The Report of API Test Tool

|  |
| --- |
| Lan Yiyuan(55614744), Wang Haipeng(55762913), Sun Jingyao(55880040), Zou Shuang(55911609),  Li Yixin(55975771), Zhao Dongyu(55980768), Du Kejun(56075096) |
| *Computing Sciences Department, City University of Hong Kong, Hong Kong, China* |

***Abstract*—The API Test Tool implements most of functions on the basis of Postman that is a popular test tool. The API Test Tool provides HTTP and Web API request debugging, which can send any type of HTTP request with parameters and headers. Moreover, it also provides the import and export of test data and environment configuration data. In the API Test Tool, there are three main parts, which are History, Collection and request builder. Any HTTP request can be created quickly and saved in History to be executed again. Through Collection function, we can well classify and manage the API provided by the test software. In the part of testing request, there are many different types of request, such as get, post, put and delete. API Test Tool has a variety of methods to define variables, which are used in different domain. No matter what kind of variable, it uses {{variable}} to represent variables, which is useful in developing and testing.**

Keywords—API Test Tool; HTTP request; Web API

# I. Introduction

## A. Background

With the increasing importance of API interface testing, the traditional manual API interface testing method is inefficient, and the quality is not guaranteed. In practice, some bugs cannot be found on web page’s operation, but interface testing can find the bugs [1]. In addition, it also can check the system's ability to handle exception and the safety and stability of the system. Interface is more stable than front end page and can test the concurrency which allows two events processed at the same time. In addition, the request parameters of interface testing can be modified to break through the front-end page input limit.

When developing or debugging web programs or web B/S mode programs, users need some methods to track web page requests. There is some web monitoring tools such as firebug, which is a web page debug tool. But in order to meet users’ further requirements, a more powerful HTTP request simulation tool Postman is selected. In this project, we also design the tool based on Postman.

## B. Objective

The API Test Tool is aimed to be applied the interactive development of multiple systems and application systems with multiple subsystems. It is used to the underlying framework system and central service system that provide services for other systems. It mainly tests the interfaces provided by these systems to the external to verify their correctness and stability. In the process of developing products, the tool acts as the bridge of this interaction mode. It can send all kinds of data simulating user’s HTTP requests to the server, so that developers can make the correct response in time, or deal with the error information before a product is released in advance, so as to ensure the security after the product is launched.

## C. Signification

Interface, also known as API, is a contract for connecting different components of a software system, such as the interface between modules, or the interface between front-end and back-end. From the perspective of system architecture, the interface is often developed by the back-end personnel, which has a wide range of influence, so when there is a problem with the interface, it often affects the progress of research and development. The interface is the boundary of each module [2]. When testing the interface, you can find more important bugs, because borders are the most problematic place, whether it is the boundary of the module or the boundary of the personnel.

The API Test Tool is a convenient web debugging tool with JSON automatic typesetting function and supports recording. It can record the parameters and return values of each post or get method. It also integrates base64 encryption and decryption to provide the security of Web Data. The development of server would be done in Java Programming Language and the html page is implemented by css technology.

II. Related Work

## A. The working principle of Firewalls

When a user develops or debugs network program and B/S mode web page, some methods are needed to track web page requests. The user can use some network monitoring tools such as the famous web debugging tool, Firebug. The most common types of automated software tests including unit testing, integration testing and end-to-end testing.[3]

As an API testing tool, Postman can not only debug simple CSS, HTML, Script and other simple web pages basic information, it can also send almost all types of HTTP requests. Postman is one of the representative products in the Chrome plug-in class in terms of sending web HTTP requests. Postman is available for different operating systems, Mac OS, Windows X32, Windows X64, Linux, and supports chrome browser extensions, and more[4]. However, it does not support searching for use cases by URL or modifying environment variables by passing parameters through the command line. At the same time, you cannot read or write the database or introduce third-party plugins (such as custom JS libraries).

As an improvement, our project uses VUE to build a web platform for the Postman server. Vue.js is a progressive framework for building user interfaces. Its core library only focuses on the user interface and is very easy to integrate with other libraries or existing projects (suitable for our project). On the other hand, VUE is fully capable of driving complex single-page applications developed with single-file components and libraries supported by the VUE ecosystem. In addition, we have made improvements in the history function. The original Postman software kept only the API address in the history record after performing the HTTP request. In our project, the response data after the execution request is also kept in the history record, which is convenient for users to consult.

