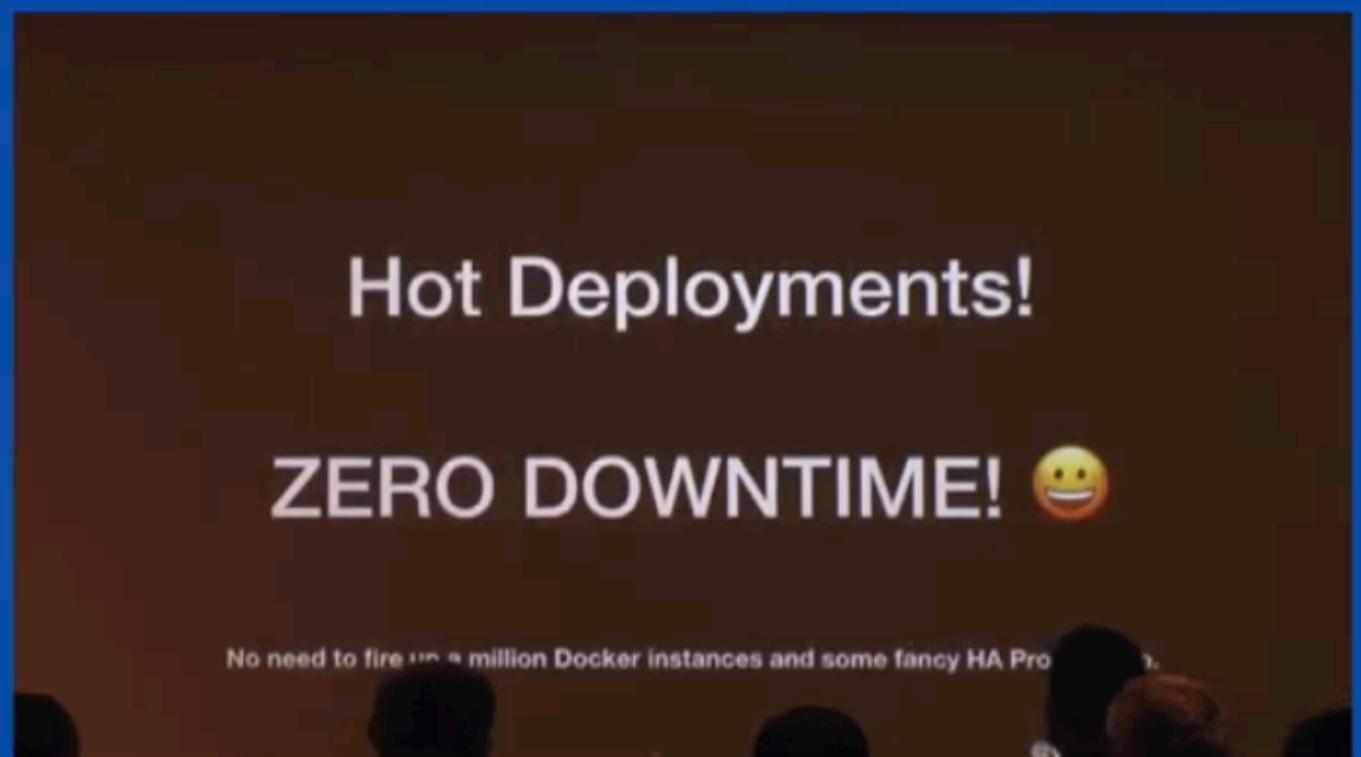
Elixir's biggest advantages
are speed, scalability and
hot deployments.

Rails vs. Hanami vs. Phoenix.



2019 Sponsors

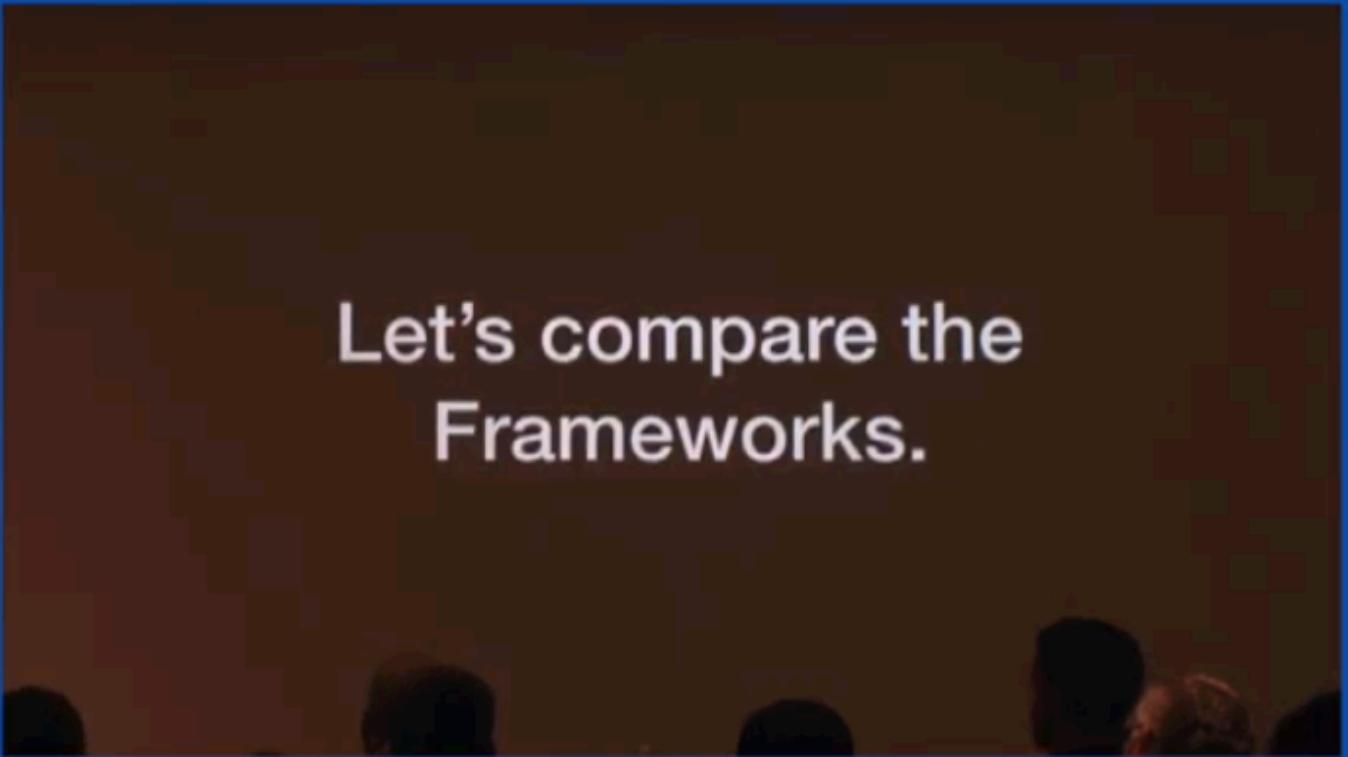
Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Let's compare the
Frameworks.

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

Continuity for Developers

Do you remember how painful Rails upgrades used to be?

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

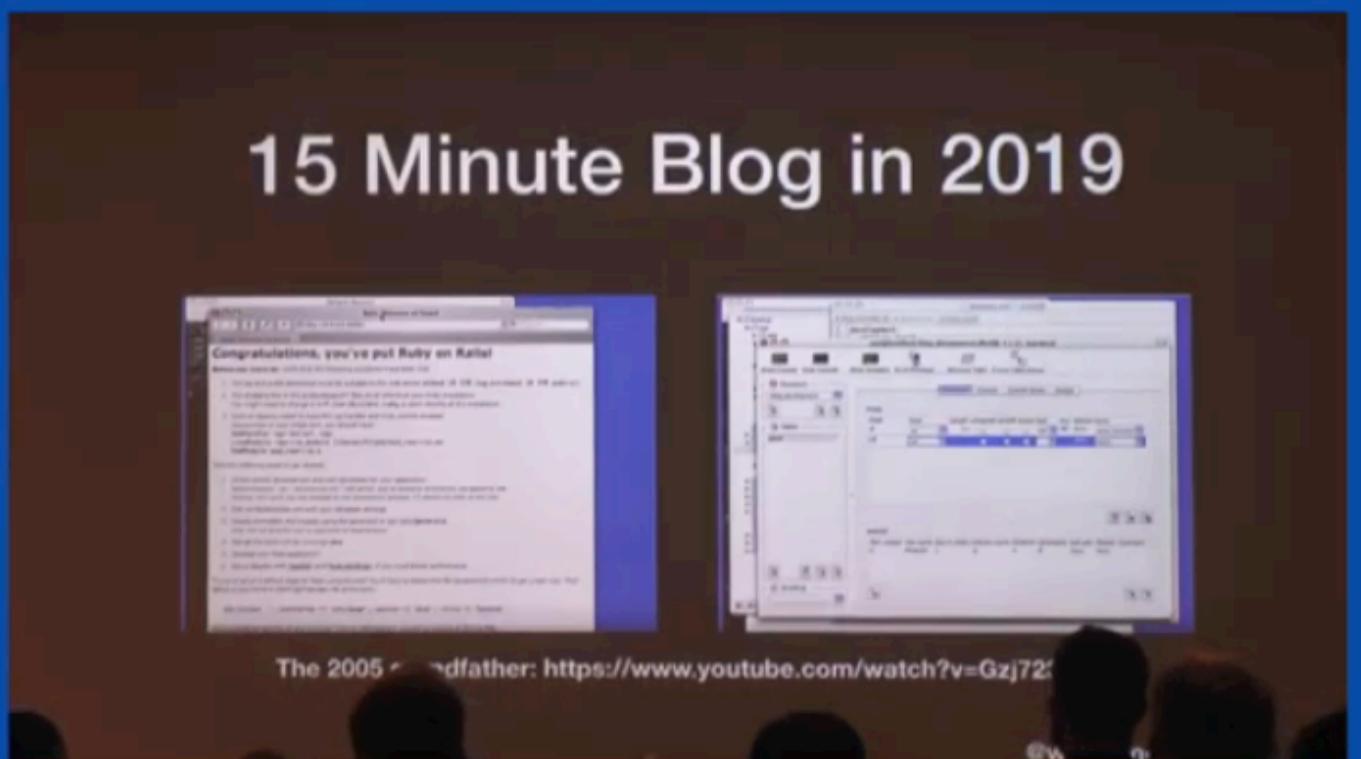
	Version	How easy is an upgrade to the next version?
Ruby on Rails	5.2.2	😊😊
Phoenix Framework	1.4.0	😊😊😊
Hanami	1.3.1	2.0 is a big change 🤯

Rails vs. Hanami vs. Phoenix.



2019 Sponsors

Stefan Wintermeyer



15 Minute Blog in 2019

The 2005 grandfather: <https://www.youtube.com/watch?v=Gzj72>

Congratulations, you've put Ruby on Railst

15 minute blog

Rails vs. Hanami vs. Phoenix.

Rails vs. Hanami vs. Phoenix.



