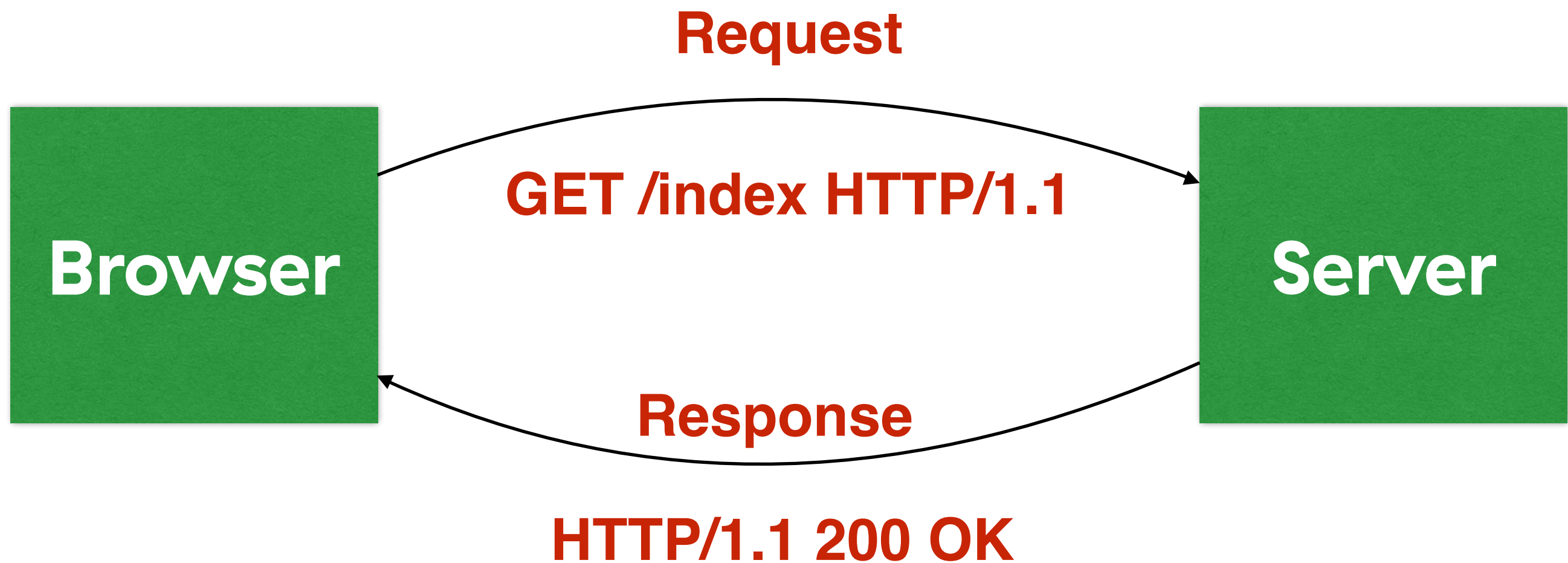


Intro to Ruby on Rails



How does the WEB work?

Request/Response cycle



Request methods (HTTP Verbs)

- ✦ GET (Read)
- ✦ POST (Create)
- ✦ PUT/PATCH (Update)
- ✦ DELETE (Destroy)



