

Views

RSS Feed

The [builder](#) gem/library for creating XML is required in this recipe.

Assume that your site url is <http://liftoff.msfc.nasa.gov/>.

```
get '/rss.xml' do
  builder do |xml|
    xml.instruct! :xml, :version => '1.0'
    xml.rss :version => "2.0" do
      xml.channel do
        xml.title "Liftoff News"
        xml.description "Liftoff to Space Exploration."
        xml.link "http://liftoff.msfc.nasa.gov/"

        @posts.each do |post|
          xml.item do
            xml.title post.title
            xml.link "http://liftoff.msfc.nasa.gov/posts/#{post.id}"
            xml.description post.body
            xml.pubDate Time.parse(post.created_at.to_s).rfc822()
            xml.guid "http://liftoff.msfc.nasa.gov/posts/#{post.id}"
          end
        end
      end
    end
  end
end
```

This will render the RSS inline, directly from the handler.

CoffeeScript

To render CoffeeScript templates you first need the [coffee-script](#) gem and [therubyracer](#), or access to the [coffee](#) binary.

Here's an example of using CoffeeScript with Sinatra's template rendering engine Tilt:

```
## You'll need to require coffee-script in your app
require 'coffee-script'

get '/application.js' do
  coffee :application
```

```
end
```

Renders `./views/application.coffee`.

This works great if you have access to [nodejs](#) or [therubyracer](#) gem on your platform of choice and hosting environment. If that's not the case, but you'd still like to use CoffeeScript, you can precompile your scripts using the `coffee` binary:

```
$ coffee -c -o public/javascripts/ src/
```

Or you can use this example [rake](#) task to compile them for you with the `coffee-script` gem, which can use either [therubyracer](#) gem or the `coffee` binary:

```
require 'coffee-script'

namespace :js do
  desc "compile coffee-scripts from ./src to ./public/javascripts"
  task :compile do
    source = "#{File.dirname(__FILE__)}/src/"
    javascripts = "#{File.dirname(__FILE__)}/public/javascripts/"

    Dir.foreach(source) do |cf|
      unless cf == '.' || cf == '..'
        js = CoffeeScript.compile File.read("#{source}#{cf}")
        open "#{javascripts}#{cf.gsub('.coffee', '.js')}", 'w' do |f|
          f.puts js
        end
      end
    end
  end
end
```

Now, with this rake task you can compile your coffee-scripts to `public/javascripts` by using the `rake js:compile` command.

Resources

If you get stuck or want to look into other ways of implementing CoffeeScript in your application, these are a great place to start:

- [coffee-script](#)
- [therubyracer](#)
- [ruby-coffee-script](#)