IRON
HACK

Rails Params

What are params?

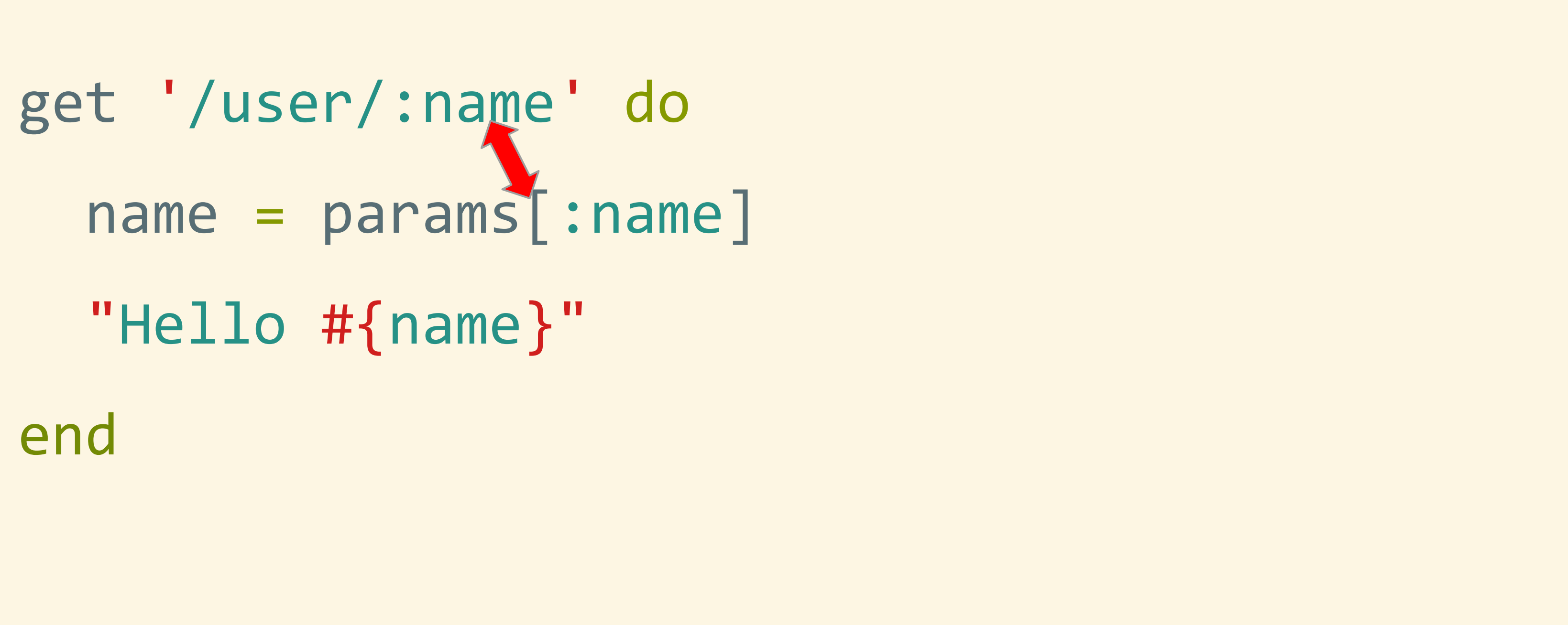
What are params?

Params (Short for parameters) is a hash used to send data from the client’s browser to the application server.

What are params?

We’ve already used params! Remember week 2 with Sinatra?

Params through a route



```
get '/user/:name' do
  name = params[:name]
  "Hello #{name}"
end
```

Params through a form

```
<form action="/user" method="POST">
  <input type="text" name="name">
  <button type="submit">Submit</button>
</form>
post '/user' do
  name = params[:name]
  "Hello #{name}"
end
```

Query String

```
# http://localhost:4567/user?name=Josh


get '/user' do
  name = params[:name]
  "Hello #{name}"
end
```

What are params?

We use params as a way to send extra information to our server. We need more information for the server to do its work.

What are params?

Think of user login. The user enters their username and password into a form, and then the server figures out if they're a valid user and the password is correct.

