

How do I POST JSON data with Curl from a terminal/commandline to Test Spring REST?

Asked 7 years, 11 months ago Active 13 days ago Viewed 2.5m times

I use Ubuntu and installed [cURL](#) on it. I want to test my Spring REST application with cURL. I wrote my POST code at the Java side. However, I want to test it with cURL. I am trying to post a JSON data. Example data is like this:

2511

```
{ "value": "30", "type": "Tip 3", "targetModule": "Target
3", "configurationGroup": null, "name": "Configuration Deneme
3", "description": null, "identity": "Configuration Deneme
3", "version": 0, "systemId": 3, "active": true }
```



647

I use this command:

```
curl -i \
-H "Accept: application/json" \
-H "X-HTTP-Method-Override: PUT" \
-X POST -d "value": "30", "type": "Tip 3", "targetModule": "Target
3", "configurationGroup": null, "name": "Configuration Deneme
3", "description": null, "identity": "Configuration Deneme
3", "version": 0, "systemId": 3, "active": true \
http://localhost:8080/xx/xxx/xxxx
```

It returns this error:

```
HTTP/1.1 415 Unsupported Media Type
Server: Apache-Coyote/1.1
Content-Type: text/html; charset=utf-8
Content-Length: 1051
Date: Wed, 24 Aug 2011 08:50:17 GMT
```

The error description is this:

The server refused this request because the request entity is in a format not supported by the requested resource for the requested method ().

Tomcat log: "POST /ui/webapp/conf/clear HTTP/1.1" 415 1051

What is the right format of the cURL command?

This is my Java side PUT code (I have tested GET and DELETE and they work):

```
@RequestMapping(method = RequestMethod.PUT)
public Configuration updateConfiguration(HttpServletResponse response, @RequestBody
Configuration configuration) { //consider @Valid tag
    configuration.setName("PUT worked");
    //todo If error occurs response.sendError(HttpServletResponse.SC_NOT_FOUND);
    return configuration;
}
```

[json](#)[rest](#)[spring-mvc](#)[curl](#)[http-headers](#)

edited Jun 24 at 13:16



YanDatsiuk

523 2 7 23

asked Aug 24 '11 at 8:51



kamaci

34.5k 60 197 318

4 checkout the [link](#) for spring 3.2.0 post requests. – AmirHd Jun 5 '13 at 9:41

5 There is a nice post [Using Curl For Ad Hoc Testing Of RESTful Microservices](#) which covers this with multiple examples. – upitau Aug 28 '16 at 11:03

22 Answers



You need to set your content-type to application/json. But `-d` sends the Content-Type `application/x-www-form-urlencoded`, which is not accepted on Spring's side.

3920



Looking at the [curl man page](#), I think you can use `-H` :

```
-H "Content-Type: application/json"
```



Full example:

```
curl --header "Content-Type: application/json" \
  --request POST \
  --data '{"username":"xyz","password":"xyz"}' \
  http://localhost:3000/api/Login
```

(-H is short for --header , -d for --data)

Note that -request POST is *optional* if you use -d , as the -d flag implies a POST request.

On Windows, things are slightly different. See the comment thread.

edited May 3 '18 at 21:43



Clemens Tolboom

963 11 19

answered Aug 24 '11 at 9:12



Sean Patrick Floyd

240k 52 405 539

4 I removed the -d option from command but still same. – [kamaci](#) Aug 24 '11 at 9:42

214 For windows, single quotes around json did not work and I ended up escaping double quotes. curl -X POST -H "Content-Type: application/json" -d "{ \"key1\": \"value1\" }" http://localhost:3000/api/method – [hlpPy](#) Sep 11 '13 at 17:34 ✎

29 For me under Windows I needed to escape quotes using quotes in this format "{ \"\"key1\"\": \"\"value1\"\" }" . Also this answer: stackoverflow.com/questions/18314796/... – [chodorowicz](#) Jan 27 '14 at 11:10 ✎

