

## Создание запроса с помощью HttpRequest

JSP & Servlets  
10 уровень, 1 лекция

ОТКРЫТА

### Методы newBuilder(), build()

Класс `HttpRequest` используется для описания http-request, что легко понять из его названия. Этот объект сам по себе ничего не делает, он только содержит разнообразную информацию по поводу http-запроса. Поэтому, как ты уже, наверное, догадываешься, для его создания тоже используется шаблон Builder.

```
1  HttpRequest request = HttpRequest.newBuilder()
2      .method1()
3      .method2()
4      .methodN()
5      .build();
```

Где между вызовами методов `newBuilder()` и `build()` нужно вызвать все методы для конструирования объекта `HttpRequest`.

Пример простейшего запроса выглядит так:

```
1  HttpClient client = HttpClient.newHttpClient();
2  HttpRequest request = HttpRequest.newBuilder()
3      .uri(URI.create("http://javarush.ru"))
4      .build();
5  HttpResponse response = client.send(request, HttpResponse.BodyHandlers.ofString());
```

Все методы класса `HttpRequest` ты можешь найти по ссылке [в официальной документации](#).

А дальше мы разберем самые популярные из них.

### Метод uri()

С помощью метода `uri()` можно задать URI (или URL), к которому будет отправлен http-запрос. Пример:

```
1  HttpRequest request = HttpRequest.newBuilder()
2      .uri( URI.create("http://javarush.ru") )
3      .build();
```

Кстати, можно записать этот код еще немного короче, передав URI прямо в метод `newBuilder()`:

```
1  HttpRequest request = HttpRequest.newBuilder( URI.create("http://javarush.ru") ).build();
```

**Важно!** URI можно создать двумя способами:

- `new URI(String)`
- `URI.create(String)`

Второй способ предпочтительнее. Первый способ, к сожалению, не очень хорош, потому-то конструктор URI объявлен так `public URI(String str) throws URISyntaxException`, а **URISyntaxException** — это checked-исключение.

# Методы GET(), POST(), PUT(), DELETE()

Задать http-метод запроса можно с помощью методов:

- GET()
- POST()
- PUT()
- DELETE()

Вот как будет выглядеть простой GET-запрос:

```
1  HttpRequest request = HttpRequest.newBuilder()  
2      .uri(new URI("https://javarush.com"))  
3      .GET()  
4      .build();
```

## Метод version()

Также можно задать версию HTTP-протокола. Их всего 2 варианта:

- HttpClient.Version.HTTP\_1\_1
- HttpClient.Version.HTTP\_2

Допустим, ты хочешь создать запрос по протоколу HTTP/2, тогда тебе нужно будет написать:

```
1  HttpRequest request = HttpRequest.newBuilder()  
2      .uri(new URI("https://javarush.com"))  
3      .version( HttpClient.Version.HTTP_2 )  
4      .GET()  
5      .build();
```

Очень просто, не правда ли? :)

## Метод timeout()

Также можно задать время выполнения запроса. Если оно пройдет, а запрос так и не будет выполнен, то выкинется исключение **HttpTimeoutException**.

Само время задается с помощью объекта `Duration` из Java DateTime API. Пример:

```
1  HttpRequest request = HttpRequest.newBuilder()  
2      .uri(new URI("https://javarush.com"))  
3      .timeout( Duration.of(5, SECONDS) )  
4      .GET()  
5      .build();
```

Наличие этого метода показывает, что классы `HttpClient` и `HttpRequest` могут решать самые разнообразные задачи. Представь, что ты выполняешь запрос, а что-то случилось с сетью и он продлился 30 секунд. Куда полезнее сразу получить исключение и правильно на него среагировать.

## Метод header()

Также к любому запросу можно добавить сколько угодно заголовков. И сделать это так же просто, как и все остальное. Для этого есть специальный метод — `header()`. Пример:

```
1  HttpRequest request = HttpRequest.newBuilder()  
2      .uri(new URI("https://javarush.com"))  
3      .header("name1", "value1")
```

```
4      .header("name2", "value2")
5      .GET()
6      .build();
```

Есть еще один альтернативный способ задавания сразу много заголовков. Может пригодиться, если, ты, допустим, преобразовал список заголовков в массив:

```
1  HttpRequest request = HttpRequest.newBuilder()
2      .uri(new URI("https://javarush.com"))
3      .headers("name1", "value1", "name2", "value2")
4      .GET()
5      .build();
```

Все элементарное просто.



Комментарии (1)

популярныеновыестарые

JavaCoder

Введите текст комментария

Саша И. Уровень 67

5 декабря 2022, 10:14

"Есть еще один альтернативный способ задавания заголовков сразу много заголовков."

Ответить

0

ОБУЧЕНИЕ

- Курсы программирования
- Курс Java
- Помощь по задачам
- Подписки
- Задачи-игры

СООБЩЕСТВО

- Пользователи
- Статьи
- Форум
- Чат
- Истории успеха
- Активности

КОМПАНИЯ

- О нас
- Контакты
- Отзывы
- FAQ
- Поддержка



JavaRush — это интерактивный онлайн-курс по изучению Java-программирования с нуля. Он содержит 1200 практических задач с проверкой решения в один клик, необходимый минимум теории по основам Java и мотивирующие фишки, которые помогут пройти курс до конца: игры, опросы, интересные проекты и статьи об эффективном обучении и карьере Java-девелопера.

ПОДПИСЫВАЙТЕСЬ

ЯЗЫК ИНТЕРФЕЙСА

Русский

СКАЧИВАЙТЕ НАШИ ПРИЛОЖЕНИЯ