Params in Rails

Params in Rails

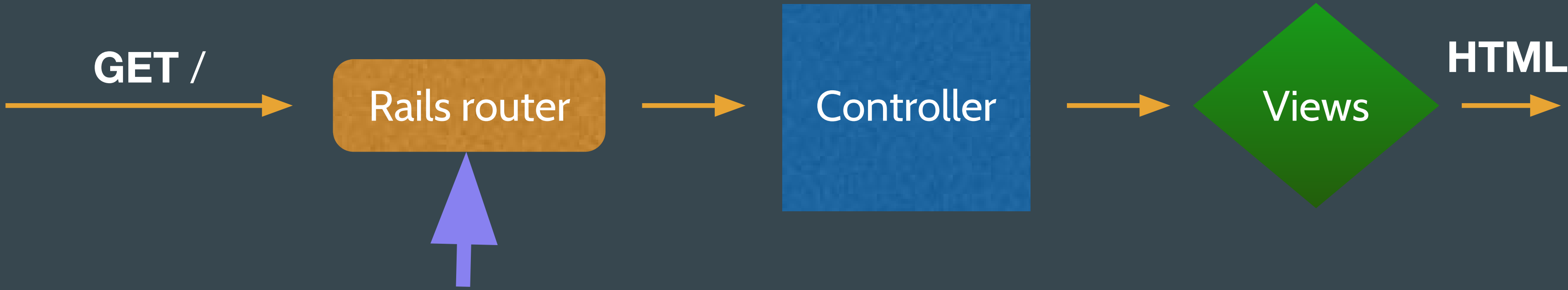
We use params in Rails in a very similar way to params in Sinatra

Params in Rails

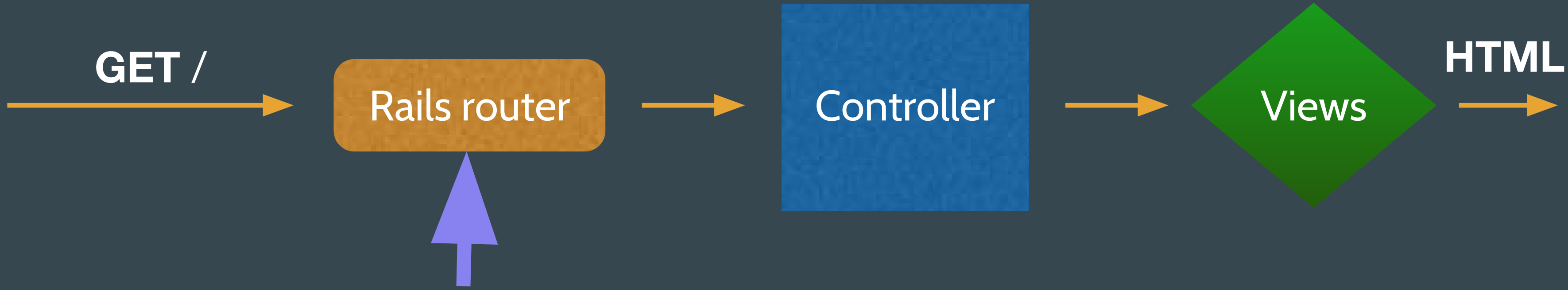
We can pass them through a route, through a form, or through a query string.

