

Fetch CSV data for data driven testing

vutil provides a REST API to fetch CSV files which is stored on your local machine. This API accepts the filePath of the CSV file as input and converts it into JSON format and respond this JSON as API response.

Now, with the help of this API, we can perform data driven testing in vREST, where external data is stored in your local system in CSV format. vutil will always fetches the latest CSV data. So, if your test data changes then vREST will always fetches the latest data. So, just change your external data, vREST will start reflecting on new data.

This API fetches the CSV file from machine on which this utility is installed.

API Endpoint:

The API format is:

```
POST {{vutilBaseURL}}/convert/csv/json
```

Here {{vutilBaseURL}} is the base URL of the vutil server.

and this API accepts the following body parameters:

1. filePath (String)
 - a. The absolute file path of the CSV file on your local system.
2. options (Object)
 - a. [Optional] The various options useful in setting the desired JSON output from the CSV file. The various options are:
 - i. **recordType**: Value can be array or object. If value is 'object' then output will be Array of JSON objects otherwise the output will be Array of array. Default value is array.
 - ii. **delimiter**: delimiter used for separating columns. Default value is ",".
 - iii. **quote**: If a column contains delimiter, it is able to use quote character to surround the column content. e.g. "hello, world" wont be split into two columns while parsing. Set to "off" will ignore all quotes. Default value is " (double quote)
 - iv. **trim**: Indicate if parser trim off spaces surrounding column content. e.g. " content " will be trimmed to "content". Default value is true.
 - v. **ignoreEmpty**: Ignore the empty value in CSV columns. If a column value is not given, set this to true to skip them. Default value is false.
 - vi. **noheader**: Indicating csv data has no header row and first row is data row. Default value is false.
 - vii. **headers**: An array to specify the headers of CSV data. If --noheader is false, this value will override CSV header row. Default: null. Example: ["my field","name"]. See [header row](#)
 - viii. **flatKeys**: Don't interpret dots (.) and square brackets in header fields as nested object or array identifiers at all (treat them like regular characters for JSON field identifiers). Default: false.
 - ix. **escape**: escape character used in quoted column. Default is double quote (") according to RFC4108. Change to back slash (\) or other chars for your own case.
 - x. **includeColumns**: This parameter instructs the parser to include only those columns as specified by an array of column indexes. Example: [0,2,3] will parse and include only columns 0, 2, and 3 in the JSON output.
 - xi. **ignoreColumns**: This parameter instructs the parser to ignore columns as specified by an array of column indexes. Example: [1,3,5] will ignore columns 1, 3, and 5 and will not return them in the JSON output.

How to use this API:

Assuming that vutil is running on port 4080 and vutil base URL is "<http://localhost:4080>".

Suppose we have a CSV file say sample.csv store in directory path {{dataDir}}/csvs/sample.csv. The content of this sample CSV file is as follows:

