In this lecture, we will discuss…

✧ Sessions and cookies

HTTP Is A Stateless Protocol

✧ HTTP is a stateless protocol

- Each new request even from the same browser knows nothing about a previous request that was made
- This means that even if a user makes a request – he will be treated as unknown on all the subsequent requests

✧ Cookies and Sessions to the rescue (keep state)

http://guides.rubyonrails.org/security.html#what-are-sessions-questionmark

Sessions in Rails

✧ Session is created and made available through a `session` hash

✧ The server sends the browser a cookie with the session information, which the browser stores and sends back to the server on all subsequent requests (until the session ends)

Restful Sessions Controller

✧ Session can be thought of as a resource – let's go ahead and create a RESTful sessions controller

```
~/sandbox/i_reviewed$ rails g controller sessions new create destroy -q
~/sandbox/i_reviewed$
```

config/routes.rb

FOLDERS

- ▼ 📂 i_reviewed
  - ▶ 📁 app
  - ▶ 📁 bin
  - ▼ 📂 config
    - ▶ 📁 environme
    - ▶ 📁 initializers
    - ▶ 📁 locales
    - 📄 applicatior
    - 📄 boot.rb
    - 📄 database.y
    - 📄 environme
    - 📄 routes.rb

routes.rb

```ruby
1  Rails.application.routes.draw do
2    root to: "books#index"
3
4    resources :books do
5      resources :notes, only: [:create, :destroy]
6    end
7    resources :sessions, only: [:new, :create, :destroy]
8
9  end
10
11
12
13
14
15
```

Sessions Controller Actions

✧ We can think of `new` action as login page and `destroy` as a logout page

✧ Thus, we'll need `new` (and `create`) actions to create a session and `destroy` action to destroy a session

✧ Let's map login/logout routes to make this more clear

config/routes.rb

```ruby
routes.rb                    ✱
Rails.application.routes.draw do
  root to: "books#index"

  resources :books do
    resources :notes, only: [:create, :destroy]
  end
  resources :sessions, only: [:new, :create, :destroy]

  get "/login" => "sessions#new", as: "login"
  delete "/logout" => "sessions#destroy", as: "logout"
end
```

This lets us refer to these routes in our code as login_path/logout_path or login_url/logout_url…

All Routes So Far

```
~/sandbox/i_reviewed$ rake routes
      Prefix Verb    URI Pattern                        Controller#Action
        root GET     /                                  books#index
  book_notes POST    /books/:book_id/notes(.:format)    notes#create
   book_note DELETE  /books/:book_id/notes/:id(.:format) notes#destroy
       books GET     /books(.:format)                   books#index
             POST    /books(.:format)                   books#create
    new_book GET     /books/new(.:format)               books#new
   edit_book GET     /books/:id/edit(.:format)          books#edit
        book GET     /books/:id(.:format)               books#show
             PATCH   /books/:id(.:format)               books#update
             PUT     /books/:id(.:format)               books#update
             DELETE  /books/:id(.:format)               books#destroy
    sessions POST    /sessions(.:format)                sessions#create
 new_session GET     /sessions/new(.:format)            sessions#new
     session DELETE  /sessions/:id(.:format)            sessions#destroy
       login GET     /login(.:format)                   sessions#new
      logout DELETE  /logout(.:format)                  sessions#destroy
```

Summary

✧ Sessions and cookies make the interaction between browser and server stateful

✧ You can think of Sessions as yet another resource

✧ Custom routes - http://guides.rubyonrails.org/routing.html

What's Next?

✧ Sessions Controller and View