Ruby / Rails

Lesson 00. Web Apps. MVC. Rails

by Yuriy Bezgachnyuk, August 2022

softserve

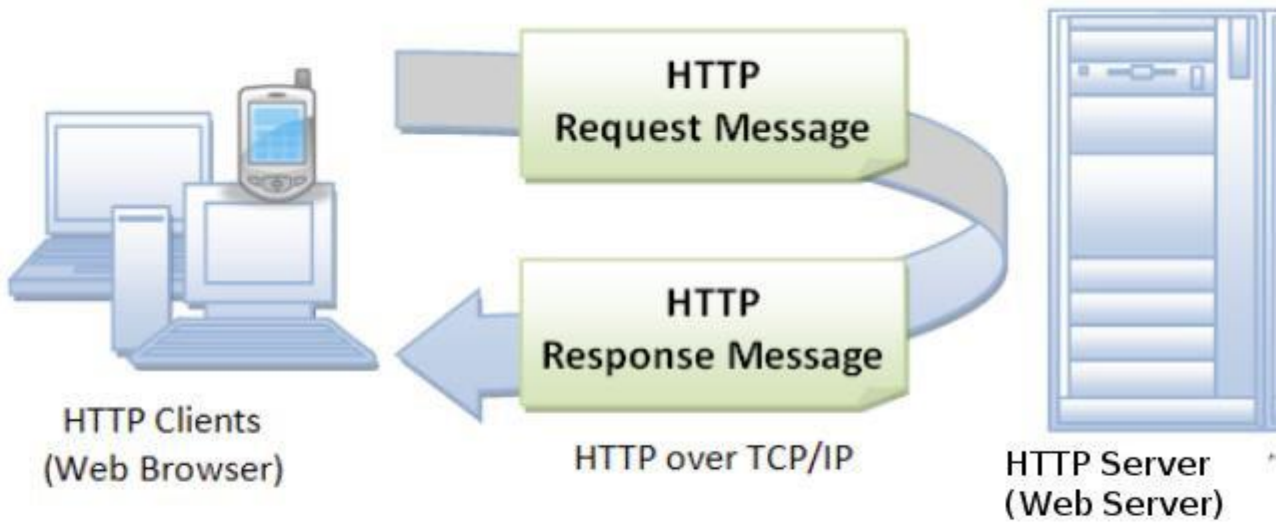
AGENDA

- HTTP Protocol (just for remind)
- Web Apps Architecture
  - 3-Tier Architecture
  - Classical Web Applications
- MVC Pattern
- Ruby on Rails framework

HTTP Protocol

- Request-response mechanism:
  - Transaction is initiated by a client sending a *request* to server
  - Server generates a *response*
- Resource Identification
  - Each HTTP request includes a **URI (Uniform Resource Identifier)**
- Statelessness
  - The server does not maintain any information about the transaction
- Meta data support

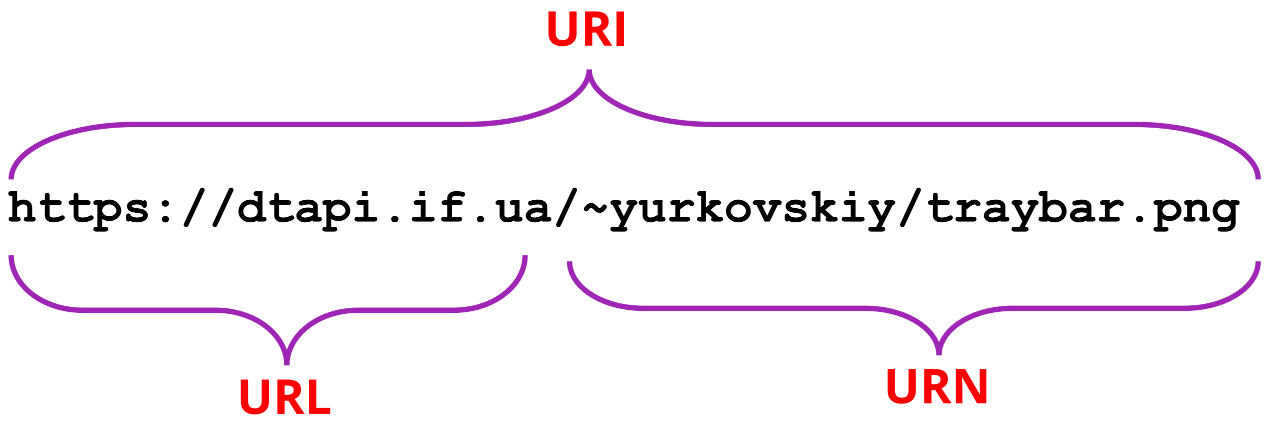
  Metadata about information can be exchanged in the messages

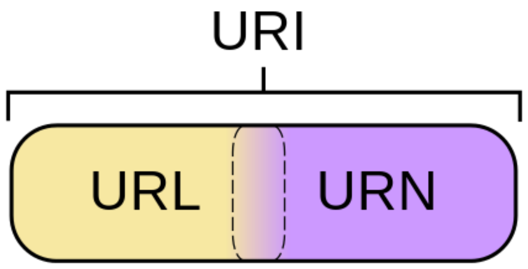


softserve

HYPERLINK

- **A uniform resource locator (URL)** – is a specific character string that constitutes a reference to a resource.
- **A uniform resource identifier (URI)** – is a string of characters used to identify a name of a web resource.





