

## Задание 7

### Сервисы SOAP: JAX-WS

#### Темы

1. Web-службы XML
2. SOAP, WSDL, UDDI
3. JAX-WS
4. Implementation: Apache CXF
5. JSON serialization: Jackson

#### Контрольные вопросы

1. Что такое сервис-ориентированная архитектура (SOA)?
2. Из каких логических частей состоит сервис согласно SOA?
3. Какова структура SOAP-сообщения?
4. Объясните значение терминов WSDL и разницу между ними: interface, binding, service, endpoint?
5. Обязательно ли для службы иметь WSDL, чтобы к ней обращаться и почему?
6. Что такое POJO и зачем как они применяются в веб-сервисах jax-ws? Перечислите основные аннотации, необходимые для описания сервиса.
7. Для чего предназначена аннотация @WebServiceRef?
8. Как в веб-сервисах применяется технология JAXB?
9. Что такое прокси-объект (проху)? Каким образом можно создать и использовать такой объект для доступа к сервису из клиентского приложения?
10. Каким образом можно, имея WSDL-документ, сгенерировать endpoint-класс, и наоборот?

#### Индивидуальные задания

1. Для созданного на предыдущих занятиях web-приложения (*по варианту*) реализуйте доступ к бизнес-логике (т.е. основным объектам БД и операциям с ними) в виде web-служб.

Необходимо повторно использовать ту же реализацию бизнес логики, которая ранее была доступна через веб-интерфейс.

2. Протестируйте работу служб с помощью клиента SoapUI.

3. Создайте потребителей этих служб в виде объектов Java Bean и используйте их в JSP/JSTL-страницах своего приложения вместо объектов Java Bean, которые вы создавали ранее для доступа к данным.

## Задание 8

### Сервисы REST: JAX-RS

#### Темы

1. REST
2. JAX-RS
3. Implementation: Apache CXF
4. (\*) Jackson
5. (\*) Documenting REST API

#### Контрольные вопросы

1. Объясните основные отличия RESTful-служб от служб SOAP.
2. Что такое content negotiating и как он реализуется?
3. В чем разница между методами PUT и POST? Какие проблемы могут возникнуть, если поменять местами аннотации @PUT и @POST?
4. Могут ли ресурсы быть вложенными?
5. Можно ли аннотацией @GET пометить не один, а несколько методов ресурса?
6. Как задать конфигурацию CXF для работы rest-сервиса?
7. Что будет если ресурс имеет параметр @QueryParam, а во входящем запросе такой параметр отсутствует?
8. Как настроить автоматический ответ на случай, если в процессе обработки запроса возникнет исключение?
9. Каким образом можно настроить автоматическое преобразование бизнес-сообщений в формат JSON?
10. Какими путями можно разрешить проблему бесконечного заикливания ссылок при сериализации объекта в JSON?

#### Индивидуальные задания

1. Для созданного на предыдущих занятиях web-приложения (*по варианту*) реализуйте бизнес-логику (доступ к основным объектам БД и операции с ними) через REST API.

Продумайте структуру своего REST API, используйте в ней под-ресурсы, path-, query-параметры и прочие возможности JAX-RS.

Необходимо повторно использовать ту же реализацию бизнес логики, которая ранее была доступна через веб-интерфейс.

2. Протестируйте работу служб двумя способами: с помощью REST Client и SoapUI.

3\*. Внедрите фреймворк Jackson для сериализации и десериализации пересылаемых сообщений в формат JSON.

4. Создайте простого потребителя этих служб, который содержит код для обращения к ресурсам сервера: получение в виде списка, добавление, удаление, изменение.

5\*. Создайте подробную документацию вашего REST API с помощью Swagger.