CS 683 Emerging Technologies
Fall Semester, 2006
Doc 18 Basic Action Pack & Active Record
Oct 19, 2006

References

Agile Web Development with Rails, Thomas & Hanson, The Pragmatic Bookshelf, 2005

Patterns of Enterprise Application Architecture, Fowler, Addison-Wesley, 2003

Action Pack Documentation, http://api.rubyonrails.com/

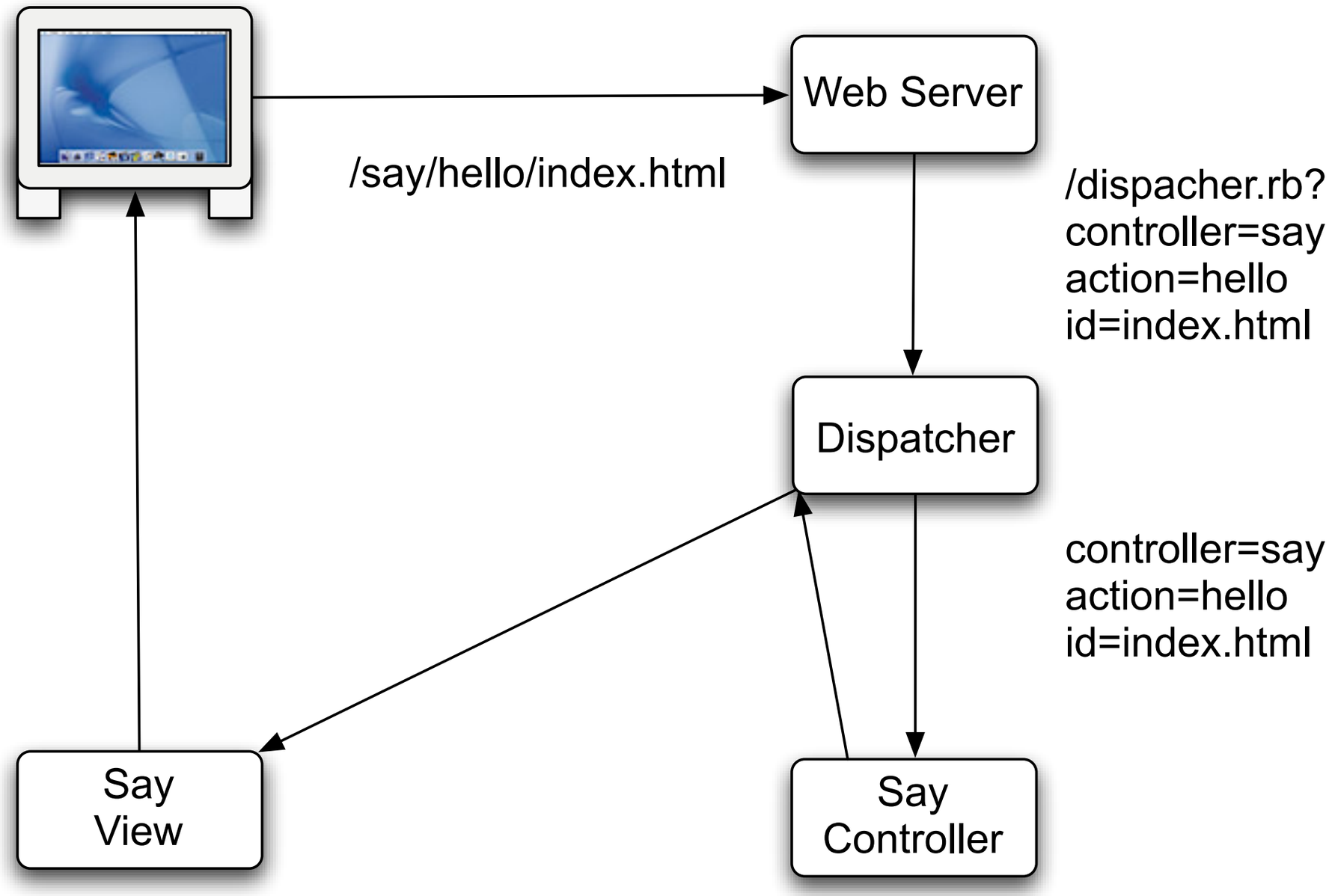
Active Record Documentation, http://api.rubyonrails.com/

