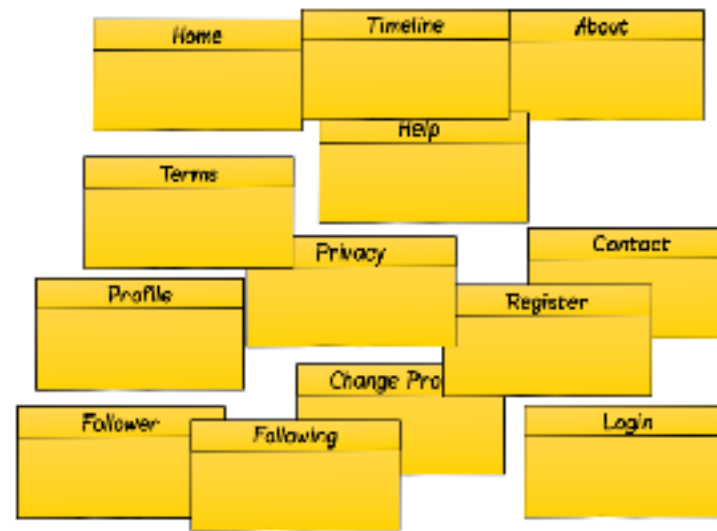


# CLIENT SIDE ROUTING

*get your kicks...*

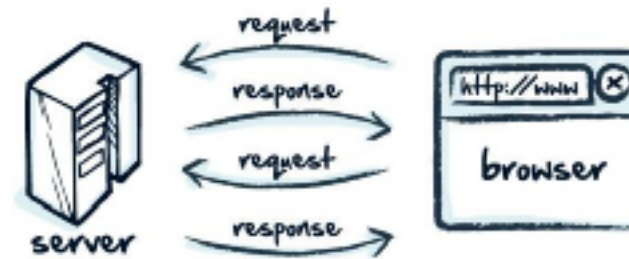


# **REGULAR WEB SITES**

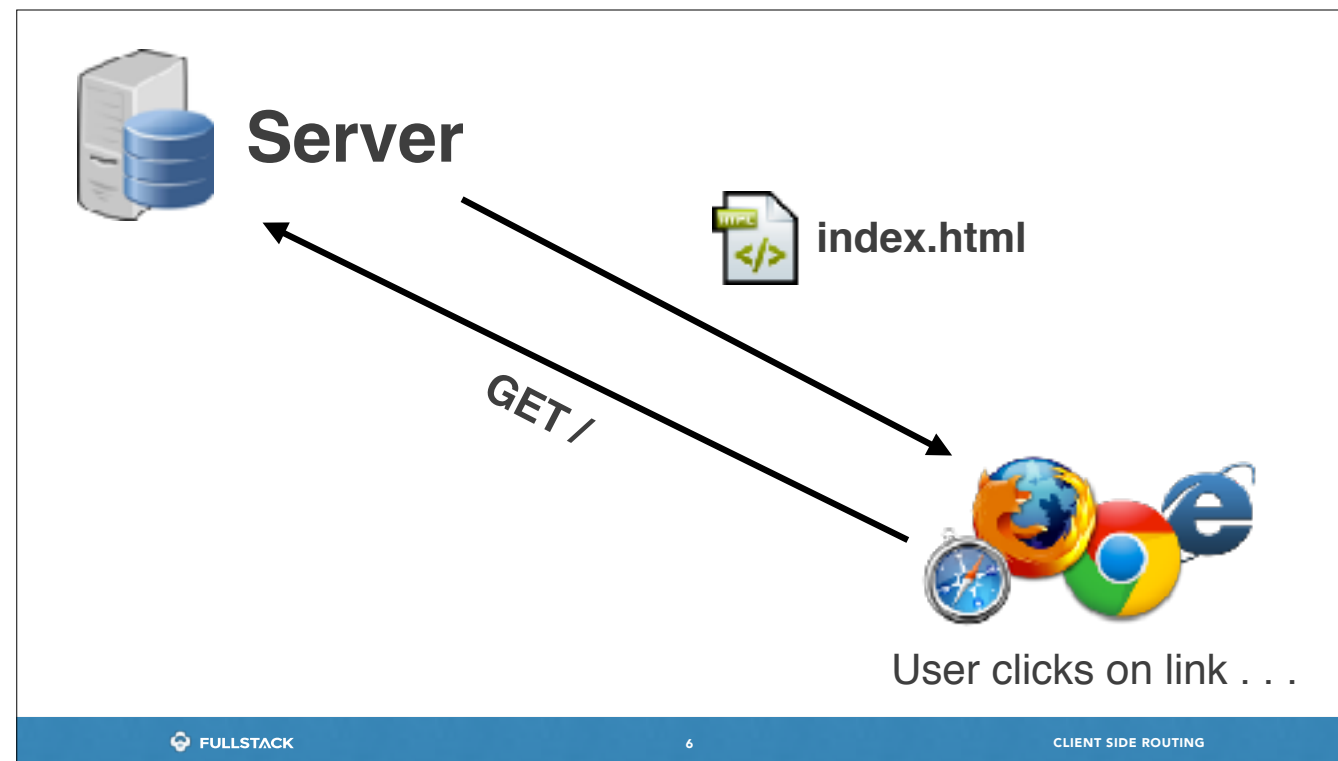
## **(NOT SINGLE PAGE APPLICATIONS)**

# NOT SINGLE PAGE APPLICATION

- Views stored on the server, served up as HTML pages.
- When user goes to a new page, the browser navigates in totality, navigating, refreshing and retrieving a brand new HTML.
- Each page, since it is a new page, retrieves stylesheets, script files, etc.



# SINGLE PAGE APPLICATION (SPA)



In Single Page applications, the browser requests the initial HTML document. As the user interacts, instead of requesting a complete new page...



... JavaScript takes over. It manipulates the DOM and may fire Ajax requests to get additional data.

