# Exercises: Creating a Blog with HTML5, JavaScript and Kinvey

Problems for exercises and homework for the [“Software Technologies” course @ SoftUni](https://softuni.bg/courses/software-technologies).

If you follow the steps of this exercise, correctly, you will implement a simple blog SPA (Single-page Application) which can be of great use to you in the future.

We have implemented all the controllers and views we need for our application. The moment to connect everything has finally come. Everything will be connected in the core of the application – **app.js.**

Before starting to finalize the application, we must write one simple line.

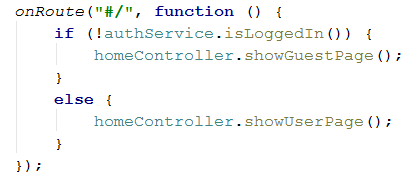


This line initializes all event services that the framework needs to load before it can be fully used. And now we can start.

## Fixing the routing

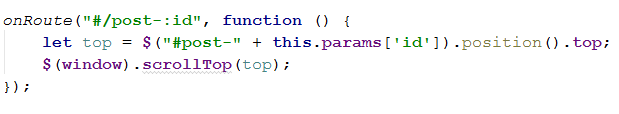
Following the comments given in the app.js, we should implement the routing. We already have **onRoute** functions written. The onRoute function is provided by the framework. It specifies a handler function for when a specified route is accessed by the user.

Let’s start with the home route. We must check whether a client has logged in or not, to know which of the pages to render.



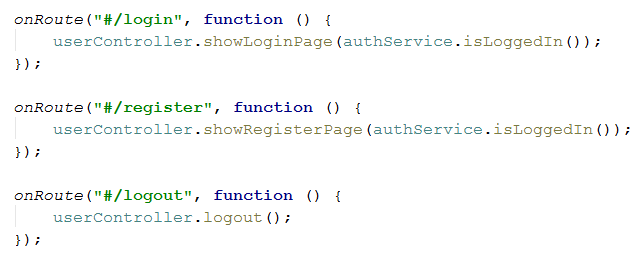
We use the authorization service to check if there is a logged in user and if so, we render the suitable page, through the home controller. With this we have implemented the home routing. But we will check it later

Now we have a bit more interesting routing here. This is the recent post routing. When you click on a recent post on the sidebar, it should focus your page view on the post that you’ve currently clicked. This is how it’s done



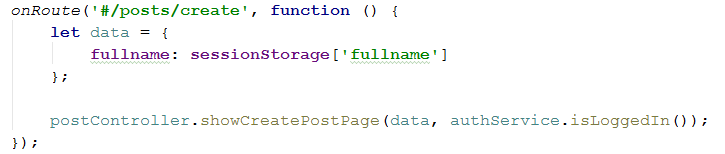
That “**:id**” in the route is a variable in the route, it can be extracted by using “**this.params”.** We extract the post, which is with that id, it’s position, and we scroll the window to that position. The id’s were set before, when were rendering the posts in the Home Controller.

Let’s fix the routing for the user logic



As you see, routing is done, by calling a specific controller action on a specific route. This is the core of the VC system we created. Those functions, with which we tested before, are now put to actual use.

And lastly, lets fix the route for the create post action.



Before calling the controller action, we form a data object and put the current user’s full name, which we extract from the current user session, stored in the session storage. This is done so that we can have an author for the post.

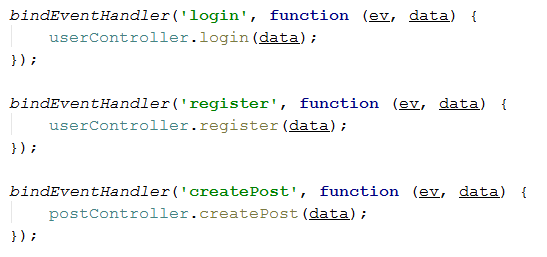
Then we pass it to the corresponding controller action.

With this we should have fixed all the needed logic around the routing for the current blog application.

## Binding event handlers

Now that we have routing, we should bind some event handlers for the events we implemented at the views – **login**, **register**, **createPost.**

This is done by using the **bindEventHandler** function provided to us by the framework. The function accepts an event-name, to which it binds the handler and a function, which will handle the event, and any data passed with it.



On every event we call the corresponding function from the suitable controllers. With this we have completed the whole logic of our application. If we have followed all steps, and implemented everything correctly, we should be able to see a fully functional SPA blog.