Action Pack
Views

Selecting the View

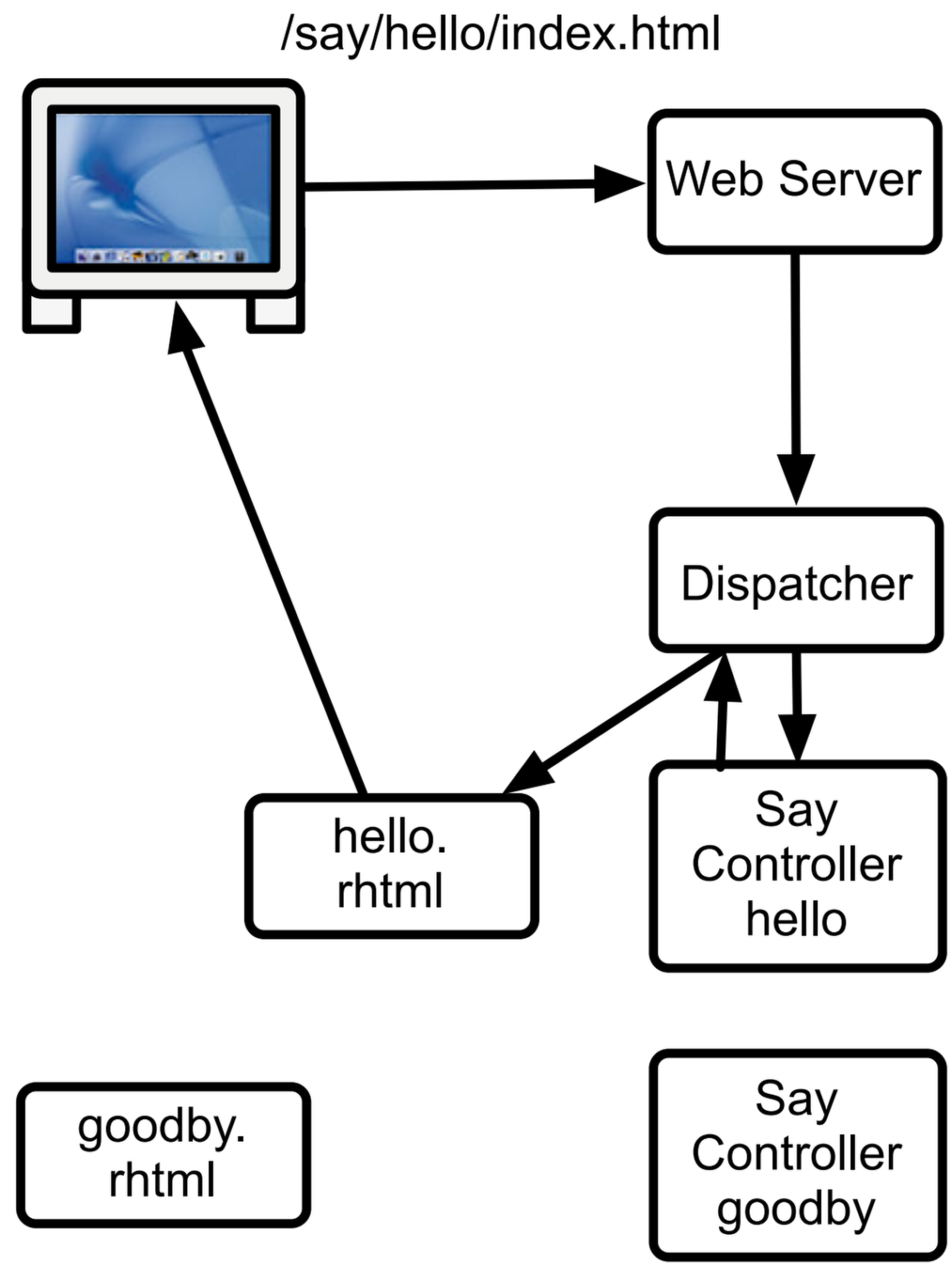
Simple Request Flow

http://127.0.0.1:3000/say/hello/index.html



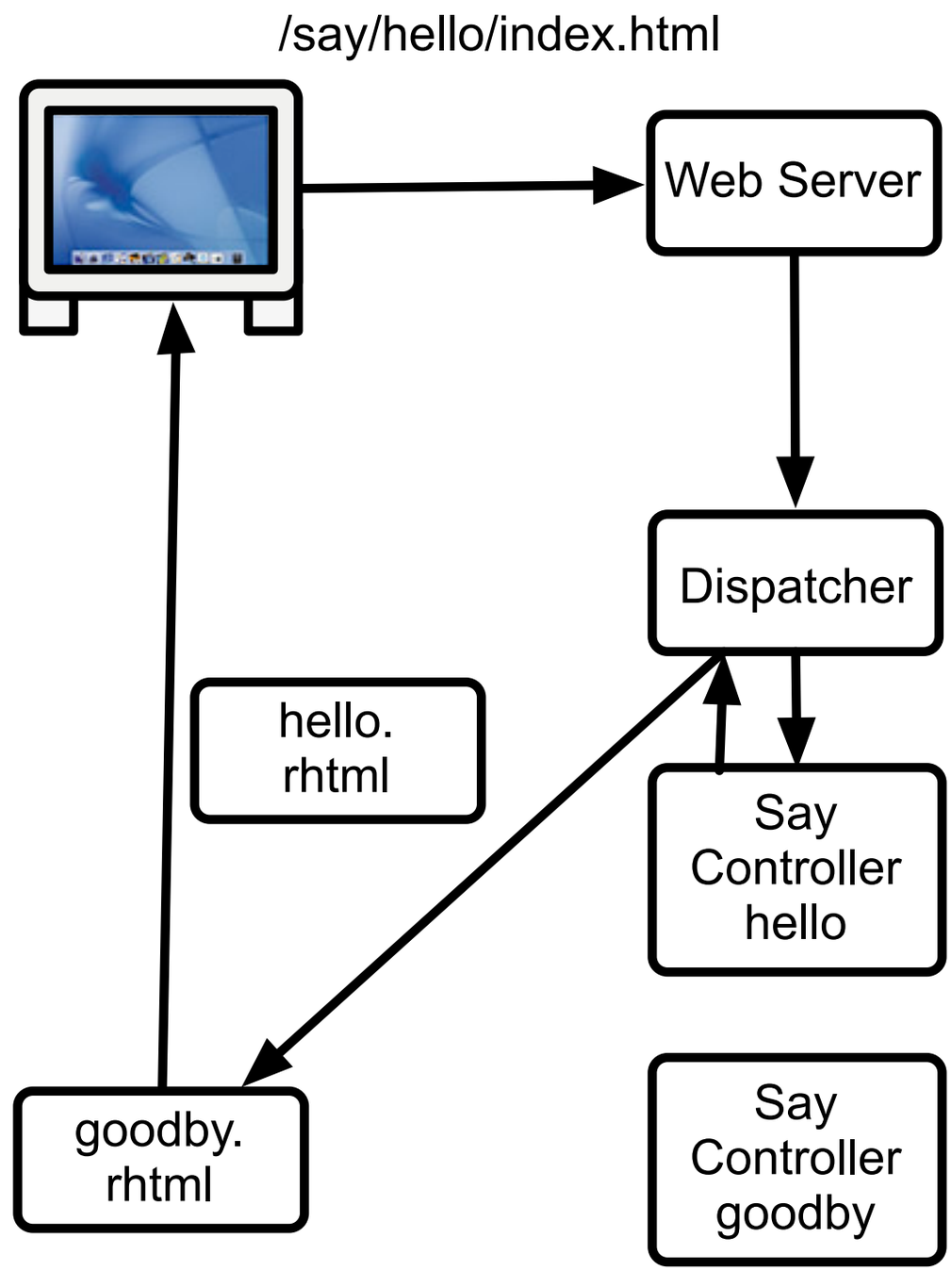
Default Action

```
class SayController < ApplicationController
  def hello
    @time = Time.now
    render :template => "say/goodbye"
  end

  def goodbye
  end
end
```



