

# Ads – SPA with AngularJS Lab

The **goal of this lab** is to continue the development of the Online Ads AngularJS application: add **filtering by category and town** when displaying the public ads.

## Problem 4. Filtering Ads by Category and Town

**Step 1.** Create a service **categoriesService** to retrieve the list of categories from the back-end REST service:

adsServices.js
<pre>... app.factory('categoriesService',   function (\$resource, baseUrl) {     var categoriesResource = \$resource(       baseUrl + '/api/categories'     );      return {       getCategories: function(success, error) {         return categoriesResource.query(success, error);       }     }   } ); ...</pre>

5 score

**Step 2.** Create a service **townsService** to retrieve the list of towns from the back-end.

5 score

**Step 3.** Load the categories and towns in the **RightSidebarController**:

RightSidebarController.js
<pre>app.controller('RightSidebarController',   function (\$scope, \$rootScope, categoriesService, townsService) {     \$scope.categories = categoriesService.getCategories();     \$scope.towns = townsService.getTowns();   } );</pre>

5 score

**Step 4.** Write a HTML template (view) to **display the categories and towns**:

right-sidebar.html
<pre>&lt;div class="box"&gt;   &lt;h2&gt;Categories&lt;/h2&gt;   &lt;ul class="sidebar-menu"&gt;     &lt;li class="special"&gt;All&lt;/li&gt;     &lt;li ng-repeat="category in categories"&gt;       &lt;!-- TODO: display category name here... --&gt;</pre>

```

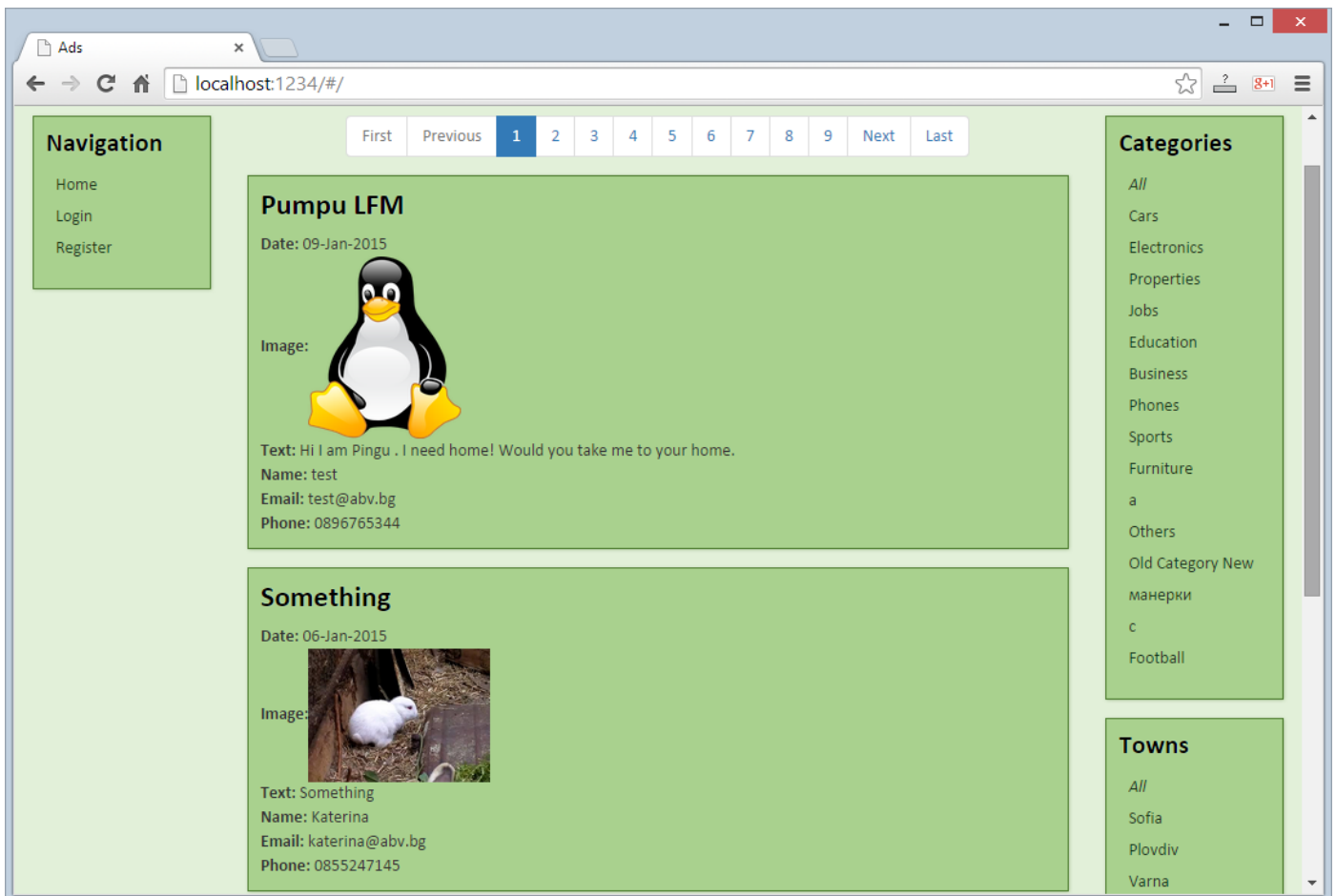
        </li>
    </ul>
</div>

<!--
    TODO: implement the Towns filter box here (just like the categories box above)
-->

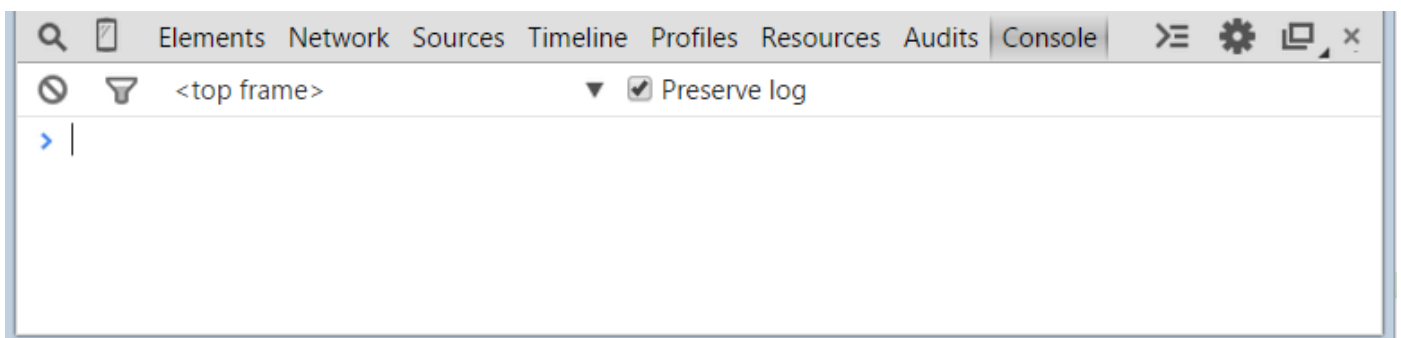
```