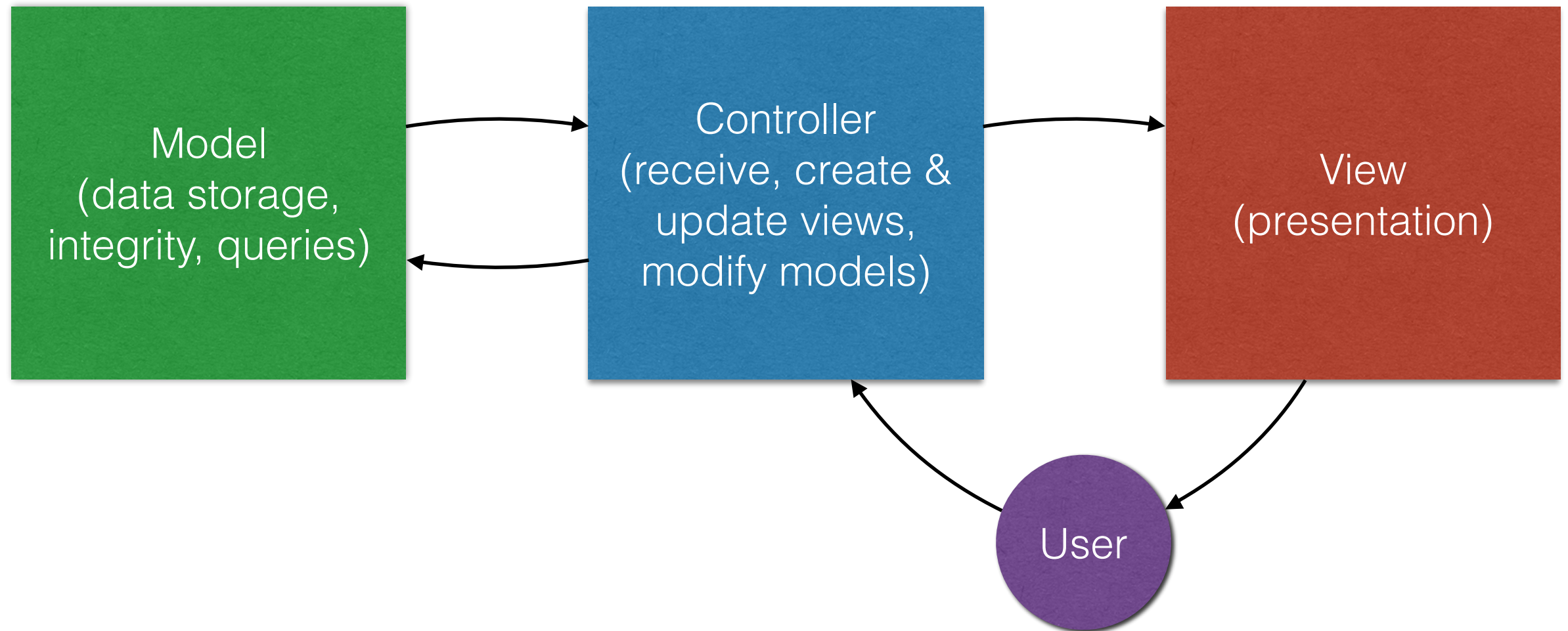
Response Status codes

- ✦ 2xx - all is well (“here you go”)
- ✦ 3xx - resource moved (“go away”)
- ✦ 4xx - access problem (“you fucked up”)
- ✦ 5xx - server error (“i fucked up”)

