

# Rails Engines.

Doing it **wrong**. And then **right**.

innoQ

@mrreynolds

# Why Engines?

**Extend your app with reusable models, controllers, views, helpers, routes, locales and tasks.**

**It's an app**

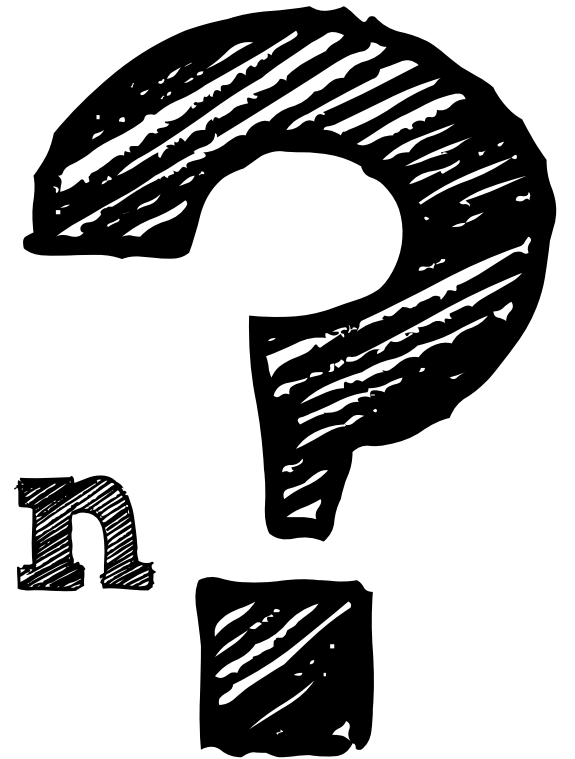
**within**

**an app.**

**Rails::Engine < Rails::Railtie**

# What's a Railtie?

- Core of the framework
- Provides hooks to extend Rails
- ActiveRecord, ActionController etc. are all Railties



What is an

Engine then



**It's just a Railtie  
with some  
defaults.**

# Typical cases

- CMS
- Admin Frontend
- Translation Frontend

# Our problem

- **Build a vocabulary and thesaurus management system**
- **Adjust and extend it for every customer without forking it**

**[github.com/innoq/iqvoc](https://github.com/innoq/iqvoc)**

# First approach

- **iQvoc as main app**
- **Vendor logic as engine**

# The problem

- Could not act as a standalone app
- Always had to be plugged into a main app

# Second approach

- Vendor logic as main app
- iQvoc as engine (and app)

# The problem

- **A Rails 3.0 Engine can not act as a standalone app by default (requires customization)**
- **No out-of-the-box support for migrations, assets etc.**



Options

**Wait for Rails 3.1**

**Just hack it.**

**What**  
**do you need**



# Act as an engine...

```
# lib/engine.rb
module Iqvoc

  class Engine < Rails::Engine
    end

end
```

**...only if we want to**

```
# config/initializers/iqvoc.rb
unless Iqvoc.const_defined?(:Application)
  require File.join(File.dirname(__FILE__), '../..../lib/engine')
end
```

```
# config/application.rb
module Iqvoc
  class Application < Rails::Application
```

# Up next: Engine tasks

```
# lib/engine.rb  
class Engine < Rails::Engine  
  
  paths.lib.tasks << "lib/engine_tasks"
```

Only available when app  
is mounted as an engine!

# What about Migrations?

```
namespace :iqvoc do
  namespace :db do

    task :migrate => :environment do
      ActiveRecord::Migration.verbose = ENV["VERBOSE"] ? ENV["VERBOSE"] == "true" : true
      path = Iqvoc::Engine.find_root_with_flag("db").join('db/migrate')
      ActiveRecord::Migrator.migrate(path, ENV["VERSION"] ? ENV["VERSION"].to_i : nil)
      Rake::Task["db:schema:dump"].invoke if ActiveRecord::Base.schema_format == :ruby
    end

  end
end
```



```
Terminal — bash — 132x44
bash
cloudseven:iqvoc_spez rglaser$ rake -T db:migrate
(in /Users/rglaser/innoc/iqvoc_spez)
rake db:migrate          # Migrate the database (options: VERSION=x, VERBOSE=false).
rake db:migrate:status   # Display status of migrations
rake iqvoc:db:migrate    # Migrate the database through scripts in db/migrate and update db/schema.rb by invoking db:schema:dump. Tar
get specific version w VERSION=x. Turn off output with VERBOSE=f...
cloudseven:iqvoc_spez rglaser$
```

# Rails 3.1

```
rake railties:copy_migrations
```

# Routes

```
Foo.application.routes.draw do
```



```
Rails.application.routes.draw do
```

# Rails 3.1

```
# Main app
Rails.application.routes.draw do
  mount Foo::Engine => "/foo"
end
```

```
# Engine
Foo::Engine.routes.draw do
  ...
```

# Rails 3.1

## Namespace isolation

```
module MyEngine
  class Foo < Rails::Engine
    isolate_namespace Foo
  end
end
```

```
# Separate routers for each Engine
foo.root_path
main_app.root_path
```

If you isolate,  
**don't forget**  
to **move** things.

```
app/controllers/foo/things_controller.rb  
app/views/foo/things/new.html.erb  
...
```

# Assets

```
task :link do
  Iqvoc.for_static_folders do |source_common_dir, target_common_dir|
    File.unlink(target_common_dir) if File.symlink?(target_common_dir) && ENV['force'] == "true"
    if !File.exists?(target_common_dir)
      puts "Linking #{source_common_dir} -> #{target_common_dir}"
      File.symlink(source_common_dir, target_common_dir)
    else
      puts "Symlink #{target_common_dir} already exists!"
    end
  end
end
end
```