sample.csv

```
request.body.name,request.body.email,request.body.designation,request.body.organization,request.body.country,
request.body.aboutMe,request.body.twitterId,request.body.facebookId,request.body.githubId,request.body.
createdOn,response.expectedBody,response.statusCode
,john.doe@example.com,Chief Technical Officer,example.com,India,Please write something about me... ,fake.john.doe,
fake.john.doe,fake.john.doe,2014-05-03T06:28:45.479Z,{"errors":{"name":"required field"}},400
Sample Name whose length is greater than the limit,john.doe@example.com,Chief Technical Officer,example.com,
India,Please write something about me...,fake.john.doe,fake.john.doe,fake.john.doe,2014-05-03T06:28:45.479Z,"
{"errors":{"name":"field length cannot be greater than 35"}},400
John Doe,john.doe@example.com,Sample designation whose length is greater than the limit,example.com,India,
Please write something about me...,fake.john.doe,fake.john.doe,fake.john.doe,2014-05-03T06:28:45.479Z,"
{"errors":{"designation":"field length cannot be greater than 35"}},400
John Doe,john.doe@example.com,Chief Technical Officer,Sample organization whose length is greater than the
limit,India,Please write something about me...,fake.john.doe,fake.john.doe,fake.john.doe,2014-05-03T06:28:45.479
Z,{"errors":{"organization":"field length cannot be greater than 35"}},400
John Doe,john.doe@example.com,Chief Technical Officer,example.com,Sample country name whose length is greater
than the limit,Please write something about me...,fake.john.doe,fake.john.doe,fake.john.doe,2014-05-03T06:28:
45.479Z,{"errors":{"country":"field length cannot be greater than 35"}},400
John Doe,john.doe@example.com,Chief Technical Officer,example.com,India,[Updated] Please write something about
me...,fake.john.doe,fake.john.doe,fake.john.doe,2014-05-03T06:28:45.479Z,{"name":"John Doe","email":
"john.doe@example.com","designation":"Chief Technical Officer","organization":"example.com","
country":"India","aboutMe":"[Updated] Please write something about me...","twitterId":"fake.john.
doe","facebookId":"fake.john.doe","githubId":"fake.john.doe","createdOn":"{*}"},200
```

Now, we will try to fetch this CSV data into JSON format by providing different options in our convert API as shown in below snapshot.

The screenshot shows a REST client interface with the following details:

- URL:** `{{vutilBaseUrl}}/convert/csv/json`
- Description:** Empty
- Method:** POST
- Condition:** Empty
- External Id:** Empty
- Loop source:** Empty
- Versions:** Empty
- Authorization:** Empty
- Tags:** Empty
- Wait for seconds:** Empty

Now, we will try to cover important options for this convert API and try to see the impact of them in the output response.

1. filePath without any options

If we set the filePath without any options

The screenshot shows a REST client interface with the following details:

- URL:** `{{vutilBaseUrl}}/convert/csv/json`
- Description:** Empty
- Method:** POST
- Condition:** Empty
- External Id:** Empty
- Loop source:** Empty
- Versions:** Empty
- Authorization:** Empty
- Tags:** Empty
- Wait for seconds:** Empty
- Parameters:** JSON, Enabled
- Raw Body:**

```
1- {
2  "filePath": "{{dataDir}}/csvs/sample.csv"
3 }
```

then the response of the API will be an Array of array:

API Output

```
{
  "output": [
    [
      " ",
      "john.doe@example.com",
      "Chief Technical Officer",
    ]
  ]
}
```