III. Preliminaries

The project uses the standard scrum for development and management. The implementation of the scrum process in the project is based on an iterative and incremental process framework. The team firstly defines project theme and requirements (which is API Test Tool) and estimates each members' advantages and capabilities. Then implement the function by sprint process, holding daily meetings to deal with new complex problems, difficulties and emergencies.

## Scrum Roles

*1) Scrum Master*

The Scrum Master is responsible for the scrum process, supervising all members to follow scrum rules and ensure that the project is completed within the expected timeframe. Help scrum members achieve the highest level of performance. Through the guidance of the scrum master, our project successfully completed the iteration and incremental of the scrum.

*2) Product owner*

The product owner is responsible for prioritizing the backlog and supervise the team to do the tasks with the highest priority. In this project, the product owner assigns the following priorities to the backlog (the following items are ranked from highest to lowest): design test cases and user stories, drawing product prototypes, function development, and testing.

*3) Team*

Team is responsible for developing software features. Our project team size is 7 people, completing subtasks with self-management, self-organization and cross-functional work.

## Scrum Implementation Process

All work is done during the sprint cycle. Before each sprint starts, scrum master holds a 1-2 hours' sprint plan meeting. The user story for each sprint plan is selected based on priority. At the sprint plan meeting, the entire user story and functional requirements according to the flow chart are showed to the group members, then team members jointly split into subtasks. After splitting the task, all members discuss and evaluate the difficulty and time consuming of each subtask, finally the scrum team members respectively pick the subtask.

Throughout the sprint process, scrum master holds a daily online meeting via WeChat every day. The time-consuming is controlled within 10 minutes, each group members are asked to simply state the tasks that have done, the tasks are in processing, and the difficulties we are currently experiencing.

We use the online tool Leangoo to track tasks. Leangoo is an online project scrum tool which provides an end-to-end project management and team collaboration platform to help achieve efficient and visual collaboration and management. Different from the traditional tabular project management tools, Leangoo supports multi-person collaboration and real-time information synchronization. In addition, a variety of project templates are built in Leangoo, covering different application scenarios such as agile development, phased projects, lightweight collaboration, and problem tracking. All subtasks are uploaded to the Leangoo's task board and labelled different time mark. These subtasks are assigned to the team members by the scrum master. Through the product backlog, burndown chart and task board, we can clearly analyse the current progress of the project and make corresponding plan. The estimated time of the task, the actual execution time, and the time required for completion are recorded in the tracking table. According to the tracking table and burndown chart, the completion of the task can be visually displayed. The task board allows the scrum master to keep track of each subtask and the current project status.

When the project member completes the subtask, it is submitted to the scrum master for checking. If there is no problem, then the subtask card would be moved to the done group. At the end of each sprint, the scrum master holds a sprint review meeting. The main purpose is to summarize the completed tasks, record happened difficulty, and develop a next sprint to-do list.

## Scrum Tools

*1) Product Backlog*

The product backlog lists all the features desired for our projects. The product owner determines the content, priority, and availability of the product backlog. This list is constantly changing as the project progresses.

*2) Sprint Backlog*

The sprint backlog is a list of tasks to complete during the sprint. The team design a preliminary task list in the sprint planning meeting. In this process, our team uses the way of discussion to estimate the duration of the task. In the event of a dispute, the scrum master makes the final decision. Only subtask responsible personnel have the right to change the sprint backlog.

TABLE I An Example of Sprint Backlog

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tasks | Build the server | Draw the UML flow chart | Draw the Timing diagram | Draw the Use case diagram | Draw the State transition diagram | Paper Writing | Video |
| Members | Wang/Zhao | Li/Du | Sun | Zou | Lan | Sun/Li/Du/Zou/Lan | Team |
| Status of progress | In Process | Done | Done | Done | Done | In Process | To do |
| Day1 | 80 | 12 | 12 | 12 | 12 | 60 | 30 |
| Day2 | 80 | 12 | 6 | 12 | 12 | 60 | 30 |
| Day3 | 80 | 6 | 6 | 0 | 6 | 50 | 30 |
| Day4 | 60 | 0 | 6 | 0 | 6 | 50 | 30 |
| Day5 | 60 | 0 | 0 | 0 | 6 | 50 | 30 |
| Day6 | 60 | 0 | 0 | 0 | 6 | 40 | 30 |
| Day7 | 40 | 0 | 0 | 0 | 0 | 40 | 30 |

