In this lecture, we will discuss…

✧ Bundler

✧ Managing gems inside your Rails application

Bundler

Bundler

✧ Lets you **specify gems** (and associated gem dependencies) for this Rails app inside `Gemfile` (in the root of your Rails app)

✧ **Preferred way** to **manage gem dependencies** in Rails

✧ Run `bundle install` or simply `bundle` after specifying a new gem in the Gemfile

✧ Run `bundle update` when **modifying** a version of a gem

Bundler

✧ You can **instruct Rails** (through `Gemfile`) to **only load certain gems** in specific Rails environments

```ruby
group :development, :test do
  # Call 'byebug' anywhere in the code to stop execution and get a debugger console
  gem 'byebug'

  # Access an IRB console on exception pages or by using <%= console %> in views
  gem 'web-console', '~> 2.0'

  # Spring speeds up development by keeping your application running in the background.
  gem 'spring'
end
```

Bundler – Which Version Of Gem?

✧ If you **don't specify** … You get the latest version

✧ Can specify an **exact version** or an **approximate version**

```
gem "nokogiri"
gem "rails", "3.0.0.beta3"
gem "rack", ">=1.0"
gem "thin", ">= 1.1", "< 2.0"
gem "thin", "~>1.1"
```

~> Pessimistic Version Constraint

Drop the final digit, then increment to get the upper limit version number

Bundler: `require`

✧ Occasionally, the name of the gem to be used inside **require** statement is **different** than the name of the gem

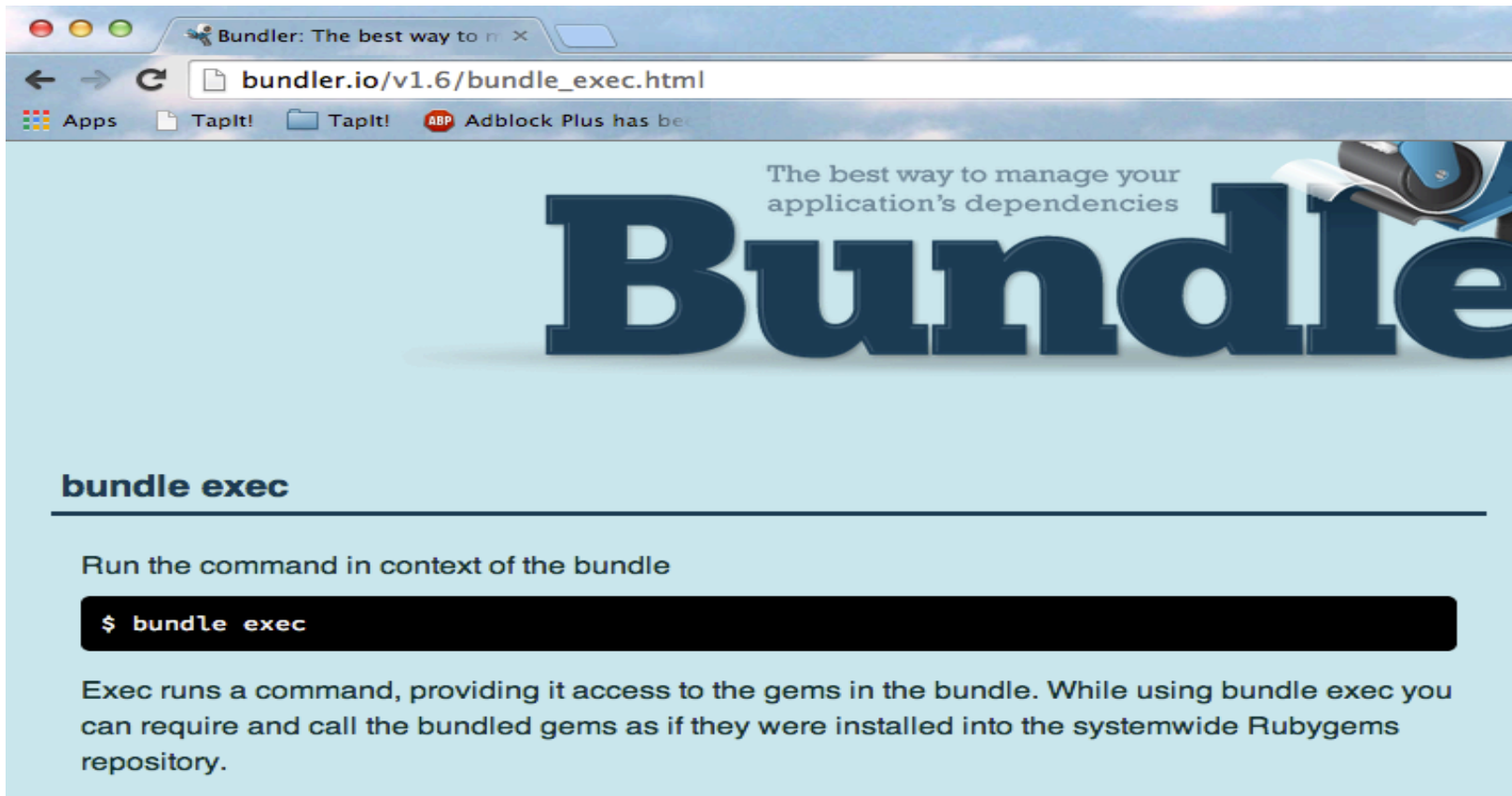```
gem 'sqlite3-ruby', require: 'sqlite3'
```

# Gemfile - Example

```
source 'http://rubygems.org'
gem 'rails', '4.2.3'
# Bundle edge Rails instead:
# gem 'rails', github:'git://github.com/rails/rails.git'
gem 'sqlite3'
…
```

✧ Our app can even use a **different version** of Rails if you **change the version** and run `bundle update`

✧ Bundler creates a `Gemfile.lock` file, which **contains the actual gem versions** your app is using with their **associated dependencies**

Bundle exec?

Summary

✧ Bundler **manages** gem dependencies

✧ **Loads** gems on application startup

What's Next

✧ Integrating HTTParty and Rails