

Практическая работа №7. Поддержка архитектуры REST в Spring

Цель работы: Познакомиться с механизмами поддержки архитектуры REST в Spring

Общая постановка задачи: Взять практическое задание №6 (или №5, на усмотрение студента, при работе с защищенным приложением могут возникнуть трудности) и добавить следующий функционал:

1) Преобразовать веб-приложение таким образом, чтобы оно поддерживало архитектуру REST. Должны поддерживаться следующие типы запросов: GET (показ (html) и извлечение (json) всех/одной записей/сущностей), PUT (добавление), DELETE (удаление).

2) Разработать REST-клиент для вашего приложения, который, используя RestTemplate позволяет выполнять базовые операции по извлечению (GET), добавлению (PUT), удалению (DELETE) ресурсов. REST-клиент не обязан иметь пользовательский интерфейс, необходим тестовый пример, который можно запускать из консоли.

3) **Обязательным** условием является сохранение всего предшествующего функционала приложения. Для удовлетворения всем характеристикам REST-архитектуры приложение может быть реорганизовано (убраны GET-запросы с параметрами) или добавлен новый функционал.

4*) PUT и DELETE запросы не обязательно делать через запросы из браузера. Достаточно реализации для клиентов-приложений.

Дополнительные комментарии: Результат работы должен представлять собой отчет составленный по СФУ СТО. В работе должен быть указан ход работы, либо исходный код, с указанием в каком файле что находится.

Работа должна быть защищена преподавателю. Защита состоит в пояснении хода действия для решения поставленной задачи и ответов на теоретические вопросы или вопросы, возникающие у преподавателя в ходе проверки работы.

Контрольные вопросы:

- 1) Поясните что собой представляет REST архитектура?
- 2) Что собой представляет контроллер, противоречащий архитектуре REST?
- 3) Как характеризуются URL, которые соответствуют архитектуре REST?
- 3) Перечислите и опишите основные методы протокола HTTP?
- 4) Для чего предназначен ContentNegotiatingViewResolver?
- 5) Для чего служит аннотация @ResponseBody?
- 6) Для чего служат преобразователи HTTP-сообщений?
- 7) Что собой представляет REST-клиент?
- 8) Для чего предназначен класс RestTemplate?
- 9) Какие методы RestTemplate вы использовали?
- 10) Каким образом (шаги) вы преобразовали приложение к REST архитектуре?