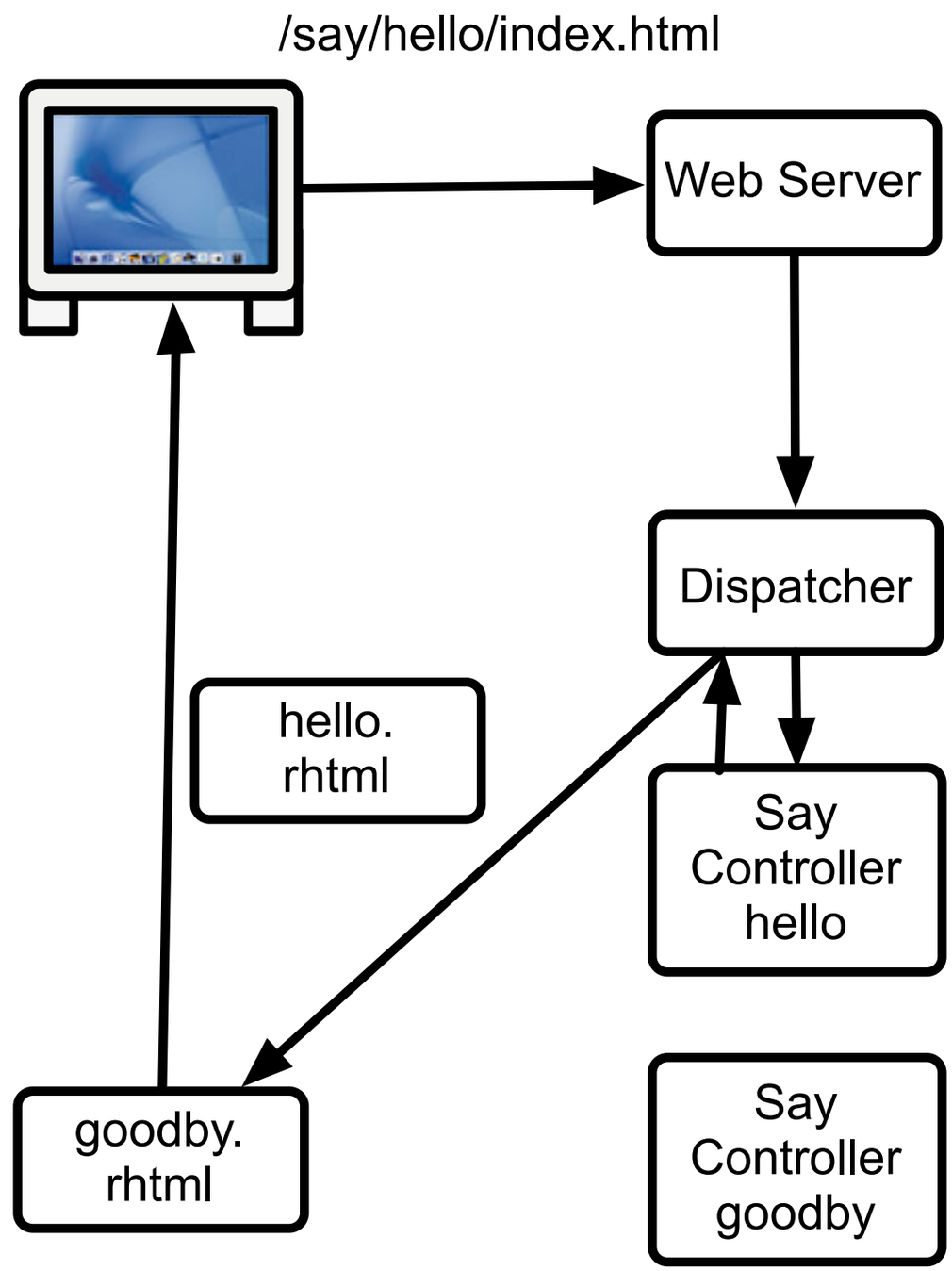
render :template

```
class SayController < ApplicationController
  def hello
    @time = Time.now
    render :template => "say/goodbye"
  end

  def goodbye
  end
end
```