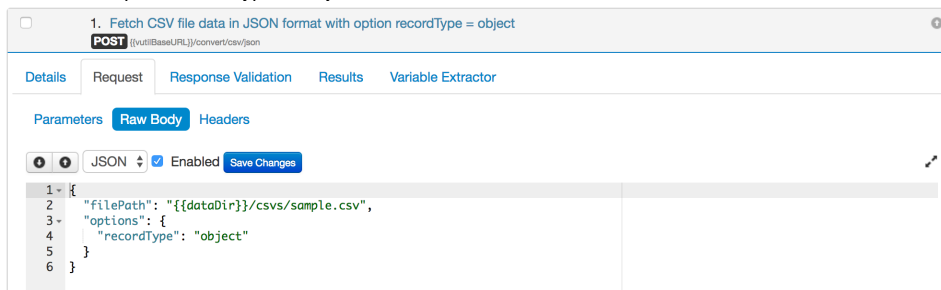
```
"example.com",
"India",
"Please write something about me...",
"fake.john.doe",
"fake.john.doe",
"fake.john.doe",
"2014-05-03T06:28:45.479Z",
{"errors": { "name": "required field"}},
"400"
],
[
"Sample Name whose length is greater than the limit",
"john.doe@example.com",
"Chief Technical Officer",
"example.com",
"India",
"Please write something about me...",
"fake.john.doe",
"fake.john.doe",
"fake.john.doe",
"2014-05-03T06:28:45.479Z",
{"errors": { "name": "field length cannot be greater than 35"}},
"400"
],
[
"John Doe",
"john.doe@example.com",
"Sample designation whose length is greater than the limit",
"example.com",
"India",
"Please write something about me...",
"fake.john.doe",
"fake.john.doe",
"fake.john.doe",
"2014-05-03T06:28:45.479Z",
{"errors": { "designation": "field length cannot be greater than 35"}},
"400"
],
[
"John Doe",
"john.doe@example.com",
"Chief Technical Officer",
"Sample organization whose length is greater than the limit",
"India",
"Please write something about me...",
"fake.john.doe",
"fake.john.doe",
"fake.john.doe",
"2014-05-03T06:28:45.479Z",
{"errors": { "organization": "field length cannot be greater than 35"}},
"400"
],
[
"John Doe",
"john.doe@example.com",
"Chief Technical Officer",
"example.com",
"Sample country name whose length is greater than the limit",
"Please write something about me...",
"fake.john.doe",
"fake.john.doe",
"fake.john.doe",
"2014-05-03T06:28:45.479Z",
{"errors": { "country": "field length cannot be greater than 35"}},
"400"
],
[
"John Doe",
"john.doe@example.com",
"Chief Technical Officer",
"example.com",
```

```

    "India",
    "[Updated] Please write something about me...",
    "fake.john.doe",
    "fake.john.doe",
    "fake.john.doe",
    "2014-05-03T06:28:45.479Z",
    "{ \"name\": \"John Doe\", \"email\": \"john.doe@example.com\", \"designation\": \"Chief Technical
Officer\", \"organization\": \"example.com\", \"country\": \"India\", \"aboutMe\": \"[Updated] Please write
something about me...\", \"twitterId\": \"fake.john.doe\", \"facebookId\": \"fake.john.doe\", \"githubId\": \"
fake.john.doe\", \"createdOn\": \"{{*}}\" }",
    "200"
  ]
}
}

```

2. If we set the option recordType = "object"



then the API output will be Array of JSON objects.

API Output

```

{
  "output": [
    {
      "request": {
        "body": {
          "name": "",
          "email": "john.doe@example.com",
          "designation": "Chief Technical Officer",
          "organization": "example.com",
          "country": "India",
          "aboutMe": "Please write something about me...",
          "twitterId": "fake.john.doe",
          "facebookId": "fake.john.doe",
          "githubId": "fake.john.doe",
          "createdOn": "2014-05-03T06:28:45.479Z"
        }
      },
      "response": {
        "expectedBody": "{ \"errors\": { \"name\": \"required field\" } }",
        "statusCode": "400"
      }
    },
    {
      "request": {
        "body": {
          "name": "Sample Name whose length is greater than the limit",
          "email": "john.doe@example.com",
          "designation": "Chief Technical Officer",
          "organization": "example.com",
          "country": "India",
          "aboutMe": "Please write something about me...",
          "twitterId": "fake.john.doe",
          "facebookId": "fake.john.doe",
          "githubId": "fake.john.doe",
          "createdOn": "2014-05-03T06:28:45.479Z"
        }
      }
    }
  ],
}

