HTTP POST using JSON in Java

I would like to make a simple HTTP POST using JSON in Java.

Let's say the URL is `www.site.com`

and it takes in the value `{"name":"myname","age":"20"}` labeled as `'details'` for example.

How would I go about creating the syntax for the POST?

I also can't seem to find a POST method in the JSON Javadocs.

java json http post request

edited Jun 9 '16 at 15:59 cricket_007 33.4k 5 18 45

asked Aug 24 '11 at 20:01 asdf007 459 1 5 3

7 Answers

Here is what you need to do:

1. Get the Apache HttpClient, this would enable you to make the required request
2. Create an HttpPost request with it and add the header "application/x-www-form-urlencoded"
3. Create a StringEntity that you will pass JSON to it
4. Execute the call

The code roughly looks like (you will still need to debug it and make it work)

```
//Deprecated
//HttpClient httpClient = new DefaultHttpClient();

HttpClient httpClient = HttpClientBuilder.create().build(); //Use this instead

try {

    HttpPost request = new HttpPost("http://yoururl");
    StringEntity params =new StringEntity("details={\"name\":\"myname\",\"age\":\"20\"}
");
    request.addHeader("content-type", "application/x-www-form-urlencoded");
    request.setEntity(params);
    HttpResponse response = httpClient.execute(request);

    //handle response here...

}catch (Exception ex) {

    //handle exception here

} finally {
    //Deprecated
    //httpClient.getConnectionManager().shutdown();
}
```

edited Dec 29 '16 at 21:48 dckuehn

answered Aug 24 '11 at 20:21 momo

hc.apache.org/httpclient-3.x/ but it seems to be down? Hmph. – asdf007 Aug 24 '11 at 20:34

3 You could but it always good practice to abstract it out as JSONObject as if you are doing directly in the string, you might program the string wrongly and causing syntax error. By using JSONObject you make sure that your serialization is always follow the right JSON structure – momo Aug 24 '11 at 20:47

3 In principal, they are both just transmitting data. The only difference is how you process it in the server. If you have only few key-value pair then a normal POST parameter with key1=value1, key2=value2, etc is probably enough, but once your data is more complex and especially containing complex structure (nested object, arrays) you would want to start consider using JSON. Sending complex structure using a key-value pair would be very nasty and difficult to parse on the server (you could try and you'll see it right away). Still remember the day when we had to do that urgh.. it wasn't pretty.. – momo Aug 24 '11 at 21:53

1 Glad to help! If this is what you are looking for, you should accept the answer so other people with similar questions have good lead to their questions. You can use the check mark on the answer. Let me know if you have further questions – momo Aug 25 '11 at 1:56

7 Shouldn't the content-type be 'application/json'. 'application/x-www-form-urlencoded' implies the string will be formatted similar to a query string. NM I see what you did, you put the json blob as a value of a property. – Matthew Ward May 2 '13 at 20:31

You can make use of Gson library to convert your java classes to JSON objects.

Create a pojo class for variables you want to send as per above Example {"name":"myname","age":"20"}

```
class pojo1
{
String name;
String age;
//generate setter and getters
}
```

once you set the variables in pojo1 class you can send that using the following code

```
String       postUrl       = "www.site.com";// put in your url
Gson         gson          = new Gson();
HttpClient   httpClient    = HttpClientBuilder.create().build();
HttpPost     post          = new HttpPost(postUrl);
StringEntity postingString = new StringEntity(gson.toJson(pojo1));//gson.tojson() converts
your pojo to json
post.setEntity(postingString);
post.setHeader("Content-type", "application/json");
HttpResponse  response = httpClient.execute(post);
```

and these are the imports

```
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.StringEntity;
```

and for GSON

```
import com.google.gson.Gson;
```

edited Feb 18 '16 at 20:43

answered Jan 17 '14 at 4:00

Prakash
1,371 2 16 26

hi, how do you create your httpClient object? It's an interface – user3290180 Jan 26 '16 at 11:19

1 Yes that is an Interface. You can create an instance using 'HttpClient httpClient = new DefaultHttpClient();' – Prakash Jan 26 '16 at 15:03

1 now that is deprecated, we must use HttpClient httpClient = HttpClientBuilder.create().build(); – user3290180 Jan 28 '16 at 9:03

3 How importing HttpClientBuilder ? – saman Apr 7 '16 at 14:11

are you missed new in line 5th of the second code example? `StringEntity postingString = new StringEntity(gson.toJson( new pojo1()));` I have a error when I removing new. – saman Apr 7 '16 at 14:21

@Prakash can you please look on this question [stackoverflow.com/questions/42024158/… – Kasun Siyambalapitiya Feb 6 at 4:41

@momo's answer for Apache HttpClient, version 4.3.1 or later. I'm using `JSON-Java` to build my JSON object:

```
JSONObject json = new JSONObject();
json.put("someKey", "someValue");

CloseableHttpClient httpClient = HttpClientBuilder.create().build();

try {
```

```
    httpClient.execute(request);
// handle response here...
} catch (Exception ex) {
    // handle exception here
} finally {
    httpClient.close();
}
```

edited Aug 19 '14 at 14:55

answered Nov 11 '13 at 18:02

Cigano Morrison Mendez
2,373 6 27 50

2 thanks, but instead of "application/x-www-form-urlencoded", content-type should be "application/json"! – Ehsan Mirsaeedi Aug 19 '14 at 6:52

