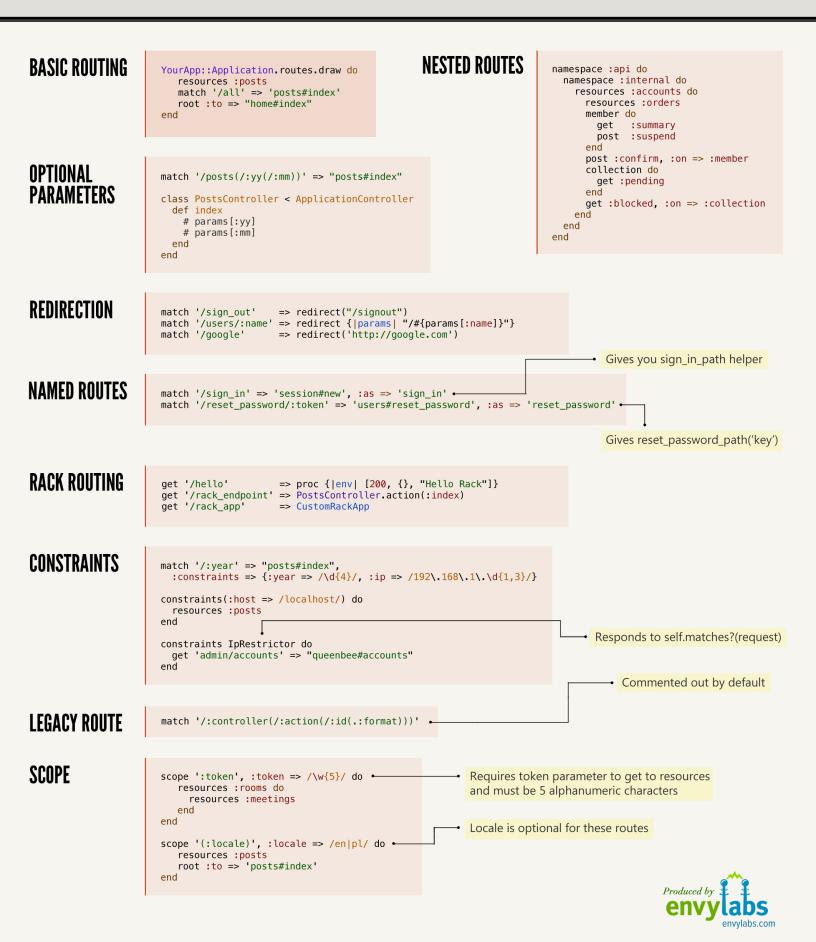
**RAILS 3** 







# BUNDLER Commands

< > required [ ] optional

#### \$ bundle

Make sure all dependencies in your Gemfile are available to your application. If they are not available, go install them.

#### \$ bundle --without <group\_name> Installs everything, except gems included in <group\_name>.

#### \$ bundle --deployment

Isolates all gems into vendor/bundle, requires up-to-date Gemfile.lock, use gems in vendor/cache if they exist.

#### \$ bundle check

Checks if the dependencies listed in Gemfile are satisfied by currently installed gems.

#### \$ bundle show [gem\_name]

Shows all libraries which are included by the Gemfile and their dependencies. If [gem\_name] is given, shows where it is located in the filesystem.

#### \$ bundle open <gem\_name>

Opens the gem source in the default editor.

\$ bundle update [gem\_name] Recreates Gemfile.lock and runs "bundle" to install new dependencies.

#### \$ bundle package

Copies all project gems to vendor/cache/ — use this if you don't want to rely on external servers for deployment.

# GEMFILE Syntax

source "http://rubygems.org"

```
gem "hpricot", "0.6"
gem "sqlite3-ruby", :require => "sqlite3"
gem "local_gem", :path => "~/Sites/local_gem"
gem "rails", :git => "git://github.com/rails/rails.git"
```

# git "git://github.com/rails/rails.git" do gem "railties"

```
group :test do
   gem "webrat"
end
```

#### Additional parameters

:branch => "branch\_name"
:tag => "tag\_number"
:ref => "ref\_number"

#### Creating a gemset for your app

- \$ rvm use \_ruby\_version\_
- \$ rvm gemset create \_name\_
- \$ rvm gemset use \_name\_
- \$ gem install bundler
- \$ bundle

http://rvm.beginrescueend.com



### WORKFLOW

#### **Developing a new application**

- 1. \$ cd new\_app/
- 2. \$ bundle init #creates a new Gemfile
- 3. Add project dependencies
- 4. Check Gemfile and Gemfile.lock into VCS