softserve

HTTP METHODS

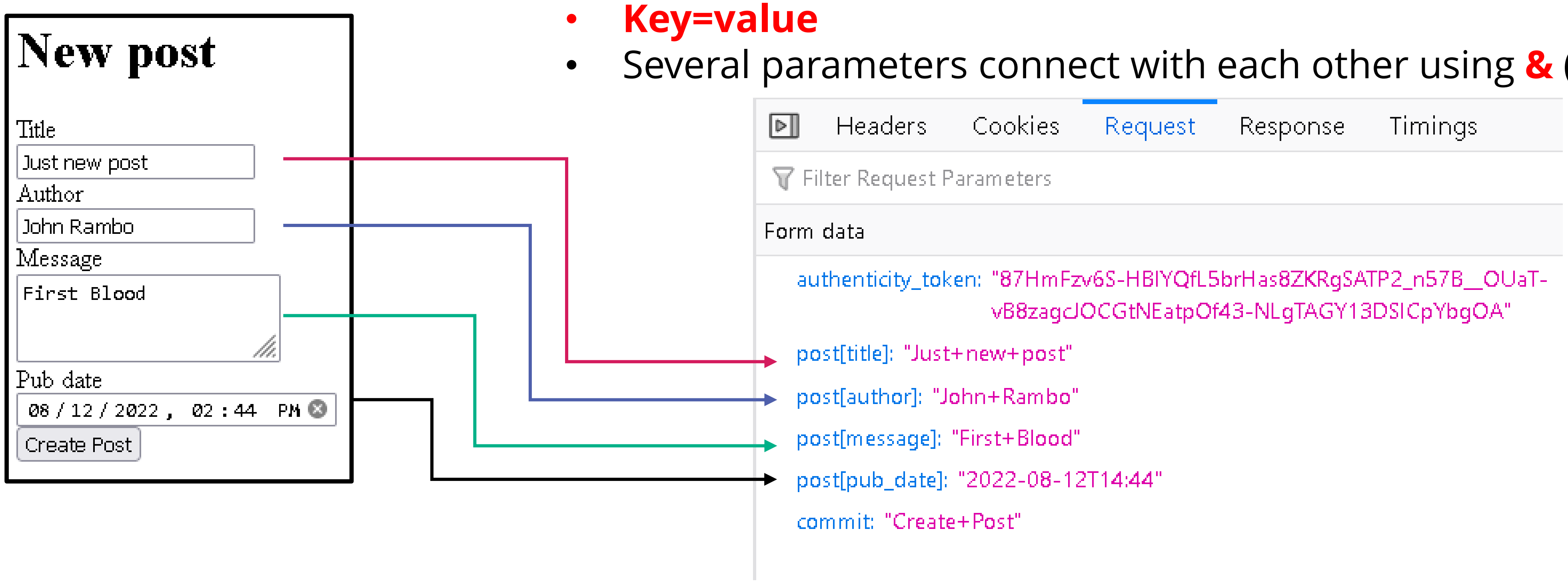
- HTTP defines a set of **request methods** to indicate the desired action to be performed for a given resource. Although they can also be nouns, these request methods are sometimes referred to as **HTTP verbs**. Each of them implements a different semantic, but some common features are shared by a group of them: e.g. a request method can be safe, idempotent, or cacheable.

GET POST PUT DELETE HEAD OPTIONS TRACE CONNECT PATCH

softserve

HOW DATA TRANSFER TO SERVER

- Data always transfers to the server as a dictionary structure
  - **Key=value**
  - Several parameters connect with each other using **&** (ampersand symbol)



New post

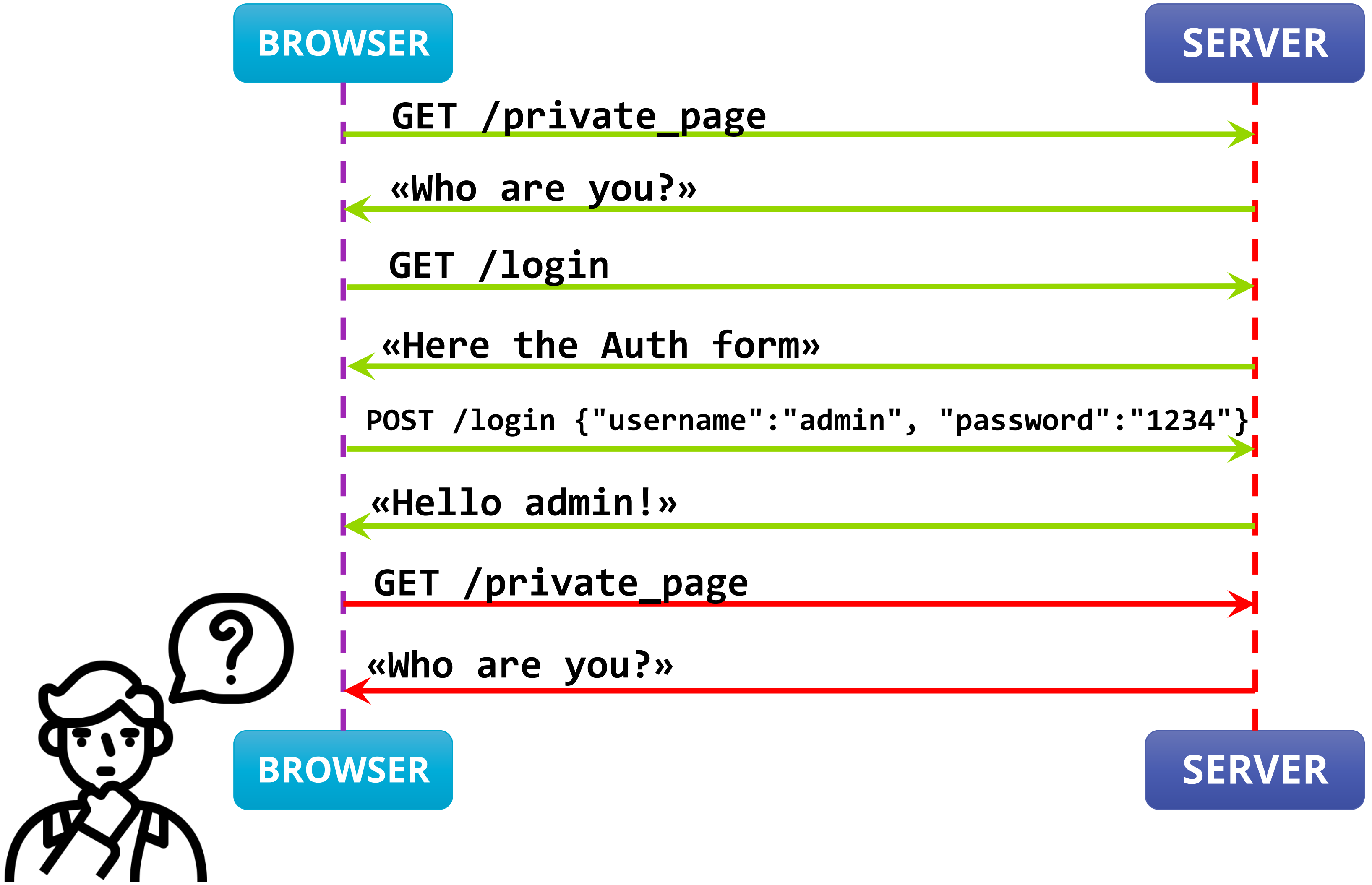
Title: Just new post
Author: John Rambo
Message: First Blood
Pub date: 08 / 12 / 2022 , 02 : 44 PM

