# Exercises: Single Page Apps with AJAX, REST and Kinvey

Problems for exercise at the [“JavaScript Applications” course @ SoftUni](https://softuni.bg/courses/javascript-applications). Submit your solution in the SoftUni [instance](https://softuni.bg/trainings/1505/js-applications-november-2016) as homework. You can use this [presentation](https://softuni.bg/downloads/svn/js-core/Sept-2016/JS-Apps-Nov-2016/7.%20JS-Apps-SPA-with-AJAX-REST-Kinvey/7.%20JS-Apps-SPA-with-AJAX-REST-Kinvey.pptx) to help you throughout this exercise.

## Prodavachnik

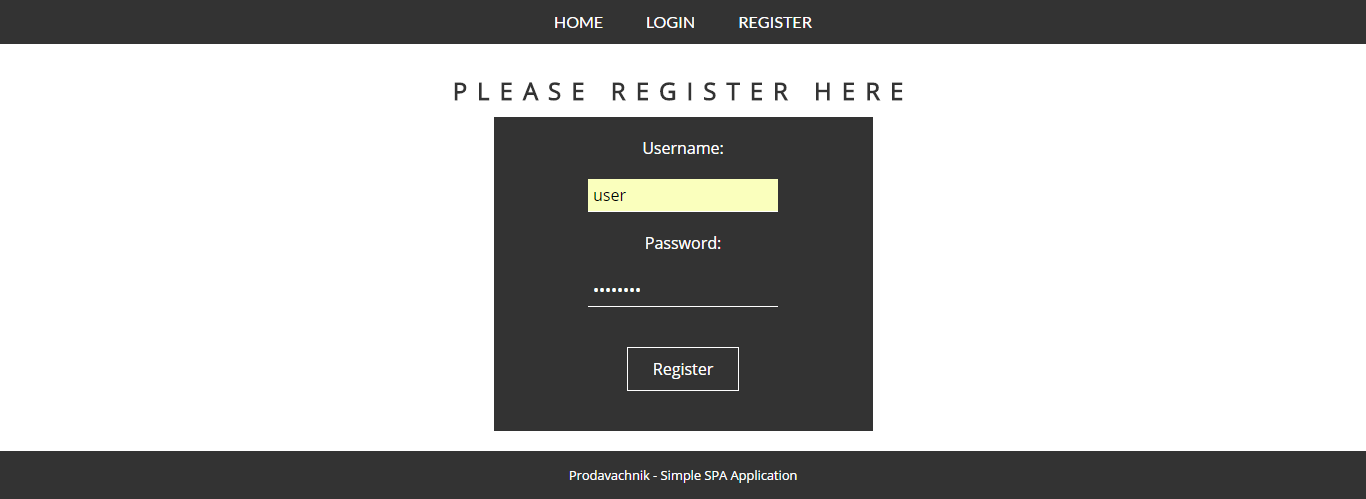
Your task is to implement a Web page that serves the functionally of an **advertisement** **site**, which specializes in **selling** **products**/**services**. The API will consist of creating, listing, editing and deleting of **advertisements**(**CRUD**). Every **advert** should contain the following properties:

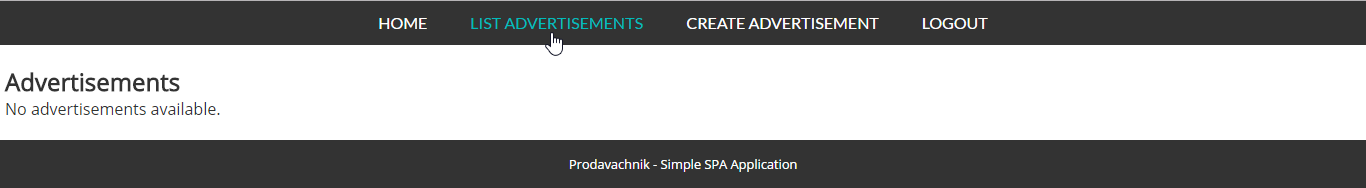
* Title - String
* Description - String
* Publisher - String
* Date of Publishing – Date (format “**mm/dd/yyyy**”)
* Price – **floating** point number, to the **second** **digit** after decimal separator.

You can download the skeleton (HTML and CSS) [here](https://softuni.bg/downloads/svn/js-core/May-2017/JS-Apps-Jun-2017/8.%20JS-Apps-Single-Page-Applications-Exercises/Prodavachnik%20-%20Skeleton.zip).

