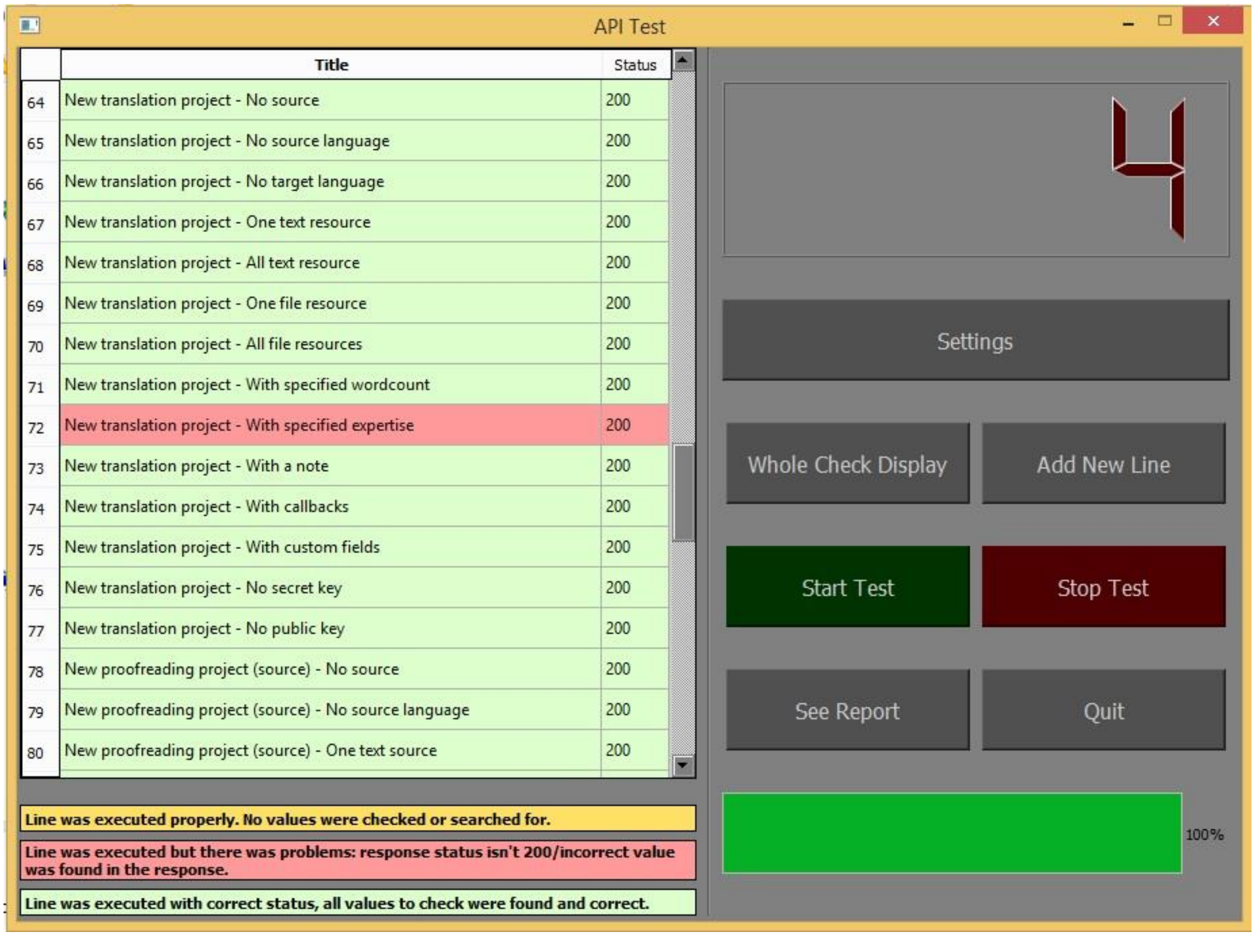
API – Test

Main Window



	Title	Status
64	New translation project - No source	200
65	New translation project - No source language	200
66	New translation project - No target language	200
67	New translation project - One text resource	200
68	New translation project - All text resource	200
69	New translation project - One file resource	200
70	New translation project - All file resources	200
71	New translation project - With specified wordcount	200
72	New translation project - With specified expertise	200
73	New translation project - With a note	200
74	New translation project - With callbacks	200
75	New translation project - With custom fields	200
76	New translation project - No secret key	200
77	New translation project - No public key	200
78	New proofreading project (source) - No source	200
79	New proofreading project (source) - No source language	200
80	New proofreading project (source) - One text source	200

🟨 Line was executed properly. No values were checked or searched for.

🟥 Line was executed but there was problems: response status isn't 200/incorrect value was found in the response.

🟩 Line was executed with correct status, all values to check were found and correct.

This window will show you live information about the test running.

While you are in the middle of a test the table on the left will be updated with every API call made. There is a title to each API call and the returned status from the server.

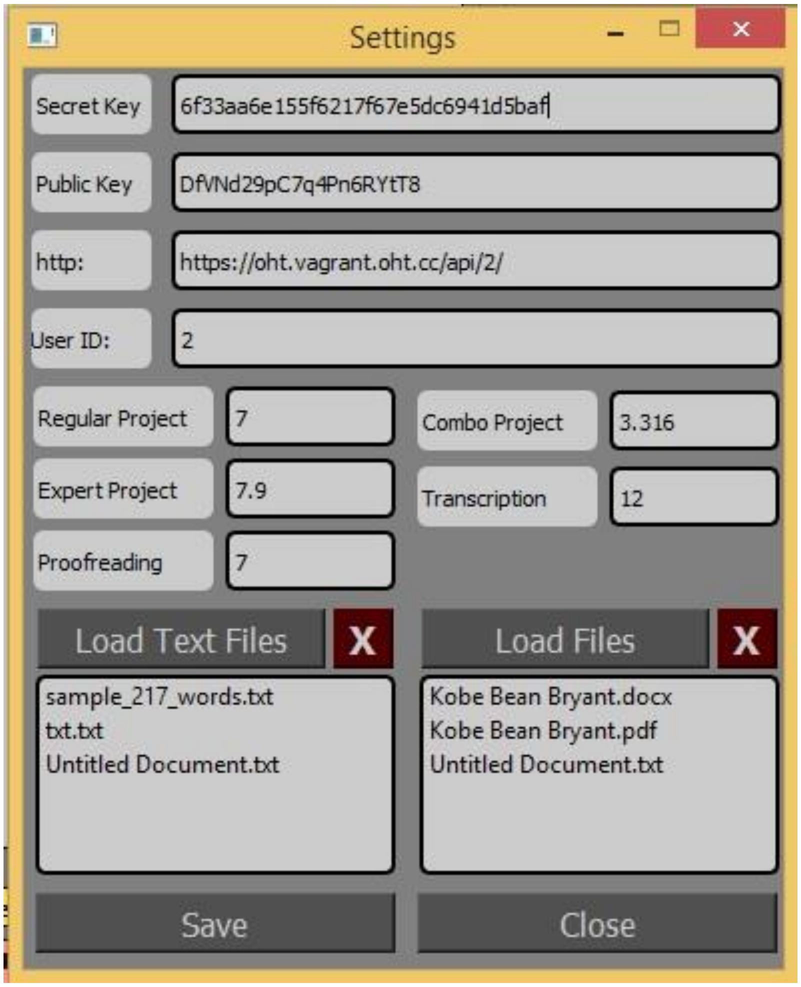
On the top-right there is a big number that count all the errors the test has encountered. After the test is finished you can click on “See Report” to see a more detailed report of each API call made.