5 score

**Step 5.** Test your application to ensure the categories and towns are loading correctly:



- Check for errors at the Web browser's development console:



10 score

**Step 6.** Make the categories and towns **selectable**. This is achieved in two steps:

- Handle the "click" event to store the ID of the selected item in the **\$scope** (or **null** if the filter is clear).
- Assign a CSS class "**selected**" to the currently selected item in the view.

First, modify the **right-sidebar.html** view template:

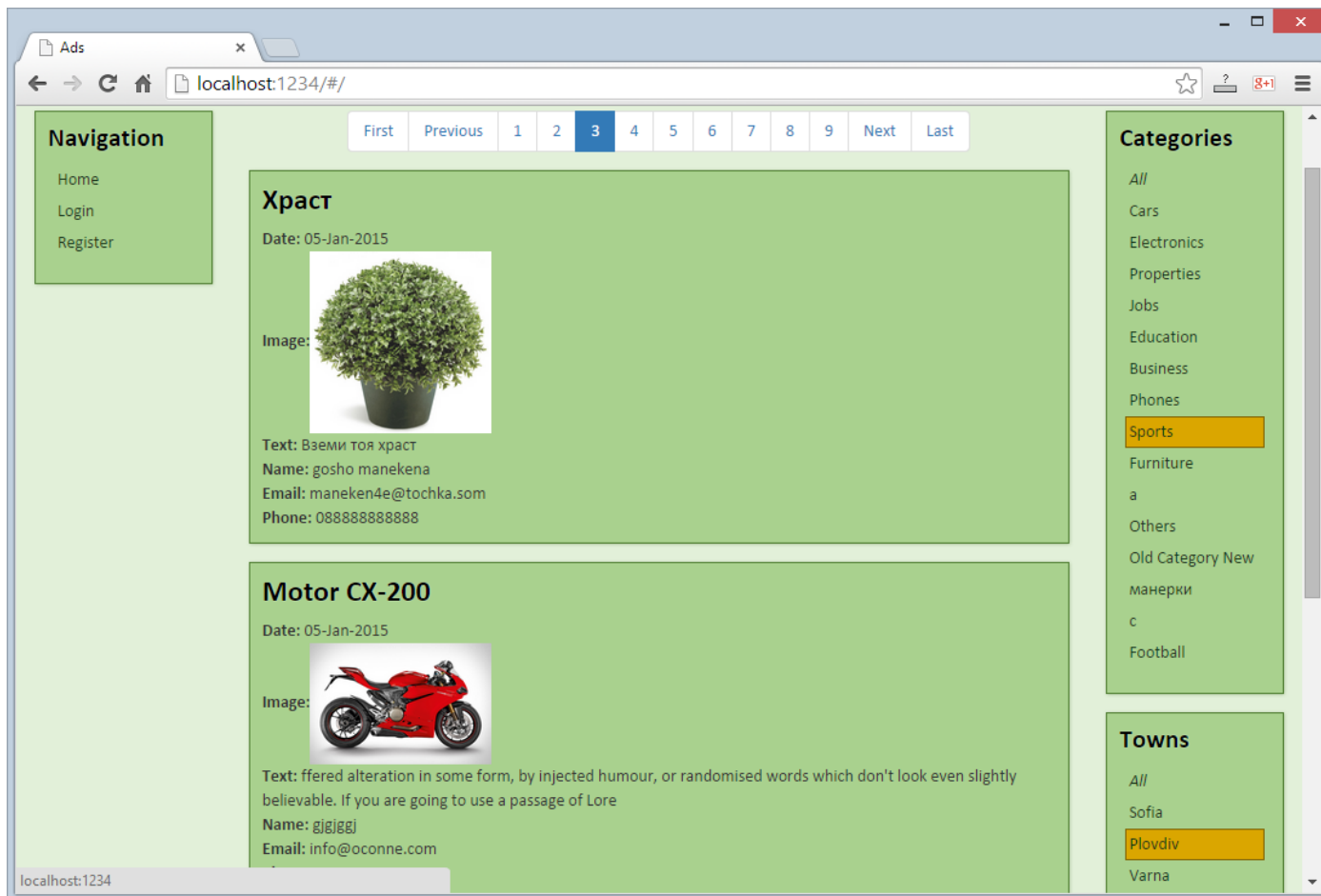
right-sidebar.html
<pre> &lt;div class="box"&gt;   &lt;h2&gt;Categories&lt;/h2&gt;   &lt;ul class="sidebar-menu"&gt;     &lt;li class="special"       ng-class="{ 'selected' : selectedCategoryId == null}"       ng-click="categoryClicked(null)"&gt;       All     &lt;/li&gt;     &lt;li ng-repeat="category in categories"       ng-click="categoryClicked(category.id)"       ng-class="{ 'selected' : selectedCategoryId == category.id}"       {{category.name}}     &lt;/li&gt;   &lt;/ul&gt; &lt;/div&gt;  &lt;!--   TODO: implement the Towns filter box here (like the Categories filter) ... --&gt; </pre>

Second, add click handlers in the **RightSidebarController**:

