

ETags в службах RESTful - Spring Boot

Семченко Регина Викторовна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Еровлев Павел Андреевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описывается процесс установки заголовков ETags. Так же использованы заголовки запроса If-None-Match, для проверки изменений в ответе. Практическим результатом является измененные заголовки Spring Boot.

Ключевые слова: Spring Boot, Java, заголовки

ETags in RESTful Services - Spring Boot

Semchenko Regina Viktorovna

Sholom-Aleichem Priamursky State University

Student

Erovlev Pavel Andreevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the installation process for the ETags headers. The If-None-Match request headers are also used to check for changes in the response. The bottom line is the changed Spring Boot headers.

Keywords: Spring Boot, Java, headlines

ETag или entity tag — один из регламентируемых спецификацией «RFC 7232» служебных заголовков протокола «HTTP/1.1», который может быть установлен web-сервером в фазе формирования ответа, на полученный от клиента запрос.

Цель данной работы изменить стандартные заголовки при загрузки приложений Spring Boot.

В своей работе А.Б.Джемалетдинов, А.А.Шевченко рассмотрели вопросы создания тестов для Spring Boot mvc контроллеров [1]. В.И.Зарайский провел обзор на разработку модуля автоматизации работы с конференциями в кафедральном приложении [2]. Р.И.Ибраимов

продемонстрировал процесс создания Docker-образа для Spring Boot проекта и развернул его на платформе AWS EC2[3]. Е.О.Кабардинский, А.Г.Ивашко провели сравнительный анализ сервисных шин предприятия, а так же сравнили некоторые ESB, одна из которых Spring Boot [4]. Так же Р.И.Ибраимов, А.Р.Зайчик, Н.С.Минзатов разработали генеалогическое дерево на языке Java с использованием фреймворка Spring Boot b ,b,kbjntrb gedcom4j[5].

ETags можно использовать для кеширования при разработке web-сервисов RESTful с Spring Boot. Платформа Spring обеспечивает поддержку ETags, предоставляя фильтры запросов. Среда Spring предоставляет фильтр ETag под названием «ShallowEtagHeaderFilter», который можно использовать для включения заголовков ETag в ответе. Этот фильтр ETag генерирует хэш для каждого ответа. А хэш добавляется в заголовок ответа и отправляется обратно клиенту. Клиенты службы могут использовать это значение заголовка ETag и отправлять заголовок запроса «If-None-Match» всякий раз, когда они запрашивают одно и то же время размещения ресурсов.

Чтобы добавить фильтр ETag необходимо создать класс конфигурации с именем «EtagConfig.java». Аннотируем класс с помощью «@EnableWebMvc» и «@Configuration» (рис.1).

```
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.Configuration;
5 import org.springframework.web.filter.ShallowEtagHeaderFilter;
6 import org.springframework.web.servlet.config.annotation.EnableWebMvc;
7
8 import java.util.logging.Filter;
9
10 @EnableWebMvc
11 @Configuration
12 public class EtagConfig {
13
14     @Bean
15     public Filter etagFilter() {
16         return (Filter) new ShallowEtagHeaderFilter();
17     }
18 }
```

Рисунок 1 – EtagConfig

В приведенном выше классе конфигурации идет создание bean-компонента фильтра заголовка HTTP с именем «ShallowEtagHeaderFilter». Этот фильтр добавляет заголовок ответа ETag с вычисленным значением хеш-функции для объекта ответа.

Далее создадим класс DTO «EmployeeDTO.java». Это простой класс POJO с двумя полями «id» и «empName» (Рис.2).

```
3     import lombok.*;
4
5     @Getter
6     @Setter
7     @AllArgsConstructor
8     @NoArgsConstructor
9     @ToString
10    public class EmployeeDTO {
11
12        private Integer id;
13        private String empName;
14    }
```

Рисунок 2 - EmployeeDTO

Создадим далее класс контроллера «RESTful» с именем «EmployeeController.java». Этот класс контроллера содержит простое отображение запроса HTTP GET, сопоставленное конечной точке «/get-json». После вызова конечной точки будет получен новый объект «EmployeeDTO» (Рис.3).

```
3     import org.springframework.http.HttpStatus;
4     import org.springframework.http.ResponseEntity;
5     import org.springframework.web.bind.annotation.GetMapping;
6     import org.springframework.web.bind.annotation.RestController;
7
8     @RestController
9     public class EmployeeController {
10
11        @GetMapping("/get-json")
12        public ResponseEntity<EmployeeDTO> getObject() {
13            EmployeeDTO dto = new EmployeeDTO(i: 1, arun: "Pavel");
14            return new ResponseEntity<>(dto, HttpStatus.OK);
15        }
16    }
```

Рисунок 3 - EmployeeController

Запустим приложение Spring Boot. Если вызвать конечную точку службы в первый раз с помощью команды «curl», то можем получить ответ JSON. Ответ содержит заголовок ответа «ETag», как показано на рисунке 4.

```
$ curl -i http://localhost:8080/get-json
HTTP/1.1 200
ETag: "0318c6db1613b8c25732cb355bdc8336"
Content-Type: application/json
Content-Length: 25

{"id":1,"empName":"Pavel"}
ASB@ASB-PC MINGW64 ~/Desktop
$
```

Рисунок 4 – Ответ

Как только «ETag» получен от службы, последовательные запросы ресурса могут быть выполнены вместе с заголовком запроса «If-None-Match» с хешем «ETag» в качестве значения заголовка.

На рисунке 5 показан HTTP-запрос «curl GET» вместе с заголовком запроса «If-None-Match».

```
ASBEASB-PC MINGW64 ~/Desktop
$ curl -i -H 'If-None-Match: "0318c6db1613b8c25732cb355bdc8336"' http://localhost:8080/get-json
HTTP/1.1 304
ETag: "0318c6db1613b8c25732cb355bdc8336"
```

Рисунок 5 - If-None-Match

После получения запроса служба загрузки Spring возвращает код состояния HTTP 304, если в значении ответа нет изменений. Служба не будет отправлять тело ответа, так как в ответе нет изменений.

В данной статье было рассмотрено, как включить заголовки ETag в Spring Boot приложений web-служб RESTful. Также было описано, как использовать заголовок запроса «If-None-Match», чтобы проверить, есть ли какие-либо изменения в полученном ответе.

Библиографический список

1. Джемалетдинов А.Б., Шевченко А.А. Spring boot: создание тестов для spring mvc контроллеров // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. №4(18). С. 104-111.
2. Зарайский В.И. Разработка модуля автоматизации работы с конференциями в кафедральном приложении // Вестник ульяновского государственного технического университета. 2019. №3. С. 74-82.
3. Ибраимов Р.И. Развертывание spring приложения с помощью сервиса aws ec2 и docker-контейнеров // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2020. №1(27). С. 138-147.
4. Кабардинский Е.О., Ивашко А.Г. Сравнительный анализ сервисных шин предприятия (esb) // Математическое и информационное моделирование. 2017. №10. С. 177-185.
5. Ибраимов Р.И., Зайчик А.Р., Минзатров Н.С. Разработка генеалогического дерева средствами фреймвока spring boot // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. 2017. №4(18). С. 18-23.