Form data

```
authenticity_token: "87HmFzv6S-HBIYQfL5brHas8ZKRgSATP2_n57B__OUaT-
                     vB8zagcJOCGtNEatpOf43-NLgTAGY13DSICpYbgOA"
post[title]: "Just+new+post"
post[author]: "John+Rambo"
post[message]: "First+Blood"
post[pub_date]: "2022-08-12T14:44"
commit: "Create+Post"
```

```
&post%5Btitle%5D=Just+new+post&post%5Bauthor%5D=John+Rambo&post%5Bmessage%5
D=First+Blood&post%5Bpub_date%5D=2022-08-12T14%3A44&commit=Create+Post
```

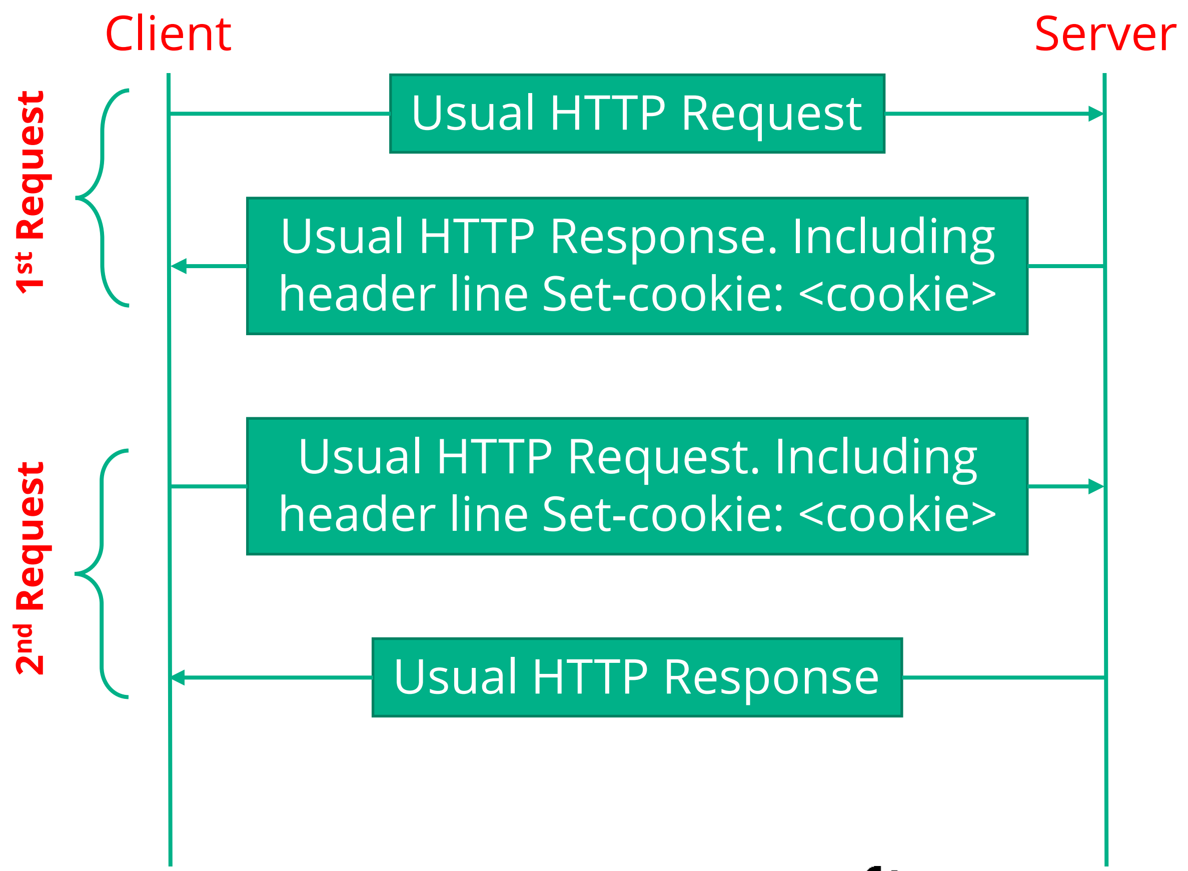
STATELESSNESS



softserve

COOKIES

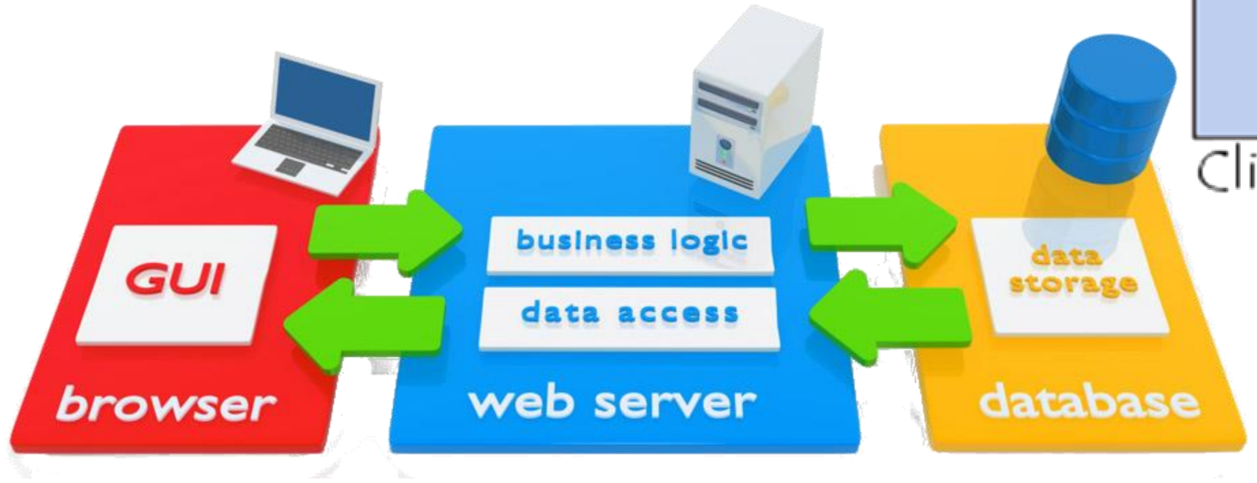
- HTTP is **statelessness** protocol server doesn't maintain information about transaction
- **Cookies** manage state maintenance by shifting the burden to client
- Cookies are transmitted in **clear text** (security issue)

