(/)

Cómo comprimir solicitudes usando la plantilla Spring Rest

Última modificación: 10 de septiembre de 2019

por baeldung (https://www.baeldung.com/author/baeldung/)
(https://www.baeldung.com/author/baeldung/)

HTTP del lado del cliente (https://www.baeldung.com/category/http/)
DESCANSO (https://www.baeldung.com/category/rest/)
Primavera (https://www.baeldung.com/category/spring/) +

1. Introducción

En este breve tutorial, veremos cómo enviar solicitudes HTTP que contienen datos comprimidos.

Además, veremos cómo configurar una aplicación web Spring para que maneje las solicitudes comprimidas.

2. Envío de solicitudes comprimidas

En primer lugar, creemos un método que comprima una matriz de bytes. Esto será útil en breve:

```
public static byte[] compress(byte[] body) throws IOException {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    try (GZIPOutputStream gzipOutputStream = new GZIPOutputStream(baos)) {
        gzipOutputStream.write(body);
    }
    return baos.toByteArray();
}
```

Luego, necesitamos implementar un *ClientHttpRequestInterceptor* para modificar la solicitud. Tenga en cuenta que ambos enviaremos los encabezados de compresión HTTP apropiados, así como también llamaremos a nuestro método de compresión corporal:

```
public ClientHttpResponse intercept(HttpRequest req, byte[] body, ClientHttpRequestExecution exec)
  throws IOException {
    HttpHeaders httpHeaders = req.getHeaders();
    httpHeaders.add(HttpHeaders.CONTENT_ENCODING, "gzip");
    httpHeaders.add(HttpHeaders.ACCEPT_ENCODING, "gzip");
    return exec.execute(req, compress(body));
}
```

Nuestro interceptor toma el cuerpo de solicitud de salida y lo comprime usando el formato GZIP. En este ejemplo, utilizamos el estándar *GZIPOutputStream de* Java para hacer el trabajo por nosotros.

Además, debemos agregar los encabezados apropiados para la codificación de datos. Esto le permite al punto final de destino saber que se trata de datos comprimidos con GZIP.

Finalmente, agregamos el interceptor a nuestra definición *RestTemplate* :

```
@Bean
public RestTemplate getRestTemplate() {
    RestTemplate restTemplate = new RestTemplate();
    restTemplate.getInterceptors().add(new CompressingClientHttpRequestInterceptor());
    return restTemplate;
}
```

3. Manejo de solicitudes comprimidas

Por defecto, la mayoría de los servidores web no entienden las solicitudes que contienen datos comprimidos. Las aplicaciones web de Spring no son diferentes. Por lo tanto, necesitamos configurarlos para manejar tales solicitudes.

Actualmente, solo los servidores web Jetty y Undertow manejan cuerpos de solicitud con datos en formato GZIP. Consulte nuestro artículo sobre la configuración de la aplicación Spring Boot (https://www.baeldung.com/spring-boot-application-configuration) para configurar un servidor web Jetty o Undertow.

3.1. Servidor web Jetty

En este ejemplo, personalizamos un servidor web Jetty agregando un Jetty *GzipHandler* . Este controlador Jetty está diseñado para comprimir respuestas y descomprimir solicitudes.

Sin embargo, agregarlo al servidor web Jetty no es suficiente. Necesitamos establecer *inflateBufferSize* en un valor mayor que cero para habilitar la descompresión:

```
@Bean
public JettyServletWebServerFactory jettyServletWebServerFactory() {
    JettyServletWebServerFactory factory = new JettyServletWebServerFactory();
    factory.addServerCustomizers(server -> {
        GzipHandler gzipHandler = new GzipHandler();
        gzipHandler.setInflateBufferSize(1);
        gzipHandler.setHandler(server.getHandler());

        HandlerCollection handlerCollection = new HandlerCollection(gzipHandler);
        server.setHandler(handlerCollection);
    });
    return factory;
}
```

3.2. Recuperar servidor web

Del mismo modo, podemos personalizar un servidor web Undertow para descomprimir automáticamente las solicitudes para nosotros. En este caso, necesitamos agregar un *RequestEncodingHandler* personalizado .