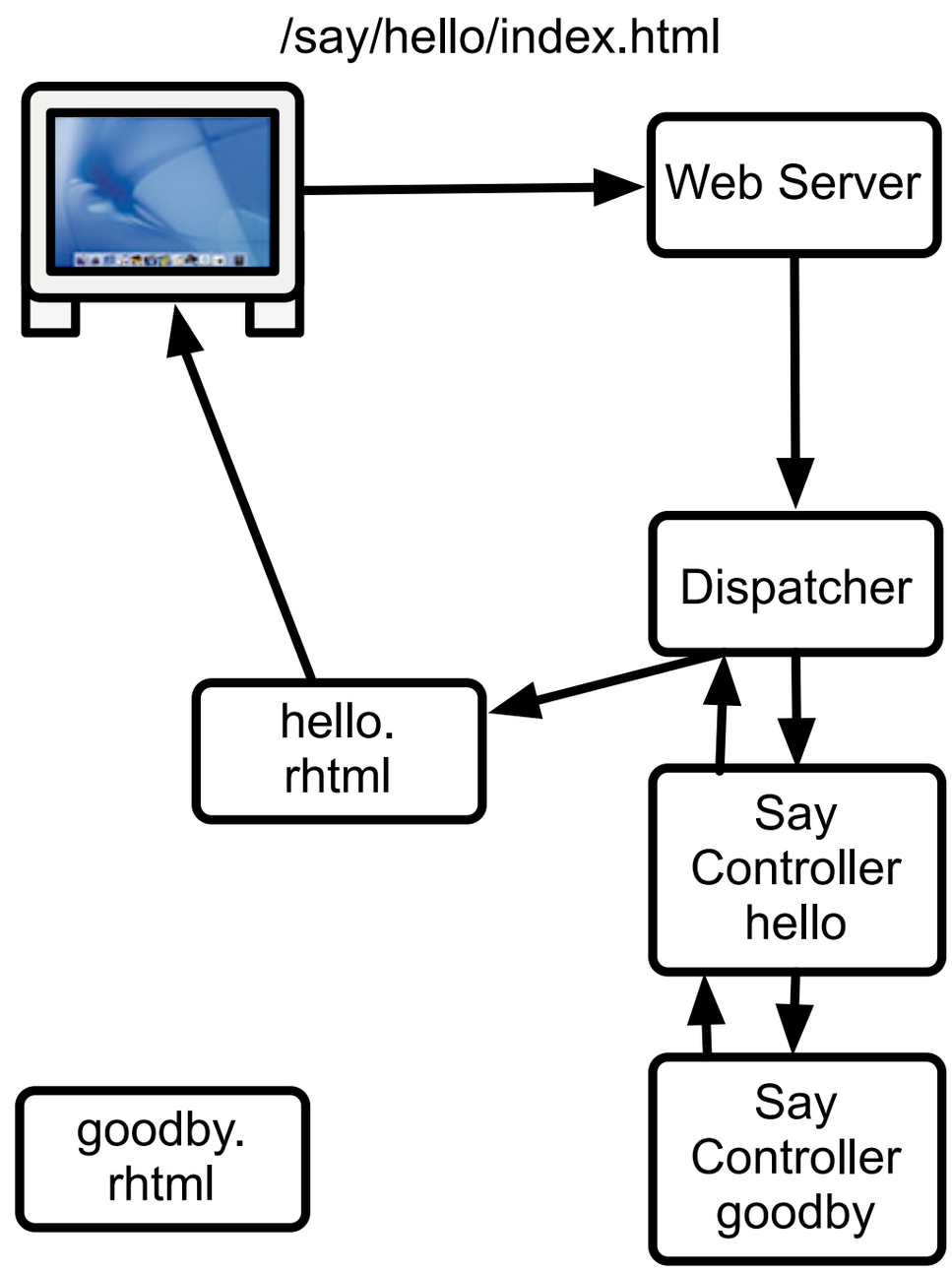
render :action

```
class SayController < ApplicationController
  def hello
    @time = Time.now
    render :action => "goodbye"
  end

  def goodbye
  end
end
```