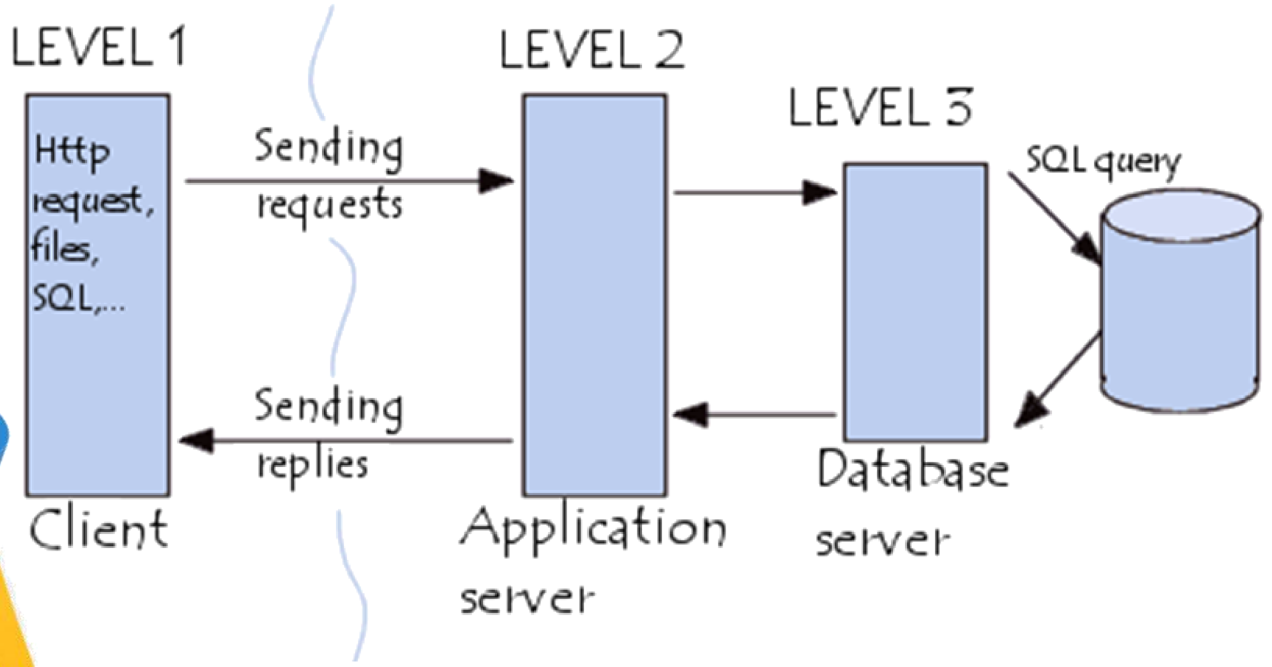

Client
Server
1st Request
Usual HTTP Request
Usual HTTP Response. Including header line Set-cookie: <cookie>
2nd Request
Usual HTTP Request. Including header line Set-cookie: <cookie>
Usual HTTP Response

softserve

3-TIER ARCHITECTURE

- Level 1: Client → Web browser
- Level 2: Web Server
- Level 3: Database Server





softserve

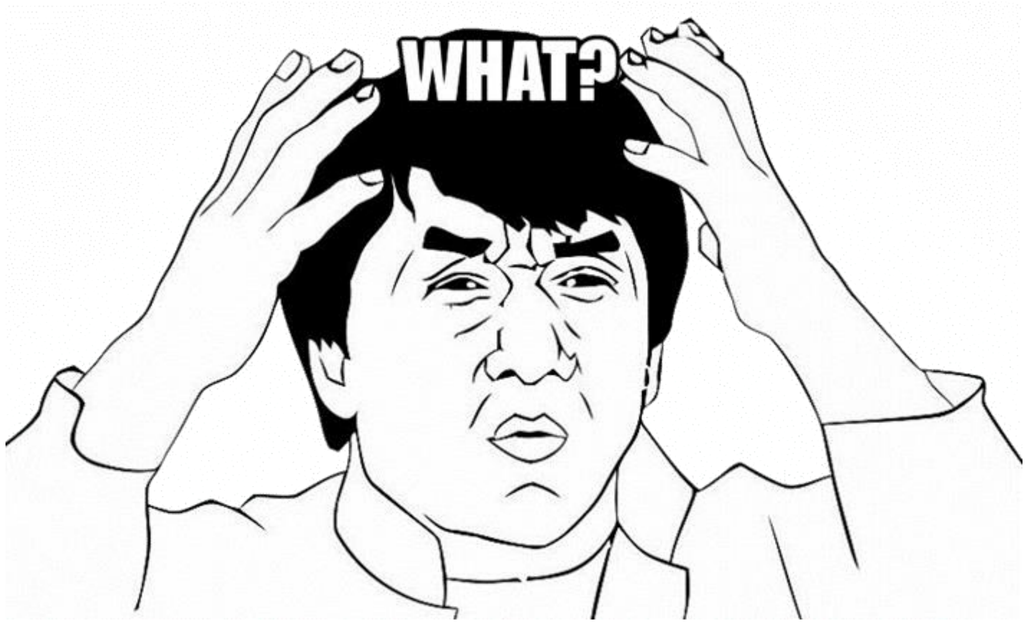
CLASSIC WEB APPLICATION

- Classic web application has a structure which was shown on the previous slide
- Database Server – store data
- Web Server – contains program code of the system which provides **business logic** and **data access** layer (interaction between web server and database)
- There is no explicit separation between **Frontend** and **Backend**
- All actions are handling on the server (Backend)
- Backend code dynamically generates web pages based on a user's data

BACKEND / FRONTEND

- **Backend** – also called as Server-side development
- **Frontend** – also called as Client-side development

- **What does it mean?**
  - Server-side – the program code is running on the **server** machine (by/under web-server)
  - Client-side – the program code is running on the **client** machine (by/under web-browser)



