**Fileformat for Batch Upload** :

emaild:userid:operation:uuid

a0;b0;1;d0

a1;b1;1;d1

a2;b2;1;d2

a3;b3;1;d3

a4;b4;1;d4

a5;b5;1;d5

a6;b6;1;d6

a7;b7;1;d7

a8;b8;1;d8

a9;b9;1;d9

a10;b10;1;d10

Operation implies 1 for userId insertion and 2 for userId Deletion.

In the file dummy strings are used and code can be improved using regex filters for emailId for data consistency.

**Executor Pool Operation -**

Data is read from files line by line and each line is passed to Worker Thread which implements Callable Interface.

Threads are initialised when Executor pool is started.

Each Thread adds UserId - Email combination to Level DB batch whose maximum capacity of batch upload is set to 10000.

This parameter can be tuned to check the point at which throttling starts to maximise throughput.

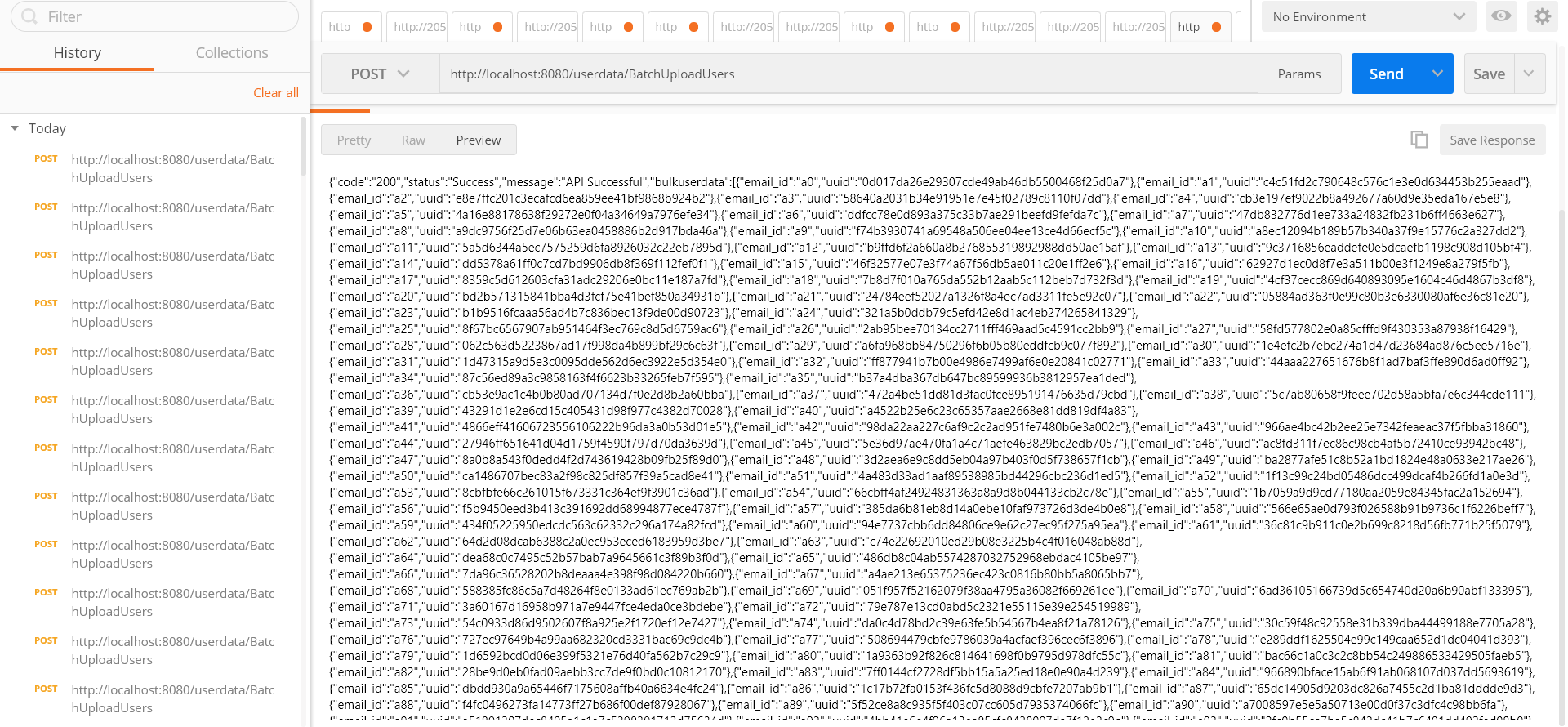
When batch is filled, insertions are performed in level after that batch is closed.

After 10000 insertions, new batch is started and this process continues till userIds List is empty and no more insertions are needed.

**Batch Upload POST Request output**

It gives a List of Key Value pairs – EmailId, UUID, which can then further be used for Querying User data from Level DB.

<http://localhost:8080/userdata/BatchUploadUsers>



Sample Log File generated for Batch Upload Operation -

Processed Batch of Size 10000

Processed Batch of Size 10000

Processed Batch of Size :998

Batch upload file path can be specified in configuration file - config.properties