code|cademy

[My Path](#)
[Catalog](#)

[Codecademy Resources](#) > ruby
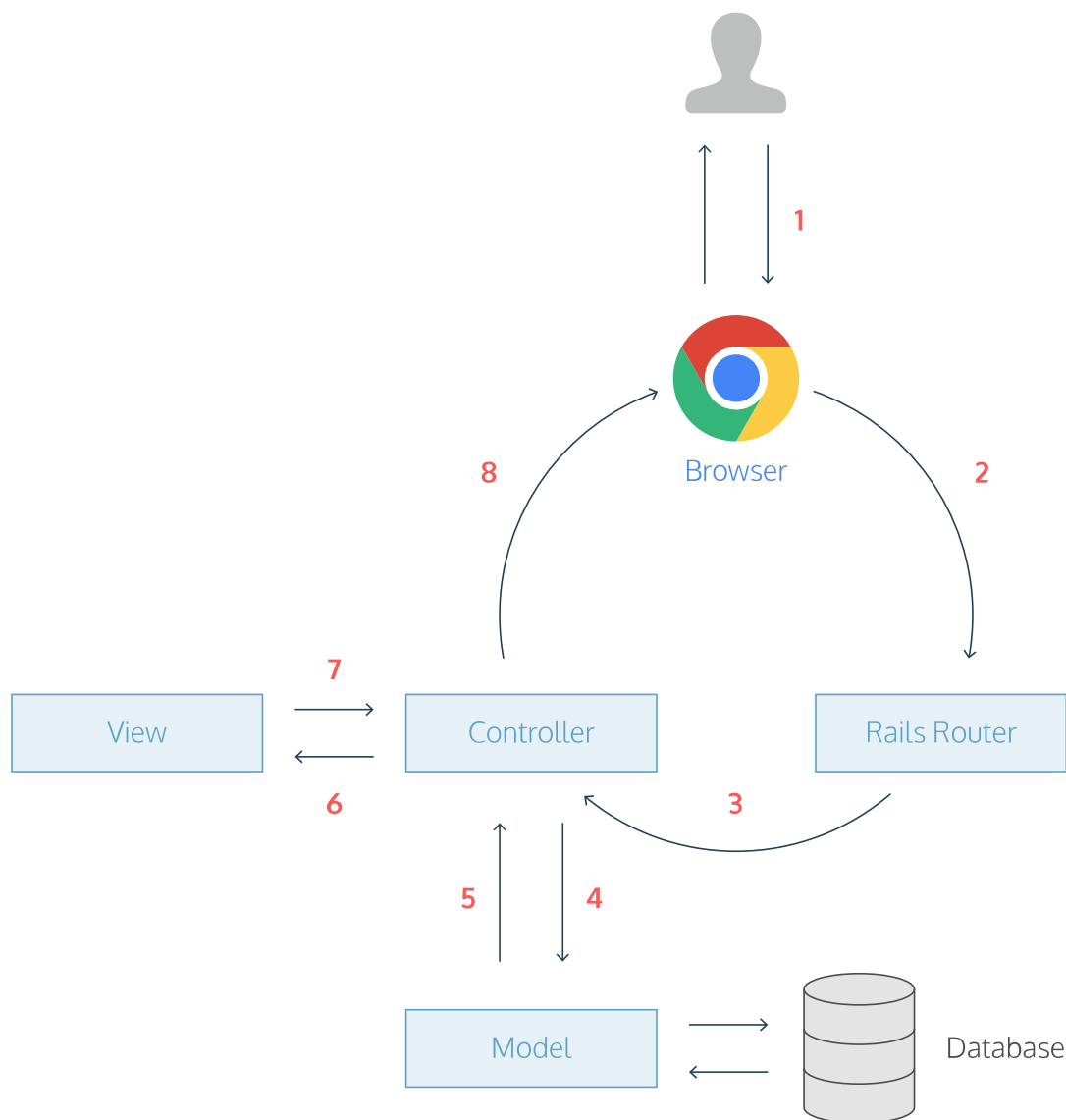
Request-Response Cycle II

Guide for navigating a dynamic Rails app.

▼

Background

When developing a Rails app, the **request/response cycle** is a useful guide to see how all the app's files and folders fit together. The request/response cycle traces how a user's request flows through the app. Understanding the request/response cycle is helpful to figure out which files to edit when developing an app (and where to look when things aren't working).

How it Works

1. A user opens his browser, types in a URL, and presses Enter. When a user presses Enter, the browser makes a request for that URL.
2. The request hits the Rails router (**config/routes.rb**).
3. The router maps the URL to the correct controller and action to handle the request.
4. The action receives the request, and asks the model to fetch data from the database.
5. The model returns a list of data to the controller action.
6. The controller action passes the data on to the view.
7. The view renders the page as HTML.
8. The controller sends the HTML back to the browser. The page loads and the user sees it.

In this way, the request/response cycle is a useful way to see how a Rails app's files and folders are for and how they fit together.

code_cademy

Teaching the world how to code.

Company

- [About](#)
- [Stories](#)
- [We're hiring](#)
- [Blog](#)

Resources

- [Articles](#)
- [Schools](#)

Learn To Code

- [Make a Website](#)
- [Make an Interactive Website](#)
- [Learn Rails](#)
- [Ruby on Rails Authentication](#)
- [Learn AngularJS](#)
- [Learn the Command Line](#)
- [Learn SQL](#)
- [SQL: Analyzing Business Metrics](#)
- [Learn Java](#)
- [Learn Git](#)

- [HTML & CSS](#)
- [JavaScript](#)
- [jQuery](#)
- [PHP](#)
- [Python](#)
- [Ruby](#)
- [Learn APIs](#)

[Privacy Policy](#) [Terms](#)
Made in NYC © 2016 Codecademy

English ▼