



TimeTracker Demo and Rails Setup

Let me introduce you
our time tracking tool



Check out Toggle to see
an example of what
we're about to build

First things first

1. change directory to your projects folder

```
$ cd projects/ironhack
```

Installing rails

1. Install the Rails gem

```
$ gem install rails
```

How to check Rails help

```
$ rails --help
```

Usage:

```
rails new APP_PATH [options]
```

Options:

```
-r, [--ruby=PATH]          # Path to the Ruby binary of your choice
                           # Default: /Users/fernando/.rvm/rubies/ruby-1.9.3-p484/bin/ruby
-m, [--template=TEMPLATE]  # Path to some application template (can be a filesystem path or URL)
  [--skip-gemfile]          # Don't create a Gemfile
-B, [--skip-bundle]         # Don't run bundle install
-G, [--skip-git]            # Skip .gitignore file
  [--skip-keeps]            # Skip source control .keep files
-O, [--skip-active-record]  # Skip Active Record files
```

...

Rails Help

- Gives you lists of commands and the parameters they accept
- Every Rails command has them!
- They're like built in documentation for your project. Use them like your life depends on it.

```
$ rails new timetracking
```



```
$ tree app/
```

```
app/
├── assets
│   ├── images
│   ├── javascripts
│   │   └── application.js
│   └── stylesheets
│       └── application.css
├── controllers
│   ├── application_controller.rb
│   └── concerns
├── helpers
│   └── application_helper.rb
├── mailers
├── models
│   └── concerns
└── views
    ├── layouts
    │   └── application.html.erb
```

Create New Database

```
$ rake db:create
```

Development environment

- Meant to be used while building new features for software
- Shows helpful stack traces and error messages for the developer
- Slower because all of your code is reloaded on each request

Git

```
$ git init  
$ git add .  
$ git commit -m "Add initial Rails structure"
```