

# Read System file via API

vutil provides a REST API to read system file from the machine on which the utility is installed. If your test data is directly in JSON format, then you may read JSON file with the help of an API. Then you may consume the response in subsequent APIs as test data.

This API fetches text data from the machine on which this utility is installed.

## API Endpoint:

The API format is

```
GET {{{vutilBaseURL}}}/file
```

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

and this API accepts a single query parameter "filePath" and its value is the absolute path of the file which you want to fetch from system.

## 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 sample JSON file sample.json stored in directory path {{{dataDir}}}/jsons/sample.json.

Now we will try to fetch this JSON file by using the following API.

The screenshot shows an API client interface for a request named "1. Read System File". The request method is GET and the URL is {{{vutilBaseURL}}}/file. The interface includes tabs for Details, Request, Response Validation, Results, and Variable Extractor. The Request tab is active, showing a list of parameters with values set to "Empty".

| Parameter        | Value                   |
|------------------|-------------------------|
| URL              | {{{vutilBaseURL}}}/file |
| Description      | Empty                   |
| Method           | GET                     |
| External Id      | Empty                   |
| Tags             | Empty                   |
| Condition        | Empty                   |
| Loop source      | Empty                   |
| Authorization    | Empty                   |
| Wait for seconds | Empty                   |

Now set the request query parameter "filePath" as follows:

The screenshot shows the same API client interface, but now the "Parameters" tab is active. A table lists the query parameters, with "filePath" set to the value {{{dataDir}}}/jsons/sample.json.

| Name     | Description | Field Type | Parameter Type | Value                           |
|----------|-------------|------------|----------------|---------------------------------|
| filePath |             | string     | query          | {{{dataDir}}}/jsons/sample.json |

And the API will produce the file content as output:

The screenshot shows the "Results" tab of the API client, displaying the JSON output of the API call. The output is a JSON object with a "request" property containing a "body" property with user information.

```
[
  {
    "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"
  }
}
]

```

That's it.