2 @chodorowicz that's awful! I only know that syntax from VB! – [Sean Patrick Floyd](#) Jan 27 '14 at 12:42

3 @ostrokach sorry it wasted your time. syntax worked fine for me on OSX when I posted it (haven't retried). Guess it is/was just a platform difference. I imagine the upvotes are from people that it helped. – [Adam Tuttle](#) Nov 1 '15 at 1:23

Try to put your data in a file, say body.json and then use

501

```
curl -H "Content-Type: application/json" --data @body.json
http://localhost:8080/ui/webapp/conf
```

answered Aug 24 '11 at 10:04



Typisch

5,011 1 8 2

- 2 It returns 404 error – [kamaci](#) Aug 24 '11 at 10:22
- 10 You probably should use the `--data-binary` option instead of `--data` . One would expect that the client sends the data as-is, but `--data` strips CR und LF from the input. – [h2stein](#) Mar 24 '14 at 8:22
- 12 Using cUrl with inline json Strings seems to be a nightmare. There's the need to scape the double quote character. Going with a file like this is nicer. – [Xtreme Biker](#) Jun 26 '14 at 10:26
- 36 It's important to add an `@` character before the name of the file, otherwise it won't work. I just spent 20 minutes banging my head at this crap... – [Radu Murzea](#) Aug 18 '15 at 11:41
- 7 I was going to upvote this answer, but its current score is 404... – [ximo](#) May 30 '18 at 16:57

You might find resty useful: <https://github.com/micha/resty>.

92

It's a wrapper round CURL which simplifies command line REST requests. You point it to your API endpoint, and it gives you PUT and POST commands. (Examples adapted from the homepage)

```
$ resty http://127.0.0.1:8080/data #Sets up resty to point at your endpoing
$ GET /blogs.json               #Gets http://127.0.0.1:8080/data/blogs.json
                                #Put JSON
$ PUT /blogs/2.json '{"id" : 2, "title" : "updated post", "body" : "This is the new."}'
                                # POST JSON from a file
$ POST /blogs/5.json < /tmp/blog.json
```

Also, it's often still necessary to add the Content Type headers. You can do this once, though, to set a default, or add config files per-method per-site: [Setting default RESTY options](#)

edited Feb 8 '13 at 14:34

answered Jan 12 '12 at 19:10



[mo-seph](#)

3,361 7 27 34

80

For Windows, having a single quote for the `-d` value did not work for me, but it did work after changing to double quote. Also I needed to escape double quotes inside curly brackets.

That is, the following did not work:

```
curl -i -X POST -H "Content-Type: application/json" -d '{"key":"val"}'
http://localhost:8080/appname/path
```

But the following worked:

```
curl -i -X POST -H "Content-Type: application/json" -d "{\"key\":\"val\"}"
http://localhost:8080/appname/path
```

edited May 27 '17 at 21:51



Peter Mortensen

14.4k 19 88 117

answered Jun 30 '14 at 22:12



venkatnz

901 6 2

-
- 4 FYI - looks like you're missing a closing double quote around the json body – [acanby](#) Mar 14 '16 at 23:10 ✎
-
- 2 For me on Windows, the " around the data does not work, no quotes works instead – [rodedo](#) Oct 17 '16 at 9:44
-
- 2 If you're using PowerShell, see [this](#) answer. – [rsenna](#) Feb 28 '18 at 19:02
-

It worked for me using:

78

```
curl -H "Accept: application/json" -H "Content-type: application/json" -X POST -d
'{"id":100}' http://localhost/api/postJsonReader.do
```

It was happily mapped to the Spring controller:

```
@RequestMapping(value = "/postJsonReader", method = RequestMethod.POST)
public @ResponseBody String processPostJsonData(@RequestBody IdOnly idOnly) throws
Exception {
    logger.debug("JsonReaderController hit! Reading JSON data!" + idOnly.getId());
    return "JSON Received";
}
```

IdOnly is a simple [POJO](#) with an id property.