Buttons:

- Settings - Here you inset all relevant information for the test. See Settings Window for more information.
- Whole Check Display – Here you can see each API call from the main test file. You can update, delete each API call in this window. For more information, see Whole Check Display Window.
- Add New Line – Here you can add new lines for the main test. See Add New Line Window for more info.

- Start Test – Will start a test.
- Stop Test – If you want to stop a test in the middle, you can click on this button to stop.
- See Report – Here you can see a report of the test, will be available once you run a test. For more info see “See Report Window”.
- Quit – will close the program.

Settings Window



Secret Key: 6f33aa6e155f6217f67e5dc6941d5baf

Public Key: DfVNd29pC7q4Pn6RYtT8

http: https://oht.vagrant.oht.cc/api/2/

User ID: 2

Regular Project: 7

Expert Project: 7.9

Proofreading: 7

Combo Project: 3.316

Transcription: 12

sample_217_words.txt
txt.txt
Untitled Document.txt

Kobe Bean Bryant.docx
Kobe Bean Bryant.pdf
Untitled Document.txt

Secret Key: Your users secret key from the site.

Public Key: Your users public key from the site.

http: your vagrant address.

User ID: user’s id

The next five sections: Regular Project, Expert Project, Proofreading, Combo Project and Transcription are inserted manually and saved. If you changed the prices on your site, please update them here so you will get correct info for the tests.

Load Text Files and Load Files buttons: You can add or delete files for the test.

The text files are for text resources, those files are located in the programs folder:

API-Test-Portable-master\Text_Files

You can add your own files by copying them to the folder before you start the program or by clicking on “Load Text Files” and selecting. You can select a file you don’t want in the list under the button and click on the red “X”, this will delete the file from the current test plan.

The program will load the files from the folder every time you click on main.exe.

To the right is the files resource upload, the folder is:

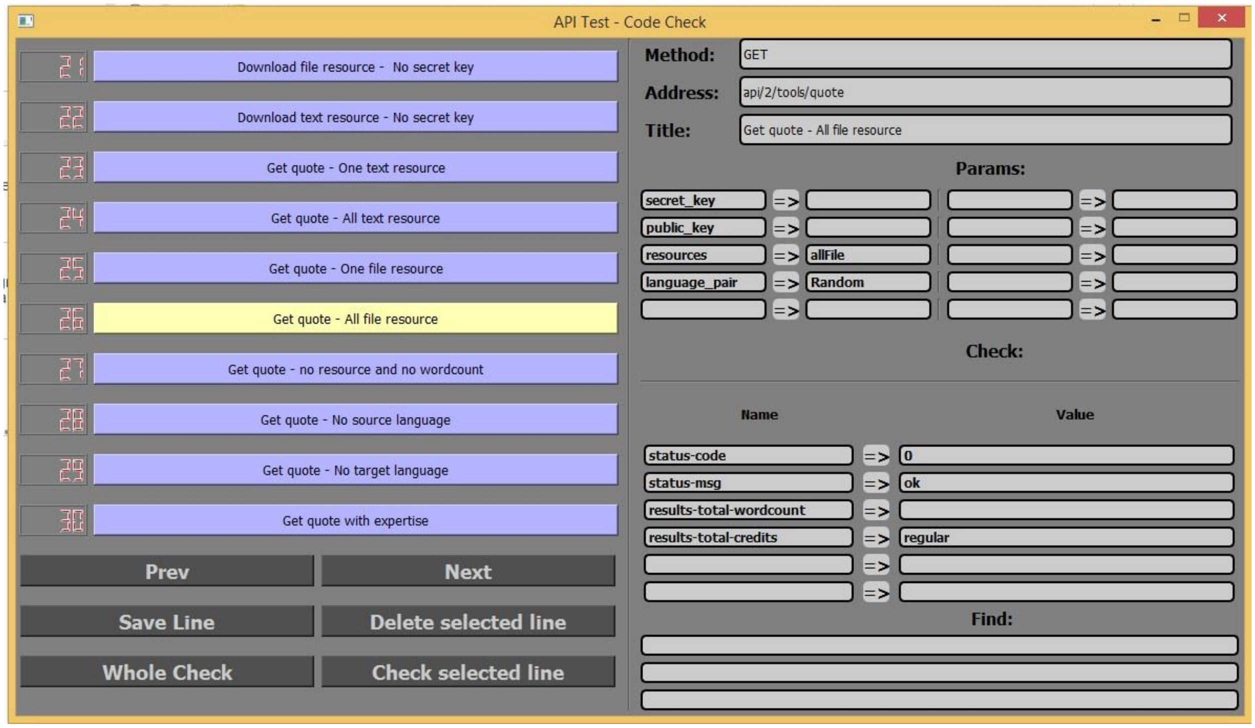
API-Test-Portable-master\Other_Files

This section is behaving the same as the text section.

Important: After you make any change to the settings, you have to click on “Save” other vice no changes will be saved.

To close the window press “Close”.

Whole Check Display Window


API Test - Code Check
21 Download file resource - No secret key
22 Download text resource - No secret key
23 Get quote - One text resource
24 Get quote - All text resource
25 Get quote - One file resource
26 Get quote - All file resource
27 Get quote - no resource and no wordcount
28 Get quote - No source language
29 Get quote - No target language
30 Get quote with expertise
Prev
Next
Save Line
Delete selected line
Whole Check
Check selected line

Method: GET

Address: api/2/tools/quote

Title: Get quote - All file resource

Params:

secret_key	=>	
public_key	=>	
resources	=>	allFile
language_pair	=>	Random

Check:

Name		Value
status-code	=>	0
status-msg	=>	ok
results-total-wordcount	=>	
results-total-credits	=>	regular

Find:

This window will show you all API calls in the test and lets you update or delete API calls from the main test file.

On the left you have 10 blue buttons, each button is showing information of an API test.

You can navigate between pages by pressing “Prev” or “Next”:

Prev – will show you previous 10 API calls.

Next – will show you the next 10 API calls.

When you want to see all the information of a specific API call. You can click the blue button of the call, the button will change its color to yellow and on the right you will see all the info of the call:

Method – Get/Post

Address – this is what to add to the end of the address from the settings, if this isn’t written right the call will fail.

Title – this is the title shown to the user in the main window while a test is running.

Params – Will show all the parameters provided with the call.

Check – Will show all the checks that are made to the response that is returned after the API call.

Find – You can search for a specific string in the response, here you will see if there is a search for a string.