It's probably easiest to use HttpURLConnection.

http://www.xyzws.com/Javafaq/how-to-use-httpurlconnection-post-data-to-web-server/139

You'll use JSONObject or whatever to construct your JSON, but not to handle the network; you need to serialize it and then pass it to an HttpURLConnection to POST.

edited Jun 17 '15 at 20:42

answered Aug 24 '11 at 20:06

Alex Churchill
2,886 3 20 40

JSONObject j = new JSONObject(); j.put("name", "myname"); j.put("age", "20"); Like that? How do I serialize it? – asdf007 Aug 24 '11 at 20:26

@asdf007 just use `j.toString()` . – Alex Churchill Aug 24 '11 at 20:42

but this connection is not async.... rite? – gumuruh Mar 23 '12 at 12:58

That's true, this connection is blocking. This probably isn't a big deal if you are sending a POST; it is much more important if you running a webserver. – Alex Churchill Apr 8 '13 at 23:10

1 @TobiasRoland Thanks for the heads-up; I've updated – Alex Churchill Jun 17 '15 at 20:42

```
    protected void sendJson(final String play, final String prop) {
    Thread t = new Thread() {
    public void run() {
       Looper.prepare(); //For Preparing Message Pool for the childThread
       HttpClient client = new DefaultHttpClient();
       HttpConnectionParams.setConnectionTimeout(client.getParams(), 1000); //Timeout
Limit
       HttpResponse response;
       JSONObject json = new JSONObject();

           try {
               HttpPost post = new HttpPost("http://192.168.0.44:80");
               json.put("play", play);
               json.put("Properties", prop);
               StringEntity se = new StringEntity(json.toString());
               se.setContentType(new BasicHeader(HTTP.CONTENT_TYPE, "application/json"));
               post.setEntity(se);
               response = client.execute(post);

               /*Checking response */
               if (response != null) {
                   InputStream in = response.getEntity().getContent(); //Get the data in
the entity
               }

           } catch (Exception e) {
               e.printStackTrace();
               showMessage("Error", "Cannot Estabilish Connection");
           }

           Looper.loop(); //Loop in the message queue
       }
    };
    t.start();
}
```

answered Sep 8 '15 at 13:58

Mohamed Elgarnaoui
138 1 7

4 Please consider editing your post to add more explanation about what your code does and why it will solve the problem. An answer that mostly just contains code (even if it's working) usually won't help the OP to understand their problem – Reeno Sep 8 '15 at 14:14

This worked for me, thanks! – Lozzano Oct 15 '15 at 18:54

you are welcome – Mohamed Elgarnaoui Oct 17 '15 at 15:50

@Reeno this code is pretty much self-explained – Mladen Adamovic Jan 20 at 13:10

Try this code:

```
HttpClient httpClient = new DefaultHttpClient();

try {
    HttpPost request = new HttpPost("http://yoururl");
    StringEntity params =new StringEntity("details={\"name\":\"myname\",\"age\":\"20\"}
");
    request.addHeader("content-type", "application/json");
    request.addHeader("Accept","application/json");
    request.setEntity(params);
    HttpResponse response = httpClient.execute(request);

    // handle response here...
}catch (Exception ex) {
    // handle exception here
} finally {
    httpClient.getConnectionManager().shutdown();
}
```

edited Sep 13 '14 at 9:46
vdenotaris
3,746 9 42 77

answered Sep 13 '14 at 9:24
Sonu Dhakar
61 1 1

Thanks! Only your answer solved the encoding issue :) – Shrikant Feb 12 '15 at 9:38

@SonuDhakar why you send `application/json` both as a accept header and as content-type – Kasun Siyambalapitiya Feb 6 at 4:26

I found this question looking for solution about how to send post request from java client to Google Endpoints. Above answers, very likely correct, but not work in case of Google Endpoints.

Solution for Google Endpoints.

1. Request body must contains only JSON string, not name=value pair.
2. Content type header must be set to "application/json".

```
post("http://localhost:8888/_ah/api/langapi/v1/createLanguage",
                   "{\"language\":\"russian\", \"description\":\"dsfsdfsdfsdfsd\"}");


public static void post(String url, String param ) throws Exception{
  String charset = "UTF-8";
  URLConnection connection = new URL(url).openConnection();
  connection.setDoOutput(true); // Triggers POST.
  connection.setRequestProperty("Accept-Charset", charset);
  connection.setRequestProperty("Content-Type", "application/json;charset=" + charset);

  try (OutputStream output = connection.getOutputStream()) {
    output.write(param.getBytes(charset));
  }

  InputStream response = connection.getInputStream();
}
```

It sure can be done using HttpClient as well.

edited Jun 13 '15 at 15:39

answered Jun 13 '15 at 15:27
yurin
1,154 12 15