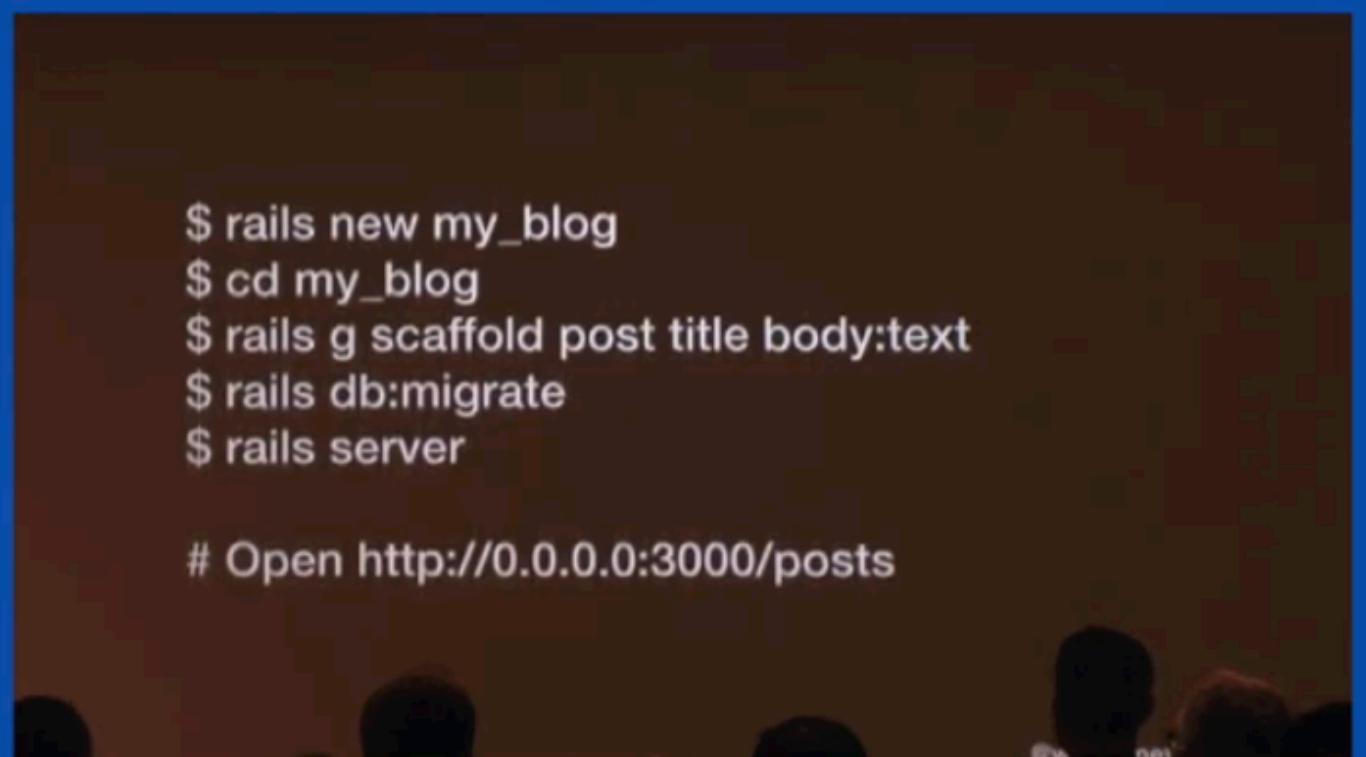
Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



RubyFuza & Friends 2019

```
app
├── assets
│   ├── javascripts
│   │   └── posts.coffee
│   ├── stylesheets
│   │   ├── posts.scss
│   │   └── scaffolds.scss
│   └── controllers
│       └── posts_controller.rb
├── helpers
│   └── posts_helper.rb
└── models
    └── post.rb
views
└── posts
    ├── _form.html.erb
    ├── _post.json.jbuilder
    ├── edit.html.erb
    ├── index.html.erb
    └── index.json.jbuilder
    └── _il.erb
    └── _nl.erb
    └── _us.il.erb
config
└── routes.rb
db
└── migrate
    └── 20190126153653_create_posts.rb
test
└── controllers
    └── posts_controller_test.rb
└── fixtures
    └── posts.yml
└── models
    └── post_test.rb
└── system
    └── posts_test.rb
```

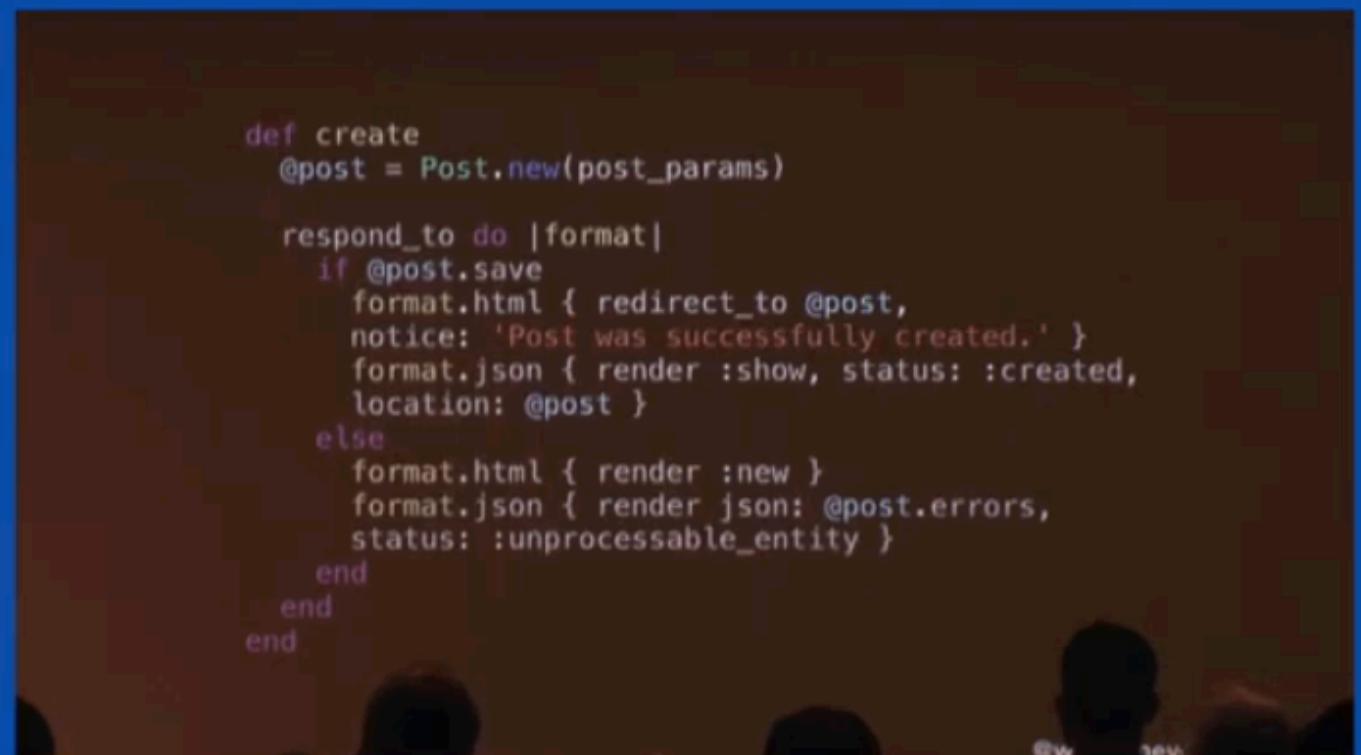
2019 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