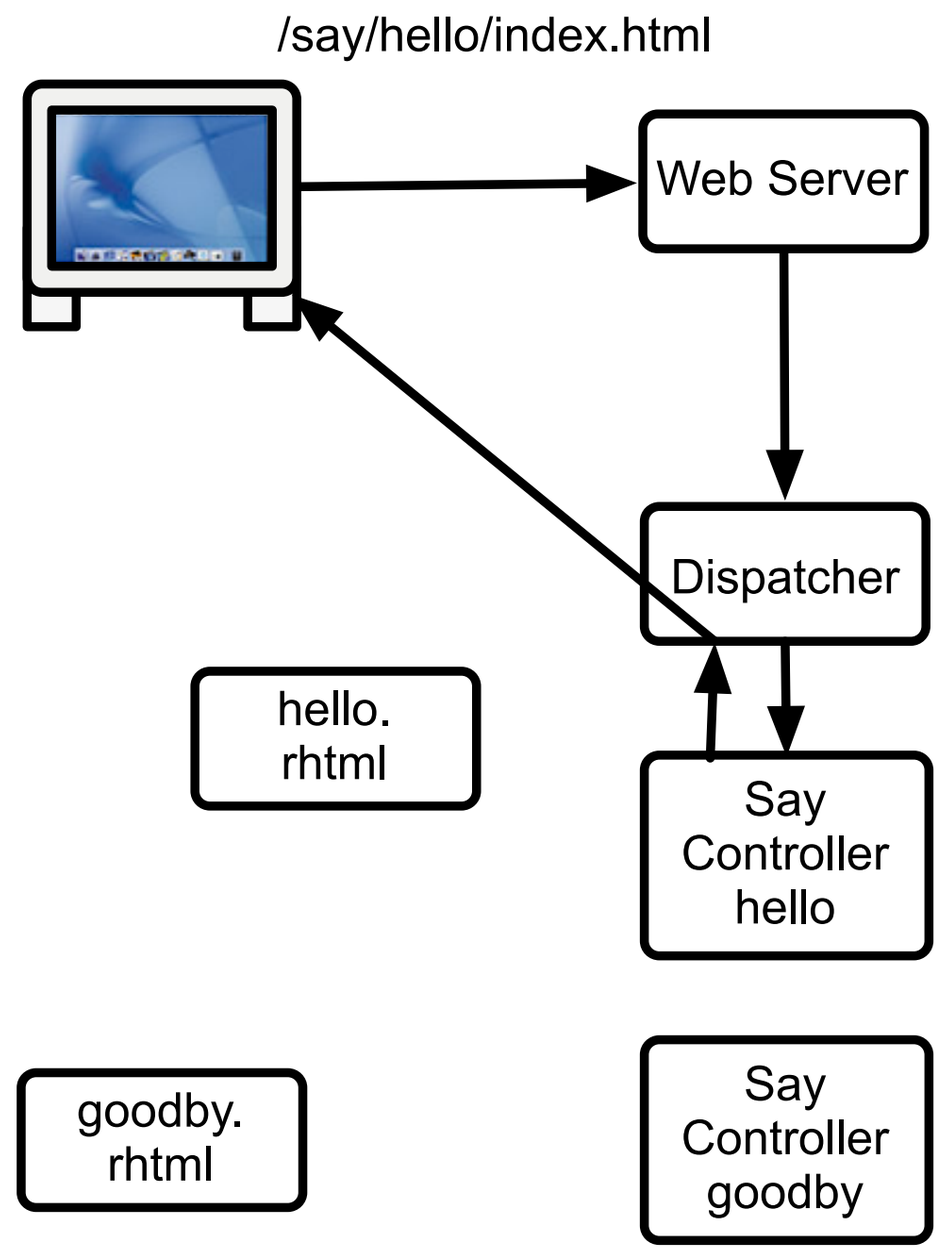
Direct Call

```
class SayController < ApplicationController
  def hello
    @time = Time.now
    goodbye
  end

  def goodbye
  end
end
```



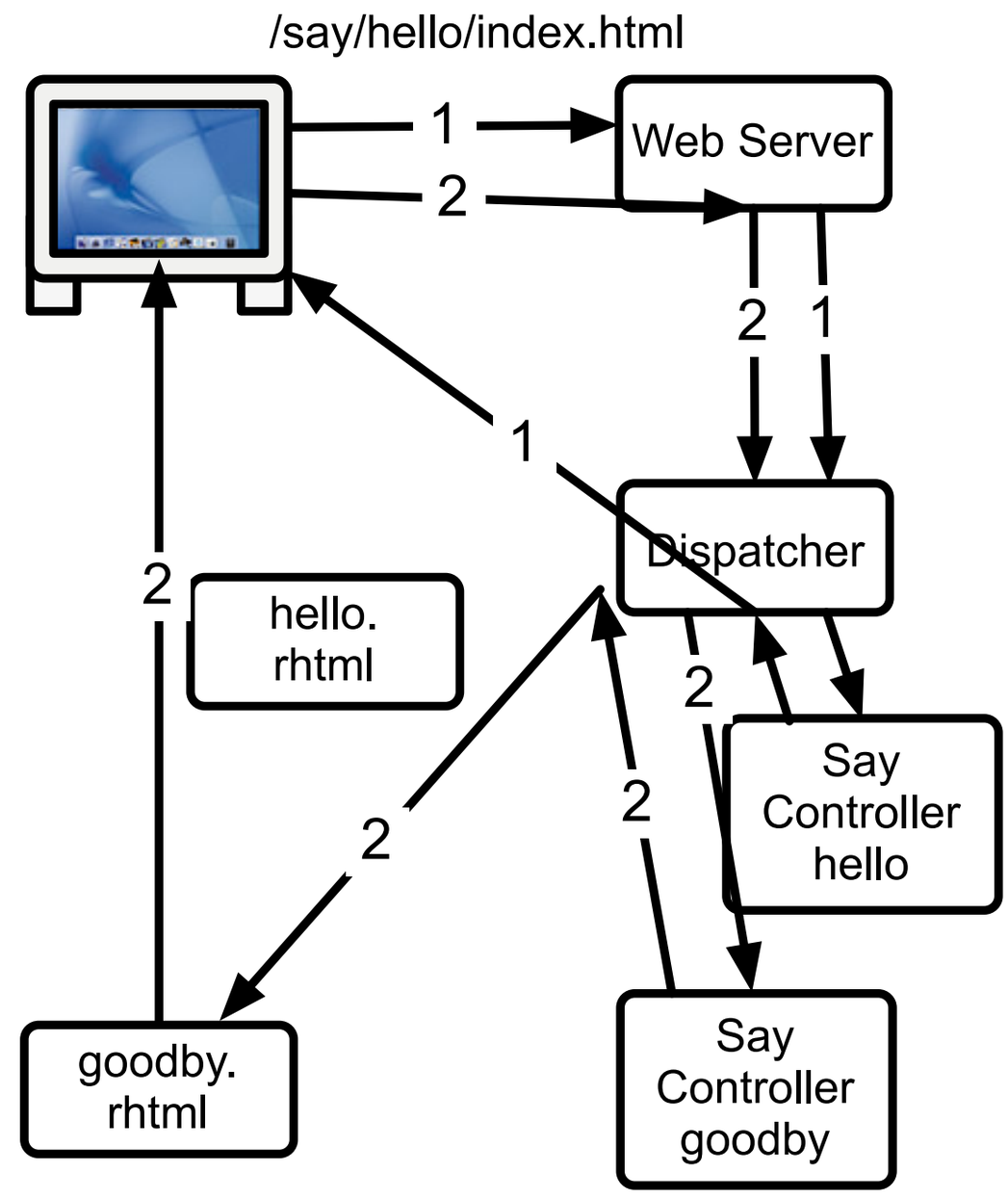
render :text

```
class SayController < ApplicationController
  def hello
    @time = Time.now
    render :text => "Hello There"
  end

  def goodbye
  end
end
```



