# Exercises: Spring Cookbook

Problems for exercises and homework for the [“Java MVC Frameworks - Spring” course @ SoftUni](https://softuni.bg/trainings/1538/java-mvc-frameworks-spring-march-2017). You are given a skeleton for the exercise.

## Get Familiar with the Project Skeleton

Take a look at the repositories, services, controllers and views that you're given. The most of the functionality is implemented for you.

## Configure Web Socket

Create configuration class that will enable the web socket message broker. Your class needs to extend the AbstractWebSocketMessageBrokerConfigurer. Override the registerStompEndpoints() and configureMessageBroker(). Configure 1 **end point** and 2 **simple message brokers**.

## Enable Schedules

Enable scheduling in the one of the application **configuration** classes.

## Implement Socket Service

Create implementation of the SocketService interface. You can send messages using SimpMessagingTemplate class.

## Task Completion Checking

Create a new class that will execute the tasks you are going to schedule. The first task should be executed on **every second**. It should get all tasks that have the status Status.IN\_PROGRESS and check if the **end** **date** of the task is before the current date. After a **task is completed**, you should **send an update** to the frond-end.

Methods that will help you:

* TaskService.getActive()
* TaskService.generateTaskOutcome()
* SocketService.sendCharUpdate()
* CharacterService.characterAjax()

## Paycheck

Create another scheduled task. It should be completed **every day** at **12:00 o'clock**. When the task is executed it should give each character the paycheck he earned. The formula that you need to use is:

(completedTasksCount \* 20L) - (failedTasksCount \* 10L);

After you give the character his money, send an update to the frond-end, using the SocketService.characterMoney().

## Visualize the Changes

Open the characters.js file. Find the connect() function and implement it, based on the TODO.