The table I above is an example of a sprint backlog in the API Test Tool project. The horizontal axis represents the pending task of the sprint, and the vertical axis represents the period of the sprint (15 days). After the tasks are determined, the relevant responsible person can mark the estimated time of completing the task on the backlog. During the sprint meeting, the scrum master encourages every member to choose tasks that they prefer or be good at. For example, the task "Building the server" is assigned to the members Wang and Zhao. This task is in progress and the remaining workload is 40 hours. All the tasks about drawing diagrams are already completed and the remaining workload is zero.

*3) Release burndown chart*

The release burndown chart is a work chart used for indicating the remaining workload and tracking progress on a project. The horizontal axis represents time and the vertical axis represents workload. This chart can be used to visually predict when work will be completed and team efficiency at each stage. Through the release burndown chart, when the slope of the open estimate curve has an upward trend, scrum master needs to focus on this stage, track the member's work efficiency, find the difficulties in technical implementation and communicate. If the open estimate curve tends to be stable, it indicates that the project is progressing favorably.

IV. The Solution of Designing API Test Tool

## Requirements

*API Test Tool* is a web application inspired by *Postman*, which is an API and network test tool. The main function is to send different types of HTTP request with parameters and headers and display all the request response information. The main purpose of this tool is to provide HTTP and Web API request debugging. The detailed functional requirements are as follows.

*1) Test API*

There are four kinds of authorization types, namely *Normal, Basic Auth, Digest Auth* and *OAuth 1.0*. Depending on the type of authorization, the inputs are different. For all the four types, the input URL, interface request method (GET, POST, PUT, PATCH, DELETE, COPY, HEAD, OPTIONS, LINK, UNLINK and PURGE), URL parameters and Headers are essential.

Besides, for the other three types except Normal, the authorization headers are needed and it will be generated and added as a custom header. Specifically, for Basic Auth, the authorization headers are Username and Password; for Digest Auth, the authorization headers are Username, Realm, Password, Nonce, Algorithm, qop, Nonce count, Client nonce and Opaque; and for OAuth 1.0, the authorization headers are Consumer key, Consumer secret, Token, Token secret, Signature Method, Timestamp, Nonce, Version and Realm.

*2) History*

HISTORY is a storage of the request history using the form of “request + URL”. It stores all the request information in *TEST API*, and each entry in it can be deleted individually or completely. The implementation of this function enables the executed requests to be executed again quickly.

*3) Collections*

In COLLECTIONS, new collections can be created, and multiple requests can be categorized and stored in different collections. Also, collections can be import from disk or URL. The existing collections can be edited, deleted and shared as a link or downloaded as a JSON file. The implementation of this function enables the requests to be ordered and well-classified.

*4) Information*

INFORMATION shows the response to the request in TEST API, which includes Body, Cookies, Headers and Status. In Body, the response information can be shown in different content-type, including Pretty (JSON or XML), Raw and Preview. The cookies information and the headers information in response are displayed in Cookies and Headers, respectively. In Status section, there is the response time and the response status in the form of HTTP status code.

*5) Settings*

In SETTINGS section, the shortcuts that can be used and the basic preferences (such as *Max history items, Auto save request, Send no-cache header, Language detection* and so on) can be set and modified.

*6) Testcase*

In summary, the testcase of the above functional requirements is shown in Figure 1.

图片包含 文字, 地图

描述已自动生成

Fig. 1. Testcase

## Design

In accordance with the requirements in 4.1, the prototyping, workflow, use case diagram, state diagram and sequence diagram is shown in detail in this section.

*1) Prototyping*

It is a web application inspired by *Postman*. An example of the prototype is shown in Figure 2., and the detailed prototype is in our Git repository.

图片包含 屏幕截图

描述已自动生成

Fig. 2. Prototyping

*2) Workflow*

The workflow generated from the prototype design is shown in Figure 3.

