About vREST

vREST is an automated REST API Testing tool which provides the following features:

1. **Regression Testing of your web / mobile application**
    a. vREST provides functions to validate your REST APIs. This is the core function of vREST. Writing API test cases will ensure you to avoid any bugs after handing over the application to the client. You will be more confident on the build quality of your web application.
    b. It will also help you to release frequently and shorten the time to market.
    c. For more information, refer writing your first test case.
2. **Record and replay of test cases**
    a. No need to write your test cases manually.
    b. vREST provides HTTP Recorder to record your web application test cases.
    c. You just need to turn on the recorder and do the manual testing of your web application for the last time.
    d. vREST will record test cases for you behind the scenes.
    e. For more information, please read HTTP recording via Chrome Extension.
3. **Speed up your development by reducing dependencies among teams**
    a. vREST provides functions to define API mocks. By defining API mocks, both teams (Frontend and Backend) can now work in parallel by using them. Frontend team don't have to wait for backend teams to implement REST APIs. They can just use API mocks to start their development.
    b. vREST provides functions to export all of your test cases to API mocks and vice versa with the help of a click.
    c. For more information, refer Mock Server.
4. **Powerful Response Validation**
    a. vREST provides built-in response validators and assertions to validate your API response.
    b. Response validators are very powerful to validate the dynamic API responses in a very easy way.
    c. For more information, please read response validation.
5. **Works for test applications deployed on localhost / intranet / internet**
    a. vREST executes all the test cases on client machine with the help of chrome extension or vrunner.
    b. So, it doesn't need to create any tunnel to validate test applications hosted on intranet / localhost.
6. **Integrate with third party tools like JIRA, Slack etc.**
    a. vREST provides hooks functionality, by which you may integrate vREST with any third party tool which support REST API.
    b. vREST also provides built-in templates to integrate with JIRA, Slack etc.
    c. For more information, please read third party tools integrations via hooks.
7. **Integrate with continuous integration tools like Jenkins, TeamCity etc.**
    a. vREST provides modules like vrunner to integrate with any continuous integration servers like Jenkins, TeamCity etc.
    b. For more information, please read Integration with Jenkins Server or Integration with TeamCity Server.
8. **Imports from Swagger, Postman ...**
    a. vREST provides functionality to import test cases from third party tools like Swagger, Postman etc.
    b. It will reduce your effort of initial heavy lifting to create test cases.
9. **Perform Data driven testing**
    a. With the help of loop functionality, user may perform data driven testing in vREST.
    b. vREST provides external utility vutil to fetch data stored in CSV/JSON files for data driven testing.
    c. For more information, please read data driven testing.
10. **DB Validation after executing API test cases.**
    a. vREST provides external utility vutil to perform database validation after executing the test cases.
    b. vutil supports a number of database vendors like MySQL, Postgres, Oracle, MSSQL, MongoDB etc. for database validation.
    c. For more information, please read Database Validation.
11. **Ability to execute / validate multipart requests efficiently.**
    a. vREST provides external utility vutil to execute / validate multipart requests / responses.
    b. You may perform separate validations on individual part of a multipart response.
    c. For more information, please read Executing multipart requests / Process multipart or complex responses.
12. **Ability to execute same test cases on multiple environments like dev, prod, staging etc.**
    a. vREST provides environments functionality to configure initial / global variables for multiple environments like dev, prod, staging etc.
    b. With the help of environments, you may switch to different environment very easily.
    c. The same test cases can be executed on multiple environments with different configurations.
    d. For more information, please read Global Variables / Environments.
13. **Ability to execute test cases on command line via vrunner module.**
    a. vREST provides an external utility vrunner, through which you may use to execute vREST test cases on command line.
    b. For more information, please read vrunner utility.
14. **Scheduling of test cases via external cron utility with the help of vrunner module.**
    a. With the help of vrunner module, you may use the vrunner command in any third party schedular like cron.
    b. vrunner command is very easy to use and can be integrated with any schedular.
    c. For more information, please read Scheduling vREST test cases.

We are well known for our customer support. So, if you face any issues while using vREST, feel free to ping us on live chat (if available) or drop your query by sending an email to support@vrest.io.