jim-clark / **guide-to-add-feature-to-web-app.md**

Last active 8 days ago • Report abuse

☆ Star

<> **Code** -o- Revisions 5 ☆ Stars 3 Forks 4

guide-to-add-feature-to-web-app.md

Guide to Add a Feature to a Web App

Intro

This guide is generic in that it does pertain to any particular web framework, e.g., Express.

When the time comes to implement a user story (feature) in a typical web application, following the steps below usually is a good approach to take in most cases...

Step 1 - Identify the "Proper" Route

If there's a RESTful/Resourceful route that's applicable - use it!

Use this guide to identify the CRUD operation or purpose similar to the feature you are implementing to find the proper route.

Of course, you will substitute the data resource you are CRUDing. For example: `GET /students` would be the proper route if I wanted to see a list of the *students* resource.

Then make note of the **HTTP method** and the **URI/Path** of the route.

Step 2 - Create the UI

Now that the route has been identified, you need to create the UI that is going to send an HTTP request that matches that route.

Examples include:

- Hyperlinks:
  See a Form to enter a new Student
  `<a href="/students/new">Add a Student</a>`
- Forms:
  Create a Student
  `<form action="/students" method="POST">`

Step 3 - Define the Route

Okay, so the UI is in place to send the HTTP request to the server when the user interacts with it.

Now you have to ensure that there's a route defined on the server that matches that route.

When the server receives an HTTP request matching a route that's been defined, the web framework is going know what code to run because you define routes for the purpose of mapping HTTP requests to controller actions/methods...

Step 4 - Code the Controller Action

So, the route definition is going to result in a controller action/method being run!

Controller actions are responsible for performing CRUD and providing data to views if necessary.

However, the last thing a controller action will do is respond to the client by...

Step 5 - Respond to the Client's HTTP Request

In the case of a CRUD **read** operation, the controller action will pass the data retrieved from the database to a view to be rendered and returned to the client. **Code that view if need be.**

In case of a CRUD **create, update or delete**; the action likely be responding to the client by sending a redirect to an appropriate URL.