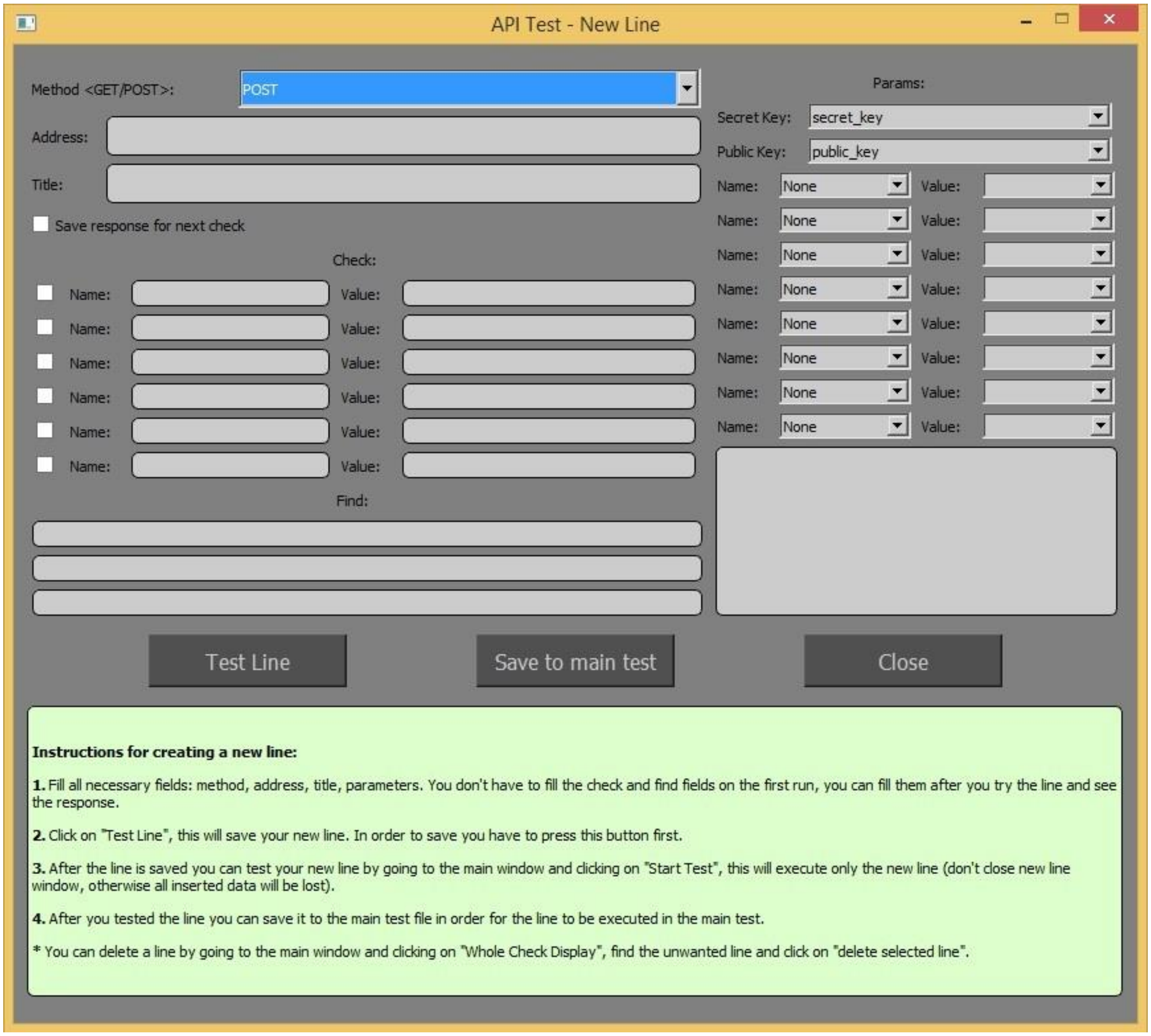
- All light grey areas on the right are editable and you can change all parameters.

Buttons on bottom-left:

- Save Line – will update the highlighted API call. If you made changes to a line you have to click on this button in order to save it.
- Delete Selected Line – this button will delete the selected line.
  Very Important – After you delete a line there is no way back, the line will be lost.

- Check Selected Line – If you want to check only one line, select the line you want to check and click this button. The window will close, when you click on Start Test only the selected line will be executed.
- Whole Check – If you experimented with one line tests and you want to go back and run the whole test, click this button. The window will be closed and when you click on Start Test the whole test will be executed.

Add New Line Window



Method <GET/POST>: POST

Address:

Title:

Secret Key: secret_key

Public Key: public_key

Name: None Value:

Instructions for creating a new line:

1. Fill all necessary fields: method, address, title, parameters. You don't have to fill the check and find fields on the first run, you can fill them after you try the line and see the response.

2. Click on "Test Line", this will save your new line. In order to save you have to press this button first.

3. After the line is saved you can test your new line by going to the main window and clicking on "Start Test", this will execute only the new line (don't close new line window, otherwise all inserted data will be lost).

4. After you tested the line you can save it to the main test file in order for the line to be executed in the main test.

* You can delete a line by going to the main window and clicking on "Whole Check Display", find the unwanted line and click on "delete selected line".

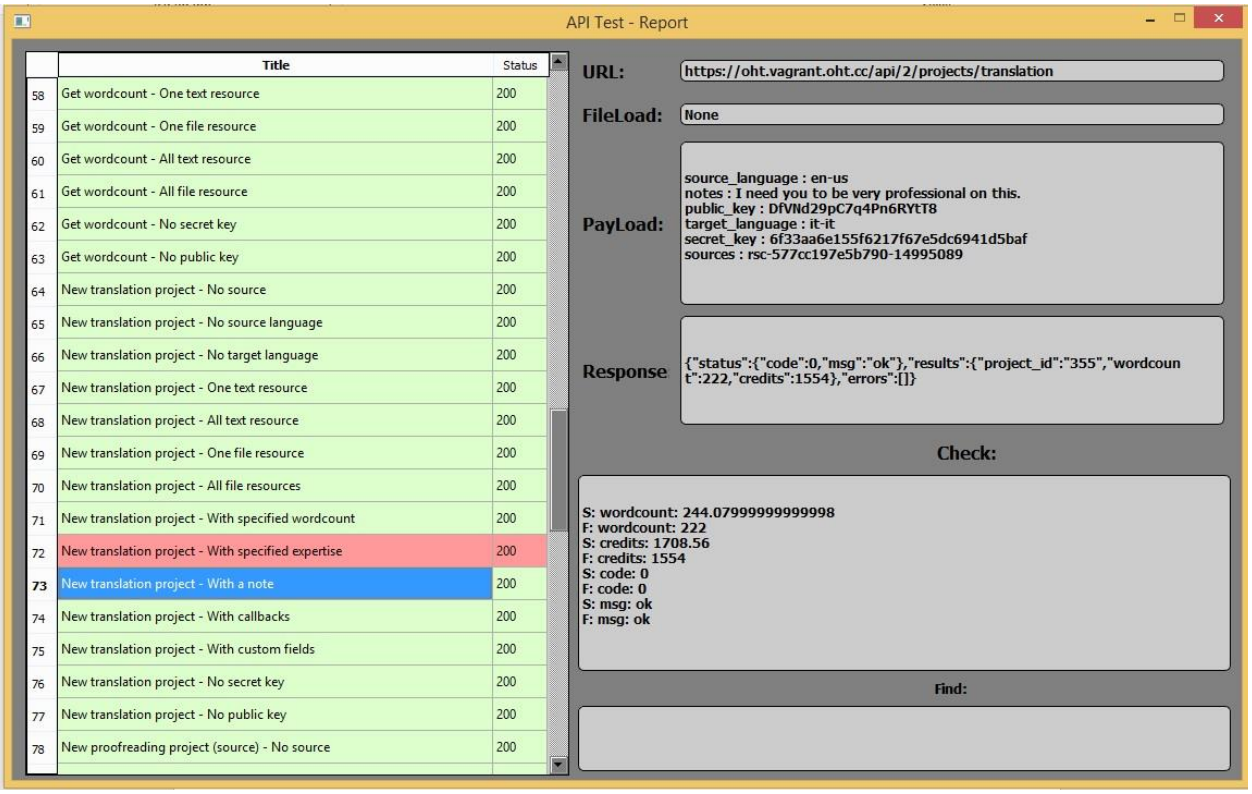
In this window you can create a new line, see more instruction on creating a new line in Creating a new line section.