Route Params

Route Params

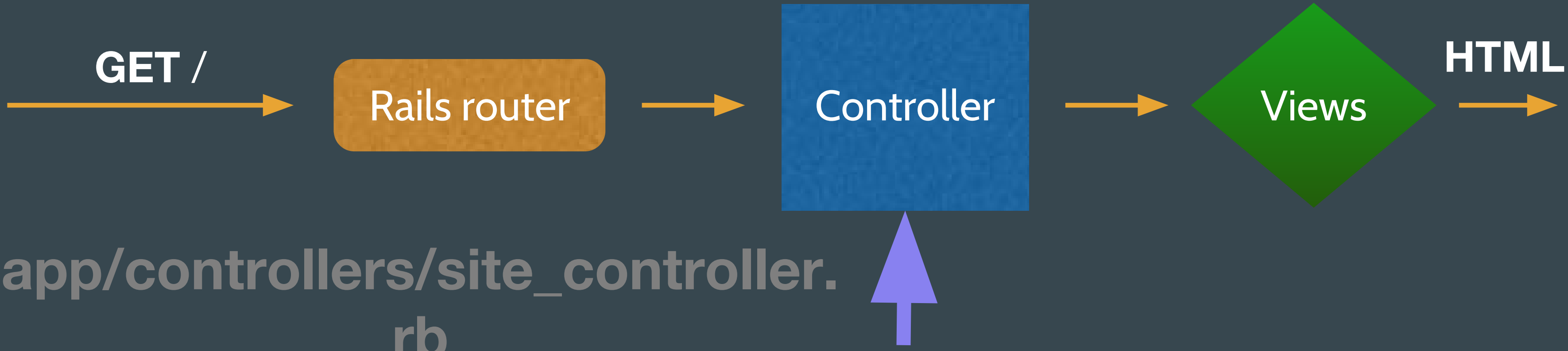


config/routes.rb



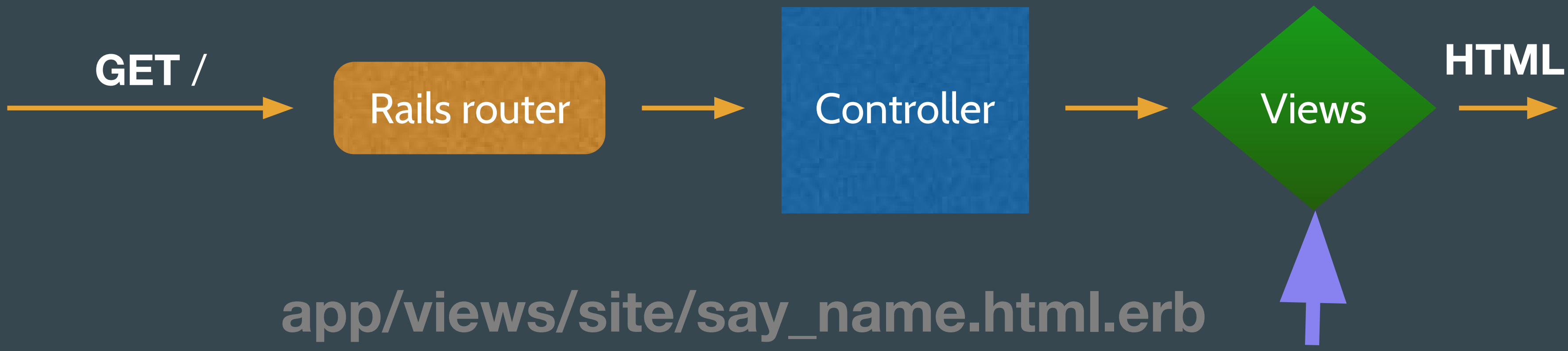
config/routes.rb

```
get '/say_name/:name', to: 'site#say_name'
```



app/controllers/site_controller.rb

```
class SiteController < ApplicationController
[...]
  def say_name
    @name = params[:name]
    render 'say_name'
  end
end
```



app/views/site/say_name.html.erb

```
<h1>Hello <%= @name %></h1>
```

Params in Rails

We can pass as many params through a route as we want.

Params through a route

(Don’t do this)

```
get '/other_route/:lots/:of/:params' => 'some_controller#some_action'
```

Form Params

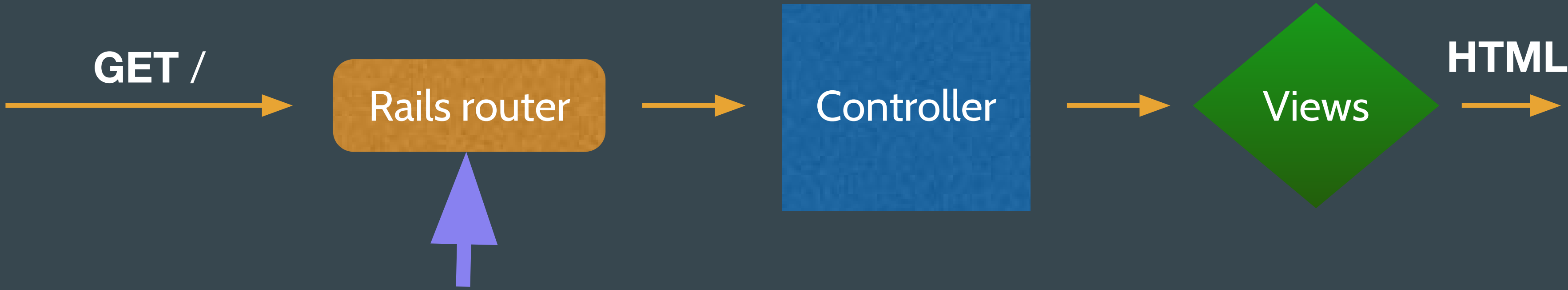
Params in Rails

Sometimes we need to get input from a user. This is where we can use a form. Let’s create our online calculator, except this time with Rails.

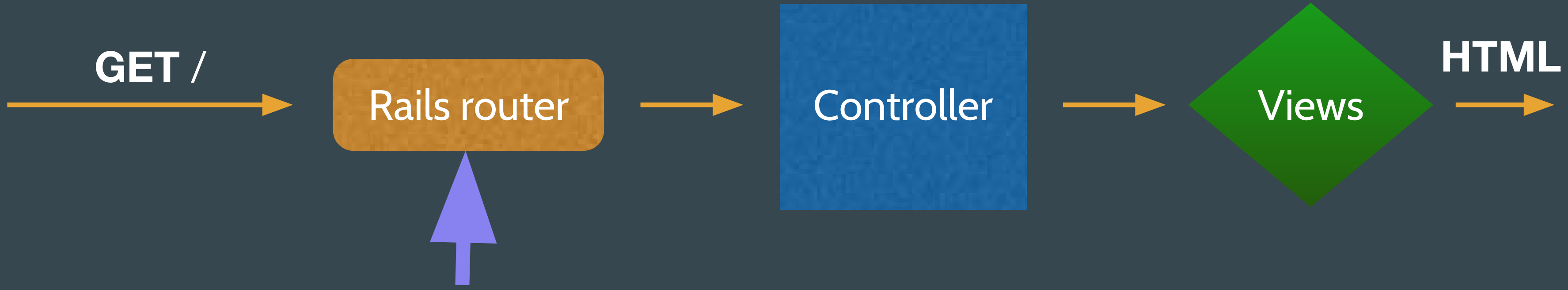
Form Params

First we need a route to render the form.

