rjs demystified

a cheat sheet by amy hoy, **www.slash7.com**

what is this RJS thing anyway?

R.J.S., acronym
Ruby JavaScript

A templating tool, built in Ruby, to output JavaScript.

synonym, *amazing*

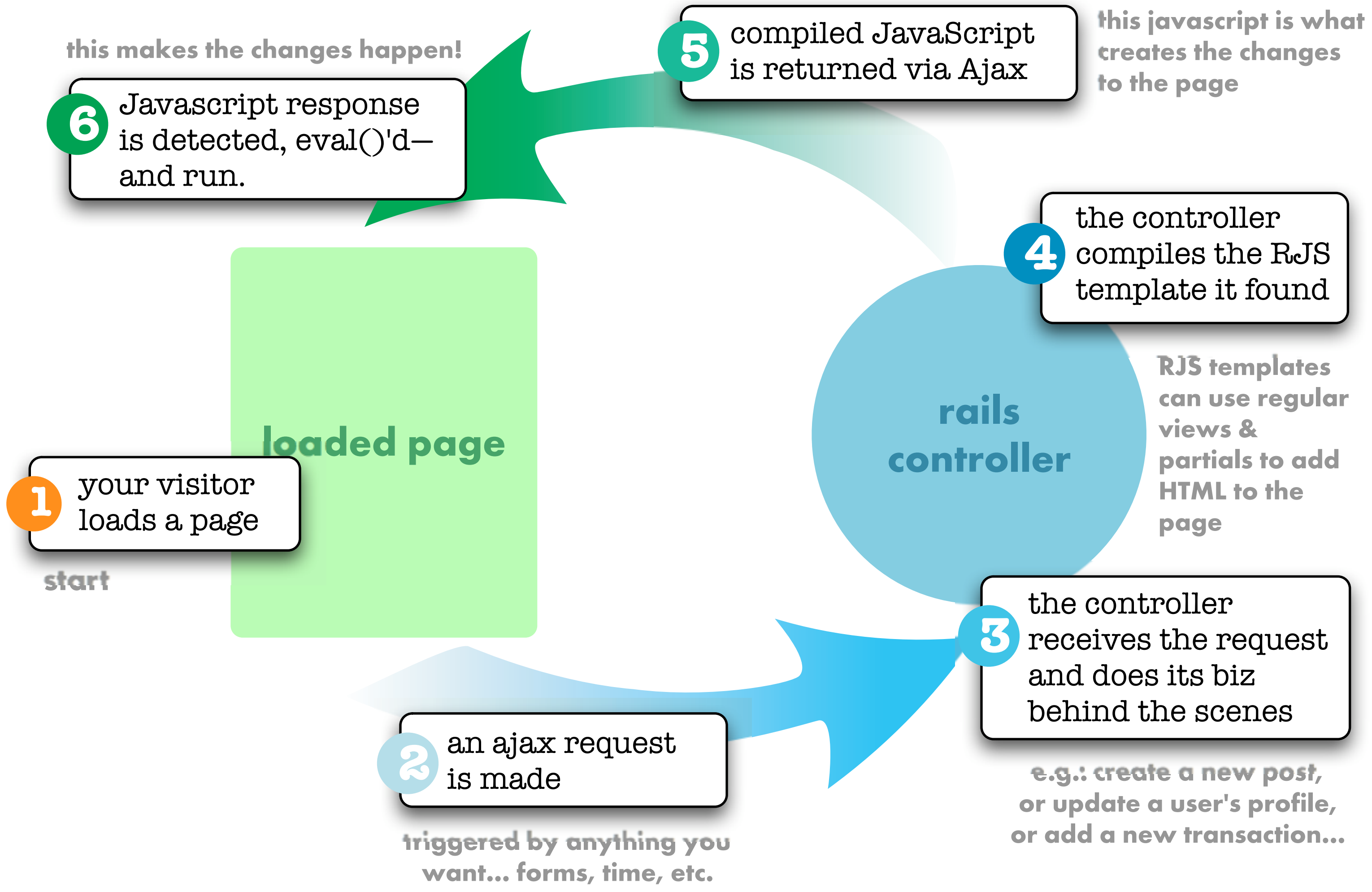
Makes it almost utterly painless to craft sophisticated Ajax responses... with sexy results!