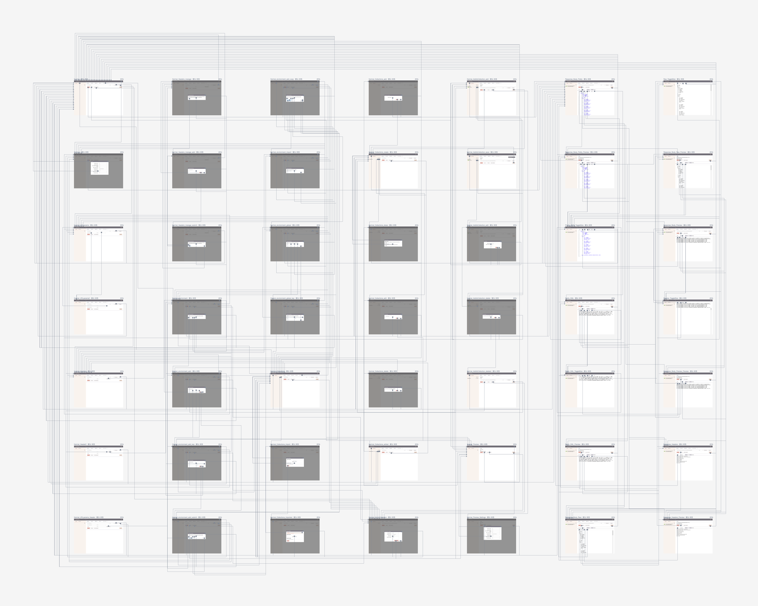


Fig. 3. Workflow

*3) Workflow*

The use case diagram is shown in Figure 4. As the use case diagram shown, there is only one participant, the User, and there is the following use case. Firstly, the user *Input URL* to choose the request method, and input the request URL and other request information (such as authentication information) to finish the whole request information. And then user Send URL to send the request to get the response information. Meanwhile, the user can also *Add Collections* to add the request to a collection. Secondly, user can *Check Collections* to do the operations (such as edit, deletion, import and share) on the existing or new collections. Thirdly, user can *Check History* to view the history list. Lastly, user *Set Settings* to set the shortcuts and basic preferences.

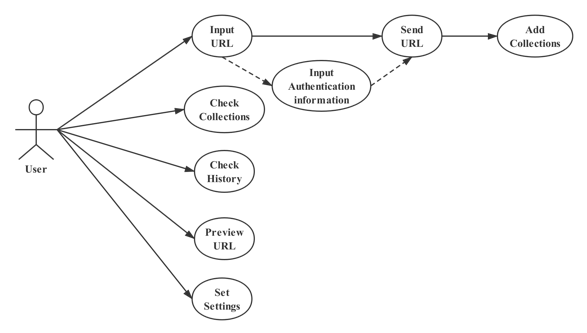


Fig. 4. Use case diagram

*4) State diagram*

The state diagram is shown in Figure 5. As the state diagram shown, the initial state is Input Pattern, and the system waits for user input in this state. After the system obtains the user’s input, it changes to Sending mode, and it waits for response from the server. If the user’s input is a wrong URL, the system *Fail to Return Response* and return to *Input Pattern* state. Otherwise, if the user’s input is correct, the state changes to *Return Response*, and the whole process finishes.

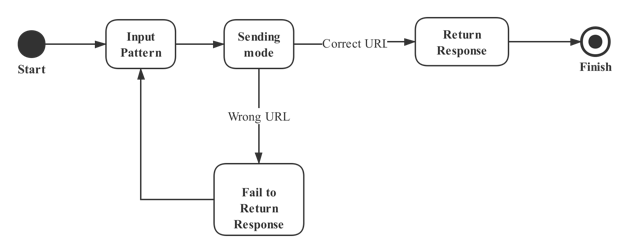


Fig. 5. State diagram

*5) Sequence diagram*

The sequence diagram is shown in Figure 6. As the sequence diagram shown, *APITestTool* sends the request message to the controller Send, and it calls *getMethod()* function to get the optional request method, and calls *getInfo(input, method)* function to get the response information, and the information is added to the history. Besides, *APITestTool* can calls *addToCollection(request)* function to add the request to an existing collection, or calls *createCollection()* to create a new collection. To set the shortcuts and basic preferences, system calls *set()* function, and the controller calls *update()* function to modify the current settings.

图片包含 地图

描述已自动生成

Fig. 6. Sequence diagram

## Implement

1. *Content*

The running environment of API Test Tool is Node server in Linux Centos 7 and the tools be used were as followed:

① npm: Node package management, a tool used to install or uninstall some common packages easily.