Form Params

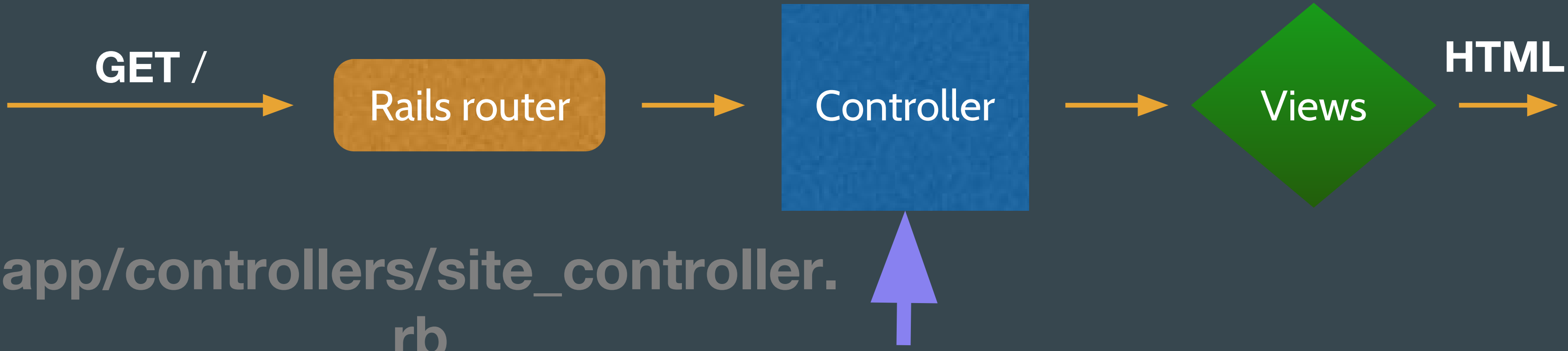


config/routes.rb



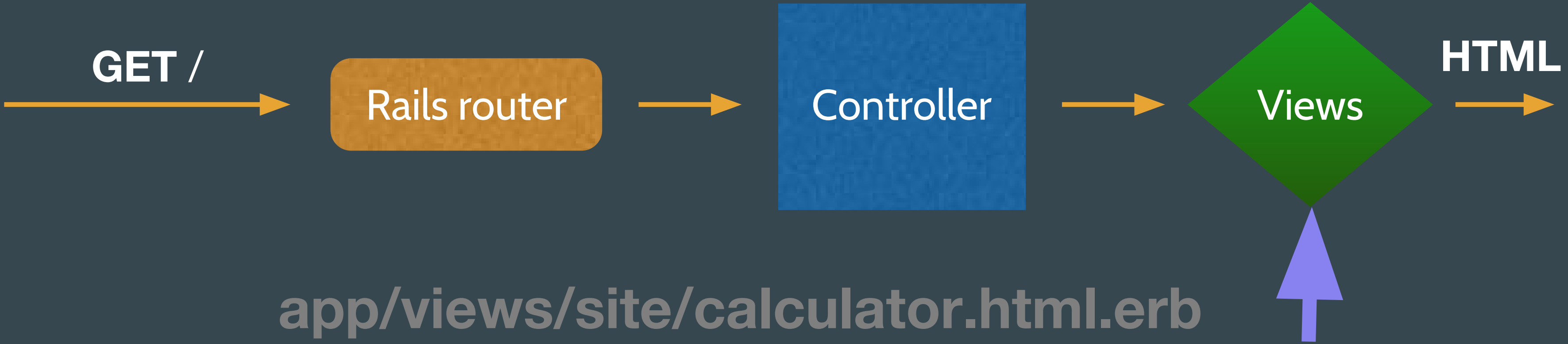
config/routes.rb

```
get '/calculator', to: 'site#calculator'
```