redirect :action

```
class SayController < ApplicationController
  def hello
    @time = Time.now
    redirect_to :action => "goodbye"
  end

  def goodbye
  end
end
```



These are method calls

```
class SayController < ApplicationController
  def hello
      redirect_to :action => "goodbye"
            puts "we do reach here"
  end

  def goodbye
  end
end
```

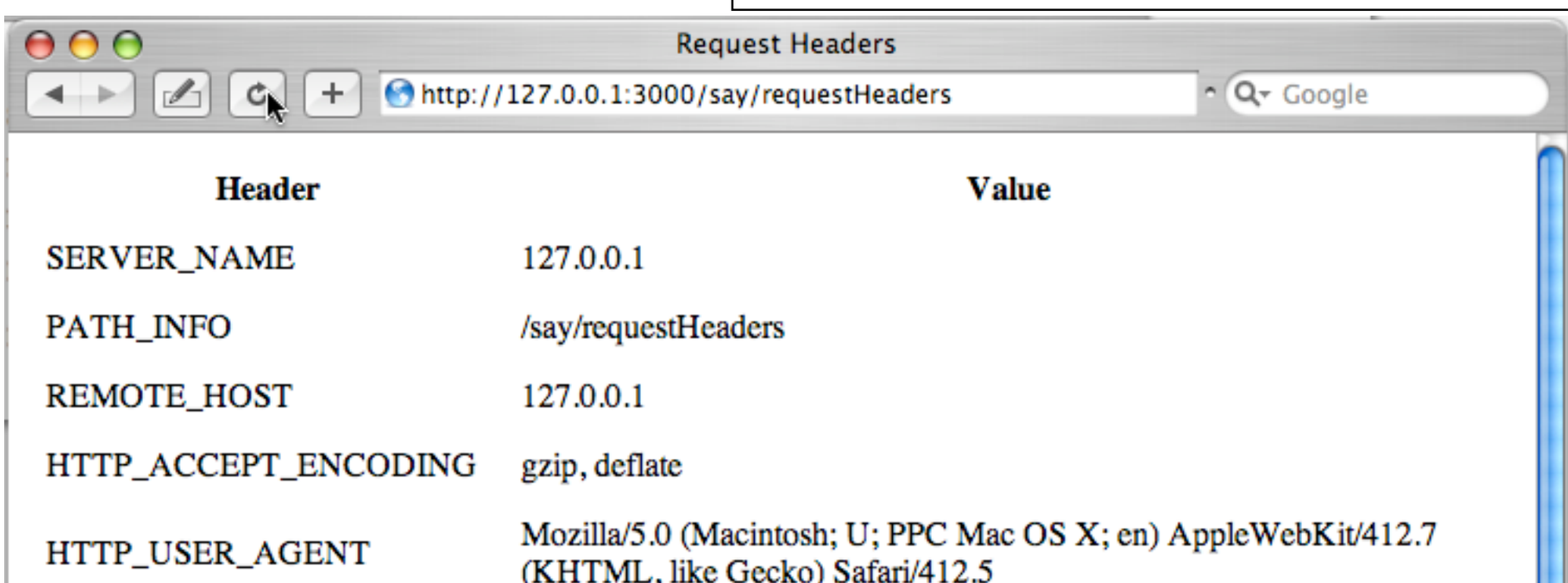
View & Information

Request Headers

app/views/say/requestHeaders.rhtml

```
class SayController < ApplicationController
  def requestHeaderInController
    remote_host = request.env['REMOTE_HOST']
  end

  def requestHeaders
  end
end
```

```
<html>
  <head> <title>Request Headers</title>
  </head>
  <body>
    <table border="0" cellspacing="5" cellpadding="5">
      <tr><th>Header</th><th>Value</th></tr>
      <% request.env.each_pair do | key, value| %></p>
        <tr><td><%= key %></td><td><%= value %></td></tr>
      <% end %>
      </table>
  </body>
</html>
```