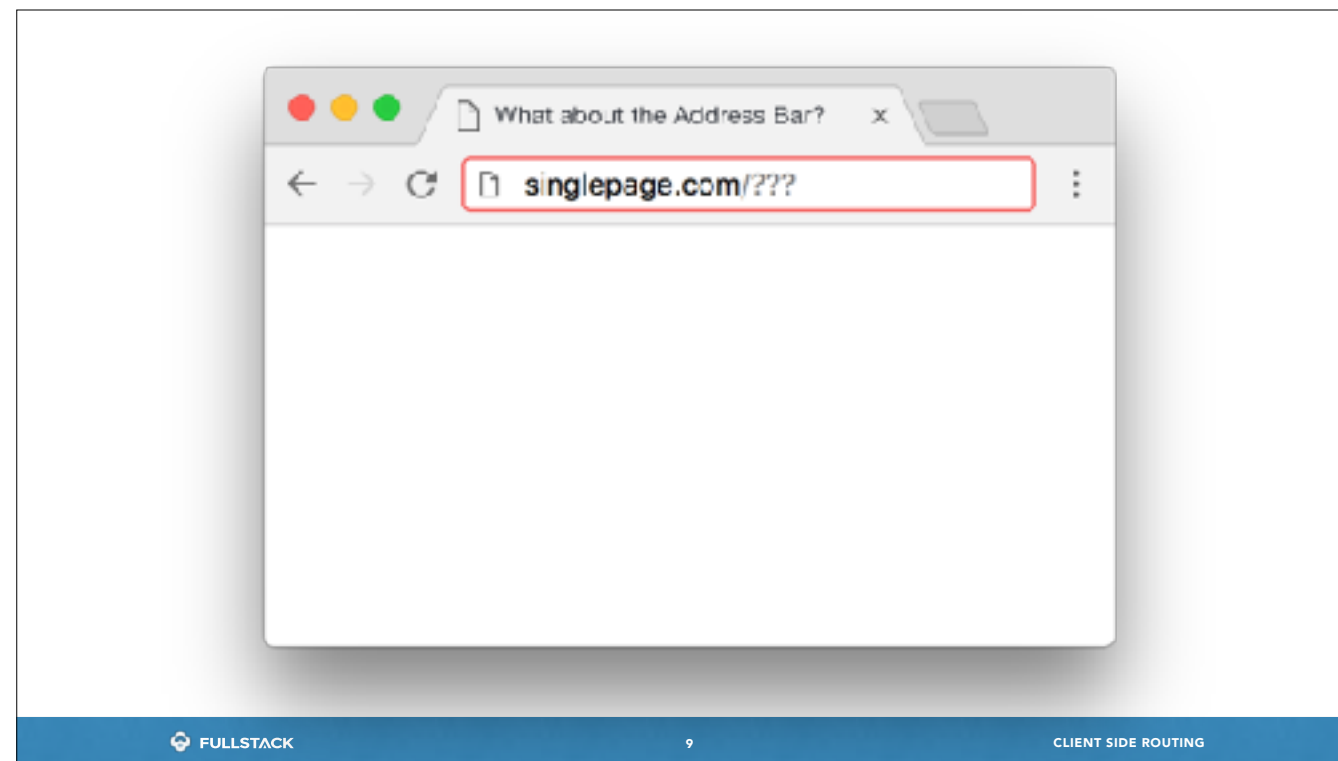
# SINGLE PAGE APPLICATIONS

- On page change, a new page is not loaded. The front-end application replaces elements on existing DOM to update view.
- AJAX plays a big part to fill in data that would normally be served up by the server.

Ok, so everything is perfect, right?

...or is it?





If we're on a "SINGLE PAGE", the address bar will never change.

## IF YOU DO NOT UPDATE THE ADDRESS BAR...

- User will lose the ability to bookmark/share some specific view of the app.
- No proper handling of the browser's "back" button.

## MANIPULATING THE ADDRESS BAR IN JS

- **Browser allows for control of URL, even though pages are not visited.**
  - Abusing “hash” (Document Fragment)
  - Browser History API

Abusing “hash”: Older, still very effective – no changes needed on the server.

History API: Do not produce URL Artifact (“#” – it’s a real URL). Require server to be aware of client-side routing.

# DOCUMENT FRAGMENT IDENTIFIER (#) DEMO

Fragment identifier, AKA the part of the URL that follows the # symbol (including the # symbol).