RJS techniques

* insert new HTML anywhere
* remove HTML
* change DOM properties
* move things around
* create special effects
* hide stuff
* show stuff
* take over the universe

RJS templates are "just" another type of **view template**, made specifically for answering Ajax requests. The code you write in an **.rjs template** is based around the page you will be changing—the page that made the Ajax request (and therefore will be getting the response). Anything you can do with JavaScript, you can do with RJS.

here's how it works!


this makes the changes happen!
6 Javascript response is detected, eval()'d— and run.
5 compiled JavaScript is returned via Ajax
this javascript is what creates the changes to the page
4 the controller compiles the RJS template it found
rails controller
RJS templates can use regular views & partials to add HTML to the page
loaded page
1 your visitor loads a page
start
3 the controller receives the request and does its biz behind the scenes
e.g.: create a new post, or update a user's profile, or add a new transaction...
2 an ajax request is made
triggered by anything you want... forms, time, etc.

a sample...

```
if @saved
  page["new_expense_form"].reset
  page.visual_effect :blind_up, :new_expense
  page.insert_html :top, :expenses, :partial => "expense"
  page.replace_html :total, "$#{@total}"
  page.delay(1) do
    page.visual_effect :highlight, "expense_#{@expense.id}"
  end
else
  page.alert("Problem saving expense: #{@expense.errors.full_messages}")
  page.visual_effect :blind_up, :new_expense
end
```

The controller compiles this into JS & the response is sent as `Content-Type: text/javascript`

rjs demystified

a cheat sheet by amy hoy, **www.slash7.com**

it? tip! support Slash7!

45-minute RJS screencast only $9! peepcode.com

use it yourself!

Gettin' Started

You might want to try a BS action that you're using just to experiment.

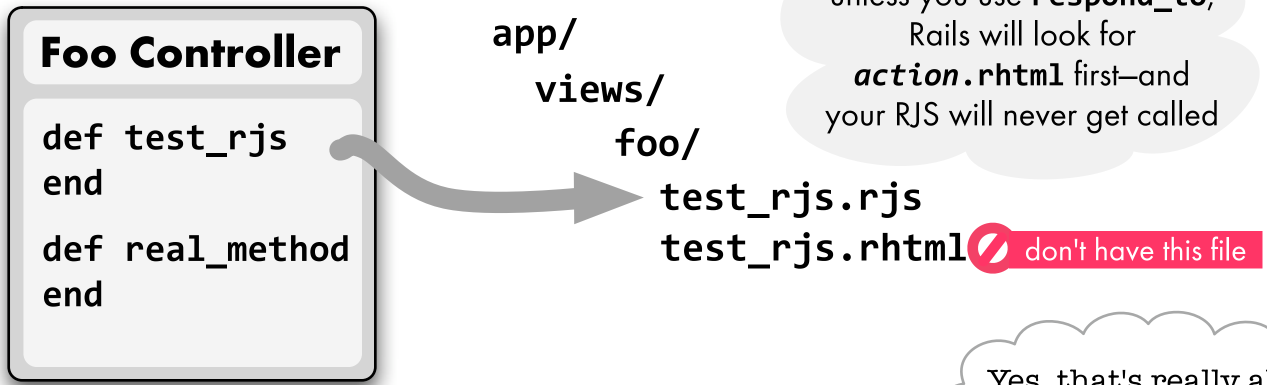
1 First thing you need to do is **create an Ajax request** that points at a controller action. Place this code in the view you want to modify with the Ajax response:

```
<%= javascript_include_tag :defaults %>

<div id="rjs-testbed">Watch this space.</div>

<%= link_to_remote "Show me RJS lovin!",
:url => { :controller => "foo", :action => "test_rjs" } %>
```

2 Create a file named ***action*.rjs** where *action* is the name of the controller action, inside the controller's view folder.



```
Foo Controller

def test_rjs
end

def real_method
end
```

```
app/
  views/
    foo/
      test_rjs.rjs
      test_rjs.rhtml
```

unless you use **respond_to**, Rails will look for ***action*.rhtml** first—and your RJS will never get called

3 Finally—it's time to write some RJS! Open up your new ***action*.rjs** file and include the following:

```
page.alert "This is a test! Warning, this is a test!"
page.insert_html :bottom, "rjs-testbed", "<br/> Wahla!"
```

Yes, that's really all to put in the file. Nothing else. That's it. Really.

4 Now you get to try it! Start your Rails app, open up your browser, and hit the action whose view houses the code from **Step 1**. Click it. Watch the...

5 **MAGIC! OMG OMG OMG SO AWESOME!!**
Now, you'll have to **read the docs**, **buy the video**, and wait for my next (reference-style) cheat sheet, where I'll list in detail all the fun stuff you can do.

rjs demystified
a cheat sheet by amy hoy, **www.slash7.com**

♥ it? tip! support Slash7!

45-minute RJS screencast only $9! peepcode.com