Configuramos el controlador de codificación para procesar datos de origen GZIP de la solicitud:

```
@Bean
public UndertowServletWebServerFactory undertowServletWebServerFactory() {
    UndertowServletWebServerFactory factory = new UndertowServletWebServerFactory();
    factory.addDeploymentInfoCustomizers((deploymentInfo) -> {
        deploymentInfo.addInitialHandlerChainWrapper(handler -> new RequestEncodingHandler(handler)
          .addEncoding("gzip", GzipStreamSourceConduit.WRAPPER));
    });
    return factory;
}
```

4. Conclusión

¡Y eso es todo lo que debemos hacer para que las solicitudes comprimidas funcionen!

En este tutorial, cubrimos cómo crear un interceptor para un *RestTemplate* que comprima el contenido de una solicitud. Además, vimos cómo descomprimir automáticamente estas solicitudes en nuestras aplicaciones web Spring.

Es importante tener en cuenta que **solo** debemos **enviar contenido comprimido a servidores web capaces de manejar tales solicitudes** .

Un ejemplo de trabajo completo para el servidor web Jetty ha terminado en GitHub (https://github.com/eugenp/tutorials/tree/master/spring-rest-compress) .

Deja una respuesta

Start the discussion...

Suscribir

CATEGORÍAS

PRIMAVERA (HTTPS://WWW.BAELDUNG.COM/CATEGORY/SPRING/)
DESCANSO (HTTPS://WWW.BAELDUNG.COM/CATEGORY/REST/)
JAVA (HTTPS://WWW.BAELDUNG.COM/CATEGORY/JAVA/)
SEGURIDAD (HTTPS://WWW.BAELDUNG.COM/CATEGORY/SECURITY-2/)
PERSISTENCIA (HTTPS://WWW.BAELDUNG.COM/CATEGORY/PERSISTENCE/)
JACKSON (HTTPS://WWW.BAELDUNG.COM/CATEGORY/JSON/JACKSON/)
HTTP DEL LADO DEL CLIENTE (HTTPS://WWW.BAELDUNG.COM/CATEGORY/HTTP/)
KOTLIN (HTTPS://WWW.BAELDUNG.COM/CATEGORY/KOTLIN/)

SERIE

TUTORIAL DE JAVA "VOLVER A LO BÁSICO" (/JAVA-TUTORIAL)
JACKSON JSON TUTORIAL (/JACKSON)
HTTPCLIENT 4 TUTORIAL (/HTTPCLIENT-GUIDE)
RESTO CON SPRING TUTORIAL (/REST-WITH-SPRING-SERIES)
TUTORIAL SPRING PERSISTENCE (/PERSISTENCE-WITH-SPRING-SERIES)
SEGURIDAD CON PRIMAVERA (/SECURITY-SPRING)

ACERCA DE

SOBRE BAELDUNG (/ABOUT)
LOS CURSOS (HTTPS://COURSES.BAELDUNG.COM)
TRABAJO DE CONSULTORÍA (/CONSULTING)
META BAELDUNG (HTTP://META.BAELDUNG.COM/)
EL ARCHIVO COMPLETO (/FULL_ARCHIVE)
ESCRIBIR PARA BAELDUNG (/CONTRIBUTION-GUIDELINES)
EDITORES (/EDITORS)
NUESTROS COMPAÑEROS (/PARTNERS)
ANUNCIE EN BAELDUNG (/ADVERTISE)

TÉRMINOS DE SERVICIO (/TERMS-OF-SERVICE)
POLÍTICA DE PRIVACIDAD (/PRIVACY-POLICY)
INFORMACIÓN DE LA COMPAÑÍA (/BAELDUNG-COMPANY-INFO)
CONTACTO (/CONTACT)