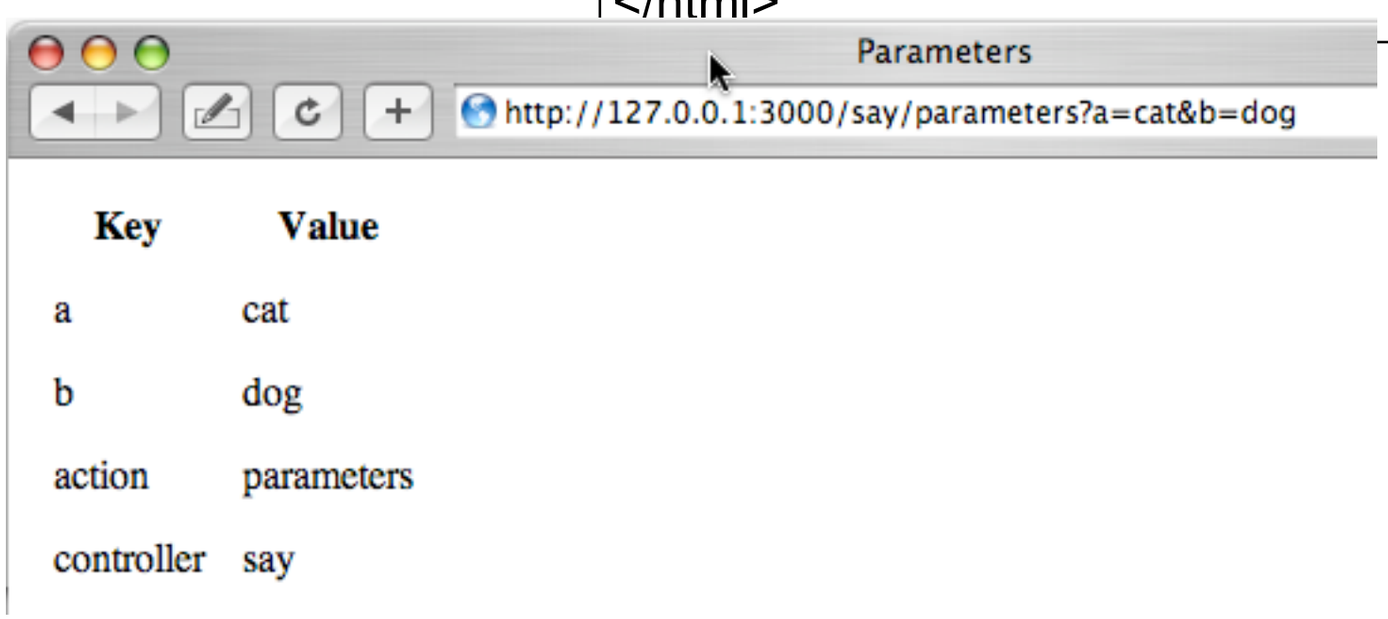


Post & Get Parameters

```
class SayController < ApplicationController
    def parameters
        a = params["a"]
  end

end
```

app/views/say/parameters.rhtml

```
<html>
  <head>
    <title>Parameters <$= params["a"] $></title>
  </head>
  <body>
    <table border="0" cellspacing="5" cellpadding="5">
      <tr><th>Key</th><th>Value</th></tr>
      <% params.each_pair do | key, value| %></p>
        <tr><td><%= key %></td><td><%= h(value) %></
td></tr>
      <% end %>
      </table>
  </body>
</html>
```



flash - Temporary variables to View

app/controllers/flash_controller.rb

```
class FlashController < ApplicationController
  def one
    flash[:a] = "Hello1"
  end
  def two
    flash[:b] = "Hello2"
  end
  def three
    flash[:c] = "Hello3"
  end
end
```

flash variable is flushed at the end of the subsequent action

app/views/flash/one.rhtml

```
<html>
  <head><title>flash[:a]</title></head>
  <body>
    <table>
      <tr><th>Header</th><th>Value</th></tr>
      <% flash.each_pair do | key, value| %></p>
        <tr><td><%= key %></td><td><%= value %></td></tr>
      <% end %>
      </table>
  </body></html>
```

app/views/flash/two.rhtml

```
<html>
  <head><title>flash[:b]</title></head>
  <body>
    <table>
      <tr><th>Header</th><th>Value</th></tr>
      <% flash.each_pair do | key, value| %></p>
        <tr><td><%= key %></td><td><%= value %></td></tr>
      <% end %>
      </table>
  </body></html>
```

flash - Example

app/controllers/flash_controller.rb

```
class FlashController < ApplicationController
  def one
    flash[:a] = "Hello1"
  end
  def two
    flash[:b] = "Hello2"
  end
  def three
    flash[:c] = "Hello3"
  end
end
```

http://127.0.0.1:3000/flash/one

Header	Value
a	Hello1

http://127.0.0.1:3000/flash/two

Header	Value
a	Hello1
b	Hello2

http://127.0.0.1:3000/flash/three

Header	Value
b	Hello2
c	Hello3

http://127.0.0.1:3000/flash/three

Header	Value
c	Hello3

flash now & keep

app/controllers/flash_controller.rb

```
class FlashController < ApplicationController
  def one
    flash[:a] = "Hello1"
  end
  def two
    flash.now[:b] = "Hello2"
    flash.keep(:a)
  end
  def three
    flash[:c] = "Hello3"
  end
end
```