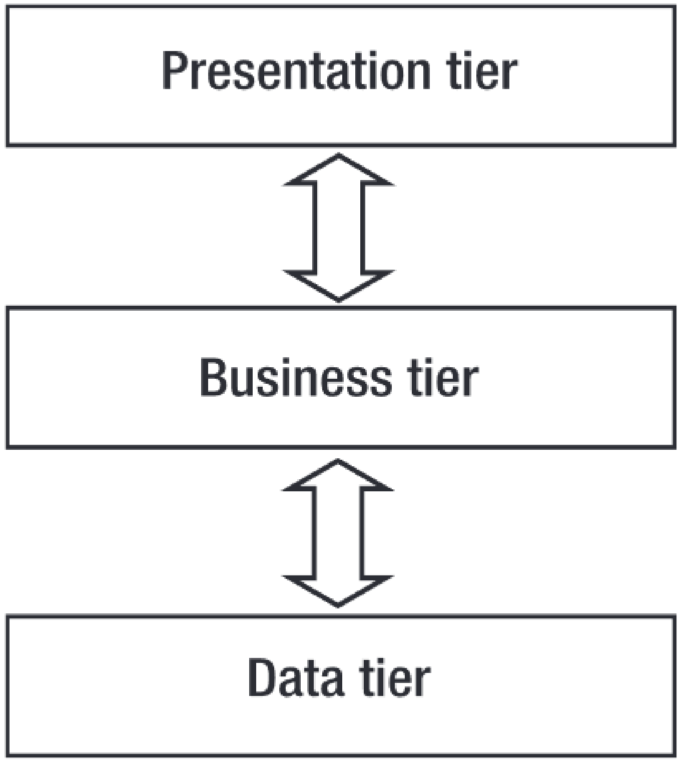
softserve

MVC PATTERN

softserve

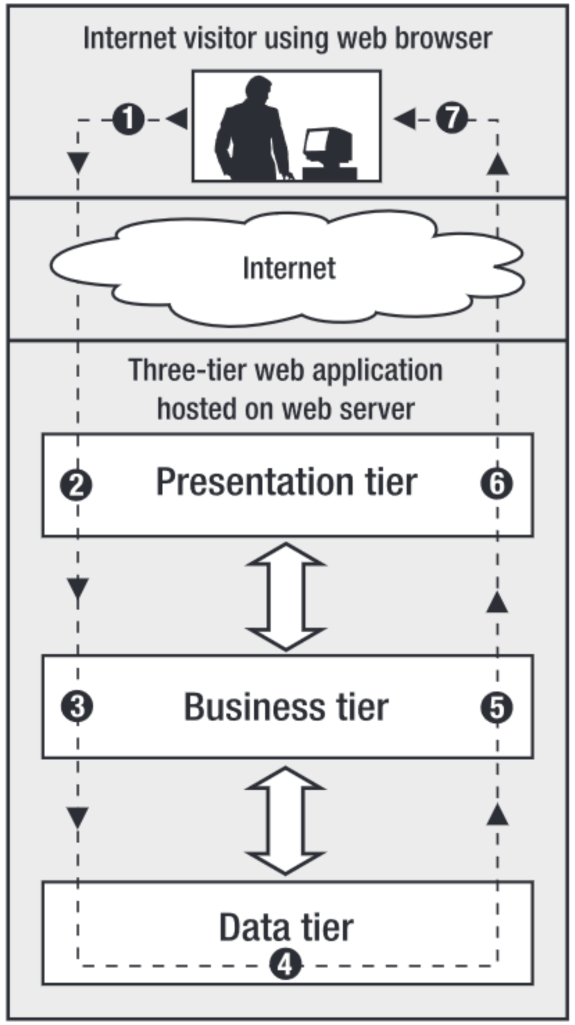
THE MAGIC OF 3-TIER ARCHITECTURE

- Generally, the architecture refers to the way we **split the code** that implements a feature of the application into **separate components** based on what they do and grouping each kind of component into a single logical tier
- 3-Tier Architecture (Layers)
  - Presentation tier
  - Business tier
  - Data tier