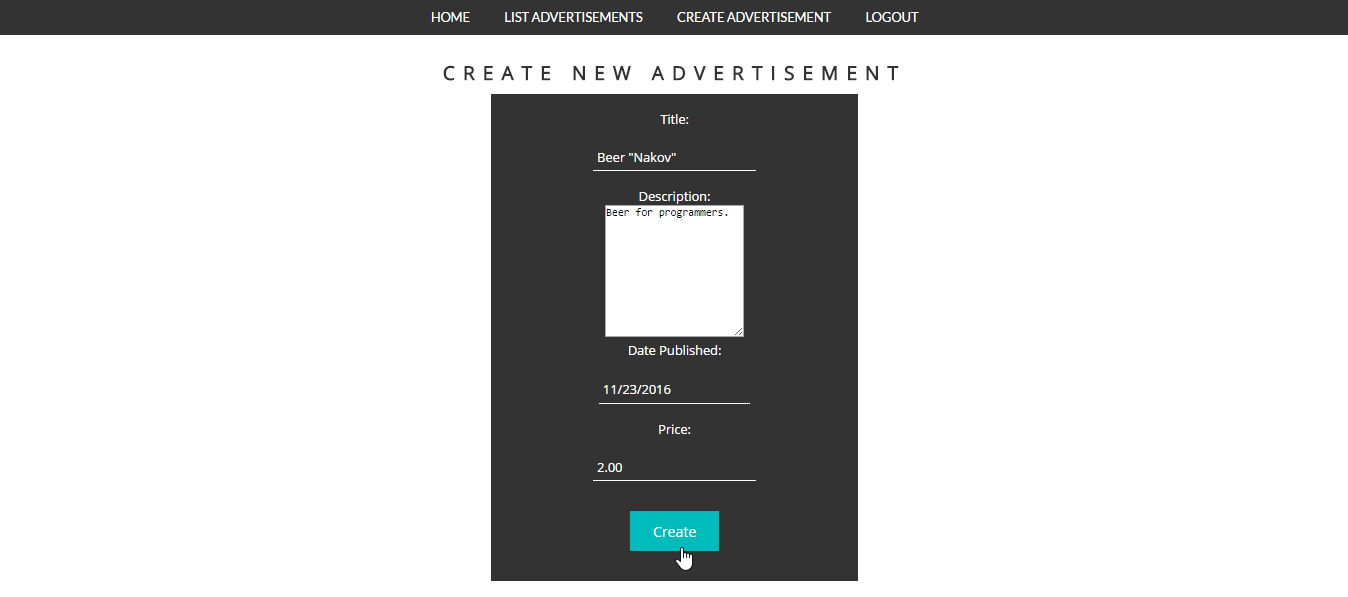
### Screenshots

Home page:

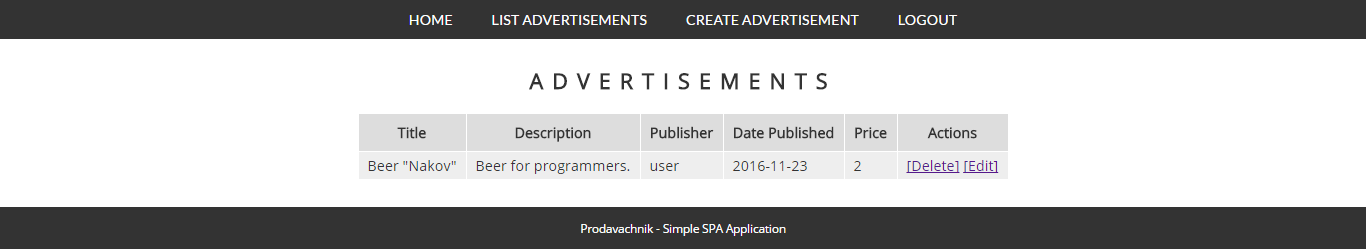
Registration page (Login page is identical):

This is how the website should look like when there are no adverts:

Let’s create one (the view for editing an advert is identical):



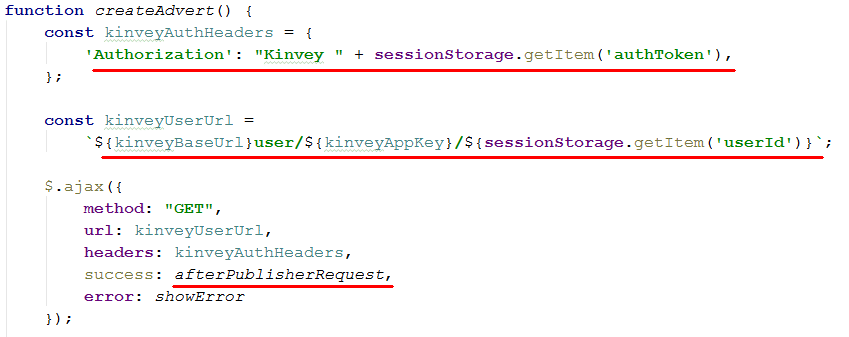
And this is how it should be listed:



The advertisement site should have the following functionality:

* Perform any **CRUD** operations over the advertisement entity.
  + Note that the **publisher** of the advertisement is the **currently** logged **user**.
  + Publisher can edit/delete his own adverts only.
* Have **Register**/**Login**/**Logout**.

### Hints

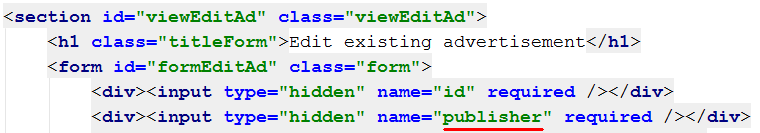
* When **create** an **advertisement** the current logged user should be it’s **publisher**. We can do it by making one request to the database (by user’s **“\_id”** property) and retrieve the whole **user** information. It may look something like that:

Once we got the user information we can use it in our “afterPublisherRequest” method:

And the rest is **parsing** the **information** from the form and sending **another** **request** – this time to **save** our **advert** in database.

**Note** that the above method should be called when the form for **advert** **creation** is **submitted**.

* When **editing** an advert in which we don’t want to change the **publisher** you can include it in the html, but mark it as “hidden”, here is how the html might look:



What is left is to **initially** **set** the **value** when printing the edit form **and** then **parsing** the whole **data** when the form is **submitted**.