edited May 27 '17 at 21:47



Peter Mortensen

14.4k 19 88 117

answered Apr 19 '12 at 16:00

Luis

1,012 8 4



As an example, create a JSON file, params.json, and add this content to it:

49

```
[
  {
    "environment": "Devel",
    "description": "Machine for test, please do not delete!"
  }
]
```

Then you run this command:

```
curl -v -H "Content-Type: application/json" -X POST --data @params.json -u
your_username:your_password http://localhost:8080/env/add_server
```

edited Jan 22 '18 at 16:05



[Engineer Dollery](#)

10.9k 3 44 71

answered Jan 5 '16 at 23:10



[Eduardo Cerqueira](#)

636 5 5

This worked well for me.

34

```
curl -X POST --data @json_out.txt http://localhost:8080/
```

Where,

-x Means the http verb.

--data Means the data you want to send.

answered Jul 1 '15 at 14:31



[felipealves.gnu](#)

658 8 20

4 The -X POST is redundant in this example – [Engineer Dollery](#) Jan 22 '18 at 16:05

I just run into the same problem. I could solve it by specifying

31 `-H "Content-Type: application/json; charset=UTF-8"`

answered Nov 15 '11 at 15:45



[Steffen Roller](#)

2,936 17 36

Using CURL Windows, try this:

26 `curl -X POST -H "Content-Type:application/json" -d "{\"firstName\": \"blablabla\", \"lastName\": \"dummy\", \"id\": \"123456\"}" http-host/_ah/api/employeeendpoint/v1/employee`

edited Mar 12 '15 at 11:37



[NorthCat](#)

6,461 15 34 43

answered Dec 20 '14 at 23:45



[Márcio Brener](#)

360 3 5

You can use [Postman](#) with its intuitive GUI to assemble your `cURL` command.

23

1. Install and Start **Postman**
2. Type in your URL, Post Body, Request Headers etc. pp.
3. Click on `Code`
4. Select `cURL` from the drop-down list
5. copy & paste your `cURL` command

Note: There are several options for automated request generation in the drop-down list, which is why I thought my post was necessary in the first place.

answered Sep 25 '17 at 12:47

[kiltek](#)



2 Didn't realize that feature was included in Postman. Thanks for pointing it out! – [ariestav](#) Mar 5 at 16:31



17



If you're testing a lot of JSON send/responses against a RESTful interface, you may want to check out the [Postman](#) plug-in for Chrome (which allows you to manually define web service tests) and its Node.js-based [Newman](#) command-line companion (which allows you to automate tests against "collections" of Postman tests.) Both free and open!

edited Feb 14 '17 at 21:40

[törzsmókus](#)

1,009 10 20

answered May 23 '14 at 13:34

[ftexperts](#)

560 5 8



16



[HTTPIe](#) is a recommended alternative to `curl` because you can do just

```
$ http POST http://example.com/some/endpoint name=value name1=value1
```

It speaks JSON by default and will handle both setting the necessary header for you as well encoding data as valid JSON. There is also:

```
Some-Header:value
```

for headers, and

```
name==value
```

for query string parameters. If you have a large chunk of data, you can also read it from a file have it be JSON encoded:

```
field=@file.txt
```

edited Aug 23 '18 at 14:13

[Mark Stosberg](#)

9,318 5 29 45

answered Nov 8 '17 at 12:45

[tosh](#)

625 8 15

This worked well for me, additionally using BASIC authentication:

14

```
curl -v --proxy '' --basic -u Administrator:password -X POST -H "Content-Type:
application/json"
--data-binary '{"value":"30","type":"Tip 3","targetModule":"Target
3","configurationGroup":null,"name":"Configuration Deneme
3","description":null,"identity":"Configuration Deneme
3","version":0,"systemId":3,"active":true}'
http://httpbin.org/post
```

Of course, you should never use BASIC authentication without SSL and a checked certificate.

I ran into this again today, using Cygwin's cURL 7.49.1 for Windows... And when using `--data` or `--data-binary` with a JSON argument, cURL got confused and would interpret the `{}` in the JSON as a URL template. Adding a `-g` argument to turn off cURL globbing fixed that.