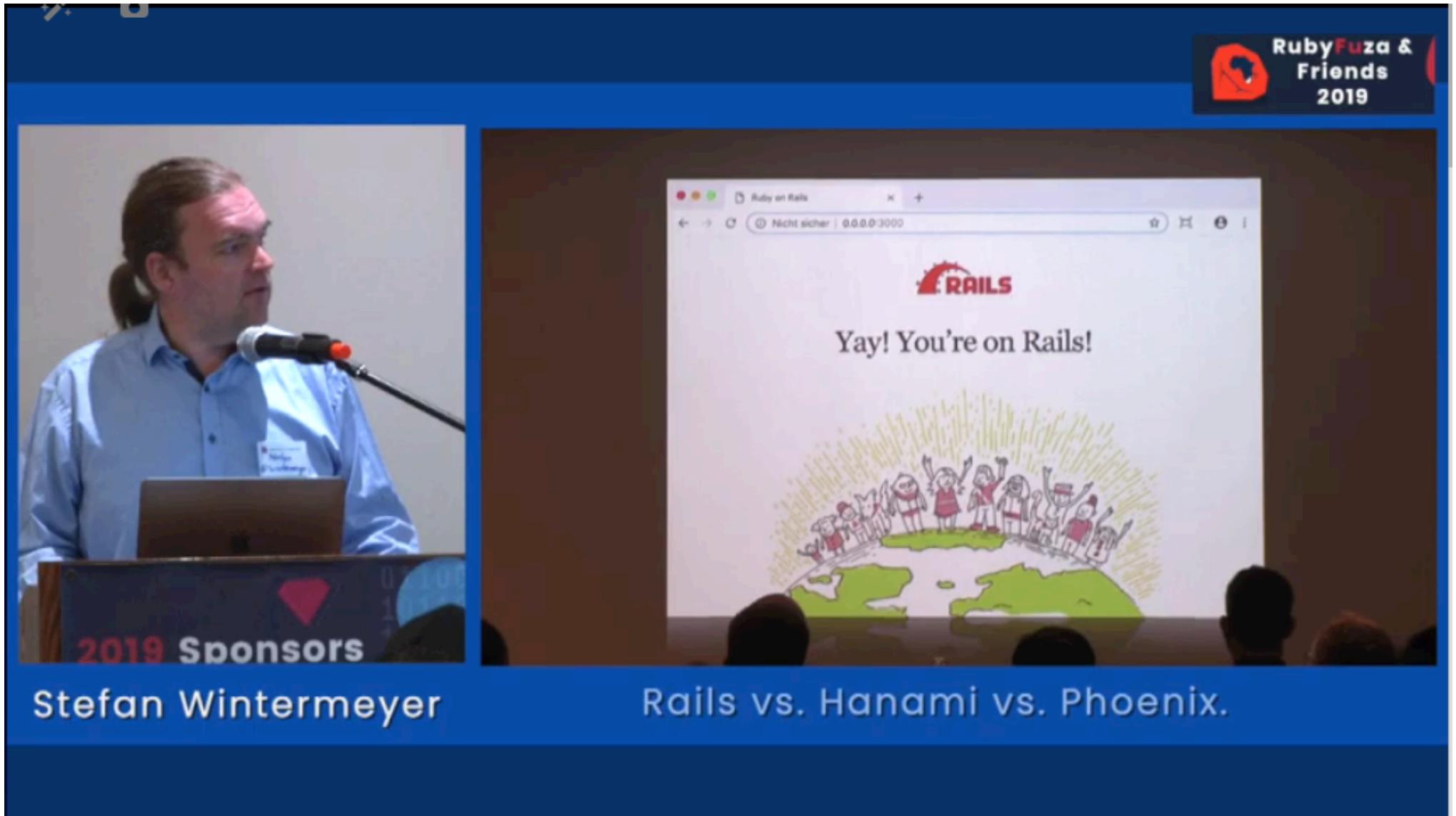


A screenshot of a presentation slide. On the left, there is a video feed of Stefan Wintermeyer speaking. On the right, a large portion of a computer screen displays a block of Ruby code. The code is a 'create' method for a 'Post' model, handling both HTML and JSON responses based on the save status of the post.

```
def create
  @post = Post.new(post_params)

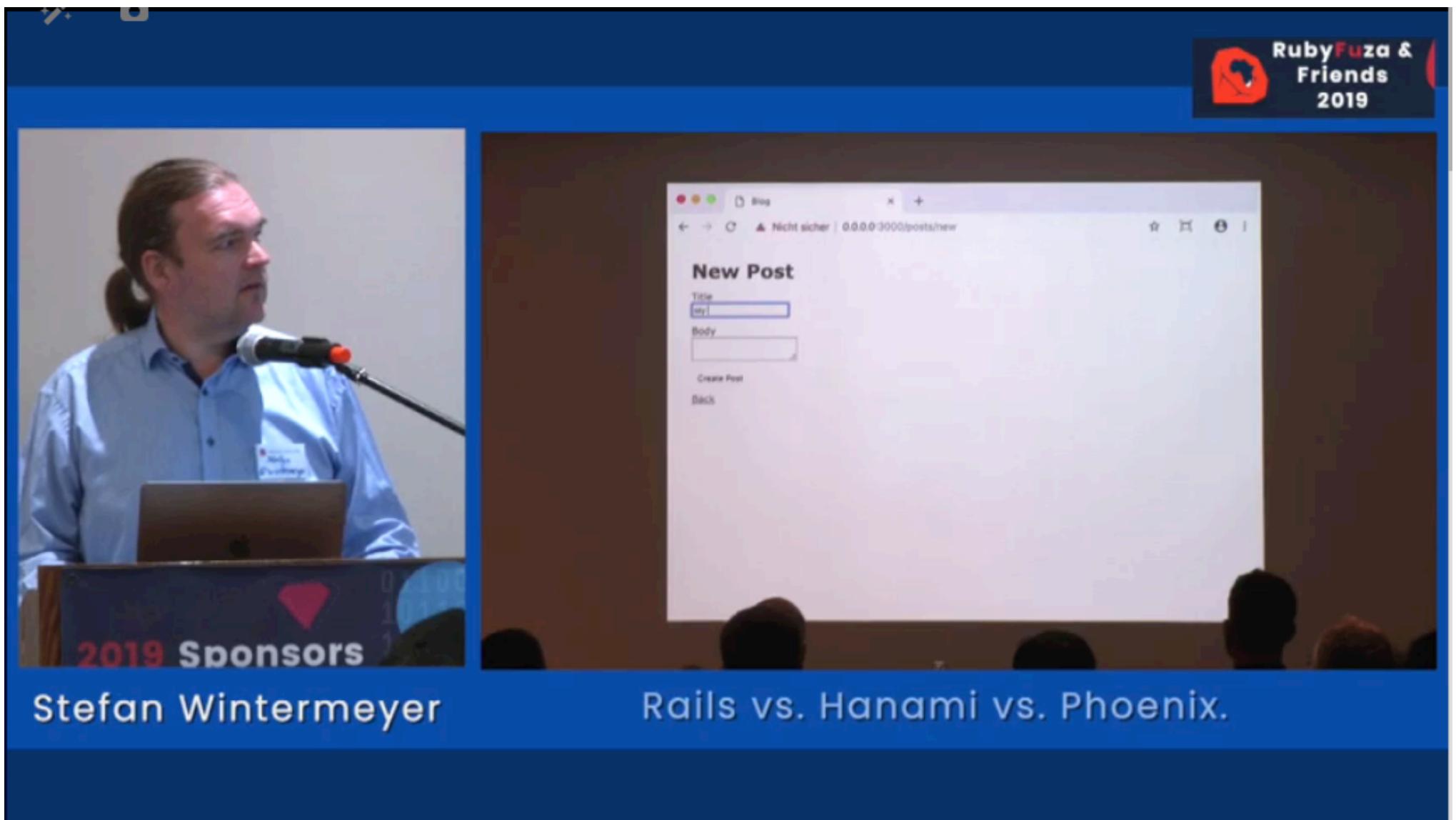
  respond_to do |format|
    if @post.save
      format.html { redirect_to @post,
        notice: 'Post was successfully created.' }
      format.json { render :show, status: :created,
        location: @post }
    else
      format.html { render :new }
      format.json { render json: @post.errors,
        status: :unprocessable_entity }
    end
  end
end
```

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.

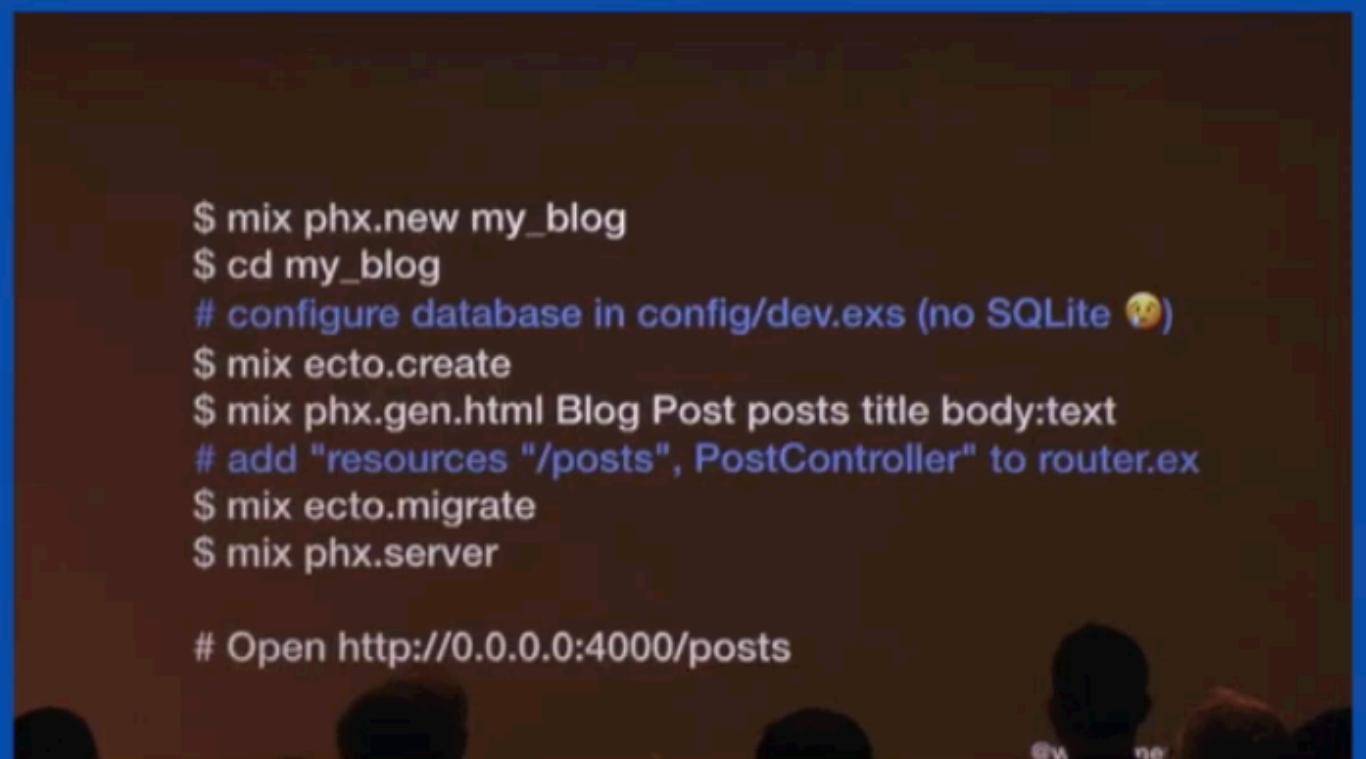


Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.



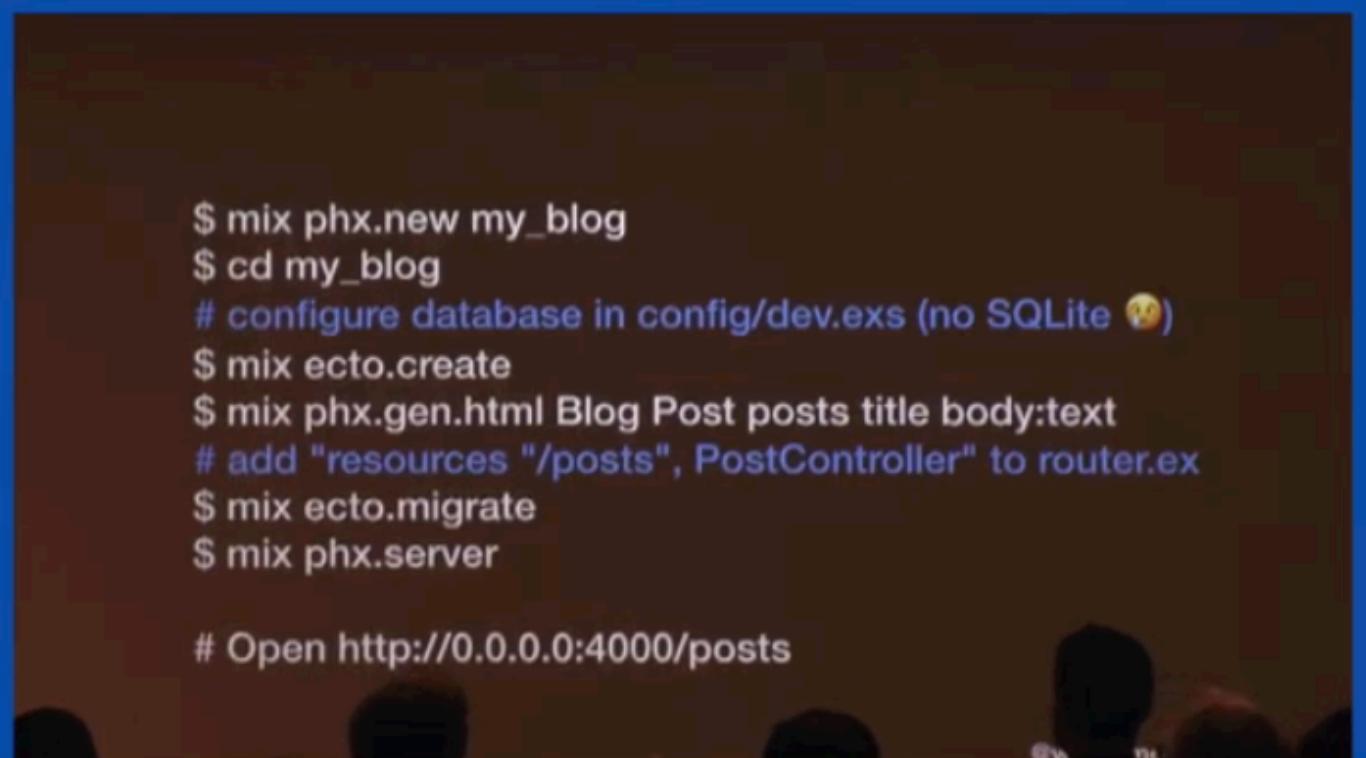
Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.

RubyFuza & Friends 2019

2019 Sponsors

Stefan Wintermeyer

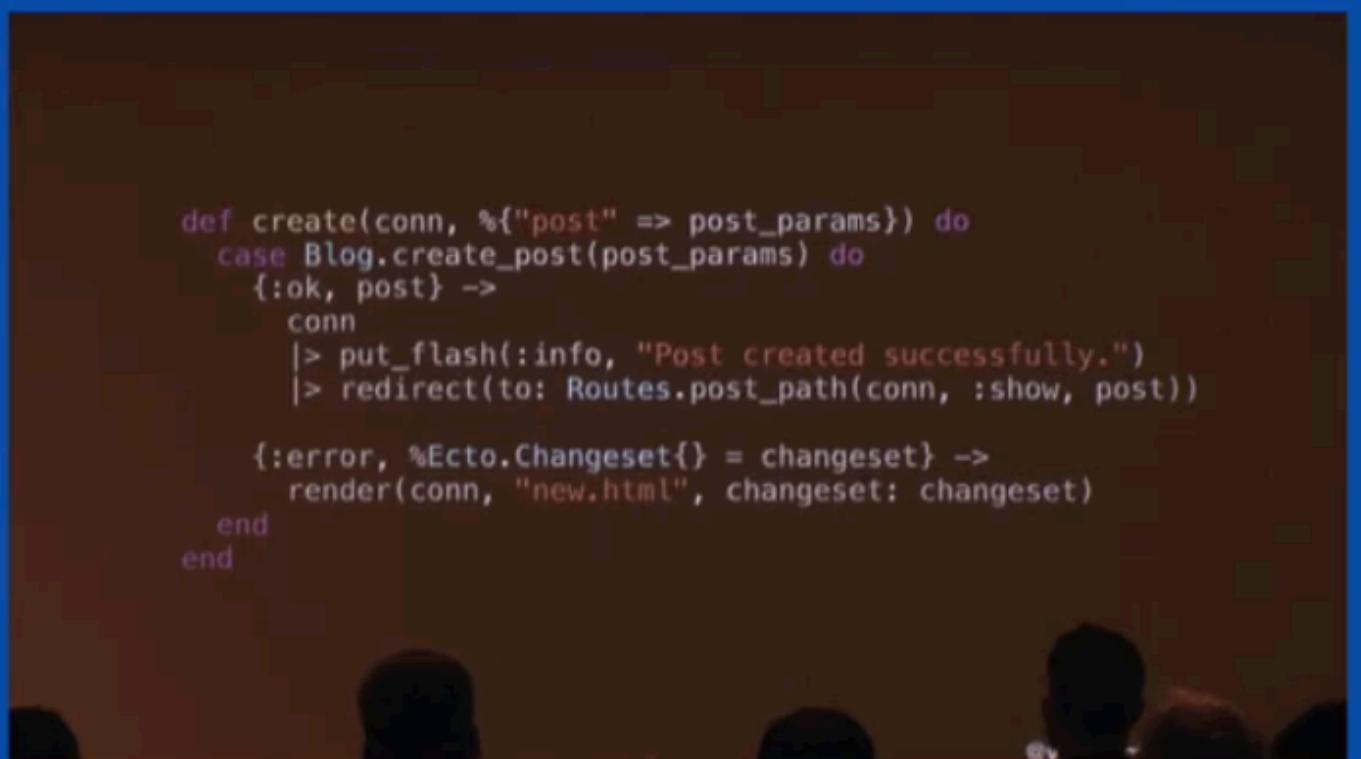
Rails vs. Hanami vs. Phoenix.

```
lib
  my_blog
    blog
      blog.ex
      post.ex
    my_blog_web
      controllers
        post_controller.ex
      templates
        post
          edit.html.eex
          form.html.eex
          index.html.eex
          new.html.eex
          show.html.eex
        views
          post_view.ex
  priv
    repo
      migrations
        20190126174858_create_posts.exs
  test
    my_blog
      blog
        blog_test.exs
      my_blog_web
        controllers
          post_controller_test.exs
```

generated 12 files 30 L



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.

