HttpClient 4 - Следуйте перенаправлениям для POST

* [HTTP Client](https://www.codeflow.site/ru/tag/HTTP%20Client)

**1. Обзор**

В этом кратком руководстве показано, как настроить Apache HttpClient 4 для автоматического следования перенаправлениям для запросов POST.

Если вы хотите копнуть глубже и узнать другие интересные вещи, которые вы можете сделать с помощью HttpClient, перейдите по ссылке [**основное руководство по HttpClient**](https://www.codeflow.site/httpclient-guide).

По умолчанию автоматически выполняются только запросы GET, приводящие к перенаправлению. Если на запросы POST отвечают либо *HTTP 301 Moved Permanently*, либо *302 Found*- **перенаправление не выполняется автоматически**.

Это указано в [HTTP RFC 2616](https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.3):

*\_\_*Если код состояния 301 получен в ответ на запрос, отличный от GET или HEAD, пользовательский агент НЕ ДОЛЖЕН автоматически перенаправлять запрос, если он не может быть подтвержден пользователем, так как это может изменить условия, при которых был выдан запрос.

Конечно, есть случаи, когда нам нужно изменить это поведение и ослабить строгую HTTP-спецификацию.

Во-первых, давайте проверим поведение по умолчанию:

@Test

**public** **void** **givenPostRequest\_\_whenConsumingUrlWhichRedirects\_\_thenNotRedirected**()

**throws** ClientProtocolException, IOException {

HttpClient instance = HttpClientBuilder.create().build();

HttpResponse response = instance.execute(**new** HttpPost("http://t.co/I5YYd9tddw"));

assertThat(response.getStatusLine().getStatusCode(), equalTo(301));

}

Как видите, **перенаправление не выполняется по умолчанию**, и мы возвращаем *301 Код состояния*.

=== **2. Перенаправление по HTTP POST**

==== \*\* 2.1. Для HttpClient 4.3 и после

\*\*

В HttpClient 4.3 был представлен API более высокого уровня для создания и настройки клиента:

@Test

**public** **void** **givenRedirectingPOST\_\_whenConsumingUrlWhichRedirectsWithPOST\_\_thenRedirected**()

**throws** ClientProtocolException, IOException {

HttpClient instance =

HttpClientBuilder.create().setRedirectStrategy(**new** LaxRedirectStrategy()).build();

HttpResponse response = instance.execute(**new** HttpPost("http://t.co/I5YYd9tddw"));

assertThat(response.getStatusLine().getStatusCode(), equalTo(200));

}

Обратите внимание ***HttpClientBuilder*теперь является отправной точкой свободного API**, который позволяет полностью настроить клиент более читабельным способом, чем раньше.

==== **2.2. Для HttpClient 4.2**

В предыдущей версии HttpClient (4.2) мы могли настроить стратегию перенаправления непосредственно на клиенте:

@SuppressWarnings("deprecation")

@Test

**public** **void** **givenRedirectingPOST\_\_whenConsumingUrlWhichRedirectsWithPOST\_\_thenRedirected**()

**throws** ClientProtocolException, IOException {

DefaultHttpClient client = **new** DefaultHttpClient();

client.setRedirectStrategy(**new** LaxRedirectStrategy());

HttpResponse response = client.execute(**new** HttpPost("http://t.co/I5YYd9tddw"));

assertThat(response.getStatusLine().getStatusCode(), equalTo(200));

}

Обратите внимание, что теперь, с новым *LaxRedirectStrategy*, ограничения HTTP ослаблены и **перенаправление также выполняется по POST**- что приводит к коду состояния *200 OK*.

==== **2.3. Предварительно HttpClient 4.2**

До HttpClient 4.2 класс *LaxRedirectStrategy*не существовал, поэтому нам нужно свернуть наш собственный:

@Test

**public** **void** **givenRedirectingPOST\_\_whenConsumingUrlWhichRedirectsWithPOST\_\_thenRedirected**()

**throws** ClientProtocolException, IOException {

DefaultHttpClient client = **new** DefaultHttpClient();

client.setRedirectStrategy(**new** DefaultRedirectStrategy() {

/\*\* \*\* Redirectable methods. \*\* / private String[]REDIRECT\_\_METHODS = new String[]{

HttpGet.METHOD\_\_NAME, HttpPost.METHOD\_\_NAME, HttpHead.METHOD\_\_NAME

};

**@Override**

protected boolean isRedirectable(String method) {

for (String m : REDIRECT\_\_METHODS) {

if (m.equalsIgnoreCase(method)) {

return true;

}

}

return false;

}

});

HttpResponse response = client.execute(new HttpPost("http://t.co/I5YYd9tddw"));

assertThat(response.getStatusLine().getStatusCode(), equalTo(200));

}

=== **3. Заключение**

В этом кратком руководстве показано, как настроить любую версию Apache HttpClient 4, чтобы она также следовала перенаправлениям для запросов HTTP POST - ослабляя строгий стандарт HTTP.

Реализация всех этих примеров и фрагментов кода **может быть найдена в [my[my project github](https://github.com/eugenp/tutorials/tree/master/httpclient" \l "readme)**- это проект на основе Eclipse, поэтому он должен легко импортировать и запускать как есть.