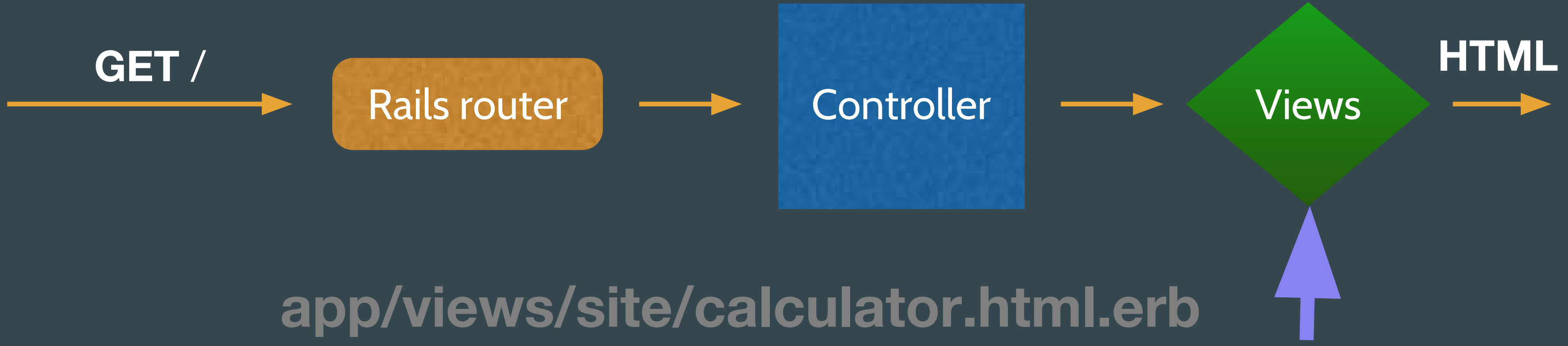
app/controllers/site_controller.rb

```
class SiteController < ApplicationController
[...]
  def calculator
    render 'calculator'
  end
end
```



app/views/site/calculator.html.erb

```
<form action="/calculate" method="POST">
  <input type="text" name="first_num" placeholder="First Number">
  <input type="text" name="second_num" placeholder="Second Number">
  <input type="hidden" name="authenticity_token" value="<%=
form_authenticity_token %>">
  <button type="submit">Add</button>
</form>
```



app/views/site/calculator.html.erb

```
<form action="/calculate" method="POST">
  <input type="text" name="first_num" placeholder="First Number">
  <input type="text" name="second_num" placeholder="Second Number">
  <input type="hidden" name="authenticity_token" value="<%=
form_authenticity_token %>">
  <button type="submit">Add</button>
</form>
```

Form Params

Forget the hidden input field here. We will learn about it later. For now, just make sure you include it in your form.

Form Params

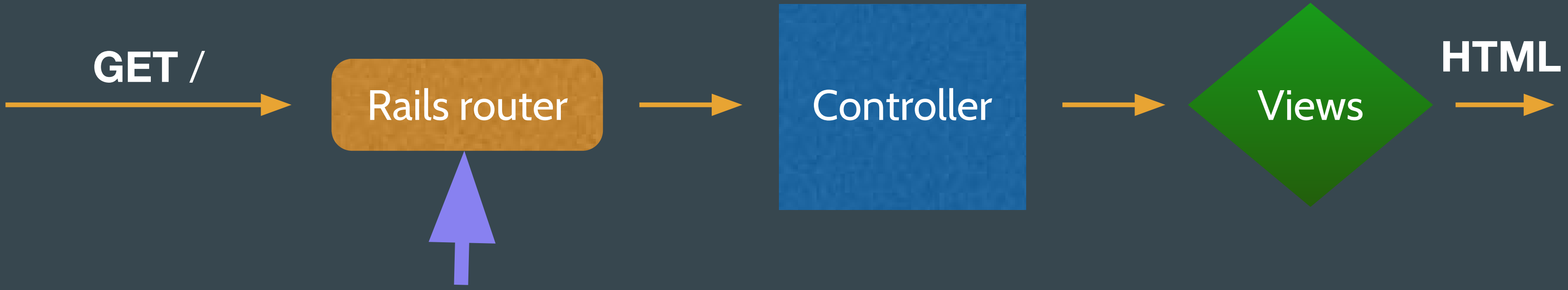
The *name* field in our form is very important. This will be accessed in our controller's action with `params[:name_field_here]`.

Form Params

Our form has an action of /calculate and a method of POST.

Form Params

Let’s handle the user’s input. We’re just going to add the two numbers together and display the result in our view.