② Vue-cli: Vue cli is a complete system based on vue.js for rapid development. There are three components, the first one is CLI providing Vue commands in the terminal. The second one is CLI Services providing serve, build, and inspect commands. The last one is CLI plugin, providing optional functions for Vue projects.

③ Webpack: A tool for packaging, resolving dependencies and optimizing structure of code.

④ Eslint: A static code checking tool that checks whether our JavaScript code meets the specified rules.

⑤ Babel: A ES6 transcoder tool, which is widely used, and can convert ES6 code to ES5 code and execute in existing environment.

The framework of API Test Tool is MVVM, which can be seen in figure 7. The first “M” means Model, which is the part of application program used to process application business logic and data, which is mainly responsible for network request, database processing, I/O and other operations. For common data used in several modules, it is stored in main store. For different submodules, the data is stored in different namespaces stores. Such as “core.normal”, “core.basic” and so forth.

The second “V” means View, which is the part of an application that processes the display of data. In web development, it generally corresponds to HTML and JS files, activity and events. It can not only apply Vue JS syntax and template in our project, but also apply Bootstrap 3 code and style in our project. In addition, some parts applied with ES6 or ES7 code in our project.

The third and forth “VM” means View Model. This part of an application creates an association to bind the model and view. In this way, once the model changes, the View Model will immediately feedback to view to automatically refresh the interface. View Model is only responsible for business logic and does nothing to do with UI. For displaying the data, the code for UI logic implements with UI views. For send the request or resolve the response, the code for business logic implements with Models.

图片包含 屏幕截图

描述已自动生成

Fig. 7. MVVM-Content Basic Architecture

When it comes to the advantages of MVVM, it can be divided into four types. The first one is data driven, in MVVM, data and business logic are in a separate View Model. As long as the View Model focuses on data and business logic, it does not need to deal with UI or control. The data automatically drives the UI to automatically update the UI, and the change of UI automatically feeds back to the data at the same time. The data becomes the dominant factor, so as long as the data is concerned in the business logic processing, it is convenient and much simpler.

The second one is low coupling degree, in MVVM, data is independent of UI. View Model is only responsible for processing and providing data. How the UI wants to process data is up to the UI itself. View Model does not involve anything related to UI and does not hold the reference of UI control.

The third one is reusability, a View Model can be reused in multiple views. The same data is displayed in different UIs. For frequent version iteration UI changes, just change the view layer.

The last advantages is unit testing. The data and business logic are in the View Model, and the UI is the focus of the view. So it is very convenient to do this test, and there is no dependence on each other at all. Both the unit test of the UI and the unit test of the business logic are low coupling.

*2) Server*

The running environment’s server is Linux CentOS 7. The frameworks and tools of API Test Tool server are as followed:

① Springboot: A stand-alone Spring applications using embed Tomcat.

② Mybatis: A first class persistence framework with support for custom SQL, stored procedures and advanced mappings.

③ Mysql: A classic database.

④ Docker: A easiest and fastest tool to use containers(mainly for Mysql5.7 in the project).

There are four layers of the API Test Tool’s server, it can be seen in figure 8. The Pojo is the definition of Java Object (History, Collection, Methods), automatic generated by Mybatis-generator. The Dao execute retrieve or insert operations in the database (base on Object defined in Pojo Layer), automatic generated by Mybatis -generator. The service includes the implement core logic in project and process requests from Controller Layer. The controller responds all kinds of web requests.

图片包含 屏幕截图

描述已自动生成

Fig. 8. Basic Architecture

The History table design, Collection table design and Method table design in database are shown in Table II, Table III and Table IV, respectively.

TABLE II History Table Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **history** | | | |
| **Attribute** | **Type** | **Null/Non-Null** | **constraint condition** | **Description** |
| **id** | int(11) | Non-Null | Primary key | The id of history  auto-increment |
| **method** | varchar(10) | Non-Null |  | method of request |
| **request** | varchar(200) | Non-Null |  | The URL of request |
| **body** | json | Null |  | content of response |
| **cookie** | json | Null |  | Cookie of response from server |
| **header** | json | Null |  | header of response from server |
| **date** | datetime | Null |  | The date of request |
| **time** | int(11) | Null |  | The time of response |