```

```

"response": {
  "expectedBody": "{\"errors\": { \"name\": \"field length cannot be greater than 35\"}}",
  "statusCode": "400"
}
},
{
  "request": {
    "body": {
      "name": "John Doe",
      "email": "john.doe@example.com",
      "designation": "Sample designation whose length is greater than the limit",
      "organization": "example.com",
      "country": "India",
      "aboutMe": "Please write something about me...",
      "twitterId": "fake.john.doe",
      "facebookId": "fake.john.doe",
      "githubId": "fake.john.doe",
      "createdOn": "2014-05-03T06:28:45.479Z"
    }
  },
  "response": {
    "expectedBody": "{\"errors\": { \"designation\": \"field length cannot be greater than 35\"}}",
    "statusCode": "400"
  }
},
{
  "request": {
    "body": {
      "name": "John Doe",
      "email": "john.doe@example.com",
      "designation": "Chief Technical Officer",
      "organization": "Sample organization whose length is greater than the limit",
      "country": "India",
      "aboutMe": "Please write something about me...",
      "twitterId": "fake.john.doe",
      "facebookId": "fake.john.doe",
      "githubId": "fake.john.doe",
      "createdOn": "2014-05-03T06:28:45.479Z"
    }
  },
  "response": {
    "expectedBody": "{\"errors\": { \"organization\": \"field length cannot be greater than 35\"}}",
    "statusCode": "400"
  }
},
{
  "request": {
    "body": {
      "name": "John Doe",
      "email": "john.doe@example.com",
      "designation": "Chief Technical Officer",
      "organization": "example.com",
      "country": "Sample country name whose length is greater than the limit",
      "aboutMe": "Please write something about me...",
      "twitterId": "fake.john.doe",
      "facebookId": "fake.john.doe",
      "githubId": "fake.john.doe",
      "createdOn": "2014-05-03T06:28:45.479Z"
    }
  },
  "response": {
    "expectedBody": "{\"errors\": { \"country\": \"field length cannot be greater than 35\"}}",
    "statusCode": "400"
  }
},
{
  "request": {
    "body": {
      "name": "John Doe",
      "email": "john.doe@example.com",
      "designation": "Chief Technical Officer",

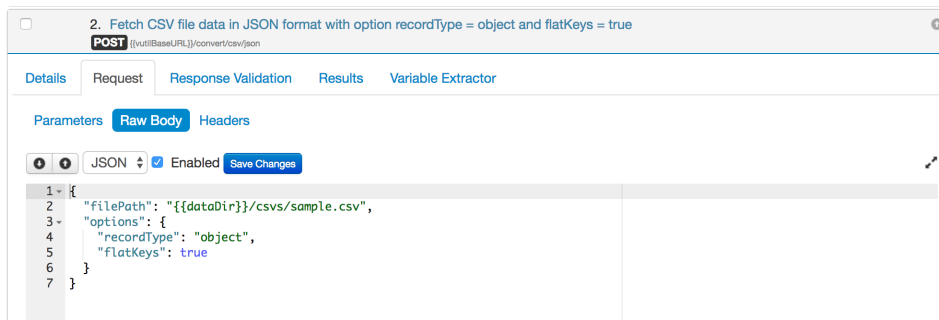
```

```

    "organization": "example.com",
    "country": "India",
    "aboutMe": "[Updated] Please write something about me...",
    "twitterId": "fake.john.doe",
    "facebookId": "fake.john.doe",
    "githubId": "fake.john.doe",
    "createdOn": "2014-05-03T06:28:45.479Z"
  }
},
"response": {
  "expectedBody": "{\"name\": \"John Doe\", \"email\": \"john.doe@example.com\", \"designation\": \"Chief Technical Officer\", \"organization\": \"example.com\", \"country\": \"India\", \"aboutMe\": \"[Updated] Please write something about me...\", \"twitterId\": \"fake.john.doe\", \"facebookId\": \"fake.john.doe\", \"githubId\": \"fake.john.doe\", \"createdOn\": \"{{*}}\"}",
  "statusCode": "200"
}
}
]
}

```

3. If we set the option recordType = "object" and flatKeys = true



then the API output will be Array of JSON objects with flat keys.