#### After adding or removing dependencies from Gemfile

- 1. \$ bundle
- 2. Commit Gemfile and Gemfile.lock

#### After modifying existing dependency versions

- 1. \$ bundle update
- 2. Commit Gemfile and Gemfile.lock

#### **Deploying your application**

\$ bundle package # locally
Copies all app gems to vendor/cache

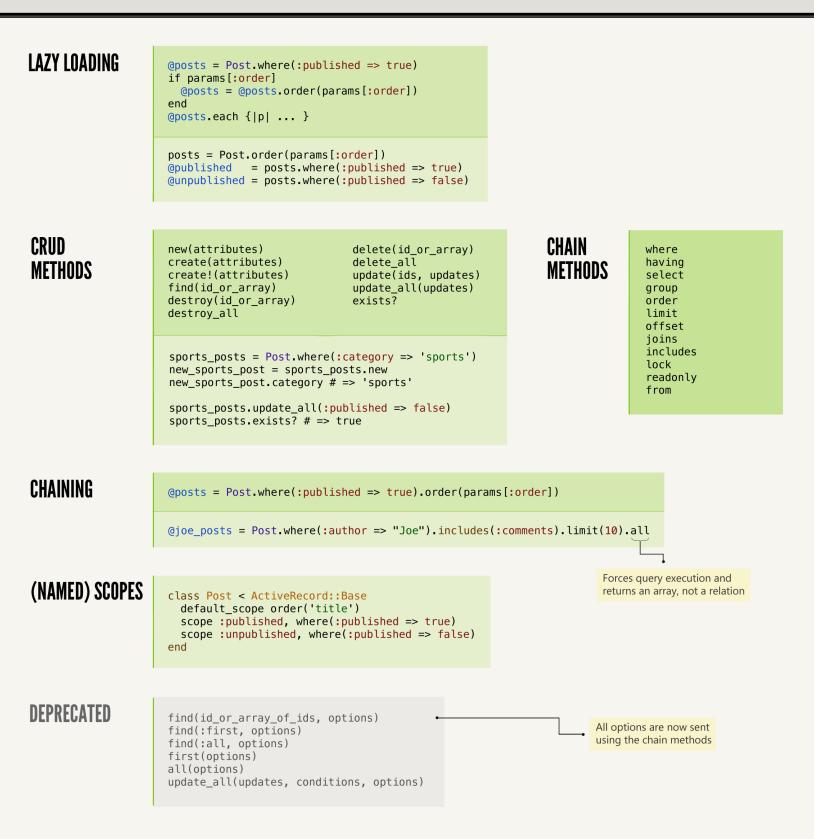
**\$ bundle** # server Installs gems from vendor/cache - no external server communication.

For more info, visit http://gembundler.com



**RAILS 3** 









# XSS PROTECTION & UJS



<%= @post.body %> ---> <%= raw(@post.body) %>
 unsafe
<%= link\_to raw("<span class='cart'>#{h @user\_input}</span>"), cart\_path %>
 not escaped not escaped

# **UNOBTRUSIVE JS**

<%= form\_for(@post, :remote => true) do |f| %>

<form action="/posts" class="new\_post" data-remote="true" id="new\_post" method="posts">

# HTML5 CUSTOM Data Attributes

data-remotedata-confirmdata-methoddata-disable-withParsed by JavaScript drivers — defaults to rails.js

# DEPRECATED

link\_to\_remote
remote\_form\_for
observe\_field
observe\_form
form\_remote\_tag
button\_to\_remote
submit\_to\_remote
link\_to\_function
periodically\_call\_remote