softserve

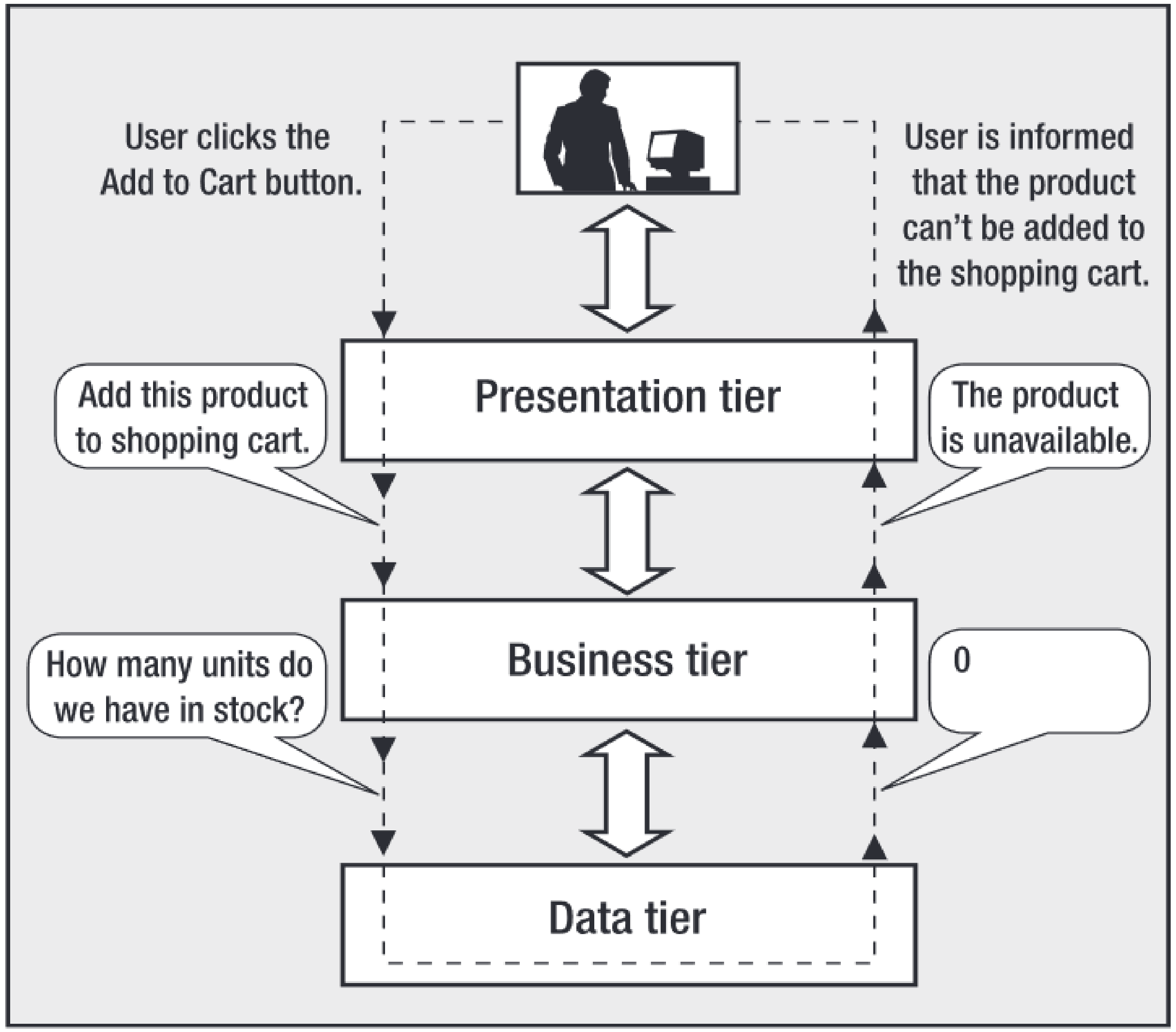
DATA FLOW

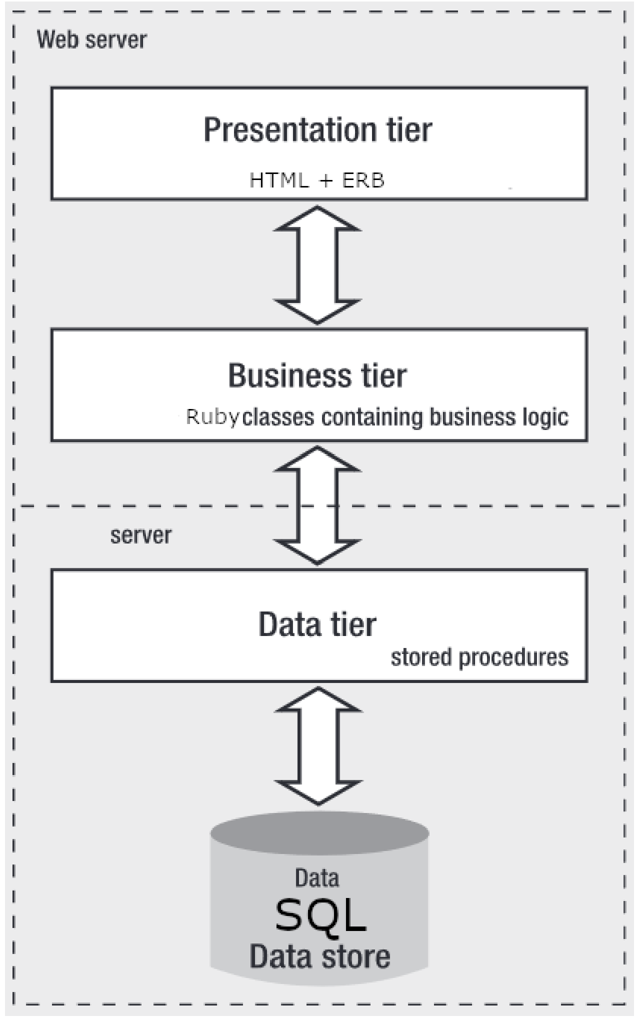

Internet visitor using web browser
1
7
Internet
Three-tier web application hosted on web server
2
Presentation tier
6
3
Business tier
5
Data tier
4

- **(1)** Client generates request to the server
- **(2)** Request is processing under presentation layer (ex.: click on button)
- **(3)** Processing the data from presentation layer
- **(4)** Communication with Database
- **(5)** Processing the data which was returned from data tier (ex.: some Result Set should be converted to another format)
- **(6)** Presentation tier dynamically generating the HTML-layout depends on data from business tier

