

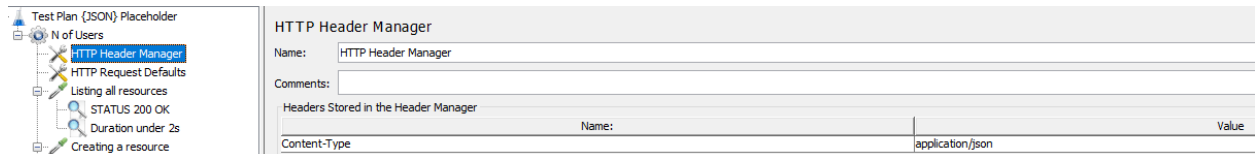
Load testing basics with Apache Jmeter

Assignment

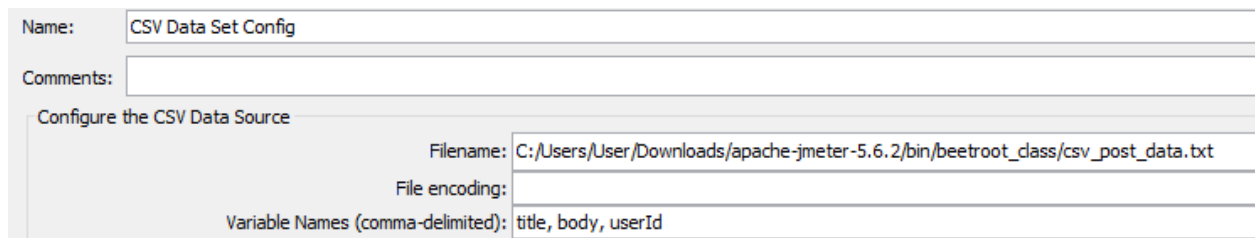
Stefania Pruteanu

Environment preparation and test data:

- Reading the API's documentation to plan each request with according method and params/body contents.
- Checking with JMeter's 16.Best practices under their user's manual
- Setting up HTTP Req defaults and Header manager: For basic url and content-type for JSON body



- Setting up a CSV Data config to simulate for each virtual user distinct requests with variables used in body, according to API's documentation

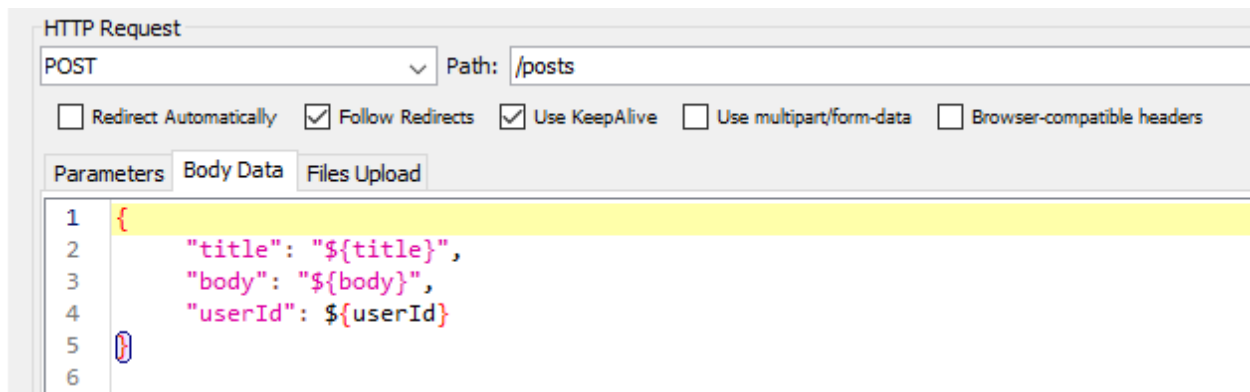


** I generated and exported my CSV file using GoogleSheets for 100 lines of variables for the load tests used later for 100 virtual users:

The screenshot shows a Google Sheet titled 'random json data for csv JMeter'. The sheet has a menu bar with 'File', 'Edit', 'View', 'Insert', 'Format', 'Data', 'Tools', 'Extensions', and 'Help'. Below the menu bar is a toolbar with various icons. The sheet contains a table with 5 columns labeled A, B, C, D, and E. The rows are numbered 98, 99, and 100. The data in the table is as follows:

	A	B	C	D	E
98	198	foo2 updated	bar2 updated	198	
99	199	foo1 updated	bar1 updated	199	
100	200	foo2 updated	bar2 updated	200	

- Used the imported variables for the required HTTP Requests:



HTTP Request

POST Path: /posts

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

```
1 {
2   "title": "${title}",
3   "body": "${body}",
4   "userId": ${userId}
5 }
6
```

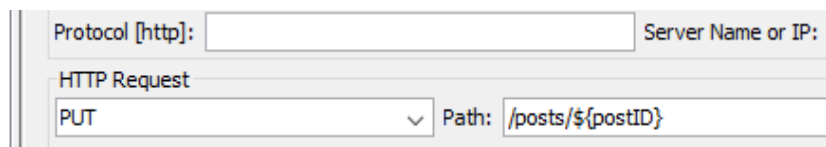
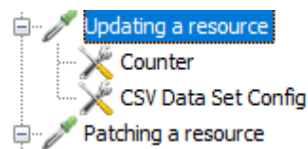
- Set up a Counter for the PUT, PATCH and DELETE methods, to simulate each virtual user a different resource id as required in the API's documentation - endpoint posts/x where x = postID :

— then used the exported variable postID as x



Starting value	1
Increment	1
Maximum value	100
Number format	
Exported Variable Name	postID

☐ Track counter independently for each user



Protocol [http]: Server Name or IP:

HTTP Request

PUT Path: /posts/\${postID}

**I defined x in range (1-100) Using the BVA technique ;)

I tested the API's overall functionality in advance to skip failed assertions and create a logic for the requests and determined that it has no logic so no need to add assertions for valid data display and processing on all reqs 😄 and It is mentioned on their documentation as well but I wanted to train my skills anyways..