# Rails 3.1

```
# ActionController::Static  
config.serve_static_assets = true
```

**or**

```
rake railties:create_symlinks
```

# **bundle in hell**

- **No support for multiple locations of a single gem**

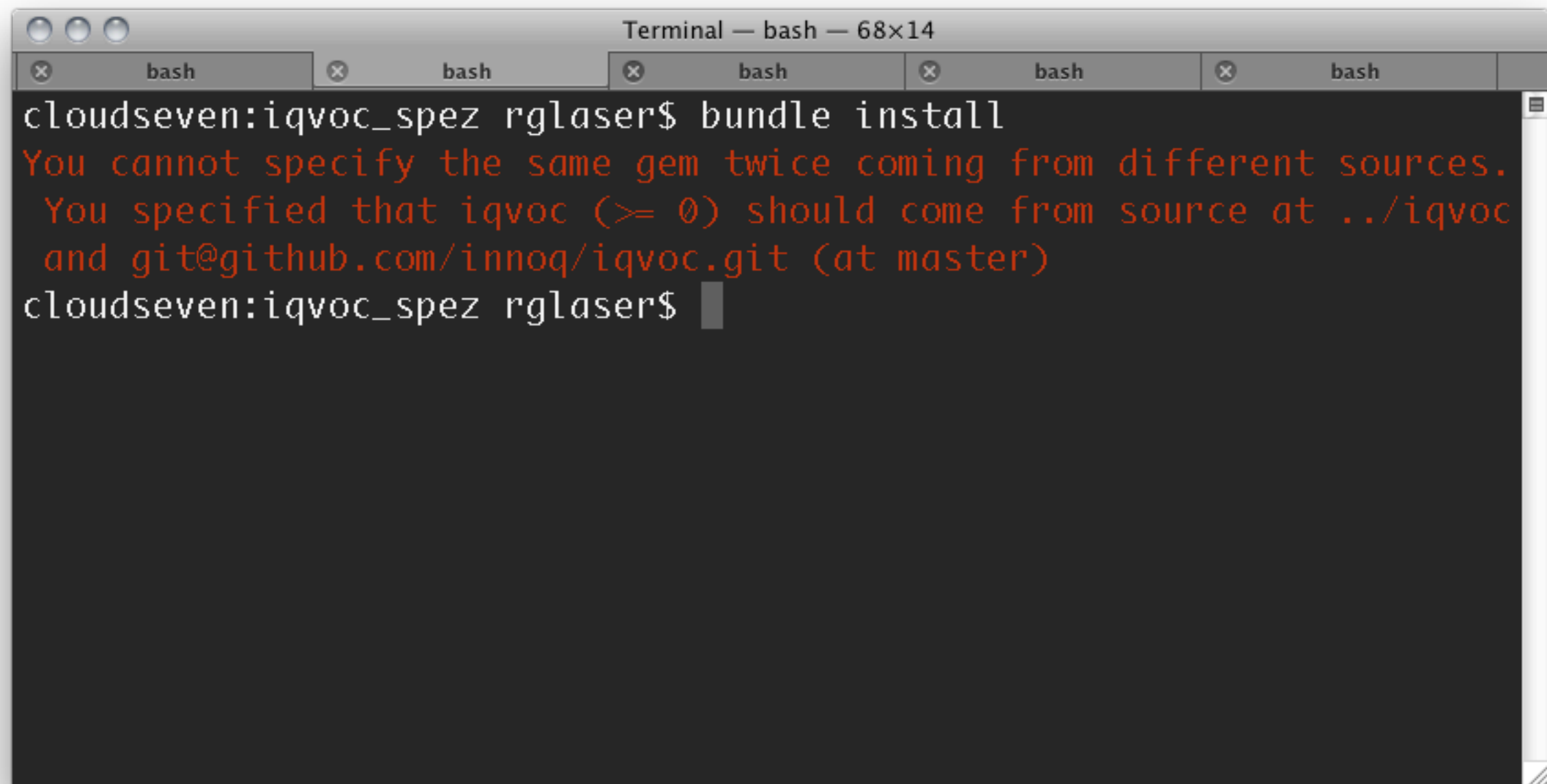
# Forget about that:

```
group :development do
  gem 'iqvoc', :path => '../iqvoc'
end
```

```
group :production do
  gem 'iqvoc', :git => 'git@github.com:innoq/iqvoc.git'
end
```

**Forget about that  
as well:**





A screenshot of a macOS Terminal window. The title bar reads "Terminal — bash — 68x14". There are five tabs, each labeled "bash". The terminal content shows a command prompt "cloudseven:iqvoc\_spez rglaser\$" followed by the command "bundle install". An error message is displayed in red text: "You cannot specify the same gem twice coming from different sources. You specified that iqvoc (>= 0) should come from source at ../iqvoc and git@github.com:innoq/iqvoc.git (at master)". The prompt "cloudseven:iqvoc\_spez rglaser\$" is shown again with a cursor.

```
cloudseven:iqvoc_spez rglaser$ bundle install
You cannot specify the same gem twice coming from different sources.
You specified that iqvoc (>= 0) should come from source at ../iqvoc
and git@github.com:innoq/iqvoc.git (at master)
cloudseven:iqvoc_spez rglaser$
```

# Instead:

## Separate your Gemfiles







**Rails 2.3 Engines**



**Rails 3.0 Engines**



**Rails 3.1 Engines**

**Engines**  
**=**  
**Mountable**  
**Apps**





Say thanks to:

**Piotr Sarnacki**  
**@drogus**

**Thanks!**