softserve

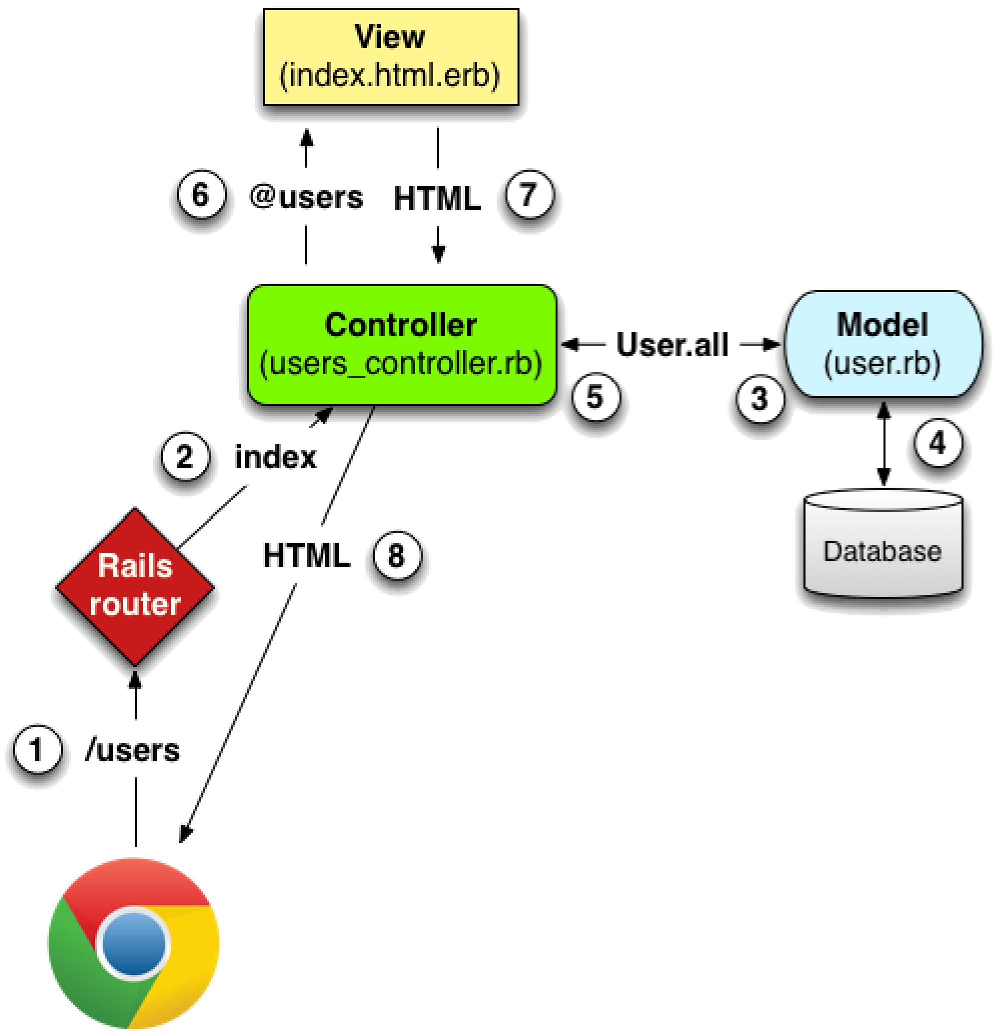
“REAL” EXAMPLE

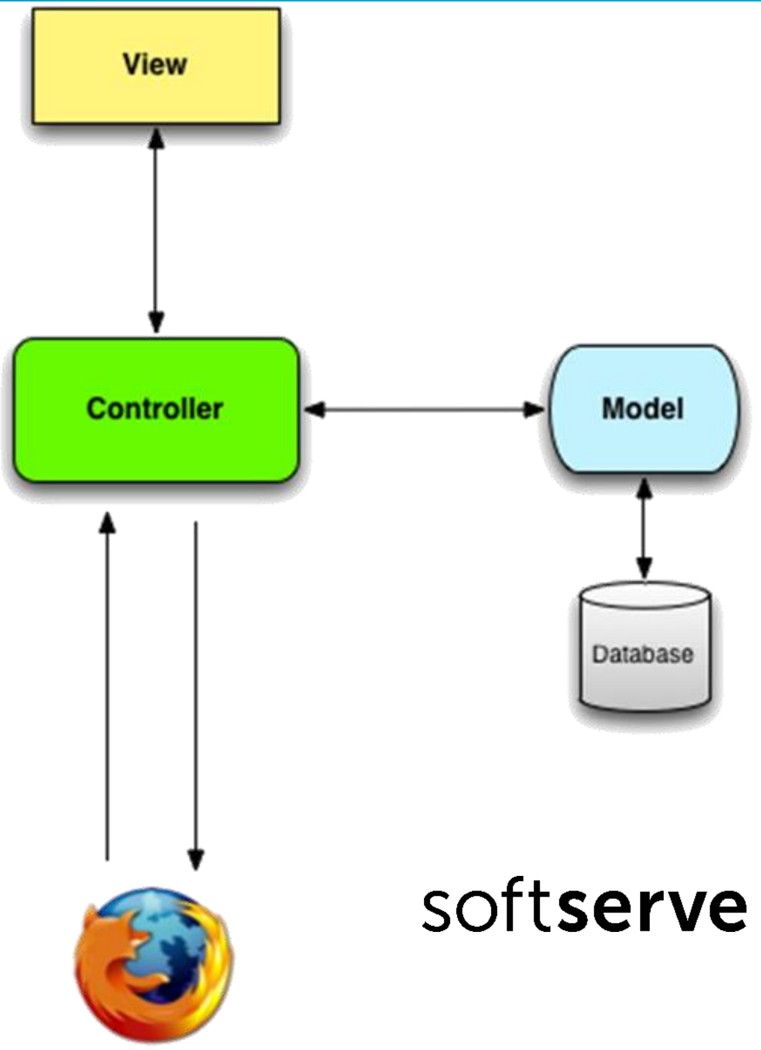

User clicks the Add to Cart button.
User is informed that the product can’t be added to the shopping cart.
Add this product to shopping cart.
Presentation tier
The product is unavailable.
How many units do we have in stock?
Business tier
0
Data tier



softserve

MVC





softserve

RUBY ON RAILS

softserve

Rails framework

- Which tools are needed for correct installation
  - Ruby interpreter (environment)
  - Node.js
  - Yarn package manager (or npm [part of node.js])
  - Run: **`gem install rails`**
- How to create new project
  - Run: **`rails new <project_name>`**

STRUCTURE [DIRECTORIES]

File / Directory	Description
`app/`	Contains main code of your project
`bin/`	Contains the rails script that starts your app and can contain other scripts you use to set up, update, deploy, or run your application.
`config/`	Contains configuration for your application's routes, database, and more.
`db/`	Contains your current database schema, as well as the database migrations
`lib/`	Extended modules for your application
`log/`	Application log files
`public/`	Contains static files and compiled assets. When your app is running, this directory will be exposed as-is
`test/`	Unit tests, fixtures, and other test apparatus
`tmp/`	Temporary files (like cache and pid files)
`Gemfile [.lock]`	These files allow you to specify what gem dependencies are needed for your Rails application. These files are used by the Bundler gem
`config.ru`	Rack configuration for Rack-based servers used to start the application

softserve

“app/” Directory

File / Directory	Description
`assets/`	CSS, JS, images
`controllers/`	Controller classes
`helpers/`	Helpers
`models/`	Model classes
`views/`	View templates, partials, layouts

- Most of your activities will be focused inside **app/** directory

softserve

Rails command line utility

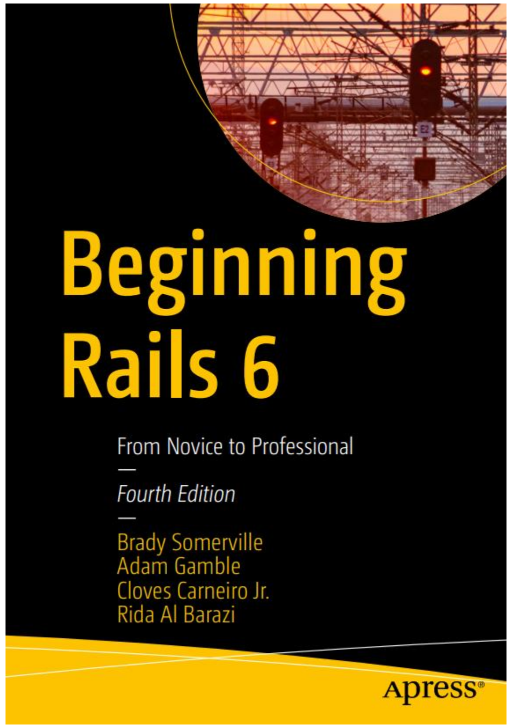
- The main purpose of **rails** utility – it's a set of generators which help to create skeleton, components of RoR application (helping to you avoiding a lot of **boring** jobs ☺)