A video frame showing a speaker at a podium and a projected slide about Phoenix Framework.

RubyFuza & Friends 2019

2019 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.

The slide displays the Phoenix Framework's "Welcome to Phoenix!" landing page. The page features a red bird logo, the URL "localhost:4000", and the text "A productive web framework that does not compromise speed and maintainability." Below the main content are "Resources" and "Help" sections.

A video frame showing a speaker at a podium and a projected slide. The speaker, Stefan Wintermeyer, is a man with long hair tied back, wearing a light blue shirt, speaking into a microphone. The podium has a sign that reads "2019 Sponsors". The projected slide shows a screenshot of a web browser window titled "MyBlog - Phoenix Framework" with the URL "localhost:4000/posts/new". The slide features the Phoenix Framework logo (a red bird) and the text "Phoenix Framework". Below the logo, it says "New Post". There are fields for "Title" and "Body", and a "SAVE" button. In the top right corner of the slide, there is a "Get Started" link. The slide is presented against a dark background with the "RubyFuza & Friends 2019" logo in the top right corner.

RubyFuza &
Friends
2019

2019 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.



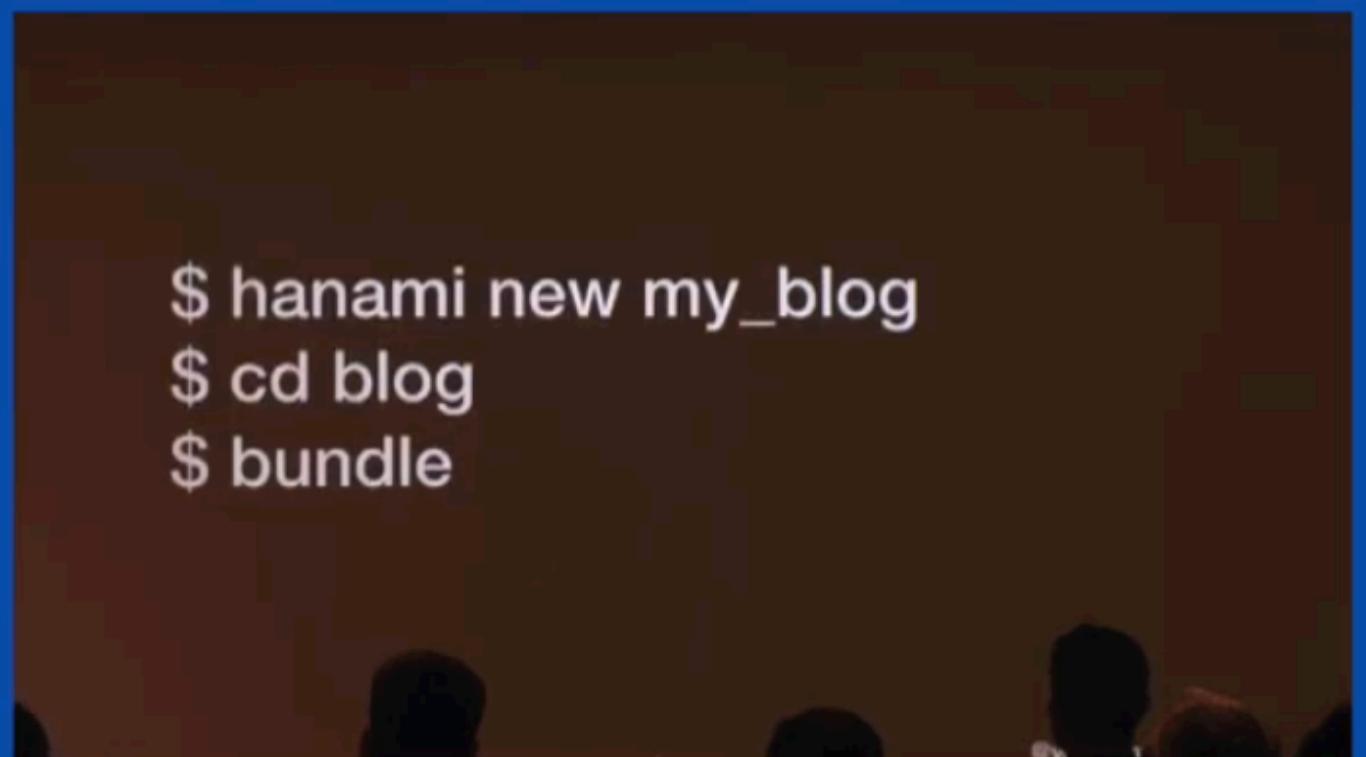
Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



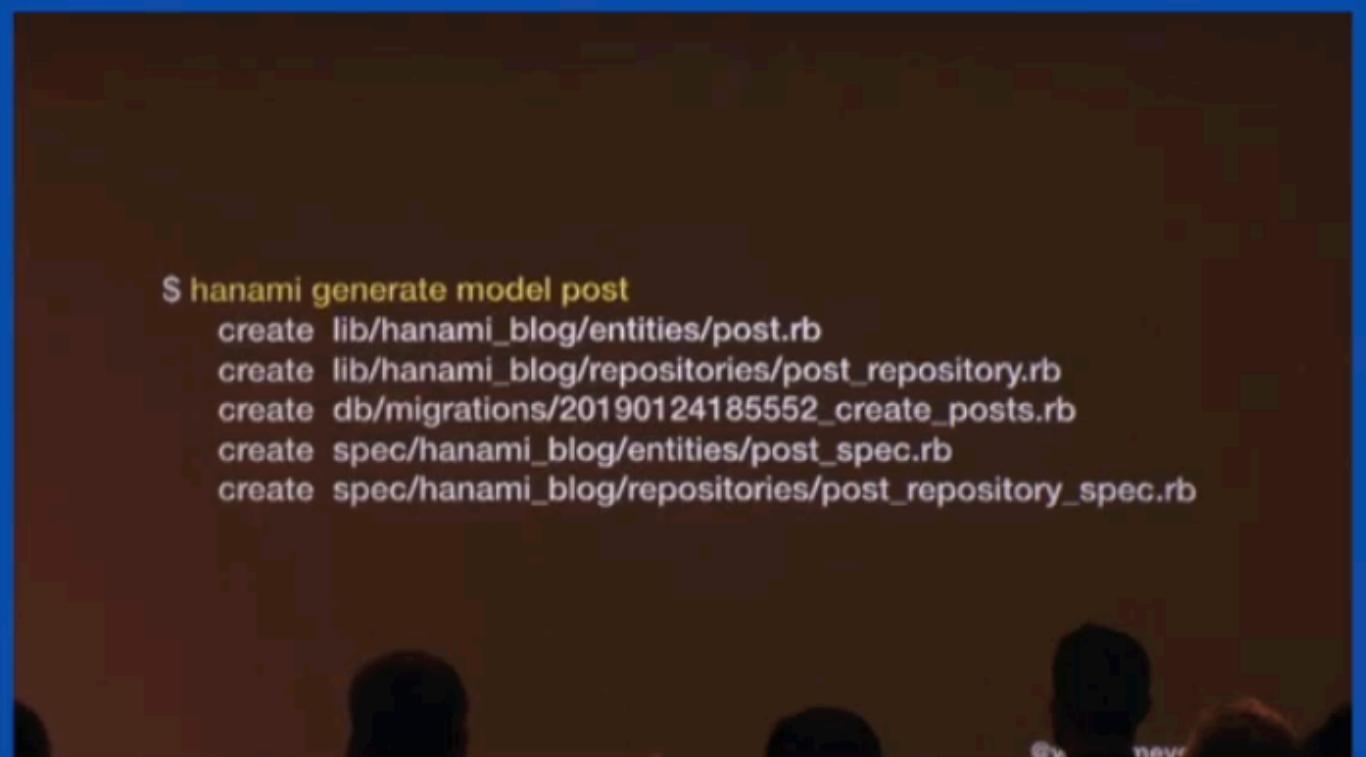
Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

```
create db/migrations/20190124185552_create_posts.rb

Hanami::Model.migration do
  change do
    create_table :posts do
      primary_key :id

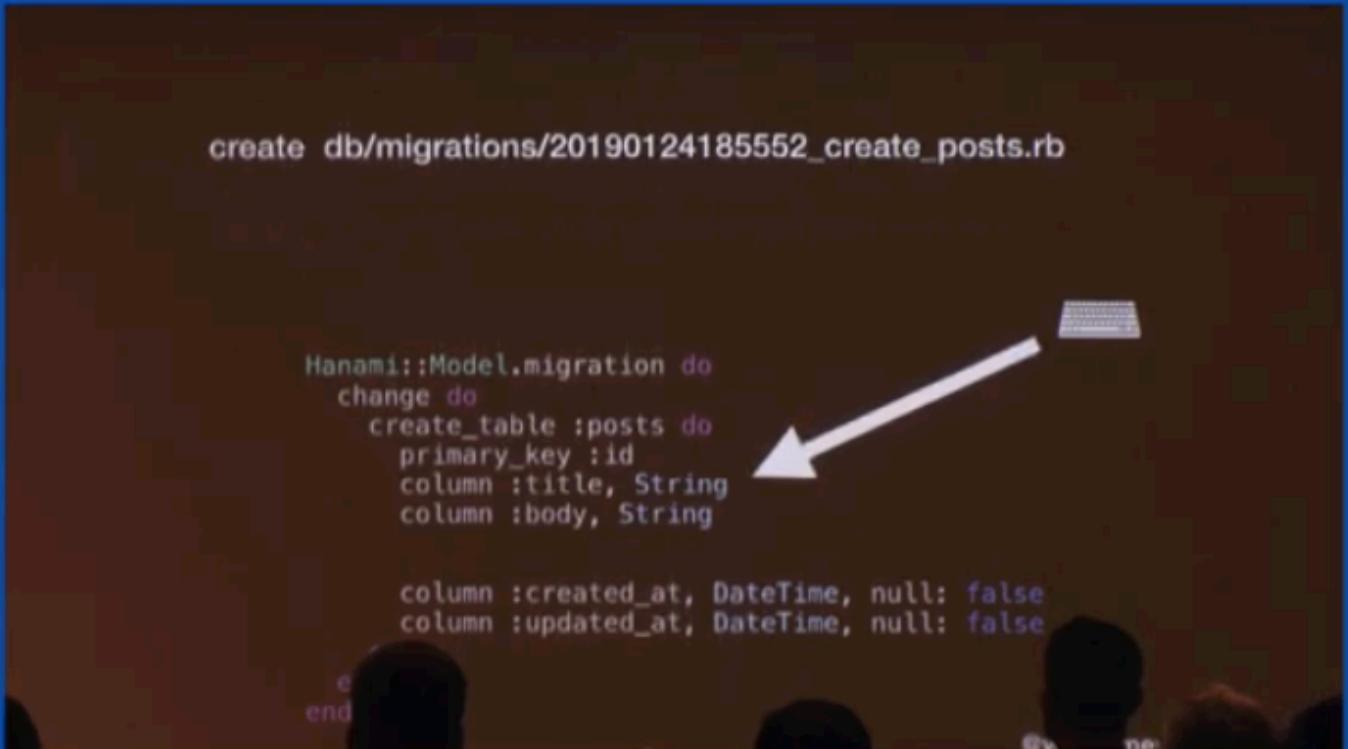
      column :created_at, DateTime, null: false
      column :updated_at, DateTime, null: false
    end
  end
end
```



