

Spring MVC Tutorial – Contact List Application

Step 13: Client Side View Functionality

Copyright © 2016 The Learning House, Inc.

All rights reserved. No part of these materials may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of The Learning House. For permission requests, write to The Learning House, addressed "Attention: Permissions Coordinator," at the address below.

The Learning House

427 S 4th Street #300

Louisville KY 40202

Step 13: Client Side View Functionality

Overview

The Stats screen of our application has been included mainly as a demonstration of how Google Charts can be used to visualize data. Our screen will consist of one bar chart that displays the number of contacts per company. In this step, we will set up everything we need to use Google Charts and write code to display the bar charts using hard-coded data. In later steps, we'll swap out the canned data for live data from the server.

Main Layout

The main layout of the Stats screen consists of the navigation bar and our one chart. Once again, we'll utilize the Bootstrap grid layout system for our screen.

Modify **stats.jsp** so that it has the following content (there's not much new here... new code is highlighted in **bold red**):

Notes:

1. We're placing the chart in a new Bootstrap grid system row, which has only one column that spans all 12 columns.
2. We define an empty div, and we will write JavaScript code to render the chart dynamically into this div.

```

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="s" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Company Contacts</title>
        <!-- Bootstrap core CSS -->
        <link href="{pageContext.request.contextPath}/css/bootstrap.min.css" rel="stylesheet">

        <!-- SWC Icon -->
        <link rel="shortcut icon" href="{pageContext.request.contextPath}/img/icon.png">

    </head>
    <body>
        <div class="container">
            <h1>Company Contacts</h1>
            <hr/>
            <div class="navbar">
                <ul class="nav nav-tabs">
                    <li role="presentation">
                        <a href="{pageContext.request.contextPath}/home">Home</a>
                    </li>
                    <li role="presentation">
                        <a href="{pageContext.request.contextPath}/search">Search</a>
                    </li>
                    <li role="presentation" class="active">
                        <a href="{pageContext.request.contextPath}/stats">Stats</a>
                    </li>
                </ul>
            </div>
            <div class="row">
                <!-- #1: Contacts Per Company Chart -->
                <div class="col-md-12">
                    <h2>Statistics</h2>
                    <!-- #2: this is where we'll render the bar chart -->
                    <div id="chart_div"></div>
                </div>
            </div>
        </div>

        <!-- Placed at the end of the document so the pages load faster -->
        <script src="{pageContext.request.contextPath}/js/jquery-1.11.1.min.js"></script>
        <script src="{pageContext.request.contextPath}/js/bootstrap.min.js"></script>

    </body>
</html>

```

This new HTML displays a heading for our chart and creates a <div> into which the chart will be rendered.

JavaScript

Google Charts are driven by JSON data and rendered via JavaScript, which means that we need to write some code. We'll put all of the Stats screen-related JavaScript into a file called **stats.js** in the **js** folder. Create that file now and enter the following code:

Notes:

1. This is a standard jQuery document-ready function that just calls the drawChart function.
2. This function draws the Google bar chart for us.
3. Use Google util function to create a hard-coded DataTable object. Eventually, we'll create this object on the server and return it via an Ajax call. Generally, Google Chart widgets consume DataTable objects.
4. Set options for the chart. We set size, main title, and axis titles here. There are many more options; see the Google Charts API documentation for details.
5. Create a new BarChart object and hand it a reference to the div where we want it to render. We created a div with an ID of "chart_div" in **stats.jsp** in the previous step.
6. Tell the bar chart to render itself.

```

// #1: document ready
$(document).ready(function () {
    drawChart();
});

// #2: draw the bar chart
function drawChart() {
    // #3: dummy data - will be replaced with Ajax call
    var data = google.visualization.arrayToDataTable([
        ['Company', '# Contacts'],
        ['Oracle', 10],
        ['Apple', 11],
        ['Microsoft', 6],
        ['SWC Guild', 18]
    ]);

    // #4: this sets up the size of the chart, the main title, and the axis titles
    var options = {
        title: 'Contacts By Company',
        vAxis: {title: 'Company', titleTextStyle: {color: 'red'}},
        hAxis: {title: 'Num Contacts', titleTextStyle: {color: 'red'}},
        'width': 500,
        'height': 400
    };

    // #5: create a new BarChart object, handing it the div into which we want it to render
    var chart = new google.visualization.BarChart(document.getElementById('chart_div'));

    // #6: tell the chart to draw itself
    chart.draw(data, options);
};

```

Including and Loading Google Charts

Now that we have our JavaScript code and HTML in place, we must reference and load the Google Charts API in **stats.jsp**. Modify **stats.jsp** so that it looks like this (changes are marked in **bold red**):

Notes:

1. Like any other JavaScript library, we must include a reference to the Google Charts API in order to use it. The Google Charts API requires the additional step of **loading** the library.
2. Include a reference to the JavaScript code we just created. Because our code relies on jQuery and Google Charts, we must put the reference to our JavaScript file **after** the references to the other libraries.

