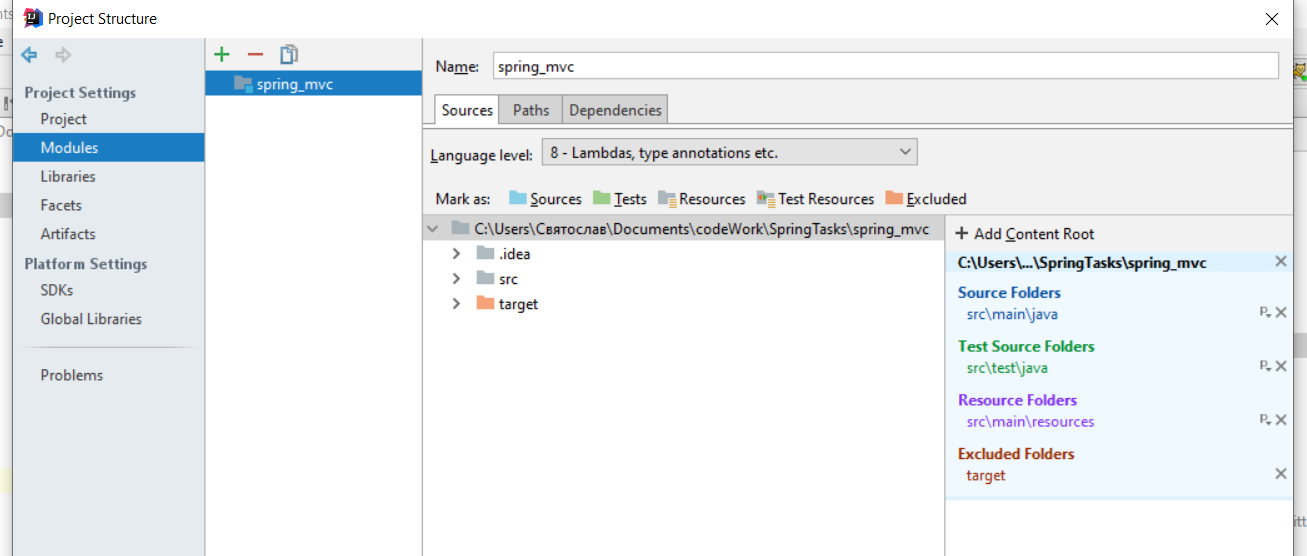
На данном этапе начинается самое интересное: ваше приложение начнет получать запросы и возвращать ответы. Плюс к этому, мы создадим приложение, которое будет запускаться и управляться с помощью сервера Tomcat. IDE поможет работать с ним без сложных консольных команд, что сильно облегчит разработку.  
В этом приложении мы используем только зависимость spring-webmvc. Это не значит, что нам больше не нужны бины или контекст, в данном случае webmvc-фреймворк будет работать самостоятельно, так как любой фреймворк Спринга содержит в себе spring-core.  
В приложении появился новый пакет controller, в котором содержатся классы, держащие группы сервлетов - контроллеры. Данная технология позволит нам избежать создания большого количества сервлетов. Контроллеры содержат методы, которые маппятся на разные url и http-методы, заменяя собой класс сервлета с методами doGet, doPost и т.д.

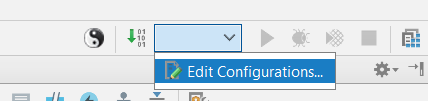


Для запуска приложения перейдите в Project Structure во вкладку Modules.  
Выберите модуль spring\_mvc и нажмите “+”. Далее следует выбрать модуль Spring и IDEA сама подтянет класс-иницилайзер WebConfig, если это не произошло автоматически, то пропишите абсолютный путь на этот файл.  
Аналогично нужно создать модуль Web, но в этом случае путь должен ссылаться на папку webapp.

