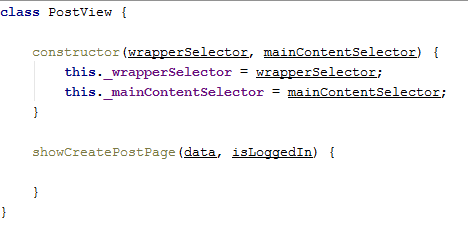
# Exercises: Creating a Blog with HTML5, JavaScript and Kinvey

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.

Most of the controllers and views are ready and fully functional. Now let’s implement a view and a controller for creating posts.

## Creating the Post View class



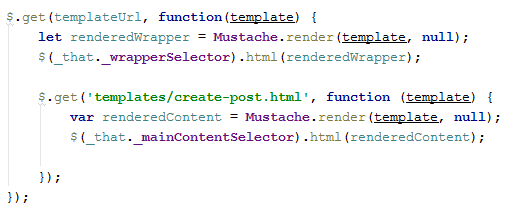
The posts also have a view and a controller. Let’s implement the view. The things look almost the same, now that we have created the user and home views and controllers.

## Implementing the Show Create Post Page logic



Same as the showLoginPage and showRegisterPage, we check if there is a logged in person, and we render the suitable wrapper with its sidebar.

## Getting the Create Post template



Same as the login forms, the create-post form depends on the wrapper, so we nest it inside the other get request. Let’s set the Author of the post defaultly to the currently logged in user. As we see the **showCreatePostPage** function accepts one more element besides the isLoggedIn Boolean. The data element is data about the currently logged in user, if there is such.

We set the author element to the current user’s full name.



Now that we’ve loaded the form, we must apply an event handler to its submission button.



The event handler again works on click for the submission button. We first extract the title, author and content of the current post. Then we use **Moment JS**, one of the external libraries we have to extract the current date, the moment at which the post was created.

Then we create a data object to hold the necessary things for our post, and we trigger an event “**createPost**” and pass the object with it.

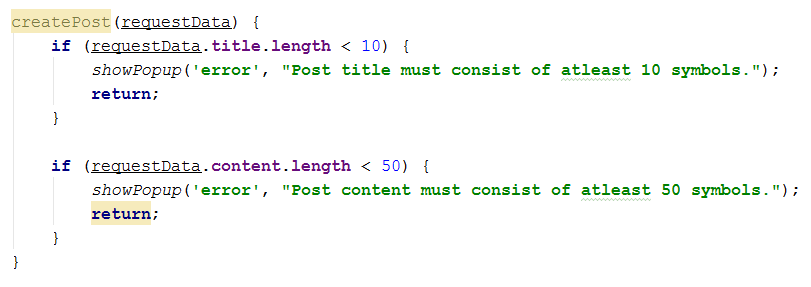
With this our Post View is ready.

## Creating the Post Controller class

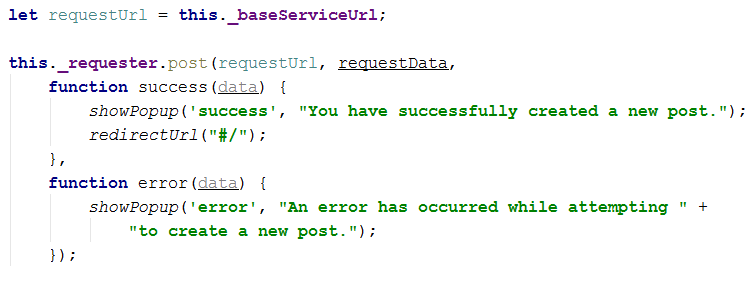


Same pattern for this controller too. The **showCreatePostPage** function accepts data and isLoggedIn. The data parameter is the current user, we explained how it works in the Post View.

## Implementing the create post logic

First we have some simple validations, same as the ones we did with the register form.

Then after everything is validated, we form the request URL and we initiate the request



And with this our Post Controller is finished.

## Revision

Instantiate the postVIew and postController in the app.js, and test the **showCreatePostPage** function. If you’ve done everything correctly, you should see the following