TABLE III Collection Table Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **collection** | | | |
| **Attribute** | **Type** | **Null/Non-Null** | **constraint condition** | **Description** |
| **id** | int(11) | Non-Null | Primary key | The id of collection  auto-increment |
| **method** | varchar(10) | Non-Null |  | method of request |
| **request** | varchar(200) | Non-Null |  | The URL of request |

TABLE IV Method Table Design

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table Name** | **method** | | | |
| **Attribute** | **Type** | **Null/Non-Null** | **constraint condition** | **Description** |
| **id** | int(11) | Non-Null | Primary key | The id of method |
| **method** | varchar(10) | Non-Null |  | method of request |

Here is the basic design for interfaces in this project:

//Get HTTP history list

@GetMapping("/GetHistory")

//Get HTTP history details by id

@GetMapping("/GetHistoryById/{id}")

//Insert HTTP history

@PostMapping("/InsertHistory/{request}/{method}/{body}/{cookie}/{header}/{time}")

//Insert HTTP history by JSON

@PostMapping(value = "/InsertHistoryByJSON",produces = "application/json;charset=UTF-8")

//Delete HTTP history by id

@DeleteMapping("/DeleteHistoryById/{id}")

//Get all the HTTP methods

@GetMapping("/GetMethods")

//Get HTTP collection list

@GetMapping("/GetCollections")

//Add HTTP collection by id

@PostMapping("/AddToCollection/{id}")

//Delete HTTP collection by id

@DeleteMapping("DeleteCollectionById/{id}")

//Analyze URL request by method and URL address

@GetMapping("/AnalyseURL/{method}/{url}")

V. Evaluation

## A. The Functional Test for API Test Tool

Figure 9 shows the testcase that we plan to realize. There are three main areas in this tool. The first one is about the URL be entered in and the kinds of interface client want to get. In this area, we design five main input condition. They are Normal, Basic Auth, Digest Auth, OAth 1.0 and No environment. The testcase of the Normal can be seen as a basic model for all the other three conditions. The client can input the URL and send it in this area. The several kinds of interface requests also can be chosen here. The preview bottom in this area can check the basic data of previous URL. The add to collection bottom can save the URL client want to collect. The Basic Auth, Digest Auth, OAth 1.0 all have different additional conditions. In this area, we realize all Normal, Basic Auth, Digest Auth and OAth 1.0 conditions. The function of No environment is what we didn’t complete. Besides, when it comes to interface request, we realize Get, Post, Put and Delete here.

图片包含 文字

描述已自动生成

Fig. 9. The testcase of API Test Tool

The second main area shows the data of the interface in different kinds of types. Client can choose the Pretty version, Raw version or click preview to see the website of the URL they enter in. There also a toggle bottom which can used to zoom in or out. The Cookies and Headers of the URL also can be checked in this area. We basically implemented all the functions here.

The third main area contains the collections and history, all the history query log and collections can be found in this area. We also complete all the functions as planned in this area.

For the testcase, we can design it from the following aspects. When it comes to the input area, we must make sure how many characters it can enter. Besides, what kinds of content can be entered should also be paid attention to. We should design the testcase including letter of alphabet, number and symbol, even the negative number should also be considered.

Therefore, first of all, we should design the testcase only including numbers. The first kind should be a very small negative number, such as -231. The second kind can be 0. The third kind should be the very large number like 231.

Then, the testcase only contains letter of alphabet should be designed. Besides, the testcase with letter and number, letter and symbol or number and letter should also be considered.

## B. The Perforance Test for API Test Tool

Apache JMeter is a Java-based stress testing tool developed by the Apache organization, which is used to stress test software, originally designed for web application testing, and later extended to other test areas. It can be used to test static and dynamic resources such as static files, Java servlets, CGI scripts, Java objects, databases, FTP servers, and more. JMeter can be used to simulate large loads on servers, networks, or objects, testing their strength and analyzing overall performance from different stress categories. In addition, JMeter can perform functional/regressive testing of the application by verifying that your program returns the results you expect by creating a script with assertions.

Our project mainly uses this tool to perform performance tests. On the http request page, we enter the server name or IP, port number, request protocol, request method, encoding, path. Through performance testing, we can view the monitored response time, error, throughput and other data. We test some of these interfaces as examples.

*1) GetMethods()*

For web server, the protocol is HTTP, server name or IP is 150.109.50.19, default port number is 8081. For HTTP request, the method is GET, path is /GetMethods.