http://127.0.0.1:3000/flash/one

Header	Value
a	Hello1

http://127.0.0.1:3000/flash/two

Header	Value
a	Hello1
b	Hello2

http://127.0.0.1:3000/flash/three

Header	Value
a	Hello1
c	Hello3

http://127.0.0.1:3000/flash/three

Header	Value
c	Hello3

Layout & Partials

Layouts

app/views/layouts/flash.rhtml

```
<html>
<head>
  <title><%= controller.action_name %></title>
</head>
<body>
<%= @content_for_layout %>
</body>
</html>
```

app/views/flash/one.rhtml

```
<table>
  <tr><th>Header</th><th>Value</th></tr>
  <% flash.each_pair do | key, value| %></p>
    <tr><td><%= key %></td><td><%= value %></td></
tr>
  <% end %>
</table>
```

app/views/flash/two.rhtml

```
<table>
  <tr><th>Header</th><th>Value</th></tr>
  <% flash.each_pair do | key, value| %></p>
    <tr><td><%= key %></td><td><%= value %></td></
tr>
  <% end %>
</table>
```

Non-default Layouts

```
class FlashController < ApplicationController
  layout 'standard'
  #...
end

class FlashController < ApplicationController
  layout 'my_layout'
  #...
end

class FlashController < ApplicationController
  layout 'my_layout', :except => [:rss, :atom]

  def rss
    render(:layout => false) #no layout
  end
end
```

```
class FlashController < ApplicationController
  layout :different_layout

  def different_layout
    if @x > 0
      "positive"
    else
      "negative"
  end
end
```

```
class FlashController < ApplicationController

  def one
    flash[:a] = "Hello1"
    render :layout => "layouts/anotherLayout"
  end
end
```

Partials

app/views/layouts/flash.rhtml

```
<html>
<head>
  <title><%= controller.action_name %></title>
</head>
<body>
<%= @content_for_layout %>
</body>
</html>
```

app/views/flash/one.rhtml

```
<h1>One</h1>
<%= render :partial => 'flashTable' %>
```

app/views/flash/two.rhtml

```
<h1>Two</h1>
<%= render :partial => 'flashTable' %>
```

app/views/flash/**_flashTable**.rhtml

```
<table>
  <tr><th>Header</th><th>Value</th></tr>
  <% flash.each_pair do | key, value| %></p>
    <tr><td><%= key %></td><td><%= value %></td></tr>
  <% end %>
</table>
```

app/controllers/flash_controller.rb

```
class FlashController < ApplicationController
  def one
    flash[:a] = "Hello1"
  end
  def two
    flash.[:b] = "Hello2"
  end
  def three
    flash[:c] = "Hello3"
  end
end
```

Active Record

Active Record

An Object that

wraps a row in a database table or view,
encapsulates the database access,
adds domain logic on that data
"Martin Fowler"

TableName	
id title author date	*primary Key*

```
CREATE TABLE `books` (
  `id` int(11) NOT NULL auto_increment,
  `title` varchar(100) NOT NULL default '',
  `author` varchar(50) NOT NULL default '',
  `date` date default '0000-00-00',
  PRIMARY KEY  (`id`)
)
```

```
require "rubygems"
require_gem "activerecord"

#Connect
ActiveRecord::Base.establish_connection(
  :adapter => "mysql",
  :host => "localhost",
  :database => "cs683BookStore_development",
  :username => "whitney")

class Book < ActiveRecord::Base
end

#Add to database
ruby = Book.new
ruby.title = 'Programming Ruby'
ruby.author = 'Thomas, Dave'
ruby.date = Time.now
ruby.save

from_database = Book.find_by_title('Programming Ruby')
puts from_database.author
```

Adding to Database

```
ruby = Book.new
ruby.title = 'Programming Ruby'
ruby.author = 'Thomas, Dave'
ruby.date = Time.now
ruby.save

aBook = Book.new(
  :title => 'Programming Ruby',
  :author => 'Thomas, Dave',
  :date => Time.now )
aBook.save

aBook = Book.create(
  :title => 'Programming Ruby',
  :author => 'Thomas, Dave',
  :date => Time.now )

params = {  :title => 'Programming Ruby',
  :author => 'Thomas, Dave',
  :date => Time.now}
aBook = Book.create(params)
```

Reading from the Database

```
a = Book.find_by_title('Programming Ruby')
a = Book.find_by_author('Thomas, Dave')
a = Book.find_all_by_author('Thomas, Dave')

a = Book.find_by_sql("select * from Books where author = 'Thomas, Dave' ")

a = Book.find(:first,
    :conditions => "author = 'Thomas, Dave'")

a = Book.find(:first,
    :conditions => [ "author = 'Thomas, Dave' and title = 'Programming Ruby'"])

a = Book.find(:all,
    :conditions => [ "author = 'Thomas, Dave'"])

name = 'Thomas, Dave'
a = Book.find(:first,
    :conditions => [ "author = ?", name])
```

Modifying & Deleting

```
a = Book.find_by_title('Programming Ruby')
a.author = 'Dave Thomas'
a.save
```

```
Book.find_by_author('Dave Thomas').destroy
```