Demo consists of opening a site such as wikipedia and showing how appending an hash character (#) to the end of the URL identifies (and scrolls) to a portion of the document.

If no element with the corresponding ID is found on the document, nothing happens (and no error is thrown).

.

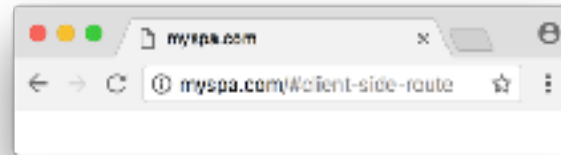
## ABUSING THE FRAGMENT IDENTIFIER

- **The “#” part of a URL:**
  - Optional
  - Ignored by the browser (if no element with such ID exists)
  - Ignored by backend routing
  - Can be manipulated through JavaScript

Because the fragment identifier (the hash) is ignored by the browser, by the express router AND can be manipulated programmatically, it is great to create CLIENT SIDE ROUTES

# LOCATION.HASH

- Gets / Sets the # part of the URL



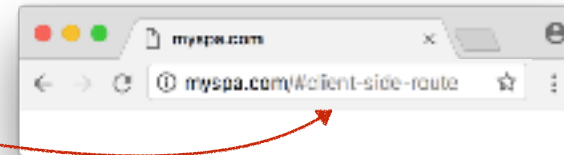
```
const route = location.hash;  
console.log(route); // #client-side-route
```

location.hash can be used as a getter, to retrieve the current fragment identifier.

# LOCATION.HASH

- Gets / Sets the # part of the URL

```
location.hash = "new-route";
```



And as a setter, to change the current fragment identifier.  
Additionally, changes in the hash don't trigger page reload.