图片包含 屏幕截图

描述已自动生成

Fig. 10. HTTP request of GetMethods()

In the aggregation report, we simulated sending 2 requests with an average response time of 325, a minimum response time of 106, a maximum response time of 545, a number of error requests (error rate) of 0, and a throughput of 4.1/min.



Fig. 11. Summary report of GetMethods()

*2) GetAllHistory()*

For web server, the protocol is HTTP, server name or IP is 150.109.50.19, default port number is 8081. For HTTP request, the method is GET, path is /GetHistory.

图片包含 屏幕截图

描述已自动生成

Fig. 12. HTTP request of GetAllHistory()

In the aggregation report, we simulated sending 2 requests with an average response time of 208, a minimum response time of 103, a maximum response time of 314, a number of error requests (error rate) of 0, and a throughput of 8.0/min.



Fig. 13. Summary report of GetAllHistory()

*3) GetHistoryById(id)*

For web server, the protocol is HTTP, server name or IP is 150.109.50.19, default port number is 8081. For HTTP request, the method is GET, path is /GetHistoryById/2. where 2 is the random id value we test.

In the aggregation report, we simulated sending 50 requests with an average response time of 27918, a minimum response time of 7144, a maximum response time of 51281, a number of error requests (error rate) of 0, and a throughput of 19.0/min.



Fig. 14. Summary report of GetHistoryById(id)

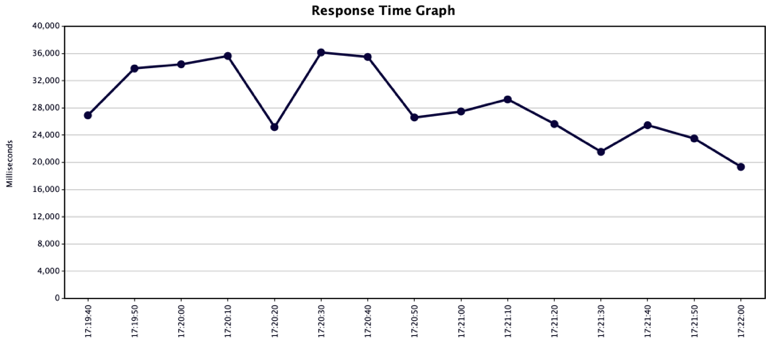


Fig. 15. Response Time Graph

*4) DeleteHistoryById(id)*

For web server, the protocol is HTTP, server name or IP is 150.109.50.19, default port number is 8081. For HTTP request, the method is DELETE, path is /DetetehistoryByid/2, where 2 is the random id value we test.

图片包含 屏幕截图

描述已自动生成

Fig. 16. HTTP request of DeleteHistoryById(id)

In the aggregation report, we simulated sending 50 requests with an average response time of 142, a minimum response time of 92, a maximum response time of 331, a number of error requests (error rate) of 0 and a throughput of 2.8/sec. At this point, throughput is meaningless.



Fig. 17. Summary report of DeleteHistoryById(id)

*5) InsertHistory(params)*

The difference is that in the post method, we add the HTTP header manager, add a header named Content-Type, the value is application/json, and enter all the parameters in the request.

图片包含 屏幕截图, 监视器

描述已自动生成

Fig. 18. HTTP request of InsertHistory(params)



Fig. 19. HTTP Header Manager

In the aggregation report, we simulated sending 50 requests with an average response time of 196, a minimum response time of 102, a maximum response time of 1206, a number of error requests (error rate) of 0, and a throughput of 2.7/sec.



Fig. 20. Summary report of InsertHistory(params)

VI. The Interface Diagram of API Test Tool

The basic design of this project can be seen in figure 21.

图片包含 屏幕截图

描述已自动生成

Fig. 21. The basic design of API Test Tool

There are 4 modes in API test:Normal,Basic Auth,Digest Auth,OAuth 1.0.

VII. Conclusion

This project aims to develop an API test tool which is essential for developing an application, especially when the development progress of front end and back end is inconsistent. In design of main web page, an interface request history and interface collection is on the left, and the interface request area is on the right, which supports get, post and other common request methods. This tool mainly aims to simulate network request package, create requests quickly, playback and manage request and set up network agent quickly. It can also process pressure test and interface function test which can set assertion, batch test interface, batch import and export, set global variables, set environment variables, etc.