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



A screenshot of a presentation slide. On the left, there is a video feed of Stefan Wintermeyer speaking. On the right, a large portion of the slide is a terminal window displaying Ruby code for a database migration:

```
create db/migrations/20190124185552_create_posts.rb

Hanami::Model.migration do
  change do
    create_table :posts do
      primary_key :id
      column :title, String
      column :body, String

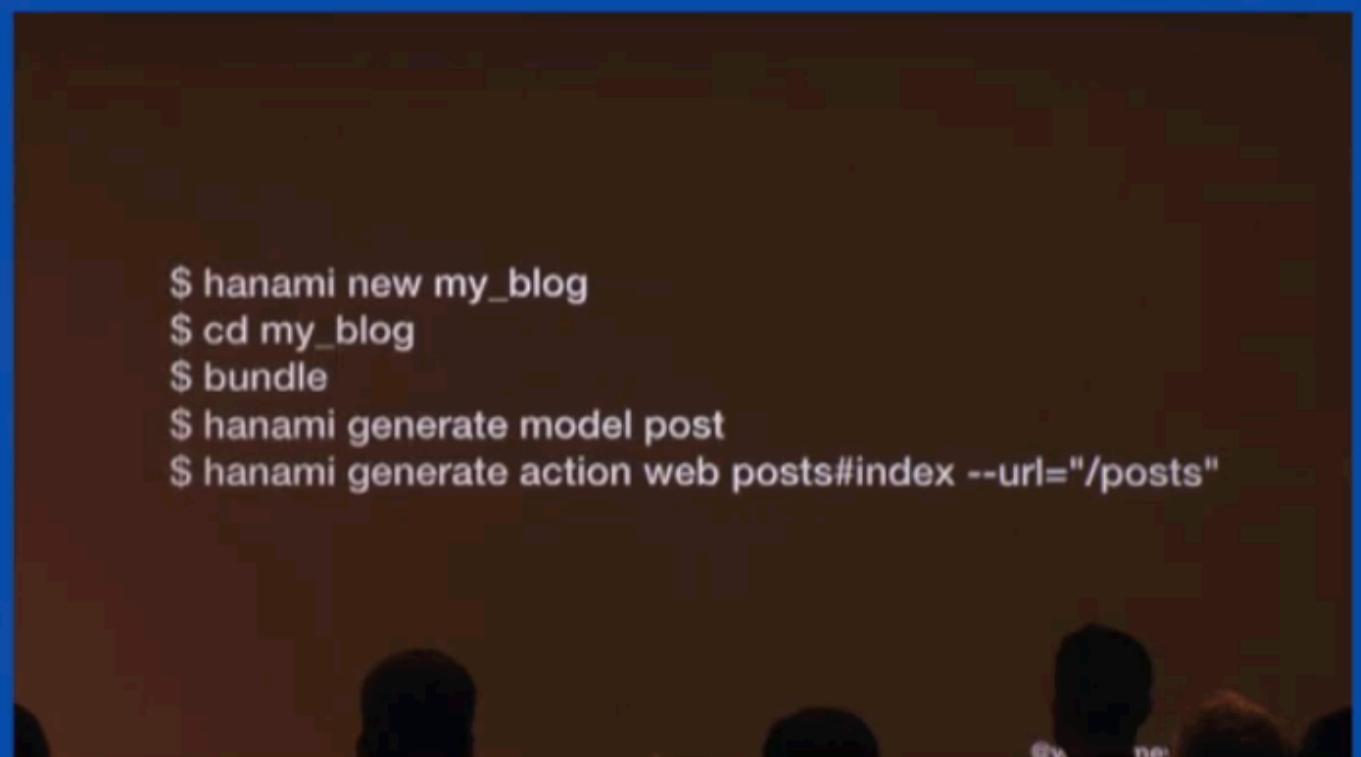
      column :created_at, DateTime, null: false
      column :updated_at, DateTime, null: false
    end
  end
end
```

An arrow points from the text "Rails vs. Hanami vs. Phoenix." below the slide to the word "Hanami" in the code.

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



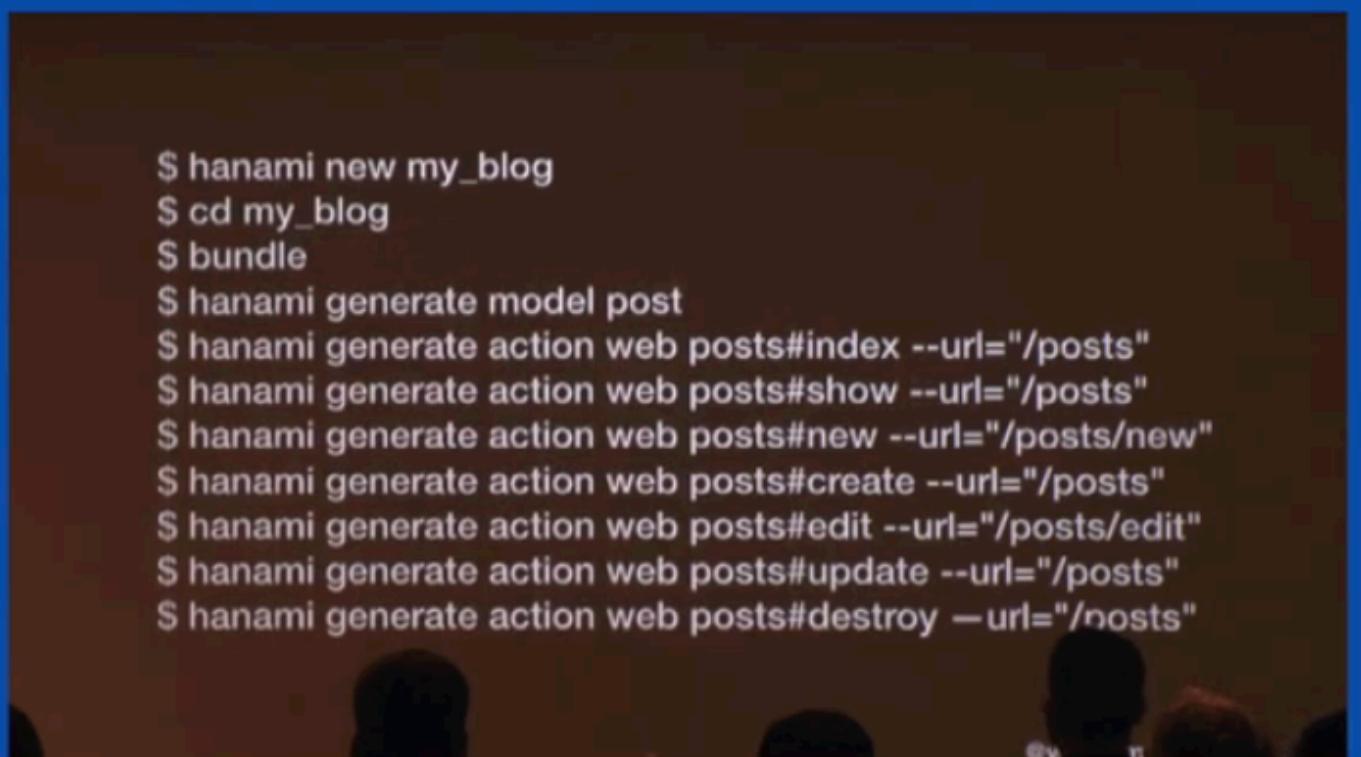
Stefan Wintermeyer

```
$ hanami generate action web posts#index --url="/posts"
  create  apps/web/controllers/posts/index.rb
  create  apps/web/views/posts/index.rb
  create  apps/web/templates/posts/index.html.erb
  create  spec/web/controllers/posts/index_spec.rb
  create  spec/web/views/posts/index_spec.rb
  insert  apps/web/config/routes.rb
```

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

RubyFuza & Friends 2019

2019 Sponsors

Rails vs. Hanami vs. Phoenix.

```
apps
  └── web
    ├── assets
    ├── config
    └── routes.rb
  ├── controllers
  │   └── posts
  │       ├── create.rb
  │       ├── destroy.rb
  │       ├── edit.rb
  │       ├── index.rb
  │       ├── new.rb
  │       ├── show.rb
  │       └── update.rb
  ├── templates
  │   └── posts
  │       ├── create.html.erb
  │       ├── destroy.html.erb
  │       ├── edit.html.erb
  │       ├── index.html.erb
  │       ├── new.html.erb
  │       ├── show.html.erb
  │       └── update.html.erb
  └── views
      └── posts
          ├── cr
          ├── de
          ├── ed
          └── in
```

```
new.rb
show.rb
update.rb
views
└── posts
    ├── create_spec.rb
    ├── destroy_spec.rb
    ├── edit_spec.rb
    ├── index_spec.rb
    ├── new_spec.rb
    ├── show_spec.rb
    └── update_spec.rb
config
db
└── migrations
    └── 20190125105614_create_posts.rb
lib
└── blog
    ├── entities
    │   └── post.rb
    ├── mailers
    └── repositories
        └── post_repository.rb
spec
└── blog
    ├── entities
    │   └── post_spec.rb
    └── repositories
        └── post_repository_spec.rb
web
└── controllers
    └── posts
        ├── create_spec.rb
        ├── destroy_spec.rb
        ├── edit_spec.rb
        ├── index_spec.rb
        ├── new_spec.rb
        └── show_spec.rb
```

RubyFuza & Friends 2019

create apps/web/controllers/posts/index.rb

index.rb — blog

```
index.rb -- blog
index.rb
apps > web > controllers > posts > index.rb
1 module Web
2   module Controllers
3     module Posts
4       class Index
5         include Web::Action
6
7         def call(params)
8           end
9         end
10      end
11    end
12  end
13
```

EXPLORER

- OPEN EDITORS
- index.rb apps/web/cont...