Command	Purpose
`rails new my_app`	Create new blank of project
`rails server | rails s`	Run development server
`rails console | rails c`	Run interactive console
`rails db:create`	Create new project's database
`rails db:migrate`	Run migration scripts (for current environment)
`rails --tasks`	List of all available commands
`rails g scaffold <...>`	Generating "full prototype" of the project

- Cool Rails developers not recommend to use **scaffolding** for project ☺

REFERENCES & SOURCES

- Official Rails guide https://guides.rubyonrails.org/
- Ruby Garage community https://rubygarage.github.io/
- Tutorialspoint https://www.tutorialspoint.com/ruby-on-rails/

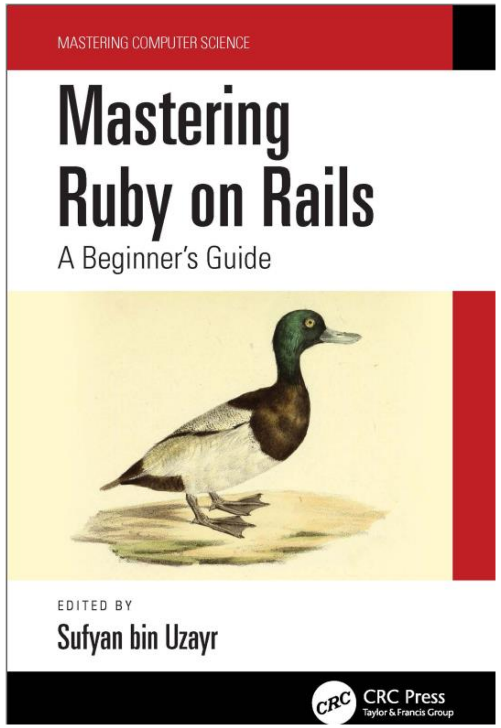

Beginning Rails 6

From Novice to Professional

Fourth Edition

Brady Somerville
Adam Gamble
Cloves Carneiro Jr.
Rida Al Barazi

Apress

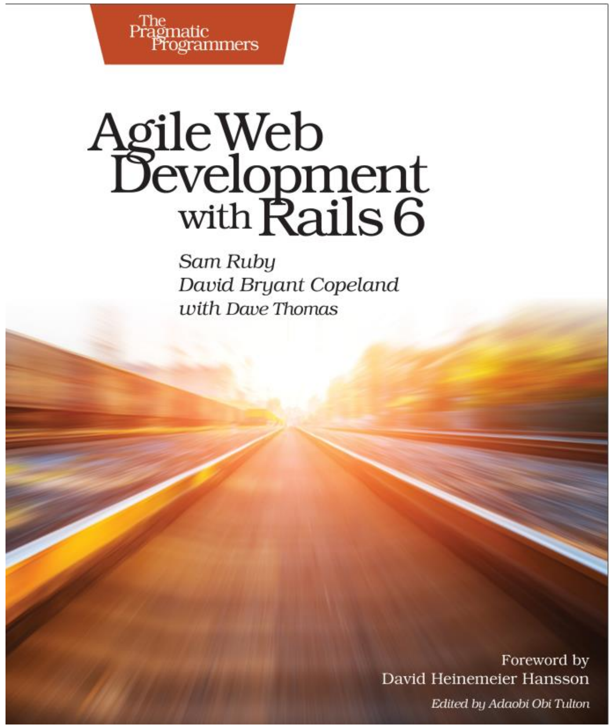

MASTERING COMPUTER SCIENCE

Mastering Ruby on Rails

A Beginner's Guide

EDITED BY
Sufyan bin Uzayr

CRC Press
Taylor & Francis Group

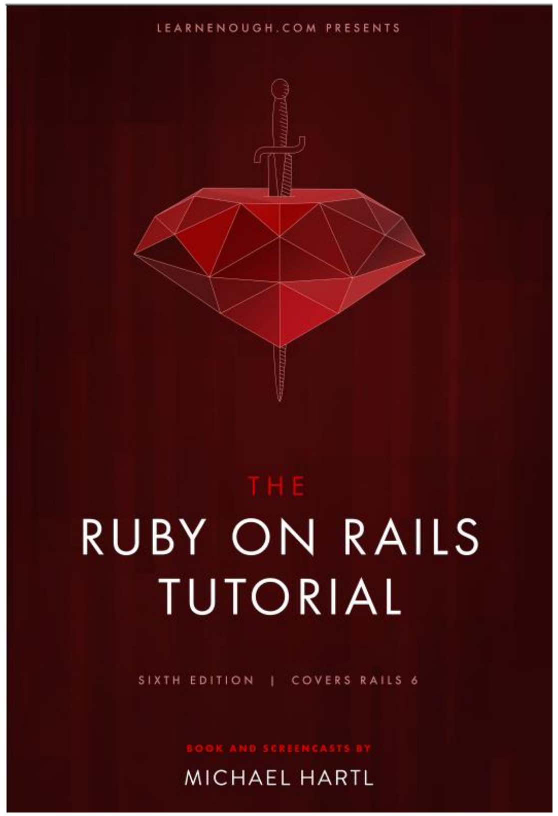

The Pragmatic Programmers

Agile Web Development with Rails 6

Sam Ruby
David Bryant Copeland
with Dave Thomas

Foreword by
David Heinemeier Hansson

Edited by Adaobi Obi Tulton


LEARNENOUGH.COM PRESENTS

THE RUBY ON RAILS TUTORIAL

SIXTH EDITION | COVERS RAILS 6

BOOK AND SCREENCASTS BY
MICHAEL HARTL

softserve

TO THE FUTURE
softserve