Although the implementation of API Test Tool has finished, it has insufficient and limitation. It implements some request methods not included options, link and purge, etc. And it can’t manipulate file related operations and use non HTTP protocol. Thorough the whole project, there are much more consideration than what we expected. For example, the technique required for server programming and the insufficient knowledge about vue used to develop front end. Every problem has driven us to brainstorm more potential ways, not limited to one single way, to improve the solution.

As for the work to be improved in the future, first test the tool’s function and find other problems on techniques, and then add other request methods to make the tool more perfect. In the development of other projects in the future, we can use the test tool to help us accelerate the exploitation plan.

##### Reference

1. Patents; Researchers Submit Patent Application, "System and Method for Automatically Testing an Application Programming Interface (Api)", for Approval (USPTO 20160283364)[J]. Computer Weekly News,2016.
2. Yang Mengmeng, Liu Meng. Design and implementation of interface test data generation tool [J]. Science and technology economics guide, 2019.
3. Colantonio*,* Joe. “Automation Testing Ultimate Guide (How, What, Why)?” Automation, <https://www.joecolantonio.com/automation-testing-resources/>
4. Postman, Inc “Automated Testing-Understanding, designing, and setting up an effective automated testing strategy”, <https://www.getpostman.com/infographics/automated-testing-whitepaper.pdf>

##### Student Bio

**Lan Yiyuan (55614744):** graduated from Sun Yat-sen University and received the B.S. degree in Software Engineering. She is currently a postgraduate student in Computer Science at City University of Hong Kong. In this project, she is mainly responsible for the design of the project and writing the requirements and the design parts in the final paper. Also, the work of collecting, arranging and typesetting the presentation slides is done by her.

**Wang Haipeng (55762913):** He graduated from Beijing University of science and technology, and after graduation, he worked in IGT company for more than one year as a software developer. His research interests include software engineering, machine learning, blockchain, etc., and he has ever obtained a patent presented by United States Patent and Trademark Office about blockchain.

In this project, he worked as a scrum master and team leader to arrange sprint works and supervisor project progress. At the same time, he attended the work of original requirement definition. What’s more, for the content of project, he worked on framework design, project configuration and development, and for the thesis, he wrote the solution part of content and reviewed the whole thesis.

**Sun Jingyao (55880040):** This author completed her undergraduate education in Nanjing University of Posts and Telecommunications and her major is Computer Science and Technology. She is pursuing the master's degree of Computer Science in City University of Hong Kong now. Sun Jingyao has great interests in network security during her undergraduate period and published three essays about the security of VPN and Firewalls as first author and Co-author. In this project, she was responsible for the Functional testing job and the corresponding part of the paper writing. In addition, the collected of the whole report and the formatting job was also done by her.

**Zou Shuang (55911609):** Received the B.S. degree in English and software engineering from Wuhan Institute of Technology , in 2019. She is currently pursuing the M.S. degree in Computer Science at City University of HongKong.

She can use C, C++, Java language and simply develop web page. In the project she is mainly responsible for designing database and server. For designing server, the data logic is simple and some methods implemented by interfaces. So the total work is relatively easy for her.

**Li Yixin(55975771):** She received the Bachelor Degree in Computer Science from Western University in 2019. She is currently pursuing the Master degree in Computer Science at City University of HongKong. In this project, she participated in designing product prototype (including user interface and interaction) and drawing UML diagrams (activity diagram). In addition, she is responsible for writing the II and III section (Related work and Preliminaries) of the final project report. Using online scrum tool Leangoo to make the sprint backlog tables.

**Zhao Dongyu(55980768):** Graduated from WuHan University of Technology and major in Computer Science. He is Very experienced in JavaWeb development. In the project,he is responsible for the implementation of Server building and Database design, the implementation of Content(mainly HTML) and solve problems in Content-Server interaction, project deployment on Linux CentOS 7 Server, presentation slides and Server Design part in final report.

**Du Kejun(56075096):** She graduated from Nanjing University of Posts and Telecommunications of Computer Science，who is now a 19fall student from Computer Science major (data science stream) in City University of HongKong. She is interested in programming and data analysis tasks and currently undertakes testing work on the project. The main contributions of her are defining requirements (including designing function modules and drawing testcase diagrams), using jmeter to test sever, drawing UML diagrams (activity diagram), responsible for making corresponding ppt and articles, etc.