API Output

```

{
  "output": [
    {
      "request.body.name": "",
      "request.body.email": "john.doe@example.com",
      "request.body.designation": "Chief Technical Officer",
      "request.body.organization": "example.com",
      "request.body.country": "India",
      "request.body.aboutMe": "Please write something about me...",
      "request.body.twitterId": "fake.john.doe",
      "request.body.facebookId": "fake.john.doe",
      "request.body.githubId": "fake.john.doe",
      "request.body.createdOn": "2014-05-03T06:28:45.479Z",
      "response.expectedBody": "{\"errors\": { \"name\": \"required field\"}}",
      "response.statusCode": "400"
    },
    {
      "request.body.name": "Sample Name whose length is greater than the limit",
      "request.body.email": "john.doe@example.com",
      "request.body.designation": "Chief Technical Officer",
      "request.body.organization": "example.com",
      "request.body.country": "India",
      "request.body.aboutMe": "Please write something about me...",
      "request.body.twitterId": "fake.john.doe",
      "request.body.facebookId": "fake.john.doe",
      "request.body.githubId": "fake.john.doe",
      "request.body.createdOn": "2014-05-03T06:28:45.479Z",
      "response.expectedBody": "{\"errors\": { \"name\": \"field length cannot be greater than 35\"}}",
      "response.statusCode": "400"
    }
  ]
}

```

```

    {
      "request.body.name": "John Doe",
      "request.body.email": "john.doe@example.com",
      "request.body.designation": "Sample designation whose length is greater than the limit",
      "request.body.organization": "example.com",
      "request.body.country": "India",
      "request.body.aboutMe": "Please write something about me...",
      "request.body.twitterId": "fake.john.doe",
      "request.body.facebookId": "fake.john.doe",
      "request.body.githubId": "fake.john.doe",
      "request.body.createdOn": "2014-05-03T06:28:45.479Z",
      "response.expectedBody": "{ \"errors\": { \"designation\": \"field length cannot be greater than
35\" } }",
      "response.statusCode": "400"
    },
    {
      "request.body.name": "John Doe",
      "request.body.email": "john.doe@example.com",
      "request.body.designation": "Chief Technical Officer",
      "request.body.organization": "Sample organization whose length is greater than the limit",
      "request.body.country": "India",
      "request.body.aboutMe": "Please write something about me...",
      "request.body.twitterId": "fake.john.doe",
      "request.body.facebookId": "fake.john.doe",
      "request.body.githubId": "fake.john.doe",
      "request.body.createdOn": "2014-05-03T06:28:45.479Z",
      "response.expectedBody": "{ \"errors\": { \"organization\": \"field length cannot be greater than
35\" } }",
      "response.statusCode": "400"
    },
    {
      "request.body.name": "John Doe",
      "request.body.email": "john.doe@example.com",
      "request.body.designation": "Chief Technical Officer",
      "request.body.organization": "example.com",
      "request.body.country": "Sample country name whose length is greater than the limit",
      "request.body.aboutMe": "Please write something about me...",
      "request.body.twitterId": "fake.john.doe",
      "request.body.facebookId": "fake.john.doe",
      "request.body.githubId": "fake.john.doe",
      "request.body.createdOn": "2014-05-03T06:28:45.479Z",
      "response.expectedBody": "{ \"errors\": { \"country\": \"field length cannot be greater than
35\" } }",
      "response.statusCode": "400"
    },
    {
      "request.body.name": "John Doe",
      "request.body.email": "john.doe@example.com",
      "request.body.designation": "Chief Technical Officer",
      "request.body.organization": "example.com",
      "request.body.country": "India",
      "request.body.aboutMe": "[Updated] Please write something about me...",
      "request.body.twitterId": "fake.john.doe",
      "request.body.facebookId": "fake.john.doe",
      "request.body.githubId": "fake.john.doe",
      "request.body.createdOn": "2014-05-03T06:28:45.479Z",
      "response.expectedBody": "{ \"name\": \"John Doe\", \"email\": \"john.doe@example.com\", \"
designation\": \"Chief Technical Officer\", \"organization\": \"example.com\", \"country\": \"India\", \"
aboutMe\": \"[Updated] Please write something about me...\", \"twitterId\": \"fake.john.doe\", \"
facebookId\": \"fake.john.doe\", \"githubId\": \"fake.john.doe\", \"createdOn\": \"{{*}}\",
      "response.statusCode": "200"
    }
  ]
}

```

That's it for the examples. You may explore other CSV conversion options as per your need.