BLOG

- assets
- config
- controllers
- posts
- create.rb
- destroy.rb
- edit.rb

OUTLINE

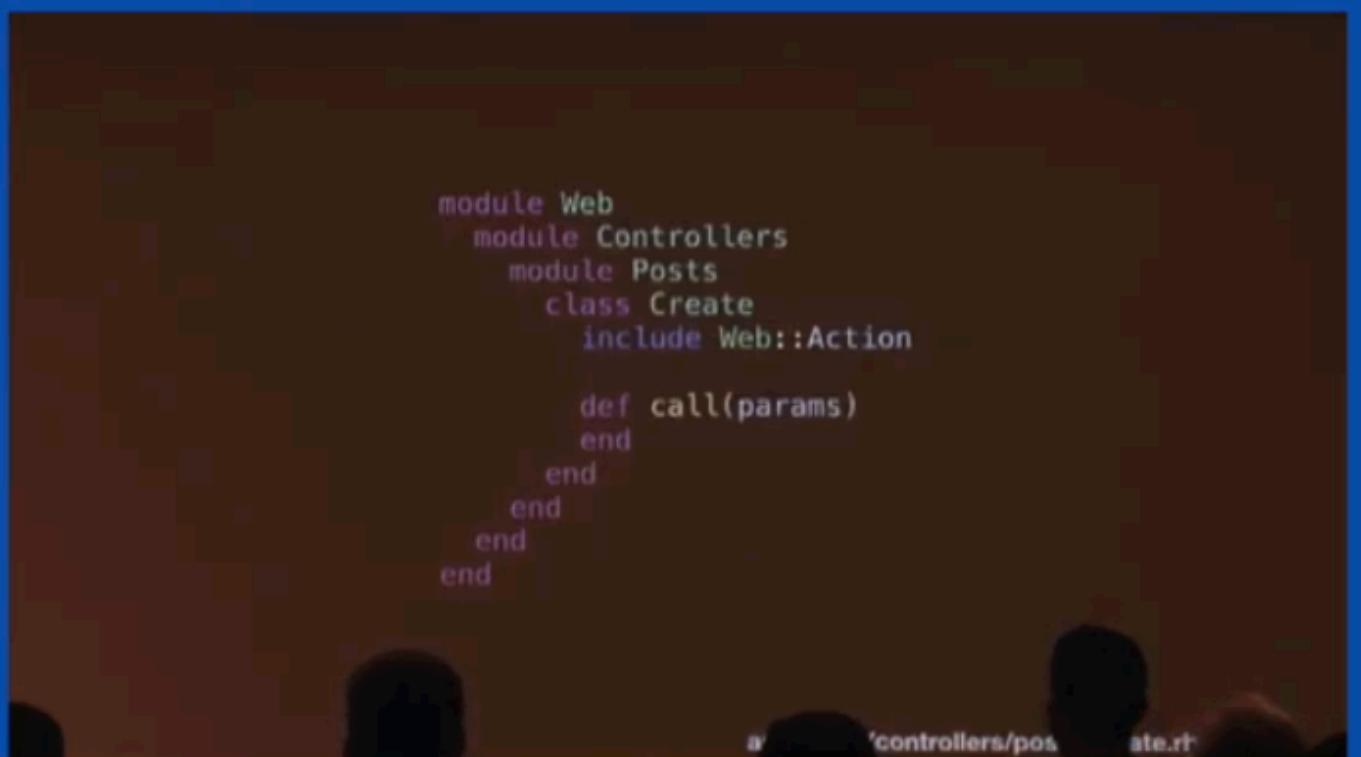
2019 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



```
module Web
  module Controllers
    module Posts
      class Create
        include Web::Action

        def call(params)
        end
      end
    end
  end
end
```

Rails vs. Hanami vs. Phoenix.



2019 Sponsors

Stefan Wintermeyer

```
$ hanami new my_blog
$ cd my_blog
$ bundle
$ hanami generate model post
$ hanami generate action web posts#index --url="/posts"
$ hanami generate action web posts#show --url="/posts"
$ hanami generate action web posts#new --url="/posts/new"
$ hanami generate action web posts#create --url="/posts"
$ hanami generate action web posts#edit --url="/posts/edit"
$ hanami generate action web posts#update --url="/posts"
$ hanami generate action web posts#destroy --url="/posts"
$ hanami db create
$ hanami db migrate
# put A LOT of Ruby code in the just generated files 😊
$ hanami server

# Open http://localhost:2300
```

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

The 15 Minute Blog

	Time (hh:mm)	generated LoC
Ruby on Rails	00:01	631
Phoenix Framework	00:02	430
Hanami	03:30	501

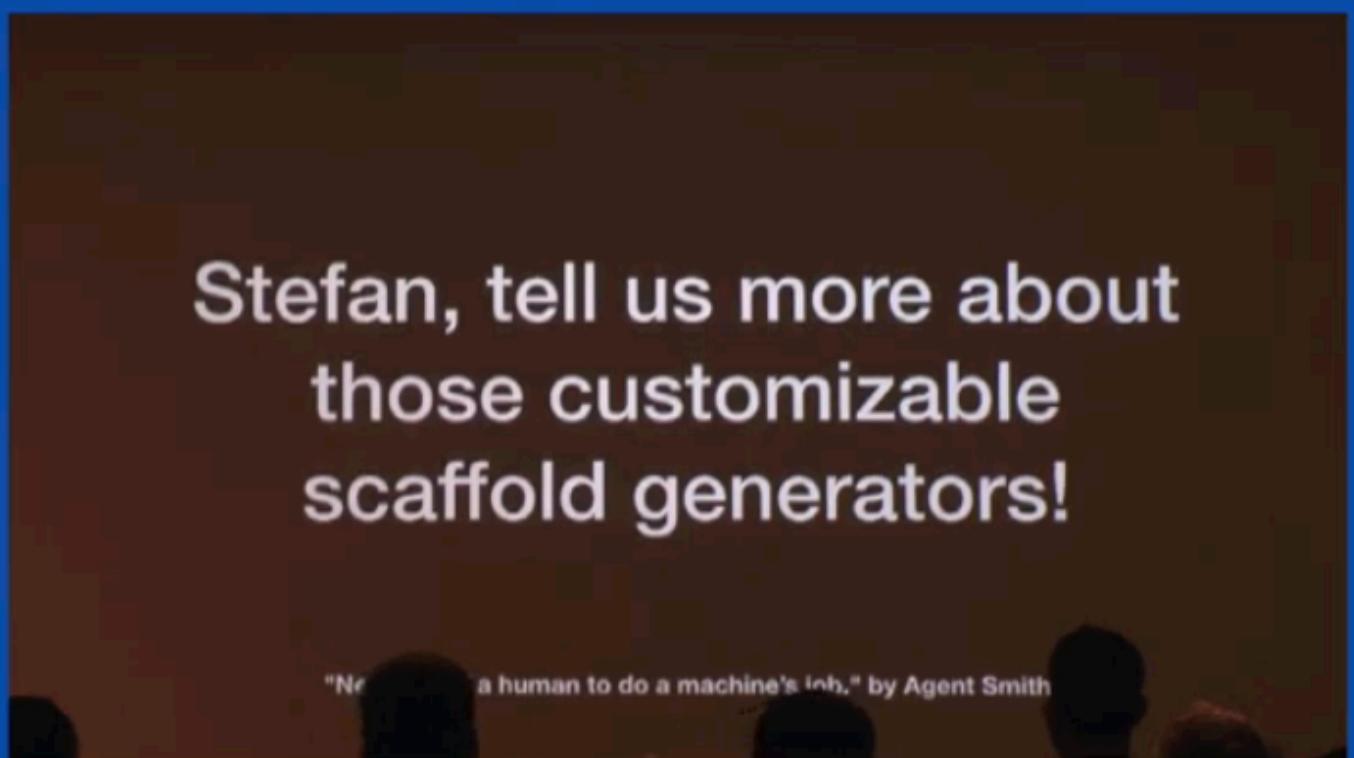
Rails vs. Hanami vs. Phoenix.



RubyFuza &
Friends
2019



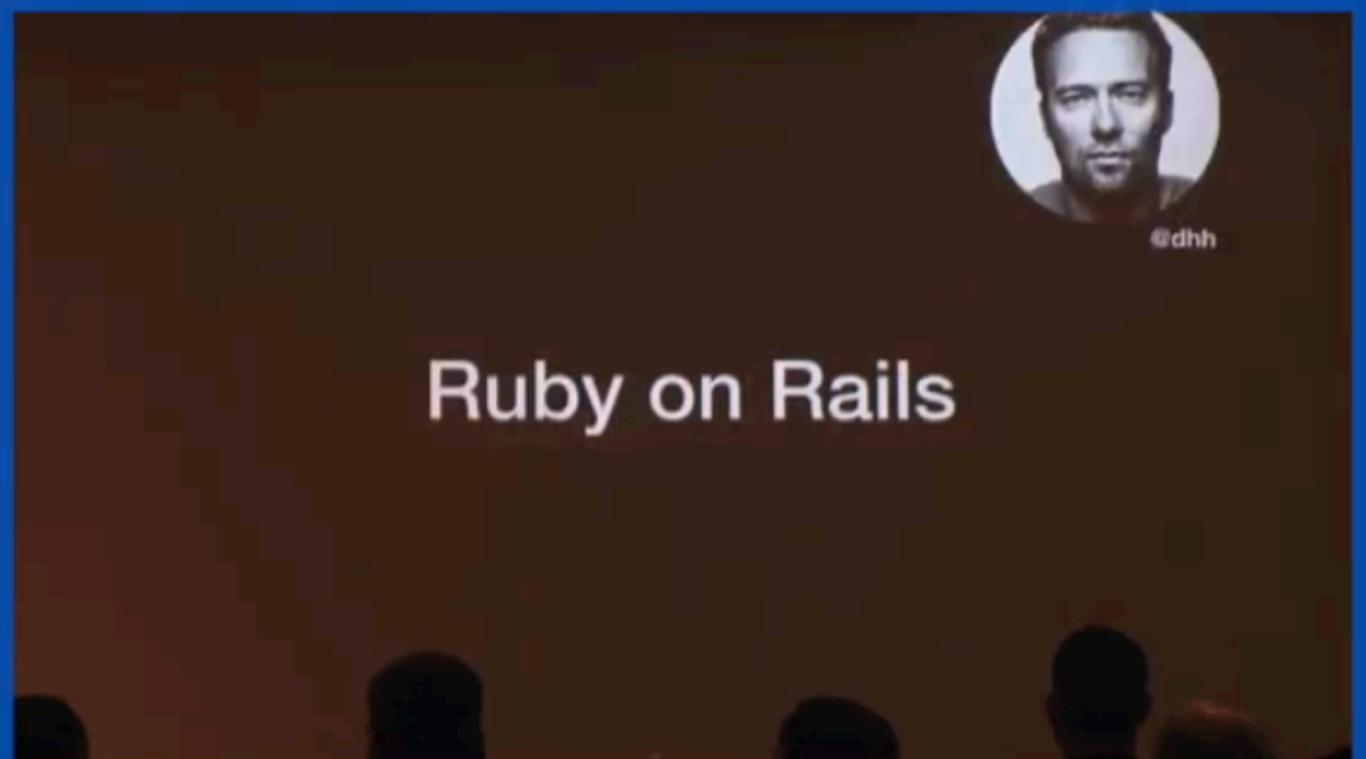
Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer





Example:
model generator

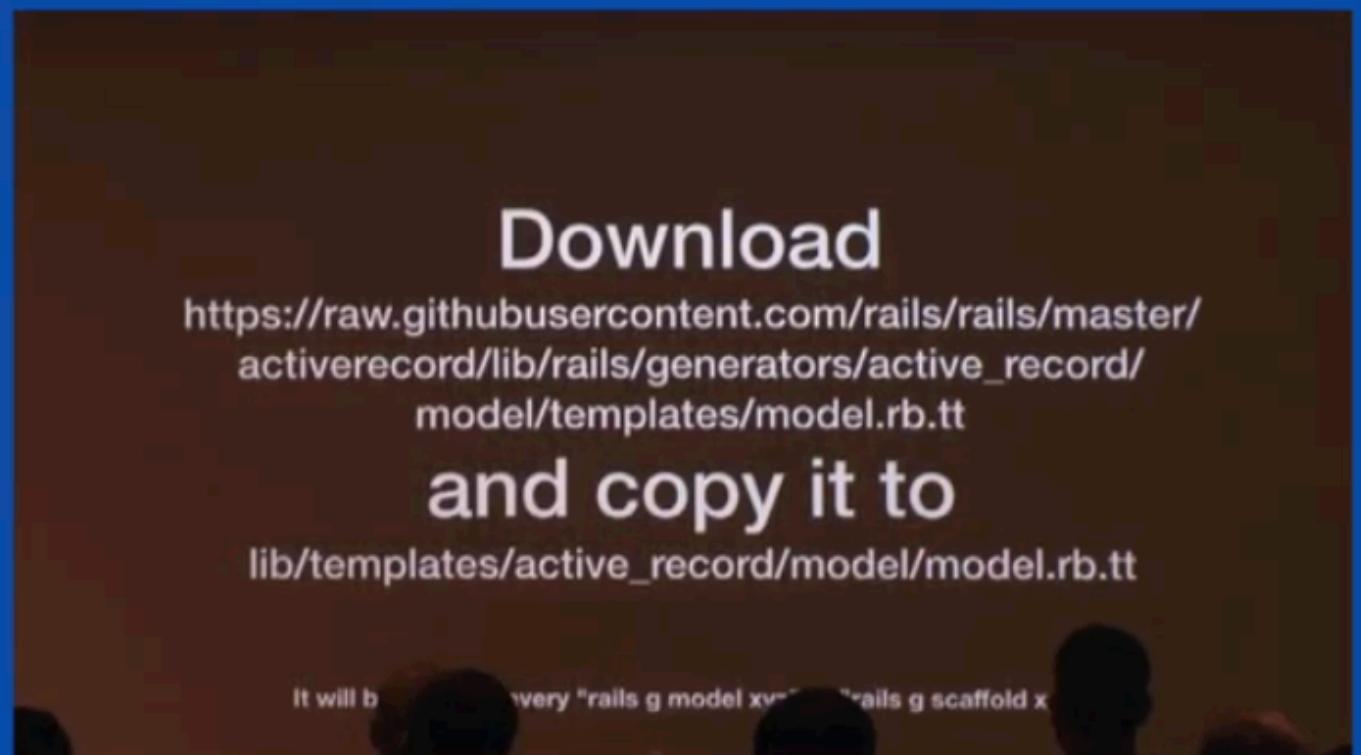
19 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

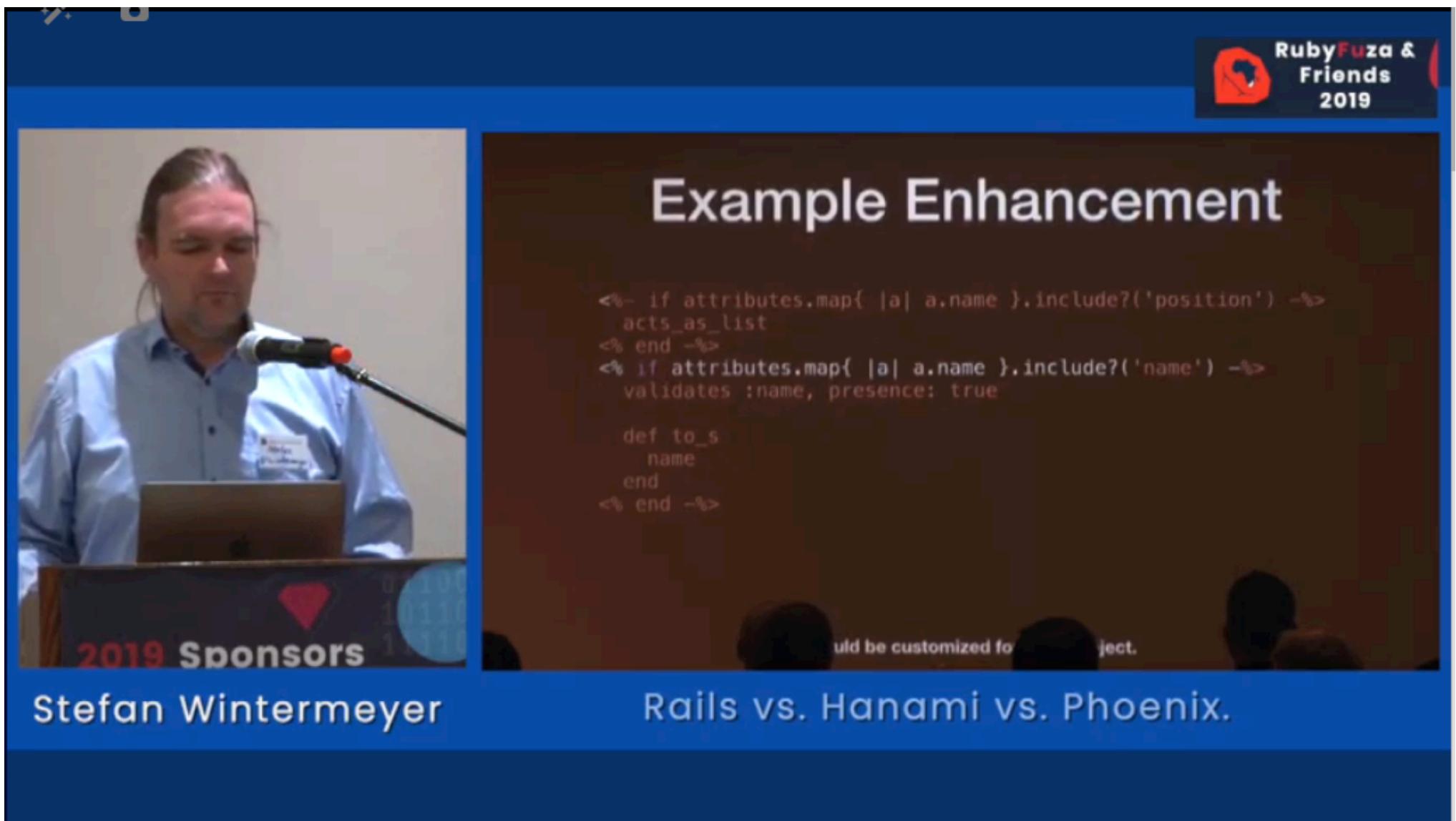


A presentation slide with a dark background featuring a pixelated pattern. The slide contains the following text:

Download
https://raw.githubusercontent.com/rails/rails/master/activerecord/lib/rails/generators/active_record/model/templates/model.rb.tt
and copy it to
`lib/templates/active_record/model/model.rb.tt`

At the bottom of the slide, there is a line of text that appears to be cut off: "It will be replaced by every "rails g model xyz" "rails g scaffold xyz".

Rails vs. Hanami vs. Phoenix.



RubyFuza & Friends 2019

Example Enhancement

```
<% if attributes.map{ |a| a.name }.include?('position') -%>
  acts_as_list
<% end -%>
<% if attributes.map{ |a| a.name }.include?('name') -%>
  validates :name, presence: true

  def to_s
    name
  end
<% end -%>
```

2019 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

Example Enhancement

```
<% if attributes.map{ |a| a.name }.include?('position') -%>
  acts_as_list
<% end -%>
<% if attributes.map{ |a| a.name }.include?('name') -%>
  validates :name, presence: true

  def to_s
    name
  end
<% end -%>
```

ould be customized for your project.

Rails vs. Hanami vs. Phoenix.

RubyFuza & Friends 2019

019 Sponsors

Stefan Wintermeyer

Rails vs. Hanami vs. Phoenix.

```
lib/templates/
└── active_record
    └── model
        └── model.rb.tt
└── erb
    └── scaffold
        ├── _form.html.erb.tt
        ├── edit.html.erb.tt
        ├── index.html.erb.tt
        ├── new.html.erb.tt
        └── show.html.erb.tt
└── rails
    └── scaffold_controller
        └── controller.rb.tt
```



Stefan Wintermeyer





Stefan Wintermeyer

The same concept as with
Rails. Just done with Elixir.

Rails vs. Hanami vs. Phoenix.

A composite image featuring a speaker at a podium on the left and a GitHub pull request interface on the right.

Speaker: Stefan Wintermeyer, wearing a blue shirt and a name tag, speaking into a microphone at a podium. The podium has a "2019 Sponsors" banner.

GitHub Pull Request: A screenshot of a GitHub pull request page for the `phoenixframework / phoenix` repository. The branch is `master`. The pull request title is `chore(formatter): Ensure that code is formatted properly when genera...`. The pull request has 22 issues and 7 pull requests. The commit message includes:

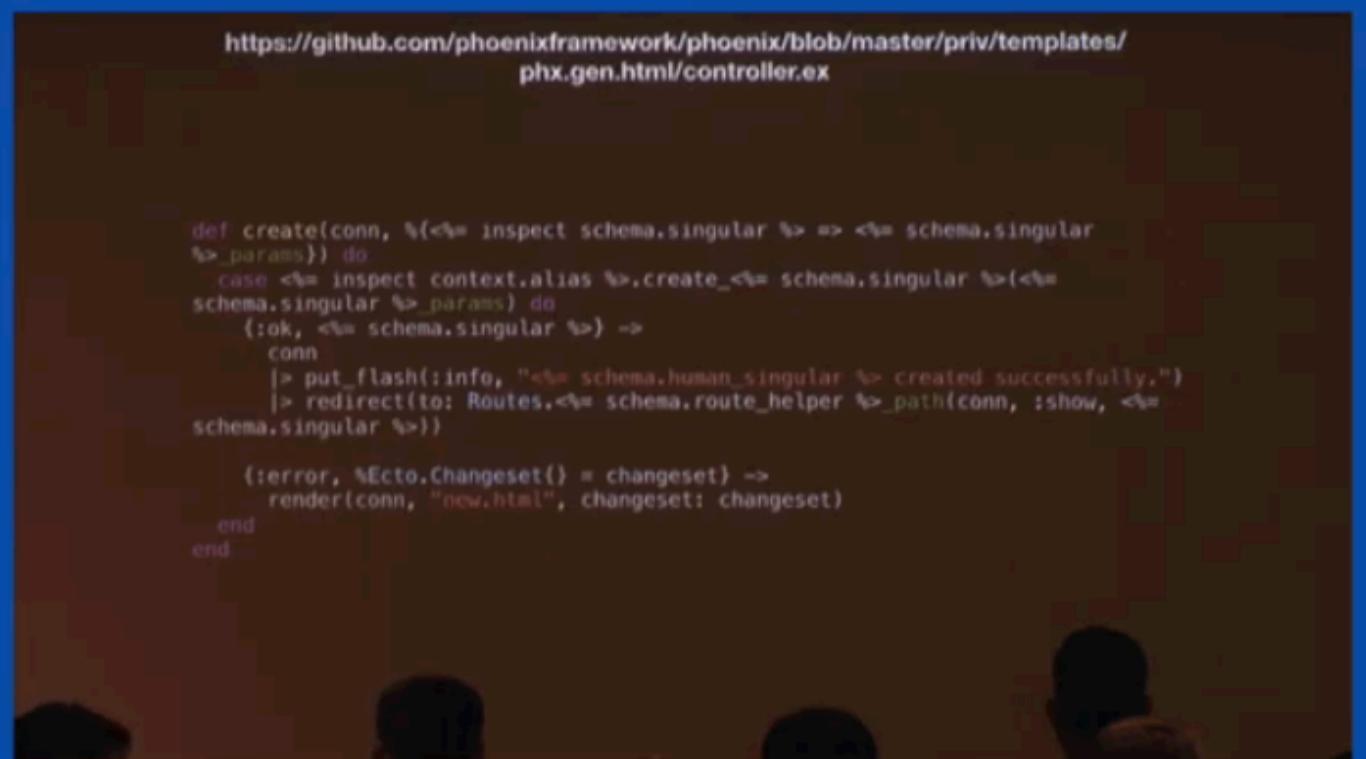
- chore(formatter): Ensure that code is formatted properly when genera...
- chore(formatter): Ensure that code is formatted properly when genera...
- Adjust level headings to form proper document outline
- Remove classes from forms, migrate to milligram
- Adjust level headings to form proper document outline
- Adjust level headings to form proper document outline
- Adjust level headings to form proper document outline
- Add --web namespace option to gen.html/json. Closes #2192

Text Labels:

- Stefan Wintermeyer** (under the speaker photo)
- Rails vs. Hanami vs. Phoenix.** (under the GitHub screenshot)



