

Control Flow for Post Creation

Control flow means the order in which control is passed or the order in which code is executed once a particular event/action is triggered.

For example, when we search for a URL inside the browser. The control first goes to the main index.js file which maps the request to the router. The router then triggers the appropriate controller based on the URL request. Further, this controller performs the associated actions like rendering views from the views folder, etc.

Now, while creating a post and hitting the submit button, an action is triggered, and further, there's a control flow associated with it.

What is the control flow once the submit button of the create post form is triggered?

When the submit button of the create-post form on the home page is triggered:

1. the control first goes to the JS file (home_posts.js) which is directly attached to home.ejs.

In home_posts.js, `e.preventDefault();` is being called whenever the submit button is clicked. This will prevent the flow from going to the action URL defined in the create post form. It means no route will be triggered, hence controller won't be called via the router.

2. after preventing the default behaviour of submitting a form, the form data will be collected using ajax, which processes it and sends it to the controller action. That is, ajax will explicitly call the controller on its own.
3. ajax uses `serialize()` function to parse the form data into JSON format before sending it to the controller.
4. once the data is fetched and parsed by ajax, control is passed to the `posts_controller.js` via the route `'/posts/create'` which is specified using ajax. This is passed as an XML HTTP request to the controller.

5. Once the control comes to the `posts_controller.js`, it creates an entry in the database for the post data, and checks if the incoming request is of type `ajax (.xhr)`, then returns a status code 200 in case of success along with the post data (whose entry is created at the backend).
6. After the success message and data are returned by the `posts_controller.js`, the control goes back to the 'success' function of the AJAX request in `home_posts.js`.

This function creates a post in the DOM which is prepended into the ul for posts in the DOM itself. Basically, DOM updation is being made by indirectly calling the router, hence updation is happening without page reloading.

7. If the `posts_controller.js` return an error then the control goes back to the 'error' function of the AJAX request in `home_posts.js` which does the error handling.

This is how AJAX is used when only some web page parts need to be changed instead of fetching the entire page again by reloading.