See also [Passing a URL with brackets to curl](#).

edited May 27 '17 at 21:49



Peter Mortensen

14.4k 19 88 117

answered Jun 5 '13 at 23:24



davenpcj

7,196 5 31 37

You could also put your JSON content in a file and pass it to curl using the `--file-upload` option via standard input, like this:

14

```
echo 'my.awesome.json.function({"do" : "whatever"})' | curl -X POST "http://url" -T -
```

edited Mar 23 at 11:41



Peter Mortensen

14.4k 19 88 117

answered Jun 13 '17 at 19:41



iloveretards

1,243 3 17 45

This worked for me:

13

```
curl -H "Content-Type: application/json" -X POST -d @./my_json_body.txt
http://192.168.1.1/json
```

edited Oct 16 '16 at 16:39



Misa Lazovic

2,686 9 27 37

answered Oct 16 '16 at 16:07



Amit Vujic

688 1 13 23

I am using the below format to test with a web server.

7 `use -F 'json data'`

Let's assume this JSON dict format:

```
{
  'comment': {
    'who': 'some_one',
    'desc': 'get it'
  }
}
```

Full example

```
curl -XPOST your_address/api -F comment='{ "who": "some_one", "desc": "get it" }'
```

edited May 27 '17 at 21:50



Peter Mortensen

14.4k 19 88 117

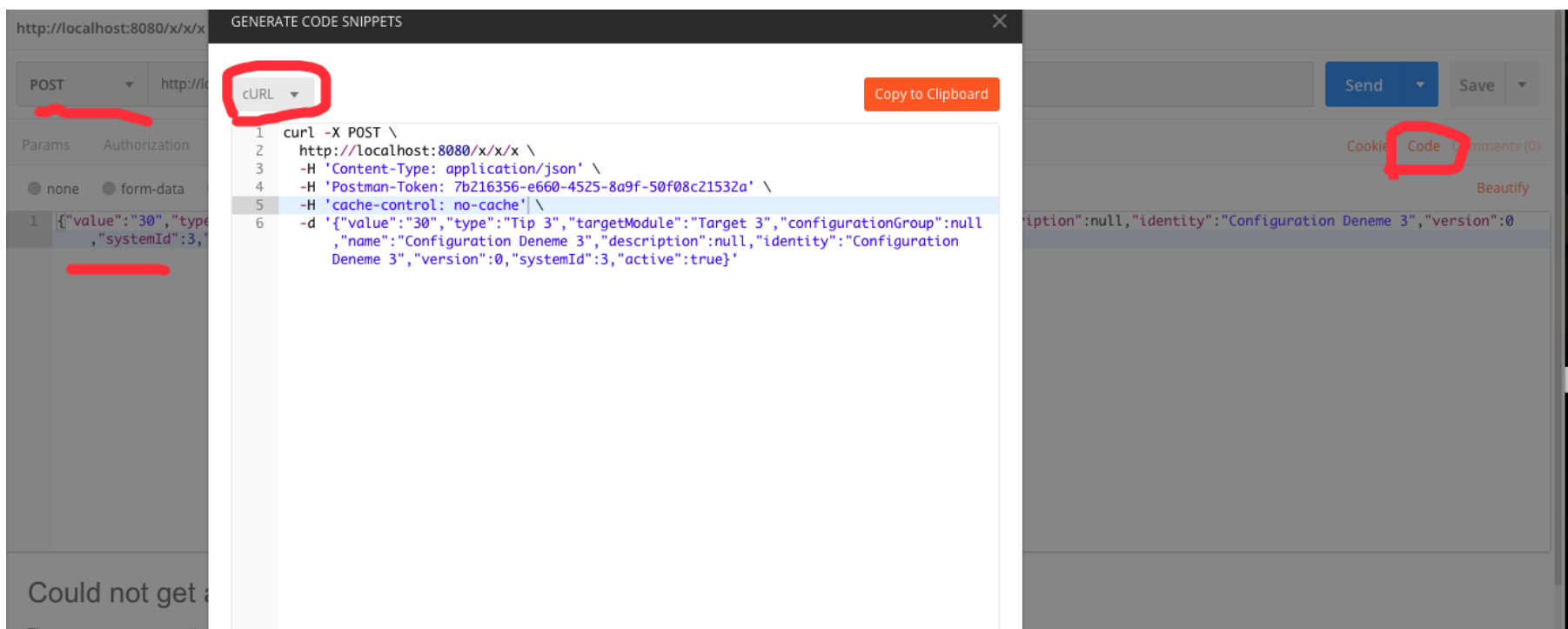
answered Jan 10 '14 at 6:49

user3180641

71 1 1

You can use postman to convert to CURL

3



answered May 30 at 6:43



forever LA

103 4

Thanks a lot @forever-LA it was a life saver tip for me – [vibs2006](#) Jun 17 at 9:37

Use -d option to add payload

```
1 curl -X POST \  
http://<host>:<port>/<path> \  
-H 'Accept: application/json' \  
-H 'Content-Type: application/json' \  
-d '{  
  "foo": "bar",  
  "lorem": "ipsum"  
'
```

In addition:

use -X POST to use POST method

use -H 'Accept: application/json' to add accept type header

use -H 'Content-Type: application/json' to add content type header

answered Apr 13 at 20:06



[Sma Ma](#)

337 1 8

This worked **for** me **for** windows10

```
1 curl -d '{"owner":"","sasdasdasdasd"}' -H "Content-Type: application/json" -X PUT  
http://localhost:8080/api/changeowner/CAR4
```

answered Jul 9 at 9:58



[sudhanshu srivastava](#)