Stefan Wintermeyer



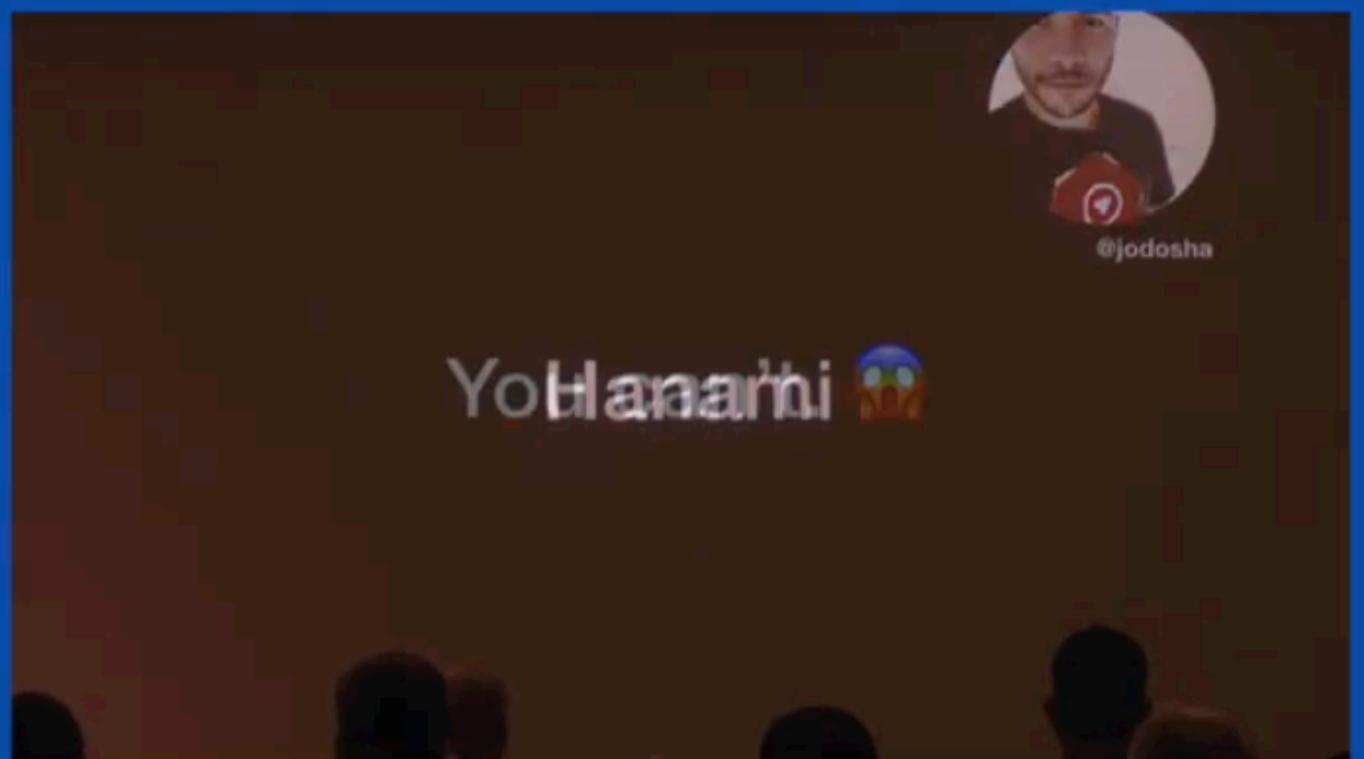
<https://github.com/phoenixframework/phoenix/blob/master/priv/templates/phx.gen.html/controller.ex>

```
def create(conn, %<%= inspect schema.singular %> => <%= schema.singular %>_params) do
  case <%= inspect context.alias %>,create_<%= schema.singular %>(<%= schema.singular %>_params) do
    {:ok, <%= schema.singular %>} =>
      conn
      |> put_flash(:info, "<%= schema.human_singular %> created successfully.")
      |> redirect(to: Routes.<%= schema.route_helper %>_path(conn, :show, <%= schema.singular %>))
    {:error, %Ecto.Changeset{} = changeset} =>
      render(conn, "new.html", changeset: changeset)
  end
end
```

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

How big is the community/visibility?

How easy can I get help?

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

	Stackoverflow Questions	GitHub Contributors
Ruby on Rails	302,236	3.742
Phoenix Framework	3,030	693
Hanami	60	140

Rails vs. Hanami vs. Phoenix.



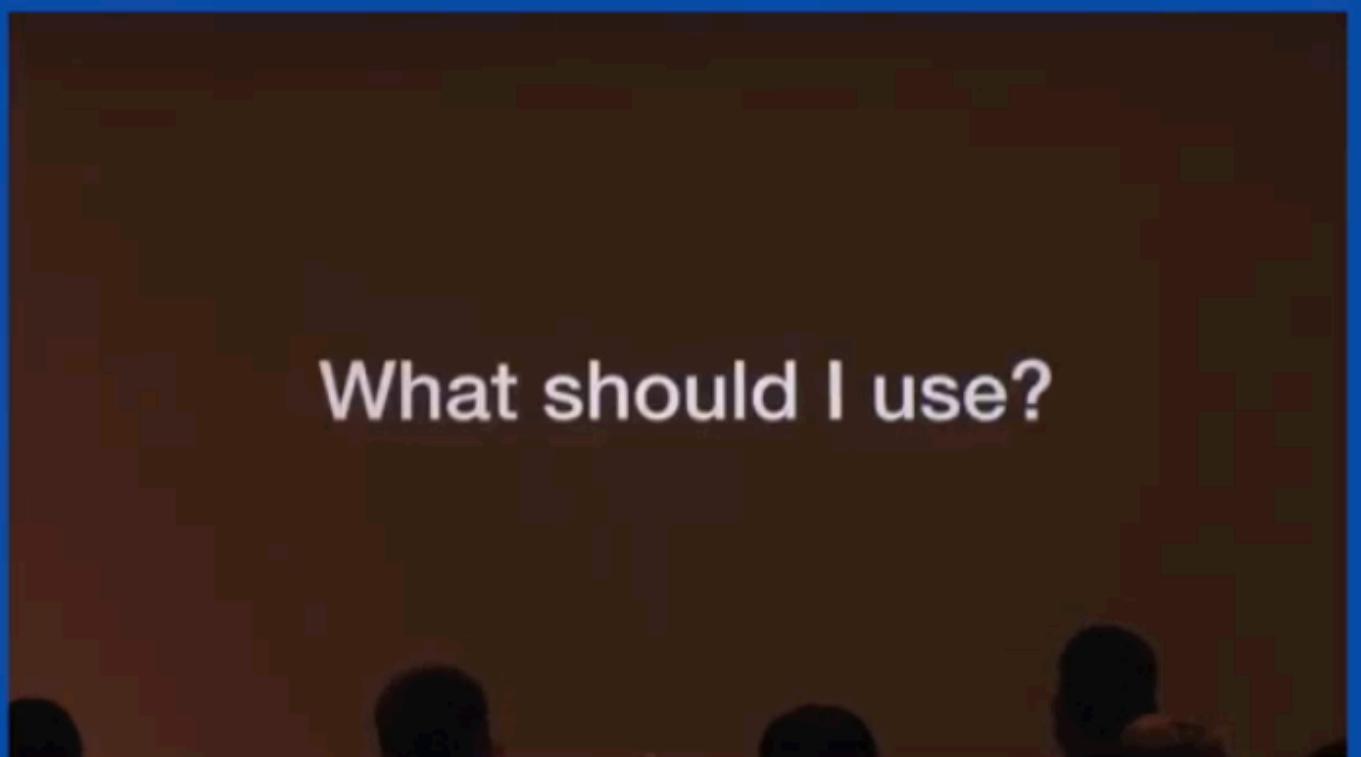
Stefan Wintermeyer

Twitter Followers	
@dhh	324 K
@chris_mccord	13.6 K
@jodosha	2.672

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

Use Phoenix if speed is
paramount.

Technically it is the best choice!

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

Use Phoenix if your team
doesn't have (much?)
Ruby knowledge yet.

Technically it is the best choice!

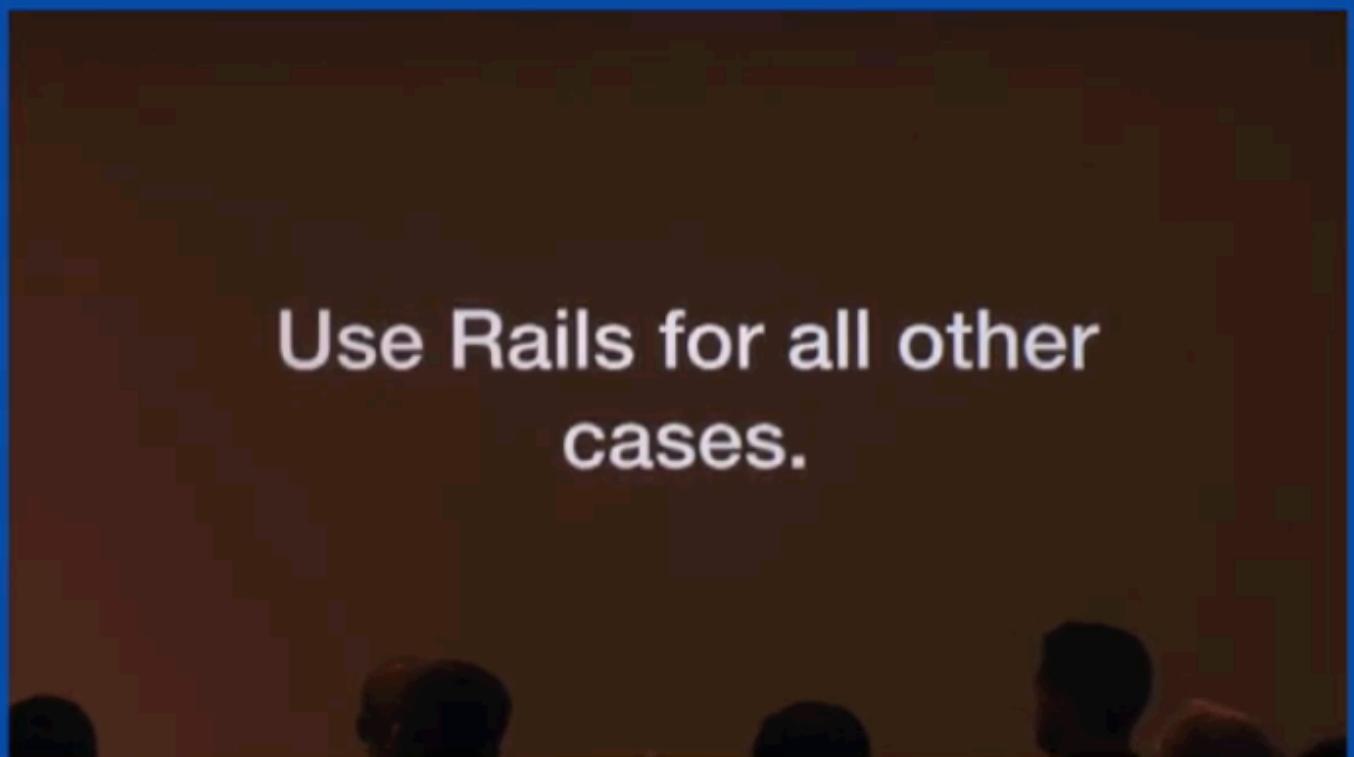
Rails vs. Hanami vs. Phoenix.



RubyFuza &
Friends
2019



Stefan Wintermeyer



Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

Don't use Hanami (for now).

Let's see what the future brings.

"Nobody / needs / and for buying IBM".gsub(/buying IBM/, 'using

Rails vs. Hanami vs. Phoenix.



Stefan Wintermeyer

Thank you!

Stefan Wintermeyer <sw@wintermeyer-consulting.de>

www.wintermeyer-consulting.de

Rails vs. Hanami vs. Phoenix.



RubyFuza & Friends 2019