

Rails Engines

Dimitris Zorbas - Athens Ruby Meetup#22



github.com/zorbash



[@_zorbash](https://twitter.com/_zorbash)



What is an Engine?

Engines can be considered **miniature** applications that provide functionality to their host **applications**

guides.rubyonrails.org/engines



An Engine

is_a?

↳ **Gem**

A Gem may contain many Engines

↳ **Plugin**

A plugin is not necessarily an Engine

↳ **Railtie**

Every major Rails component is one



Engine < Railtie

Is practically a **Railtie**

```
require 'rails/railtie'  
require 'rails/engine/railties'  
  
module Rails  
  class Engine < Railtie  
    autoload :Configuration, "rails/engine/configuration"  
  end  
end
```

railties/lib/rails/engine.rb



Engine < Raittie

With some initializers already present

```
Rails::Engine.initializers.map(&:name)
# =>

[:set_load_path, :set_autoload_paths, :add_routing_paths,
 :add_locales, :add_view_paths, :load_environment_config,
 :append_assets_path, :prepend_helpers_path,
 :load_config_initializers, :engined_blank_point]
```



Application < Engine

Is practically an **Engine** with app bootstrapping

```
require 'rails/engine'

module Rails
  class Application < Engine
    autoload :Bootstrap, "rails/application/bootstrap"
    autoload :Configuration, "rails/application/configuration"
  end
end
```

railties/lib/rails/application.rb



Popular **Engines**

- ◆ **devise**
- ◆ **refinerycms**
- ◆ **spree**
- ◆ **kaminari**
- ◆ **doorkeeper**



Why create an Engine?



Separation of concerns



Tested independently



Can be reused



Obligatory GIF



Caveats of an Engine?



Slow prototyping velocity



Dependency management



More repositories



Requires host app for dev / test



Adequate Monolith™

Using Engines for sane application growth



Adequate Monolith™

Step 1: Get an Architectural overview



Monolith

You



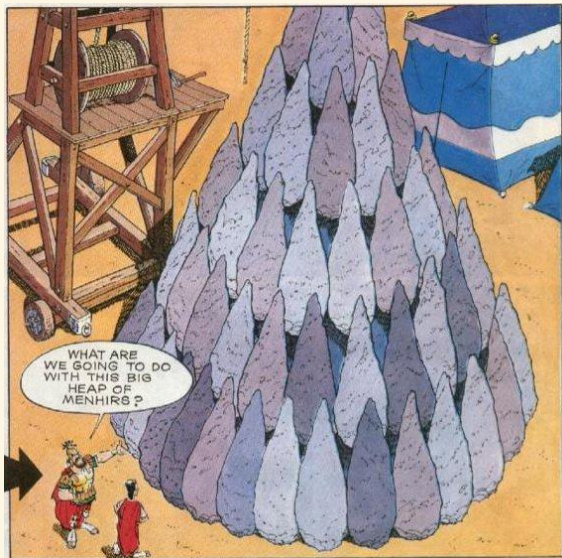
Adequate Monolith™

Step 2: Extract smaller components



Adequate Monolith™

Step 3: Compose



Engineering.initialize!

Creating an Engine

```
rails plugin new ENGINE_NAME --full
```



Engine Structure

Creating an Engine

```
▼ app/  
  ▸ {assets,controllers,helpers,mailers,models,views}/  
▼ config/  
  routes.rb  
▼ lib/  
  ▸ engine_name/  
  ▸ tasks/  
    engine_name.rb  
  Gemfile  
  Rakefile  
  engine_name.gemspec
```



Engine Configuration

```
module Chat
  class Engine < ::Rails::Engine
    isolate_namespace Chat # !important

    initializer 'chat.settings' do |app|
      app.config.x.chat = OpenStruct.new(room: 'newbies')
    end
  end
end
```

lib/chat/engine.rb

Regular **config/initializers/*** can also be used for configuration



Engine Routes

Setting the routes

```
Chat::Engine.routes.draw do
  root 'rooms#index'

  resources :rooms do
    resources :messages, only: %i[create]
  end
end
```

config/routes.rb



Engine Routes

Mounting from main app

```
Rails.application.routes.draw do
  mount Chat::Engine, at: '/chat'

  # other routes
end
```

config/routes.rb



Hint: From inside the **engine's** views, refer to the host app routing helpers using `main_app.resource_path`



Engine Development

Starting a console

```
cd ./spec/dummy  
./bin/rails console
```

shell

Bundling on the host app

```
gem 'engine_name', path: '../path/to/engine'
```

host Gemfile



Engines Best Practices

Controllers

Inherit from ApplicationController when it is absolutely required (eg auth logic)

Example:

```
class Chat::RoomsController < ApplicationController
  # default inheritance Chat::ApplicationController

  # actions..
end
```



Engines Best Practices

Models

For maximum reusability, have host app model name associations configurable.