In order to create a new line you have to enter all mandatory fields: address, title and parameters (depending on the API call you want to create).

Buttons:

- Test Line – will give you the opportunity to check the call before saving it to the main file.
- Save to main test – will save the new line to the main test file.
- Close – will close the window.

Report Window



	Title	Status
58	Get wordcount - One text resource	200
59	Get wordcount - One file resource	200
60	Get wordcount - All text resource	200
61	Get wordcount - All file resource	200
62	Get wordcount - No secret key	200
63	Get wordcount - No public key	200
64	New translation project - No source	200
65	New translation project - No source language	200
66	New translation project - No target language	200
67	New translation project - One text resource	200
68	New translation project - All text resource	200
69	New translation project - One file resource	200
70	New translation project - All file resources	200
71	New translation project - With specified wordcount	200
72	New translation project - With specified expertise	200
73	New translation project - With a note	200
74	New translation project - With callbacks	200
75	New translation project - With custom fields	200
76	New translation project - No secret key	200
77	New translation project - No public key	200
78	New proofreading project (source) - No source	200

URL: https://oht.vagrant.oht.cc/api/2/projects/translation

FileLoad: None

PayLoad:

```
source_language : en-us
notes : I need you to be very professional on this.
public_key : DfVNd29pC7q4Pn6RYtT8
target_language : it-it
secret_key : 6f33aa6e155f6217f67e5dc6941d5baf
sources : rsc-577cc197e5b790-14995089
```

Response

```
{"status":{"code":0,"msg":"ok"},"results":{"project_id":"355","wordcoun
t":222,"credits":1554},"errors":[]}
```

Check:

```
S: wordcount: 244.07999999999998
F: wordcount: 222
S: credits: 1708.56
F: credits: 1554
S: code: 0
F: code: 0
S: msg: ok
F: msg: ok
```

Find:

After a test is finished you can open this window to see detailed stats of the call.

The table on the left is the same table as in the main window. You can click any line, after you select a line all the details of the call will be shown in the fields on the right.

URL field – will show the address line executed by the call.

FileLoad field – will show the path of the uploaded file if there was an uploaded file.

PayLoad field – will show all sent parameters like secret key, public key, languages etc.

Response field – will show the response of the server to the call that was executed. This is where the program checks for correct response values.

Check field – Here you can see what the program looked for and what it found. In the picture above you can see that one of the searches was:

S: credits: 1708.56

“S” means searched.

“credits” means searched for credits in the response.

1708.56 – searched for value of credits.

Then we have

F: credits: 1554

“F” means found in response.

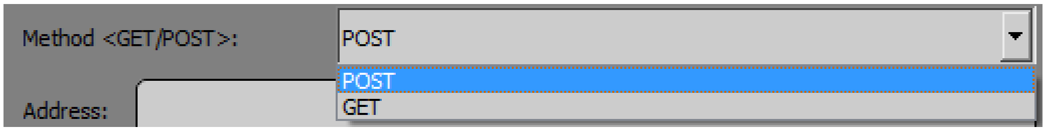
The other two fields are the same as search, each check has two fields one for searched and one for found.

There is a difference in credits amount because the program counts and calculates all values and word count by itself. If found wordcount/credits are larger than the amount calculated, you will see an error in the test table and the whole line will be painted red.

Find field – If used will show what string was searched for in the response. If no such string was found will paint the line red and report as error.

Creating a new line

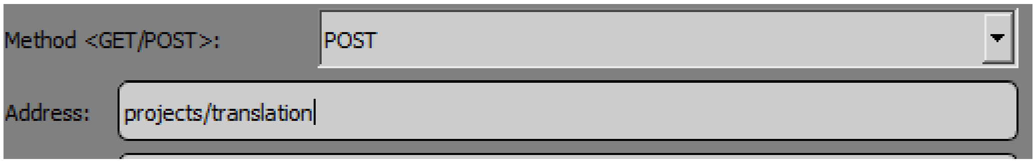
1. Click on Add New Line button in the main menu.
2. Choose Method Get/Post



Method <GET/POST>: POST

Address:

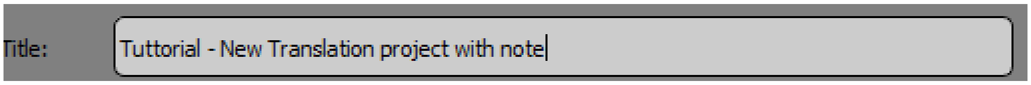
3. Enter address for the call:



Method <GET/POST>: POST

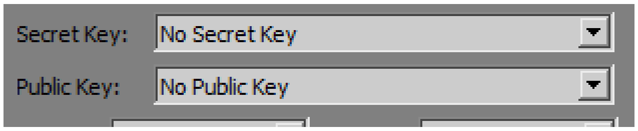
Address: projects/translation

4. Give your new line a proper title so when you see it in the test you will understand what it does.



Title: Tuttorial - New Translation project with note

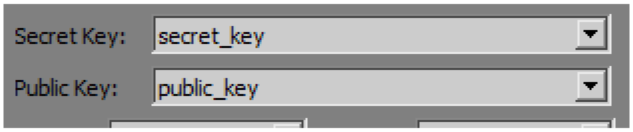
5. Secret key and public keys are added automatically, if you want to execute a call without one of the keys you can select “No Secret Key” or “No Public Key”



Secret Key: No Secret Key

Public Key: No Public Key

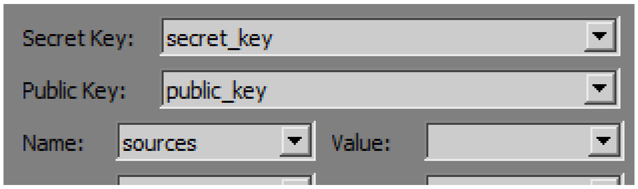
But for the new line we need keys:



Secret Key: secret_key

Public Key: public_key

6. To create a new translation project we need to supply sources, languages and a note. Select Sources in the dropdown menu:

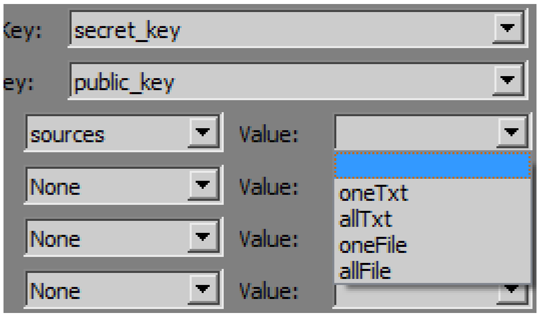


Secret Key: secret_key

Public Key: public_key

Name: sources Value:

Now we need to select the value of sources:



Key: secret_key

ey: public_key

sources Value:

None Value:

None Value:

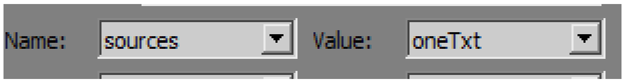
None Value:

<u>The values:</u>

- oneTxt – will send one uploaded text resource.
- allTxt – will send all uploaded text resources.
- oneFile – will send one uploaded file.
- allFile – will send all uploaded files.

- All of the resources (Text/File) are found in the settings window. Every time the program starts a new test it will upload all resources and when any call needs a resource, instead of uploading a new one, the program will take an existing one.

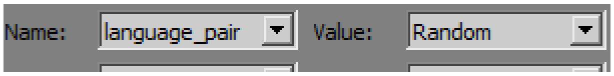
Pick any one of the four, I pick oneTxt



Name: sources Value: oneTxt

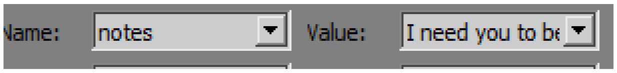
7. Now we need a language pair. Like the resources there is an API call to see which language pairs you have in your vagrant environment. The program saves the pairs. If you select: source_language, target_language or language_pair all will give you the value “Random”. This means that the languages are chosen randomly from what you have.
   - If you need both target and source languages pick “language_pair” this will create target and source language parameters from the existing pairs.
   - If you need one out of source and target select the one you need, this will give you randomly one of the existing target or source languages.

   We need both source and target:



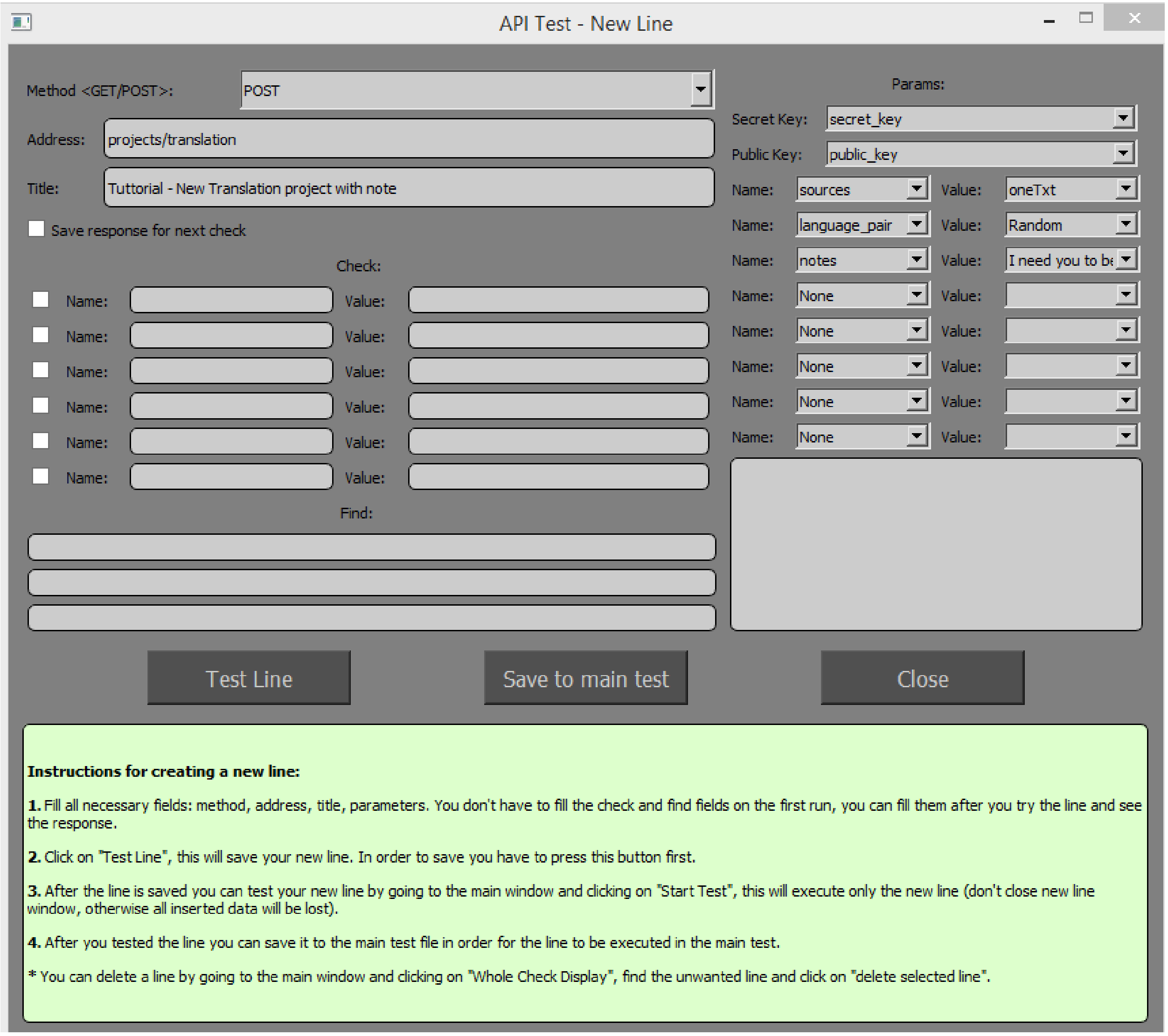
Name: language_pair Value: Random

8. Now we have to select a note, for now there is only prefixed notes, lets select note and value:



Name: notes Value: I need you to b

9. Now we have everything we need to start a new project and for the API call to execute correctly. This is hove everything looks:



Method <GET/POST>: POST

Address: projects/translation

Title: Tuttorial - New Translation project with note

□ Save response for next check

Params:

Secret Key: secret_key

Public Key: public_key

Name:	sources	Value:	oneTxt
Name:	language_pair	Value:	Random
Name:	notes	Value:	I need you to be
Name:	None	Value:	
Name:	None	Value:	
Name:	None	Value:	
Name:	None	Value:	
Name:	None	Value:	

Instructions for creating a new line:

1. Fill all necessary fields: method, address, title, parameters. You don't have to fill the check and find fields on the first run, you can fill them after you try the line and see the response.