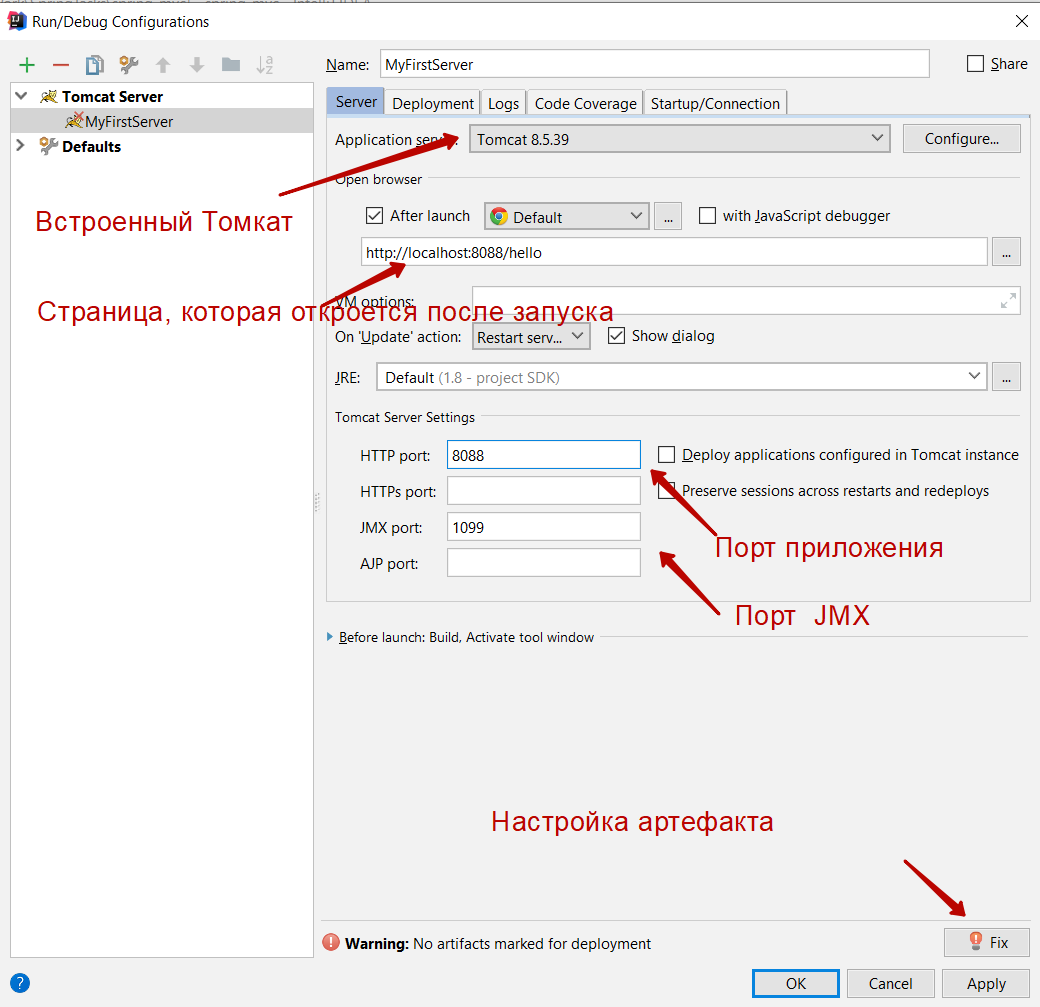
Теперь обратимся к сборке проекта, в файл pom.xml:



Для правильной интерпретации структуры проекта требуется указать следующие параметры:  
1) failOnMissingWebXml позволит нам запускать проект с помощью маппинга контроллерами, без использования конфигурации в web.xml.  
2) warSourceDirectory должен ссылаться на папку, содержащую WEB-INF/  
3) В конфигурации maven-compiler-plugin явно нужно прописать версию Java.  
Для запуска перейдите в конфигурацию сервера:



Нажмите edit, затем плюс в левом верхнем углу и выберите Tomcat Server -> Local Server.



Настройте запуск приложения на свободный порт, выберите дефолтный Tomcat, измените адрес, который будет вызван после запуска, на /, так как в нашем приложении для этой страницы есть контроллер.  
Далее следует настроить артефакт. Нажмите кнопку Fix, далее «+», выберите артефакт с именем “название проекта”-war exploded.  
После этого подтвердите изменения и запускайте приложение.  
Ваш браузер откроется автоматически и вы увидите страницу приветствия.

Отдельное внимание стоит уделить тому, что в приложении нет метода main, его запуск происходит из-под Tomcat и для этого требуется отдельный класс AppInit, который ссылается на корневой конфигурационный файл и обозначает, на каком url будет находиться наше приложение.

**Задание:**  
1. Создайте еще один контроллер, замаппленный на /cars.  
2. Создайте модель Car с тремя произвольными полями.  
3. Создайте список из 5 машин.  
4. Создайте сервис с методом, который будет возвращать указанное число машин из созданного списка.  
5. Создайте страницу cars.html. Реализуйте создание таблицы с машинами из сервиса с помощью thymeleaf.  
6. При запросе /cars выводить весь список. При запросе /cars?count=2 должен отобразиться список из 2 машин,  
при /cars?count=3 - из 3, и тд. При count ≥ 5 выводить весь список машин.