Ruby on Rails

Sessions Controller & View

login/new.html.erb

```
<div class="bs-docs-section">
  <div class="row">
                                              Es para obtener un
    <div class="col-lg-12">
      <div class="page-header">
                                              reviewer, y se hace
        <h1 id="forms">Login</h1>
                                              referencia a los fields
                                                                          Se hace submit a la
      </div>
                                              de reviewer
                                                                          url de session
    </div>
  </div>
  <div class="row">
    <div class="col-lg-6">
      <div class="well be component">
        <%= form for (:reviewer, url: sessions path, class: "form-horizontal") do |f| %>
          <div class="Torm group">
            <div><%= f.label :name , class: "col-lg-2 control-label" %> <br/> <%= f.text field :</pre>
              name , class: "form-control"%></div>
            <div><%= f.label :password %> <br/> <%= f.password field :password , class: "form-
              control"%></div>
          </div>
          <div class="actions"><%= f.submit "Login" , class: "btn btn-primary"%></div>
        <% end %>
      </div>
    </div>
  </div>
</div>
```

reviewer[name] / reviewer[password]

```
Login
                                        Z/CCOTF VOJDT WY--
                                      ▼<div class="form-group">
                                        ▼<div>
                                            <label class="col-lg-2 control-label" for=</pre>
  Name
                                            "reviewer name">Name</label>
 div | 238.68×71.67
                                            <br>
 Password
                                            <input class="form-control" type="text" name={\( 'reviewer[name \) \)</pre>
                      •••
  .....
                                            id="reviewer name"> == $0
                                          </div>
  Login
                                        ▶ <div>...</div>
                                       </div>
```

Session Controller

```
sessions controller.rb
    class SessionsController < ApplicationController</pre>
2
      def new
 3
        # Login Page - new.html.erb
      end
      def create
        reviewer = Reviewer.find by(name: params[:reviewer][:name])
 8
 9
        password = params[:reviewer][:password]
10
        if reviewer && reviewer.authenticate(password)
11
          session[:reviewer id] = reviewer.id
12
          redirect to root path, notice: "Logged in successfully"
13
        else
14
15
          redirect to login path, alert: "Invalid username/password combination"
16
        end
17
      end
18
      def destroy
19
20
       reset session # wipe out session and everything in it
        redirect to login path, notice: "You have been logged out"
21
      end
    end
```



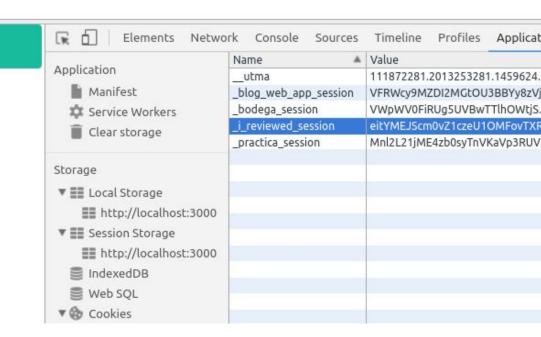
Logged in successfully

Logged in successfully

Listing Books

Name	Author	
Eloquent Ruby	Russ Olsen	ShowEditDestroy
Beginning Ruby	Peter Cooper	ShowEditDestroy
Metaprogramming Ruby 2	Paolo Perrotta	ShowEditDestroy
Design Patterns in Ruby	Russ Olsen	ShowEditDestroy
The Ruby Programming Langu	ageDavid Flanagar	ShowEditDestroy

New Book



Llaveando nuestra aplicación

Ya tenemos el login!

- Pero siempre podemos acceder via url a los demas recursos.
- Vamos a agregar un before_action en el ApplicationController (del cual todos los demas controllers heredan) que enviará de vuelta al login si todavía no estamos logeados.
- Pero si todo está bloqueado... el **SessionController** también hereda de Application controller..
- Los Controllers pueden sobre escribir el before_action con skip_before_action.

```
application controller.rb ×
    class ApplicationController < ActionController::Base</pre>
      # Prevent CSRF attacks by raising an exception.
      # For APIs, you may want to use :null session instead.
      protect from forgery with: :exception
      before action :ensure login
      protected
      def ensure login
        # Always go to login page unless session contains
        # reviewer id
        redirect to login path unless session[:reviewer id]
      end
13
    end
 sessions controller.rb .
    class SessionsController < ApplicationController</pre>
     Skip before action:ensure login, only: [:new, :create]
      def new
        # Login Page - new.html.erb
      end
```

10

11

Login



Entonces...

La página de login corresponde a un action new en el SessionController, pero utiliza atributos de reviewer.

Para "llavear" nuestra aplicación utilizamos el before_action en el ApplicationController.