

HTTP-методы

JSP & Servlets
9 уровень, 3 лекция

ОТКРЫТА

4.1 Список HTTP-методов

Самым первым словом в HTTP-запросе идет **имя метода**. Тут даже есть некоторая аналогия с вызовом методов в Java. Метод в HTTP-запросе определяет основную операцию, которую нужно выполнить над ресурсом.

Над каким же ресурсом? Все дело в том, что на заре всемирной паутины сервера просто хранили HTML-файлы, соответственно, запрос был к такому файлу и описывал некоторое действие, которое нужно произвести с ресурсом/файлом.

Стандарт HTTP задает такие методы:

#	Метод	Описание
1	GET	Используется для запроса содержимого указанного ресурса.
2	POST	Используется для передачи данных с клиента на сервер. Изменяет состояние ресурса на сервере.
3	PUT	Используется для передачи данных с клиента на сервер. Создает новый ресурс на сервере.
4	DELETE	Удаляет указанный ресурс на сервере.
5	HEAD	Аналогичен GET, но тело ответа отсутствует. Нужен для получения заголовков ответа
6	OPTIONS	Запрашивает у сервера список поддерживаемых методов для указанного ресурса.
7	TRACE	Служебный метод. Позволяет узнать, не меняют ли запрос серверы, через которые он проходит.
8	CONNECT	Служебный метод. Используется для установления защищенного соединения.

4.2 Метод GET

Метод GET — самый популярный HTTP-метод. Именно его вызывает браузер, когда отправляет запрос на сервер за очередной страницей.

Например, вы перешли в браузере по ссылке `http://javarush.ru/path/resource?param1=value1¶m2=value2`, — тогда браузер отправит на сервер JavaRush HTTP-запрос который будет начинатся с такой **starting line**:

GET /path/resource?param1=value1¶m2=value2 **HTTP/1.1**

В результате сервер должен будет отправить браузеру HTTP-ответ, в котором написать статус запроса, а также выслать запрашиваемый ресурс.

Подразумевается, что **многократный вызов метода GET не меняет состояние сервера, и сервер каждый раз должен возвращать один и тот же ответ**. Поэтому в протоколе заложено хитрое управление кешированием объекта.

Во-первых, ресурсы, полученные с помощью запроса GET, **браузер может кешировать на своей стороне** по своему усмотрению (есть нюансы).

Во-вторых, при отправке запроса на сервер можно указать специальный заголовок `If-Modified-Since` и `дату`. Если запрашиваемые ресурс/документ изменился с указанной даты, то сервер его пришлет. Если не изменился, то тело ресурса не передается. Подразумевается, что оно закешировано на клиенте.

Кеширование страниц (GET-запросов) используется постоянно, так что советую разобраться в этом вопросе.

4.3 Методы POST и PUT

Метод POST используется для того, чтобы изменить ресурс на сервере. Например, когда вы загружаете картинку на сервер, ваш браузер отправляет POST-запрос.

Рассмотрим HTTP-запрос, который будет начинаться с такой starting line:

```
1  POST /path/resource?param1=value1&param2=value2 HTTP/1.1
2  headers...
3
4  <request body>
```

В результате сервер должен будет отправить браузеру HTTP-ответ, в котором напишет статус запроса, а также выслать измененный ресурс. **Многократный вызов метода POST изменяет состояние сервера и сервер каждый раз может возвращать разный ответ.**

GET и **POST** — два самых распространенных запроса в вебе. Чтобы тебе было проще запомнить, как работают методы, рассмотрим такую таблицу:

	GET	POST	PUT
Запрос	Только URL	URL и тело запроса	URL и тело запроса
Ответ	Код и тело ответа	Код и тело ответа	Код ответа

Более детально про POST-запрос ты можешь почитать [по ссылке](#).

4.4 Метод DELETE

Ну и наконец информация по **методу DELETE**. Тут вообще все просто.

Например, мы хотим удалить определенный ресурс на сервере. Посылаем ему запрос вида:

```
DELETE /path/resource?param1=value1&param2=value2 HTTP/1.1
```

После получения этого запроса сервер удалит указанный ресурс. Если, конечно, у тебя есть права на его удаление.



JavaCoder

Введите текст комментария

ЕвгенийFrontend Developer

14 октября, 13:52

...

Где PATCH?

Ответить

0

GrekZorbaQA Automation Engineer в EPAM

10 октября, 10:23

...

Перепутали POST and PUT

Ответить

+3

Эллеонора КерриУровень 41

8 ноября, 16:56

...

Где именно, подскажите.

Ответить

0

GrekZorbaQA Automation Engineer в EPAM

18 ноября, 12:14

...

Ответить

0

Andrey PanchenkoМоет полы в Яндекс

17 сентября, 11:24

...

Из [вики](#): Фундаментальное различие методов **POST** и **PUT** заключается в понимании предназначений **URI** ресурсов. Метод **POST** предполагает, что по указанному **URI** будет производиться обработка передаваемого клиентом содержимого. Используя **PUT**, клиент предполагает, что загружаемое содержимое соответствует находящемуся по данному **URI** ресурсу.

Ответить

+2

egorabramov216Уровень 23

23 июня, 14:09

...

"Чтобы тебе было было запомнить, как работают методы, рассмотрим такую таблицу..."

Ответить

+2

Pavel Bely.Net Developer в Brusnika

3 июня, 11:48

...

по моему вы пут и пост перепутали

Ответить

+13

Herman KulikJava Developer в EPAM

28 июня, 11:40

...

тоже хотел написать

Ответить

+2

Oleg KhilkoУровень 51

14 августа, 09:43

...

я с вами)

Ответить

+2

Эллеонора КерриУровень 41

8 ноября, 16:55

...

Уточните, пожалуйста, где именно?

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 



"Программистами не рождаются" © 2022 JavaRush