56 5

Here is another way to do it, if you have dynamic data to be included.

```
0 #!/bin/bash
```

```
version=$1  
text=$2
```

```
branch=$(git rev-parse --abbrev-ref HEAD)
repo_full_name=$(git config --get remote.origin.url | sed
's/.*/:\/github.com\/;s/.git$//')
token=$(git config --global github.token)

generate_post_data()
{
  cat <<EOF
{
  "tag_name": "$version",
  "target_commitish": "$branch",
  "name": "$version",
  "body": "$text",
  "draft": false,
  "prerelease": false
}
EOF
}

echo "Create release $version for repo: $repo_full_name branch: $branch"
curl --data "$(generate_post_data)"
"https://api.github.com/repos/$repo_full_name/releases?access_token=$token"
```

answered Aug 6 at 6:02

**Anand Rockzz****2,596** 2 33 45

Please check this [tool](#). It helps you to easily create curl snippets.

-1

```
curl -XGET -H "Accept: application/json" -d '{"value":"30","type":"Tip
3","targetModule":"Target 3","configurationGroup":null,"name":"Configuration
Deneme 3","description":null,"identity":"Configuration Deneme
3","version":0,"systemId":3,"active":true}' "http://localhost:8080/xx/xxx/xxxx"
```

edited May 27 at 12:35

answered May 27 at 11:40

**Pranay Kumar****854** 6 12

You can pass the extension of the format you want as the end of the url. like <http://localhost:8080/xx/xxx/xxxx.json>



or

<http://localhost:8080/xx/xxx/xxxx.xml>

Note: you need to add jackson and JAXB maven dependencies in your pom.

answered Jul 16 '18 at 9:57

[Saurabh Oza](#)

257 3 15

3 That would only work if the server-side accepted the URLs with those extensions. – [Mark Stosberg](#) Aug 23 '18 at 14:01

1 Right, this will work with server side. – [Saurabh Oza](#) Oct 12 '18 at 7:59

protected by [Starx](#) May 13 '15 at 10:49

Thank you for your interest in this question. Because it has attracted low-quality or spam answers that had to be removed, posting an answer now requires 10 [reputation](#) on this site (the [association bonus does not count](#)).

Would you like to answer one of these [unanswered questions](#) instead?

