

Добавление body в HttpRequest

JSP & Servlets
10 уровень, 2 лекция

ОТКРЫТА

3.1 BodyPublishers

Я надеюсь, вы не забыли, что кроме **GET-запросов**, есть еще **PUT** и **POST-запросы**, когда к запросу нужно еще добавить `request body`, то есть тело запроса.

Для этого у класса `HttpRequest` есть специальный внутренний класс `BodyPublisher`. Хотя технически это интерфейс, у которого есть несколько реализаций, о которых мы поговорим ниже

И начнем мы с самой простой – отсутствие тела запроса. Да, бывает и так.

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

Просто и красиво.

3.2 ofString()

Второй самый распространенный вариант – это передать некую строку в качестве request body. Делается это очень просто:

```
1  HttpRequest request = HttpRequest.newBuilder()
2      .uri(new URI("https://javarush.com"))
3      .headers("Content-Type", "text/plain;charset=UTF-8")
4      .POST(HttpRequest.BodyPublishers.ofString("Привет"))
5      .build();
```

Кстати, можно задать и кодировку передаваемой строки. Бывает очень полезно, если http-сервер, к которому отправляется запрос, работает не на UTF8.

```
1  HttpRequest request = HttpRequest.newBuilder()
2      .uri(new URI("https://javarush.com"))
3      .POST(HttpRequest.BodyPublishers.ofString("Привет", Charset.US-ASCII))
4      .build();
```

3.3 fromFile()

Ну и наконец, ты можешь захотеть приложить к **POST-запросу** файл. Именно таким образом, обычно твои аватарки и заливаются на сервер. Для этого нужно вызвать метод `fromFile()`, куда передать `Path` локального файла:

```
1  Path avatar = Path.of("c://avatar.jpeg");
2
3  HttpRequest request = HttpRequest.newBuilder()
4      .uri(new URI("https://javarush.com"))
5      .headers("Content-Type", "image/jpeg")
6
```

7	<pre>.POST(HttpRequest.BodyPublishers.fromFile(avatar)) .build();</pre>
---	---

3.4 ofByteArray()

Еще один распространенный сценарий – отправить на сервер набор байт. Например, ты сериализовал некий объект в виде набора байт, что-то зашифровал, или просто хочешь переслать некий буфер данных. Для этого понадобится метод `.ofByteArray()`.

В качестве параметра этот метод принимает массив байт. Пример:

```
1 byte[] data = "Мое секретное сообщение".getBytes();
2 byte[] dataEncrypted = SuperEncryptor.encrypt(data);
3
4 HttpRequest request = HttpRequest.newBuilder()
5     .uri(new URI("https://javarush.com"))
6     .headers("Content-Type", "application/octet-stream")
7     .POST(HttpRequest.BodyPublishers.ofByteArray(dataEncrypted))
8     .build();
```

3.5 ofInputStream()

Наконец последний, но не менее интересный сценарий – это прикрепление `InputStream` к **POST-запросу**.

Для этого есть специальная функция – `ofInputStream()`. И она очень хитрая. Она позволяет организовать пересылку данных из потока в поток, прикрепить к **POS-запросу** поток данных, который еще даже не открыт

В функцию `ofInputStream()` нужно передать функцию, которая в качестве результата вернет поток `InputStream`.

Пример:

```
1 byte[] data = "Мое секретное сообщение".getBytes();
2 //заворачиваем массив байт в поток.
3 InputStream is = new ByteArrayInputStream(data);
4
5 HttpRequest request = HttpRequest.newBuilder()
6     .uri(new URI("https://javarush.com"))
7     .headers("Content-Type", "application/octet-stream")
8     .POST(HttpRequest.BodyPublishers.ofInputStream (() -> is;))
9     .build();
```

Я не стал тут городить огород, но думаю, принцип ты понял. Зачем так сделали? Почти всегда ты можешь воспользоваться методом `ofByteArray()`. Но если хочешь послать данные асинхронно или нужно реализовать какой-то особо сложный сценарий, то можешь воспользоваться этим методом.

JavaCoder

--	--	--	--	--	--	--	--	--	--

Введите текст комментария

Отправить

JavaCoder Уровень 51 меньше 5 секунд назад

Может не fromFile(), а ofFile() ?

Ответить

0

Евгений Frontend Developer 14 октября 2022, 17:00

В п.3.2 можно передать не просто строку, а строку в виде JSON объекта. Как раз такой вариант и применяется на практике.

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



"Программистами не рождаются" © 2023 JavaRush