

Практика Java, задание №2.2. Сервлет-фильтры, контроль доступа к ресурсам сервиса

Цели задания

- Глубже познакомиться с понятием Java Servlet как компонента, расширяющего возможности web-сервера и изучить жизненный цикл сервлета
- Изучить механизм Servlet Filter, как инструмент предобработки запроса
- Познакомится с аннотацией Spring @WebFilter и добавить новый фильтр в проект из задачи 2.1
- Познакомится со способом контроля доступа к ресурсам сервиса, реализуемого с помощью фильтров

Задание

Необходимо расширить программу, разработанную в задаче №2.1 практики таким образом, чтобы перед точкой входа, реализуемой в виде сервлета добавился сервлет-фильтр. Данный фильтр должен применяться для контроля доступа к ресурсам сервиса со стороны клиента.

1. Необходимо в базу данных проекта добавить новую таблицу SESSION с полями:

- `ID integer`, первичный ключ, счетчик сессий
- `SESSION_ID varchar(24)`, уникальный идентификатор сессии
- `START_TIME datetime`, время открытия сессии
- `TIMEOUT_MINUTES integer`, время действия сессии в минутах

2. Фильтр должен считывать из заголовка клиентского запроса уникальный идентификатор сессии, представленный в формате ключ-значение. Ключ - константа "SESSION-ID". Значение - некая строка из 4 групп по 4 символа. Например:

SESSION-ID=1234-AAFF-BB55-DD22

3. Фильтр должен проверять как наличие самого поля SESSION-ID в заголовке, так и проверку значения этого поля. Если поле SESSION-ID в заголовке отсутствует, необходимо вернуть ошибку "Invalid session".

4. Если поле SESSION-ID в заголовке присутствует, то необходимо проанализировать его значение, сравнив с активными сессиями из базы данных:

- Сессия с таким значением SESSION_ID ищется в базе данных
- Если такая запись не находится, то возвращаем ошибку "Unknown session"
- Если такая запись находится, то необходимо проверить не истекло ли время сессии. Если истекло, то возвращаем ошибку "Session expired"
- Если время не истекло, то доступ разрешен, передаем управление из фильтра в сервлет

5. Приветствуется оптимизация количества запросов к базе данных путем создания в памяти web-приложения кэша сессий.