GET /

Rails router

Controller

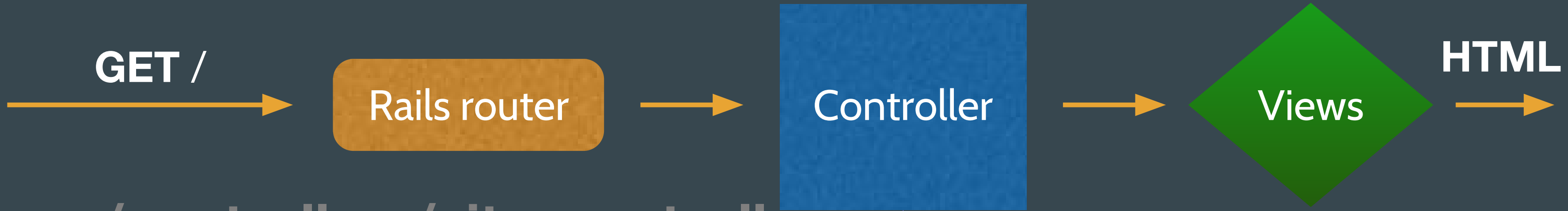
Views

HTML

config/routes.rb

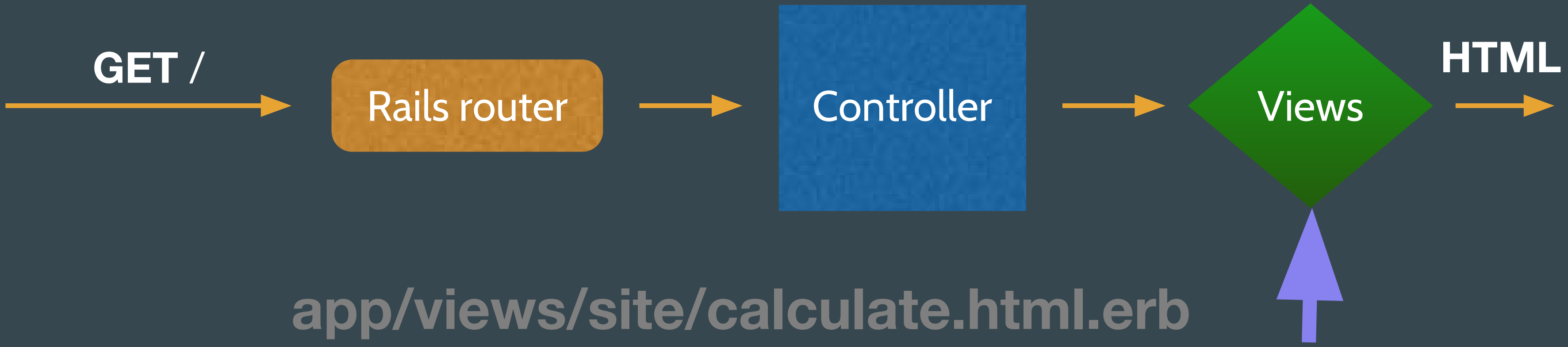
```
get '/calculator', to: 'site#calculator'

post '/calculate', to: 'site#calculate'
```



app/controllers/site_controller.rb

```
class SiteController < ApplicationController
[...]
  def calculate
    @num1 = params[:first_num].to_f
    @num2 = params[:second_num].to_f
    @result = @num1 + @num2
    render 'calculate'
  end
end
```



GET /

Rails router

Controller

Views

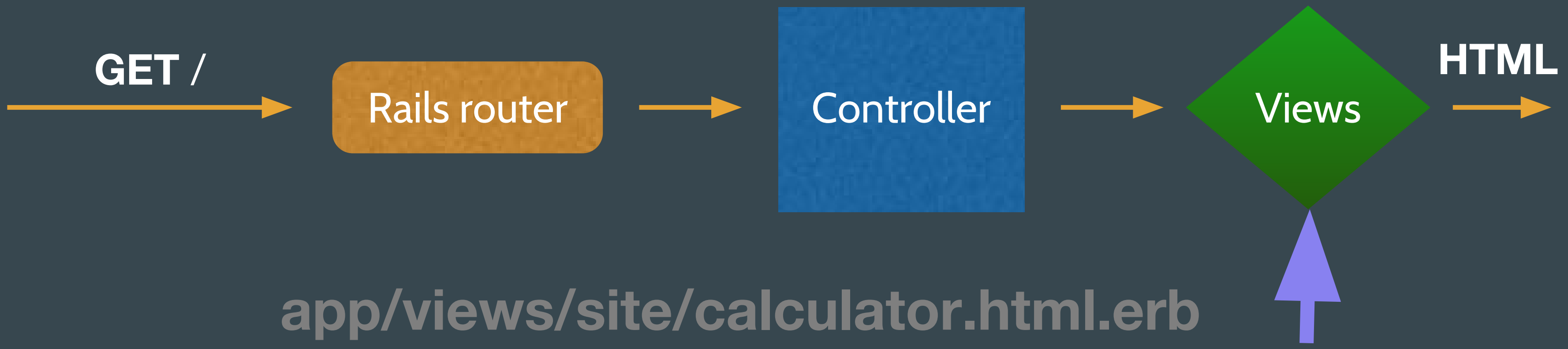
HTML

app/views/site/calculate.html.erb

```erb
The result of <%= @num1 %> + <%= @num2 %> is <%= @result %>
```

Form Params

But what happened in between?