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ taglib prefix="s" uri="http://www.springframework.org/tags"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Company Contacts</title>
        <!-- Bootstrap core CSS -->
        <link href="{pageContext.request.contextPath}/css/bootstrap.min.css" rel="stylesheet">

        <!-- SWC Icon -->
        <link rel="shortcut icon" href="{pageContext.request.contextPath}/img/icon.png">

    </head>
    <body>
        <div class="container">
            <h1>Company Contacts</h1>
            <hr/>
            <div class="navbar">
                <ul class="nav nav-tabs">
                    <li role="presentation">
                        <a href="{pageContext.request.contextPath}/home">Home</a>
                    </li>
                    <li role="presentation">
                        <a href="{pageContext.request.contextPath}/search">Search</a>
                    </li>
                    <li role="presentation" class="active">
                        <a href="{pageContext.request.contextPath}/stats">Stats</a>
                    </li>
                </ul>
            </div>
        </div>
```

```

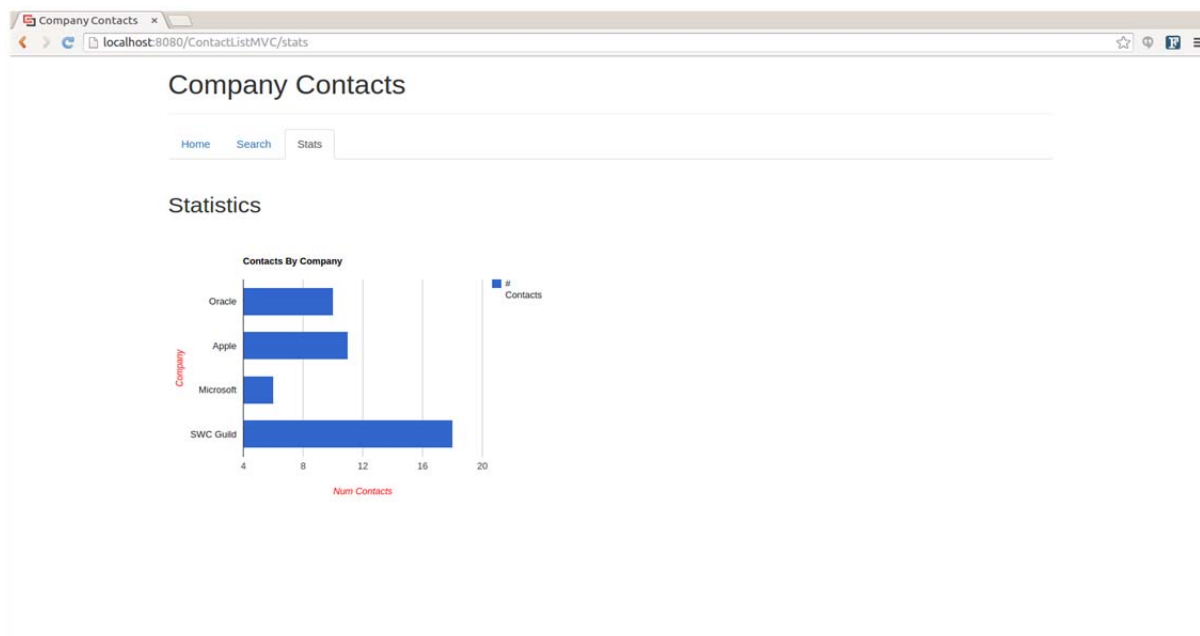
<div class="row">
  <!-- Contacts Per Company Chart -->
  <div class="col-md-12">
    <h2>Statistics</h2>
    <!-- this is where we'll render the bar chart -->
    <div id="chart_div"></div>
  </div>
</div>

<!-- Placed at the end of the document so the pages load faster -->
<!-- #1: Include and load Google Charts - both steps are required -->
<script type="text/javascript" src="http://www.google.com/jsapi"></script>
<script>
  google.load('visualization', '1.0', {'packages':['corechart']});
</script>
<script src="{pageContext.request.contextPath}/js/jquery-1.11.1.min.js"></script>
<script src="{pageContext.request.contextPath}/js/bootstrap.min.js"></script>
<!-- #2: Include and load Google Charts - both steps are required -->
<script src="{pageContext.request.contextPath}/js/stats.js"></script>

</body>
</html>

```

If everything went well, we should be done. Your Stats screen should look like this:



Wrap-up

That does it for this step. In this portion of the tutorial, we did the following:

1. Modified the **stats.jsp** layout to include our bar chart
2. Created JavaScript code to render our bar chart
3. Modified **stats.jsp** to reference and load the Google Charts API

In the next step, we will move to the server and begin the process of creating our model and persistence layer.