Example:

```
Chat.user_class = 'User' # configurable

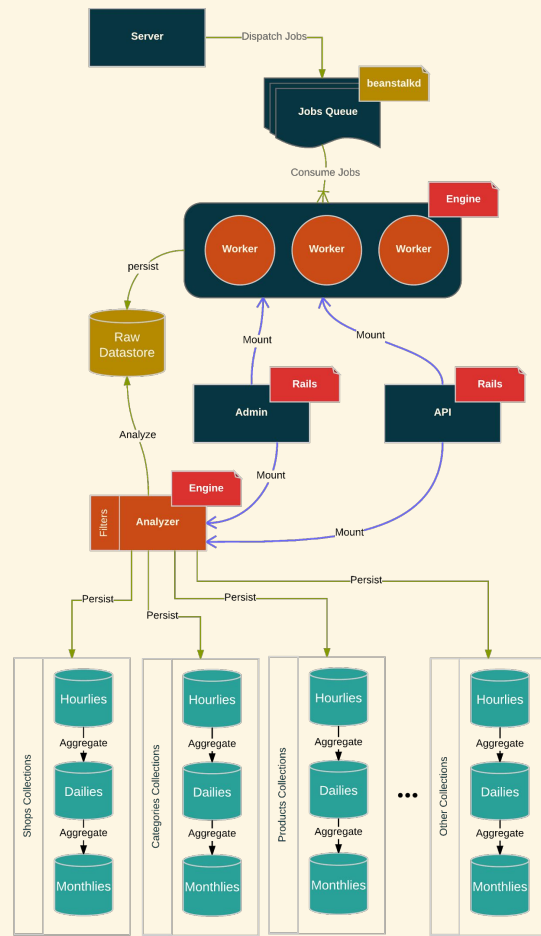
class Chat::Message < ActiveRecord::Base
  belongs_to :user, class_name: Chat.user_class
end
```



Engines on Production

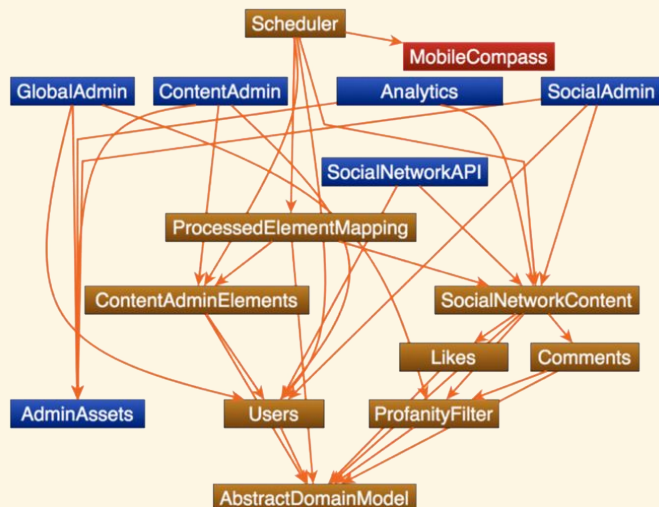
Case:  **skroutz** Analytics





Engines on Production

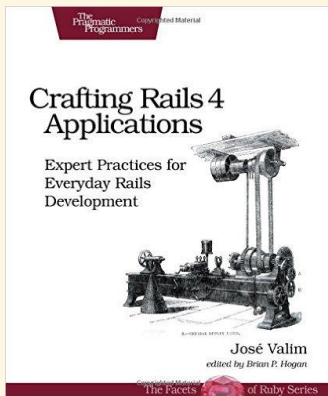
Case:  Pivotal Labs[®]



Resources

<http://guides.rubyonrails.org/engines.html>

<http://api.rubyonrails.org/classes/Rails/Engine.html>



END

Questions?