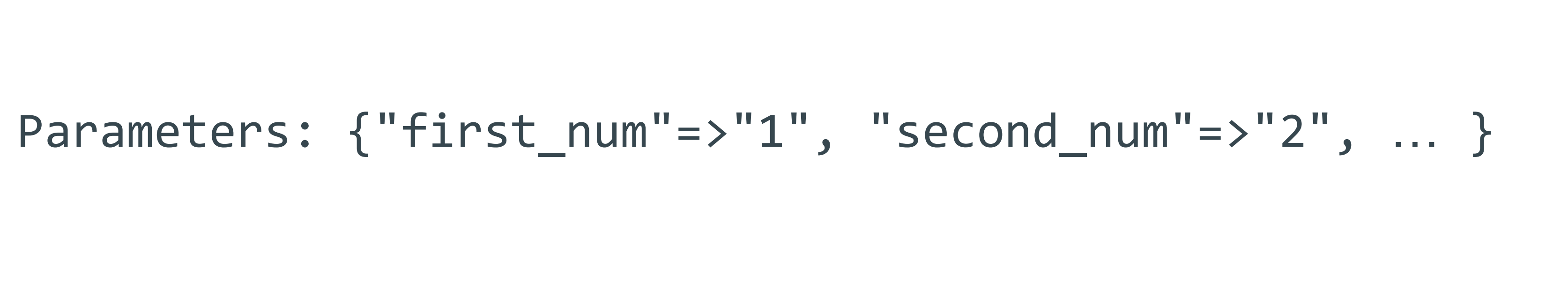
app/views/site/calculator.html.erb

```
<form action="/calculate" method="POST">
  <input type="text" name="first_num" placeholder="First Number">
  <input type="text" name="second_num" placeholder="Second Number">
  <input type="hidden" name="authenticity_token" value="<%=
form_authenticity_token %>">
  <button type="submit">Add</button>
</form>
```

Form Params

Your form packages up the information and sends it to the server in the form of params.

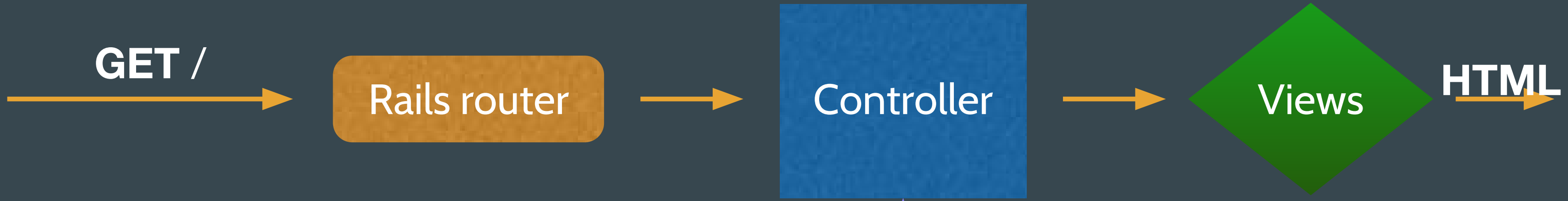
The Params Hash



```
Parameters: {"first_num"=>"1", "second_num"=>"2", … }
```

Form Params

And then we access the information in our controller action via params[:key].