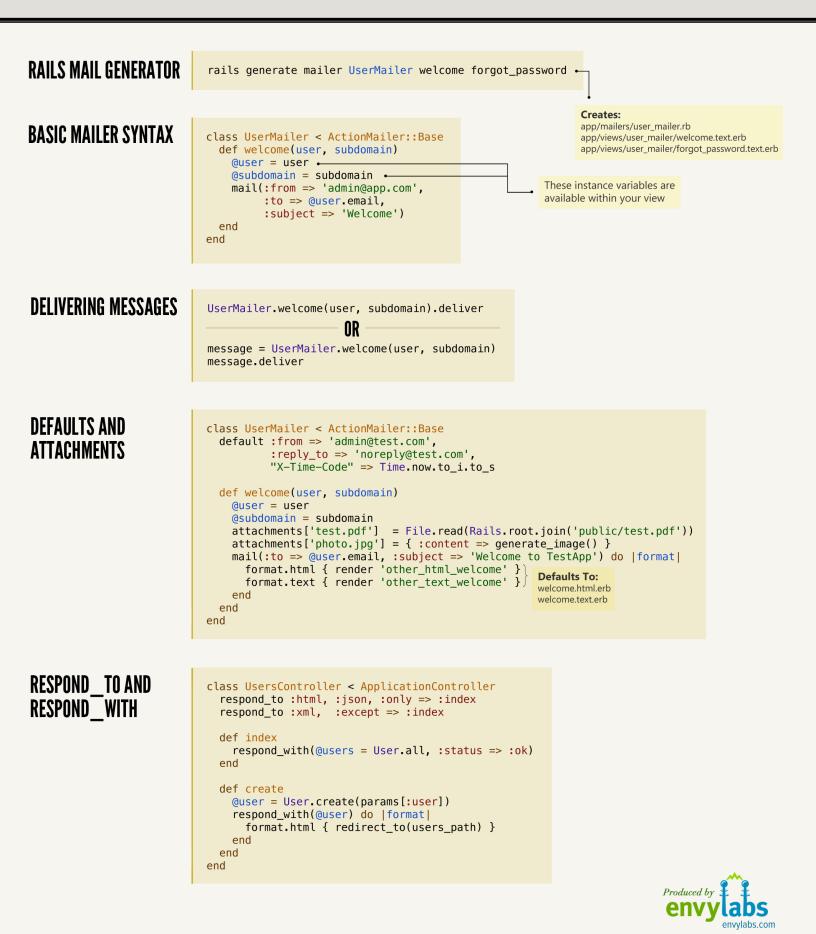
If you need to use any of the deprecated helpers, visit http://github.com/rails/prototype\_legacy\_helper

# **USING JQUERY**

Go to https://github.com/rails/jquery-ujs for instructions



# RAILS 3 ACTIONMAILER & ACTIONCONTROLLER



**RAILS 3** 



end end

DIRTY	<pre>class Person include ActiveModel::Dirty define_attribute_methods [:name] def name @name end def name=(val) name_will_change! @name = val end def save @previously_changed = changes @changed_attributes.clear end end</pre>	$\begin{array}{c} & & \\$	person.name_changed?		<b>MODULES</b> AttributeMethods Callbacks Dirty Errors Naming Observing Serialization Validations
VALIDATIONS	<pre>class Person include ActiveModel::Validations attr_accessor :email validates_presence_of :email end &gt;&gt; p = Person.new =&gt; #<person:0x000001021b7198> &gt;&gt; p.valid? =&gt; false &gt;&gt; p.errors =&gt; {:email=&gt;["can't be blank"]}</person:0x000001021b7198></pre>	va va va va va va	<pre>Shortcuts validates :terms, :acceptance =&gt; true validates :password, :confirmation =&gt; true validates :username, :exclusion =&gt; { :in =&gt; %w(admin) } validates :email, :format =&gt; {     :with =&gt; /\A([^0(\s]+)@((?:[-a-z0-9]+\.)+[a-z]{2,})\Z/i,     :on =&gt; :create } validates :age, :inclusion =&gt; { :in =&gt; 09 } validates :first_name, :length =&gt; { :maximum =&gt; 30 } validates :username, :presence =&gt; true validates :username, :uniqueness =&gt; true</pre>		
SERIALIZATION	<pre>class Person include ActiveModel::Serializers::: attr_accessor :name def attributes {:name =&gt; name} end end</pre>	JSON	<pre>&gt;&gt; p = Person.new =&gt; #<person:0x000001021;>&gt; p.name = "Gregg" =&gt; "Gregg" &gt;&gt; p.to_json =&gt; "{\"name\":\"Gregg\";</person:0x000001021;></pre>		
CALLBACKS	<pre>class Person extend ActiveModel::Callbacks define_model_callbacks :save before_save :action_before_save def save _run_save_callbacks do # Your save action methods here end end private def action_before_save # Your code here end end</pre>	e			Produced by