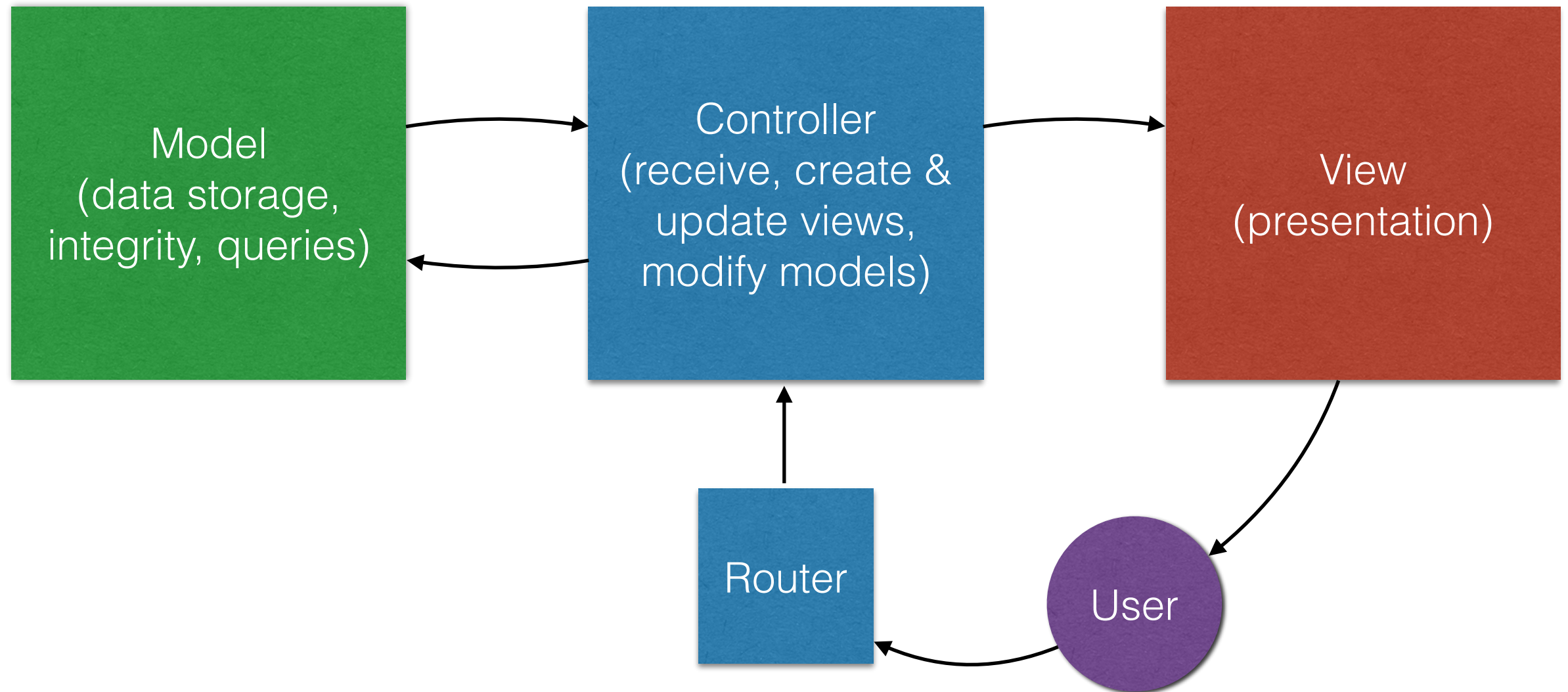
MVC

Model View Controller

Essence of MVC



Essence of Rails MVC



Project: Reddit

Shema

User

email: string
name: string
password: string

Post

belongs_to :user
belongs_to :subreddit

title: string
image: file

Vote

belongs_to :post
belongs_to :user

direction: integer

Subscribe

name: string

Favorites

belongs_to :subreddit
belongs_to :user

Comment

belongs_to :post
belongs_to :user

content: string



rails new

Installing rails

```
$ gem install bundler
```

```
$ gem install rails
```

```
$ rails new reddit_01
```

```
$ cd reddit01
```

```
$ rails server
```



Folder structure

1. Create a folder named `src`

2. Create a folder named `main`
























3. Create a folder named `test`

4. Create a folder named `docs`

5. Create a folder named `assets`

6. Create a folder named `config`

7. Create a folder named `scripts`

- ▼  reddit_01
 - ▼  app
 - ›  assets
 - ›  controllers
 - ›  helpers
 - ›  mailers
 - ›  models
 - ›  views
 - ›  bin
 - ›  config
 - ›  db
 - ›  lib
 - ›  log
 - ›  public
 - ›  test
 - ›  tmp
 - ›  vendor
 -  .gitignore
 -  config.ru
 -  Gemfile
 -  Gemfile.lock
 -  Rakefile
 -  README.rdoc



Rails philosophy

DRY (“Don’t Repeat Yourself”)

DRY is a principle of software development which states that "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system." By not writing the same information over and over again, our code is more maintainable, more extensible, and less buggy.



Convention over configuration

Rails has opinions about the best way to do many things in a web application, and defaults to this set of conventions, rather than require that you specify every minutiae through endless configuration files.



REST

Representational State Transfer is a software architecture style consisting of guidelines and best practices for creating scalable web services.



Bundler

Ruby dependency manager

Adding new gems

```
gem 'quiet_assets'
```

```
$ bundle install
```

Generators

Rails generators

```
$ rails generate
```

```
$ rails generate controller welcome index
```



Routes

config/routes.rb

```
Rails.application.routes.draw do
  get 'homepage' => 'welcome#index'

  root 'welcome#index'
end
```



Controllers

app/controllers/welcome_controller.rb

```
class WelcomeController < ApplicationController
  def index
  end
end
```



Views

app/views/welcome/index.html.erb

<h1>Welcome#index</h1>

<p>Find me in app/views/welcome/index.html.erb</p>

Rails magic

100%

app/controllers/welcome_controller.rb

```
class WelcomeController < ApplicationController
  def index
    @name = 'Rails'
  end
end
```



app/views/welcome/index.html.erb

```
<h1>Welcome to <%= @name %> course</h1>
```

rake routes

app/views/welcome/index.html.erb

```
<%= link_to 'Link na pocetnu stranicu', homepage_path %>
```