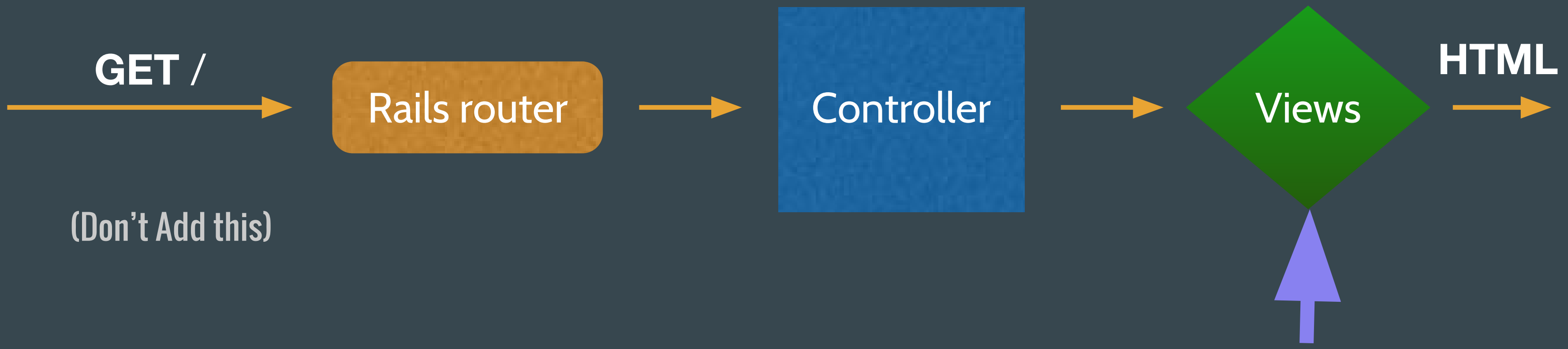


app/controllers/site_controller.rb

```
class SiteController < ApplicationController
[...]
  def calculate
    @num1 = params[:first_num].to_f
    @num2 = params[:second_num].to_f
    @result = @num1 + @num2
    render 'calculate'
  end
end
```

Form Params

How this hash is structured depends on the way our name field is structured in our form.

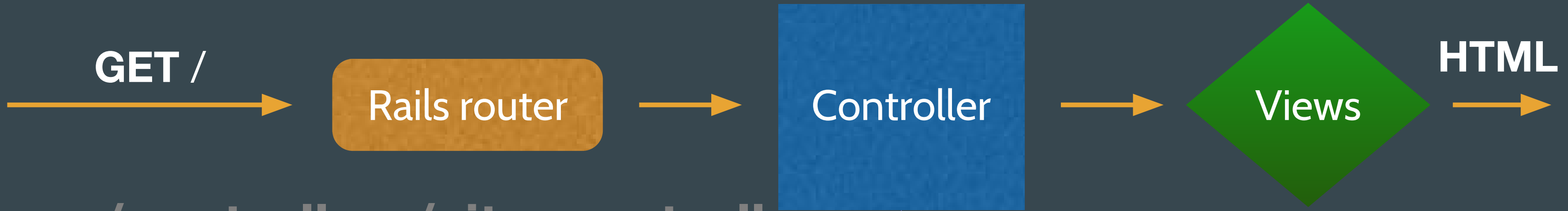


(Don’t Add this)

```
<form action="/login" method="POST">
  <input type="text" name="user[email]" placeholder="Email">
  <input type="password" name="user[password]" placeholder="Pass">
  <input type="hidden" name="authenticity_token" value="<%=
form_authenticity_token %>">
  <button type="submit">Login</button>
</form>
```

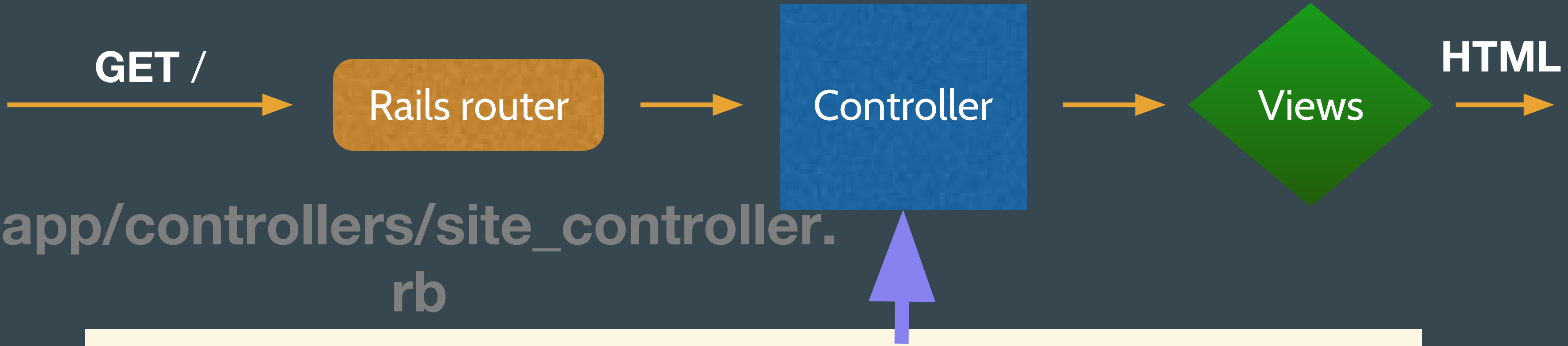
Nested Parameters

```
  Parameters: {"user"=>{"email"=>"izzy@ironhack.com",
"password"=>"[FILTERED]"}, … }
```



app/controllers/site_controller.rb

```
class SiteController < ApplicationController
[...]
  def login
    @email = params[:user][:email]
    @password = params[:user][:password]
    [...]
  end
end
```



app/controllers/site_controller.rb

```
class SiteController < ApplicationController
[...]
  def login
    @email = params[:user][:email]
    @password = params[:user][:password]
    [...]
  end
end
```

Form Params

This will come in handy shortly. Just remember, the name field is how you will reference the user’s input on the server.

Params