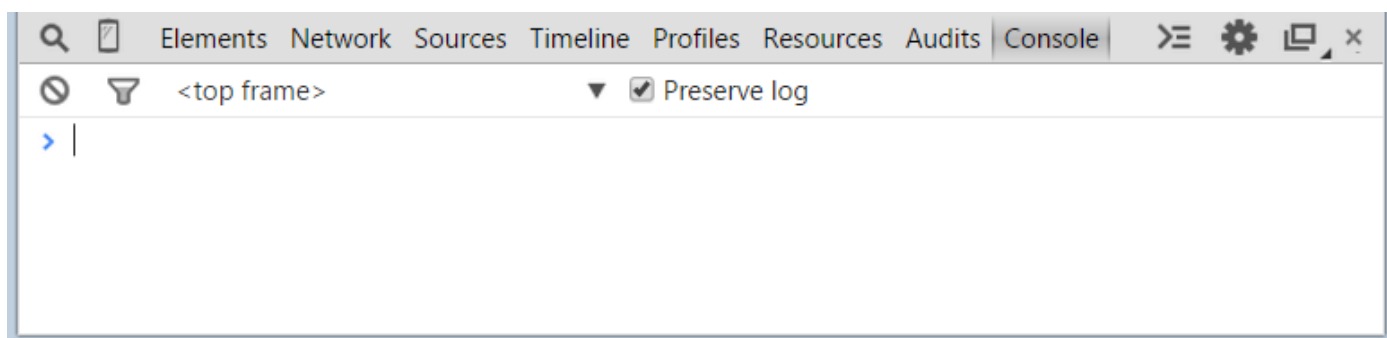
RightSidebarController.js
<pre> ... \$scope.categoryClicked = function(clickedCategoryId) {   \$scope.selectedCategoryId = clickedCategoryId; };  // TODO: implement the same logic for the "townClicked" event handler </pre>

10 score

**Step 7.** Test your application to ensure **the categories and towns are selectable**:



- Check for errors at the Web browser's development console:



5 score

**Step 8.** Filter ads by the selected category and town. This is achieved in two steps:

- **Broadcast an event "categorySelectionChanged" / "townSelectionChanged"** through the **\$rootScope** from the **RightSidebarController** when the category / town selection is changed.
- **Handle the "categorySelectionChanged" / "townSelectionChanged" events** in the **HomeController** and filter the ads by invoking the REST service at the back-end with **categoryId** and **townId** parameters.

First, modify the **RightSidebarController** to broadcast the category and town selection change events:



```

$scope.categoryClicked = function(clickedCategoryId) {
    $scope.selectedCategoryId = clickedCategoryId;
    $rootScope.$broadcast("categorySelectionChanged", clickedCategoryId);
};

$scope.townClicked = function(clickedTownId) {
    $scope.selectedTownId = clickedTownId;
    $rootScope.$broadcast("townSelectionChanged", clickedTownId);
};

```

Second, modify the **HomeController** to implement ads filtering by calling the REST service:

### HomeController.js

```

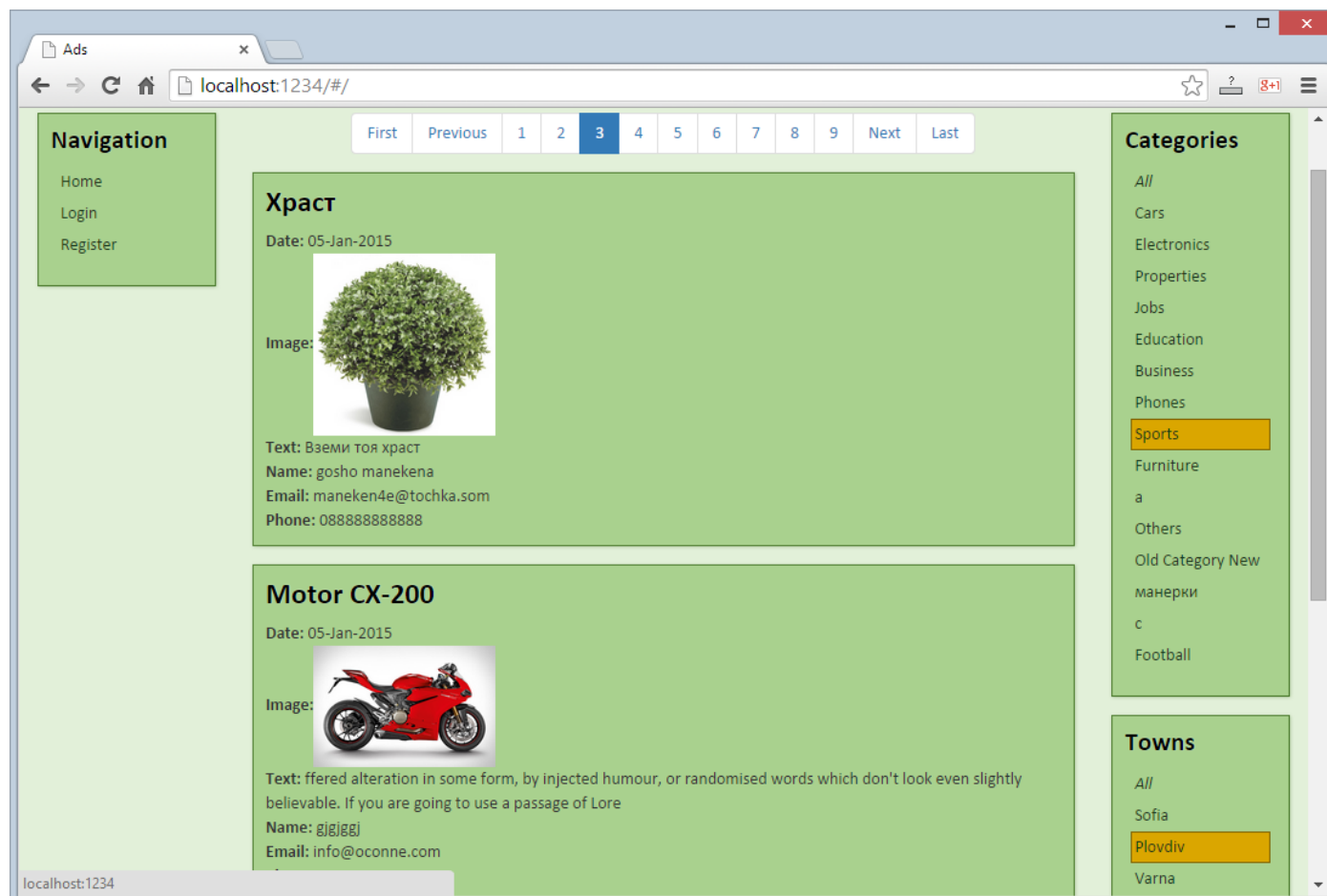
...
// This event is sent by RightSideBarController when the current category is changed
$scope.$on("categorySelectionChanged", function(event, selectedCategoryId) {
    $scope.adsParams.categoryId = selectedCategoryId;
    $scope.adsParams.startPage = 1;
    $scope.reloadAds();
});

// TODO: implement similar logic for filtering by category

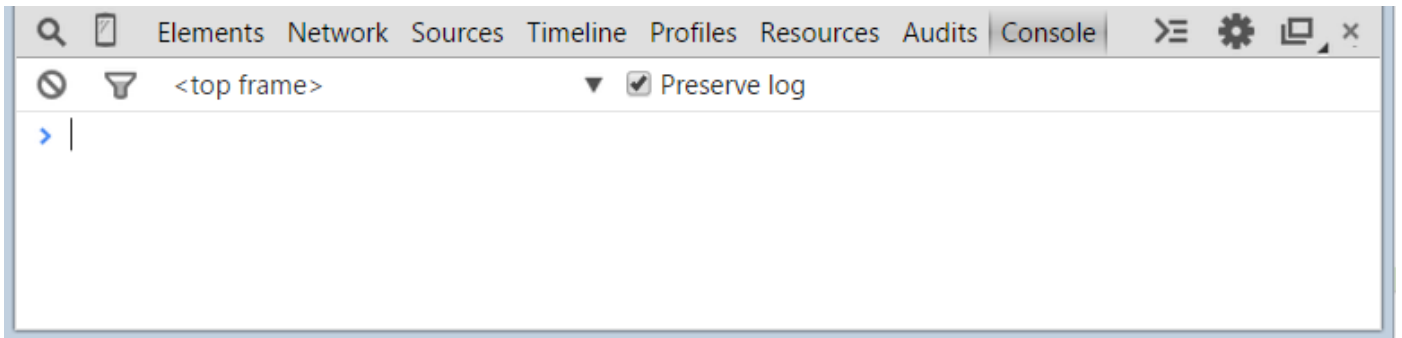
```

15 score

**Step 9.** Test your application to ensure **the filtering by category and town works correctly**:



- Check for errors at the Web browser's development console:



5 score