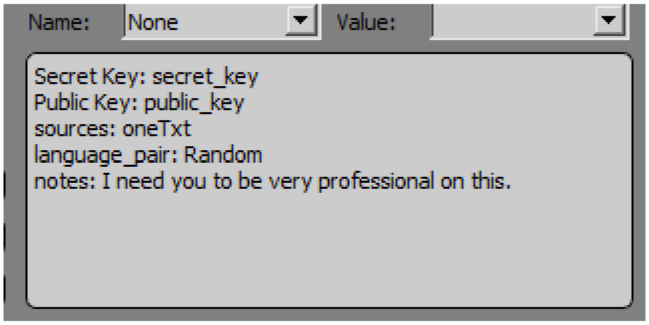
2. Click on "Test Line", this will save your new line. In order to save you have to press this button first.

3. After the line is saved you can test your new line by going to the main window and clicking on "Start Test", this will execute only the new line (don't close new line window, otherwise all inserted data will be lost).

4. After you tested the line you can save it to the main test file in order for the line to be executed in the main test.

* You can delete a line by going to the main window and clicking on "Whole Check Display", find the unwanted line and click on "delete selected line".

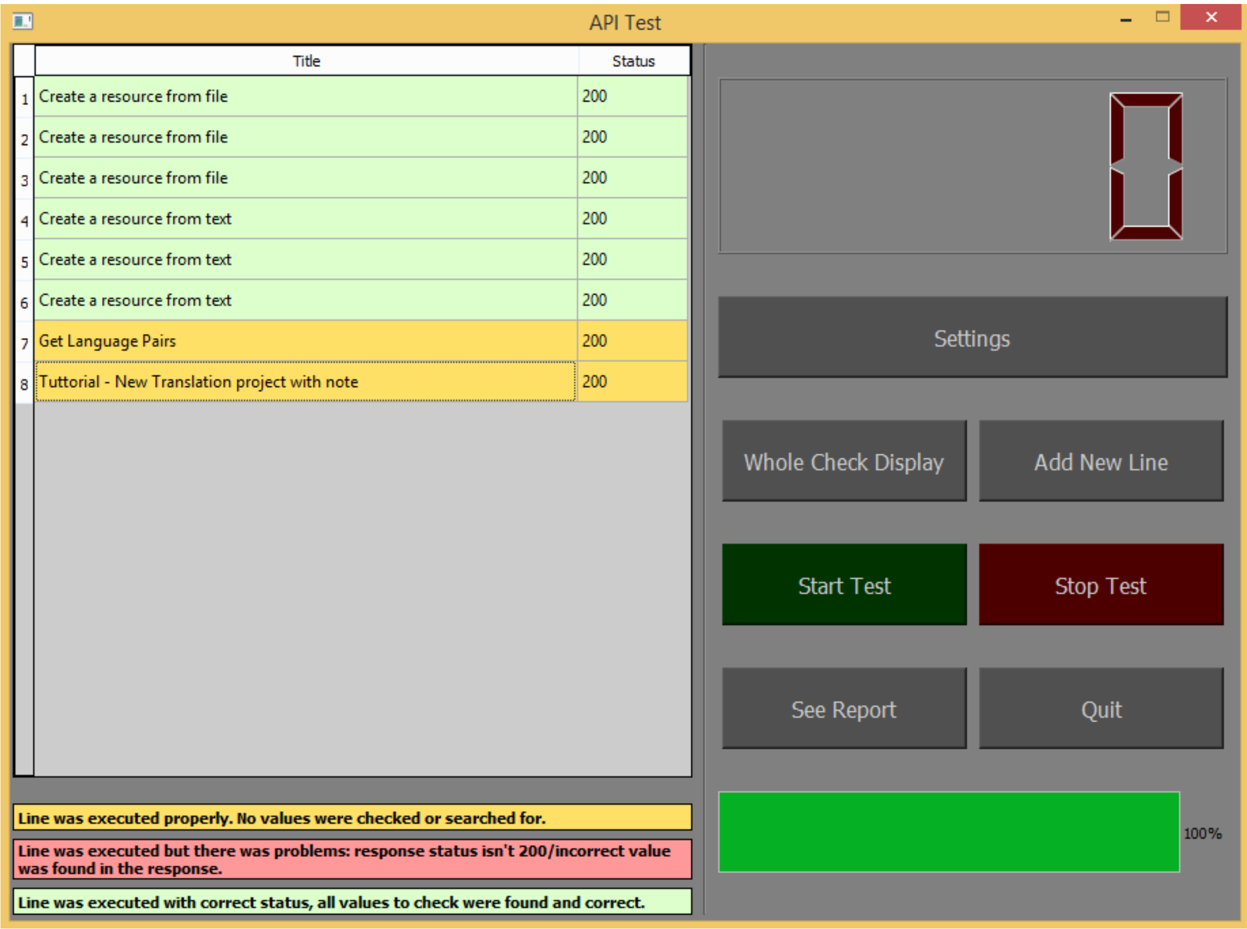
10. We can test the call just to see that everything is correct. Click on "Test Line", after you click the button you will see all the params underneath the params sections (those are the saved params)



Name: None Value:

```
Secret Key: secret_key
Public Key: public_key
sources: oneTxt
language_pair: Random
notes: I need you to be very professional on this.
```

11. Don’t close the “New Line” window (If you do the new line will be lost).
    Go to the main window and click “Start Test” and the results are:



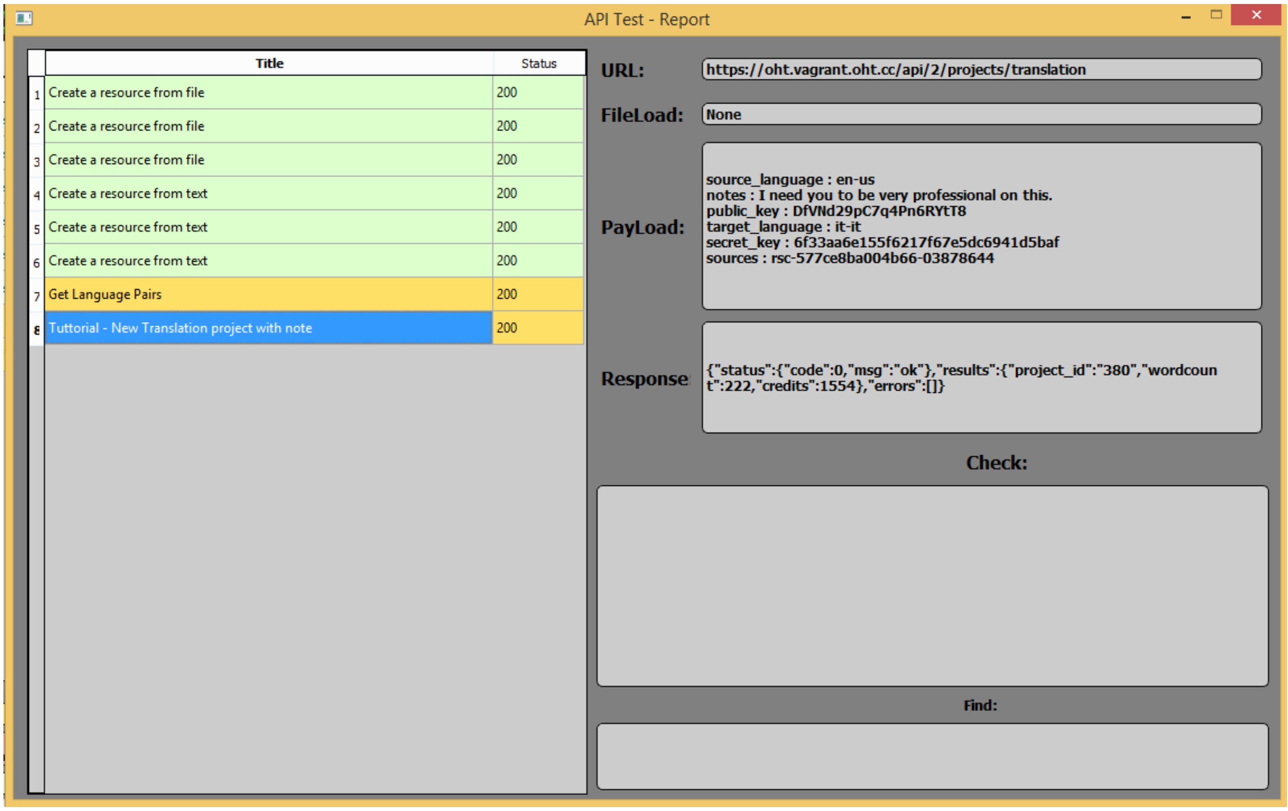
	Title	Status
1	Create a resource from file	200
2	Create a resource from file	200
3	Create a resource from file	200
4	Create a resource from text	200
5	Create a resource from text	200
6	Create a resource from text	200
7	Get Language Pairs	200
8	Tuttorial - New Translation project with note	200

Line was executed properly. No values were checked or searched for.

Line was executed but there was problems: response status isn't 200/incorrect value was found in the response.

Line was executed with correct status, all values to check were found and correct.

You can see the last line is our new line with our title, but the line is yellow which means nothing was checked. Let's see the report of the line, click on "See Report":



	Title	Status
1	Create a resource from file	200
2	Create a resource from file	200
3	Create a resource from file	200
4	Create a resource from text	200
5	Create a resource from text	200
6	Create a resource from text	200
7	Get Language Pairs	200
8	Tuttorial - New Translation project with note	200

URL: https://oht.vagrant.oht.cc/api/2/projects/translation

FileLoad: None

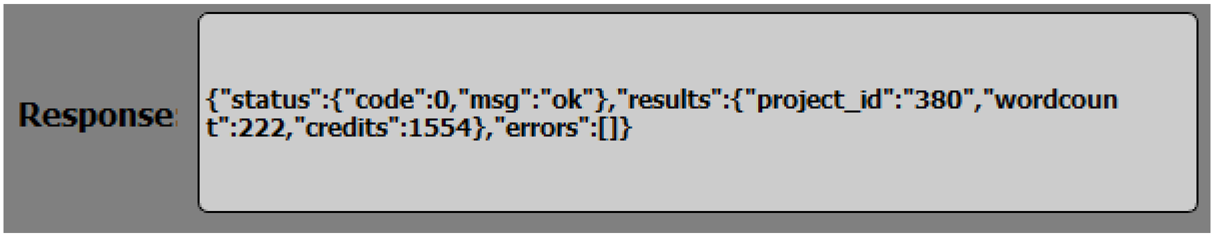
PayLoad:

```
source_language : en-us
notes : I need you to be very professional on this.
public_key : DfVNd29pC7q4Pn6RYtT8
target_language : it-it
secret_key : 6f33aa6e155f6217f67e5dc6941d5baf
sources : rsc-577ce8ba004b66-03878644
```

Response

```
{"status":{"code":0,"msg":"ok"},"results":{"project_id":"380","wordcoun
t":222,"credits":1554},"errors":[]}
```

12. Let's zoom in to see the response:



Response

```
{"status":{"code":0,"msg":"ok"},"results":{"project_id":"380","wordcoun
t":222,"credits":1554},"errors":[]}
```

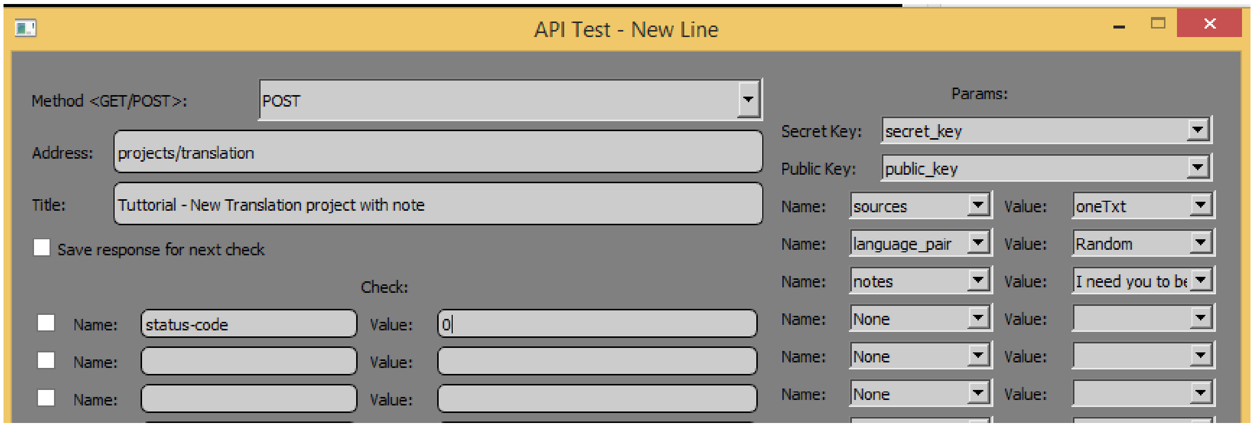
Now the logic behind the search:

The response is like a json file, this means if I want to check that the code = 0 (like in the response) I have to write the path to the "0":

status-code

If I put this line in check section and input 0 as the value to check, the program will try to find this path in the response, once the path found it will compare the value of code in the response to the value we provided. Let's try it out:

13. Go to the New Line window again and add “status-code” and value “0” to the check field:



Method <GET/POST>: POST

Address: projects/translation

Title: Tuttorial - New Translation project with note

□ Save response for next check

Check:

□ Name: status-code Value: 0

□ Name: Value:

□ Name: Value:

Params:

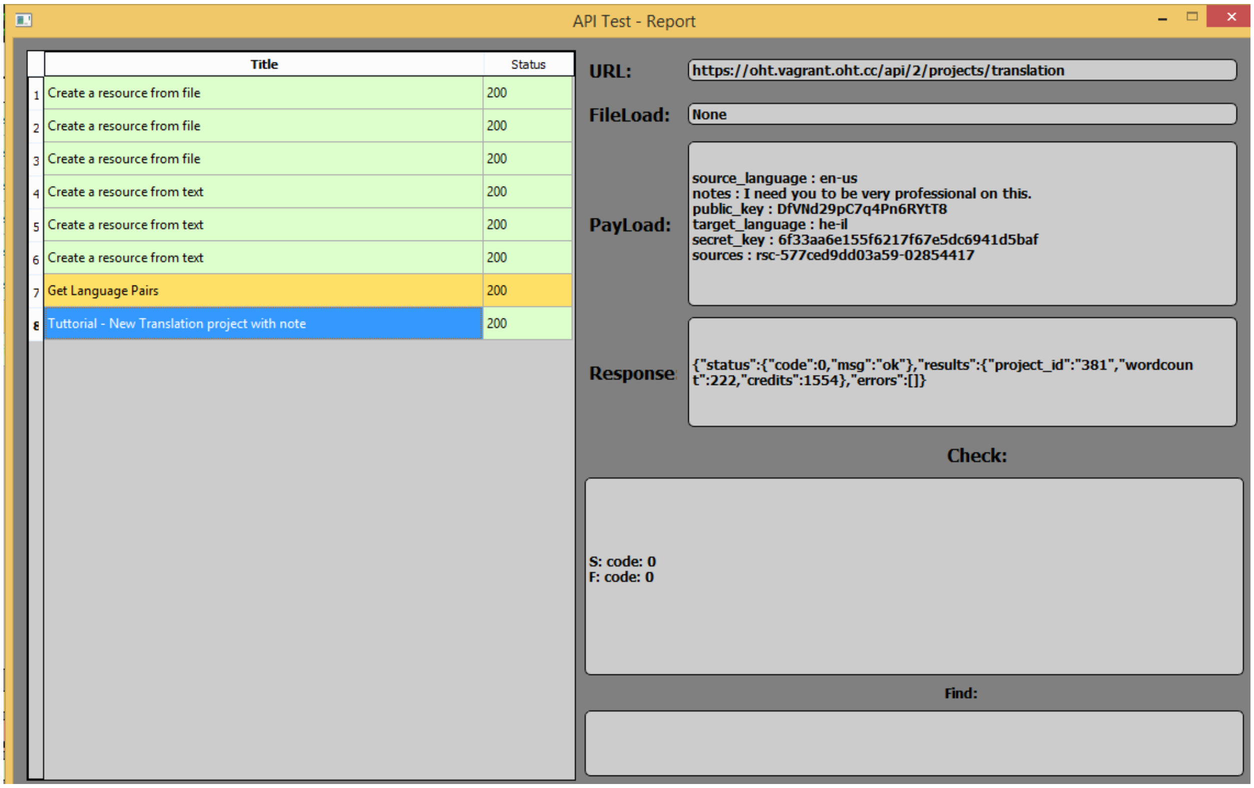
Secret Key: secret_key

Public Key: public_key

Name:	sources	Value:	oneTxt
Name:	language_pair	Value:	Random
Name:	notes	Value:	I need you to be
Name:	None	Value:	
Name:	None	Value:	
Name:	None	Value:	

Now click on “Test Line” and go to the main window, click on “Start Test”.

After test is finished click “See Report” and select the new line:



	Title	Status
1	Create a resource from file	200
2	Create a resource from file	200
3	Create a resource from file	200
4	Create a resource from text	200
5	Create a resource from text	200
6	Create a resource from text	200
7	Get Language Pairs	200
8	Tuttorial - New Translation project with note	200

URL: https://oht.vagrant.oht.cc/api/2/projects/translation

FileLoad: None

PayLoad:

```
source_language : en-us
notes : I need you to be very professional on this.
public_key : DfVNd29pC7q4Pn6RYtT8
target_language : he-il
secret_key : 6f33aa6e155f6217f67e5dc6941d5baf
sources : rsc-577ced9dd03a59-02854417
```

Response

```
{"status":{"code":0,"msg":"ok"},"results":{"project_id":"381","wordcoun
t":222,"credits":1554},"errors":[]}
```

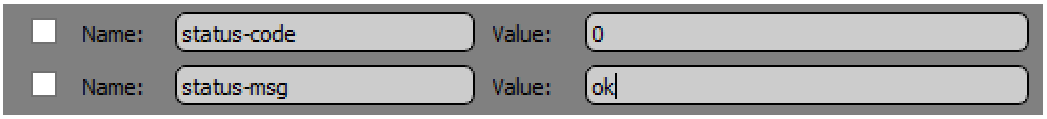
Check:

```
S: code: 0
F: code: 0
```

Find:

Notice that the line’s color changed to green, this is because now we did a check and the returned value was correct. See that now we have values shown in the check field.

14. Let’s add another check for the “msg”:”ok”:

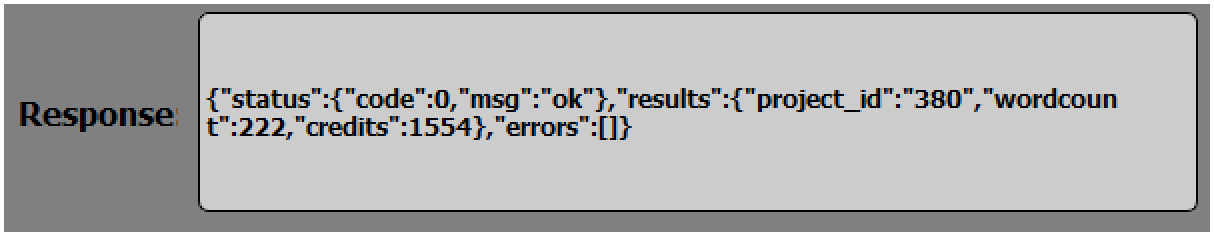


☐ Name: status-code Value: 0

☐ Name: status-msg Value: ok

This will do the same as the code check just for different property and value.

15. If you want to check the wordcount in the response all is calculated automatically and there is no need to insert a value, only the path and the path is:

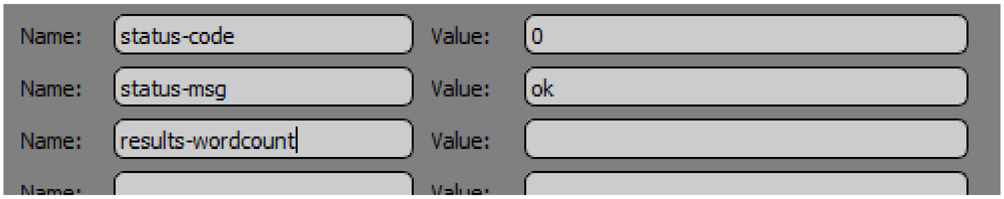


Response:

```
{"status":{"code":0,"msg":"ok"},"results":{"project_id":"380","wordcoun
t":222,"credits":1554},"errors":[]}
```

results-wordcount

the program will automatically search for the resource calculated wordcount and will compare the value in the response to the calculated value.



Name: status-code Value: 0

Name: status-msg Value: ok

Name: results-wordcount Value:

Name: Value:

16. If you want to check credits amount this also is calculated automatically but you have to specify the type of project you creating or getting a quote for. There are 5 types (the prices are defined in the settings window):
    1. regular
    2. expert
    3. proof
    4. combo
    5. transcript

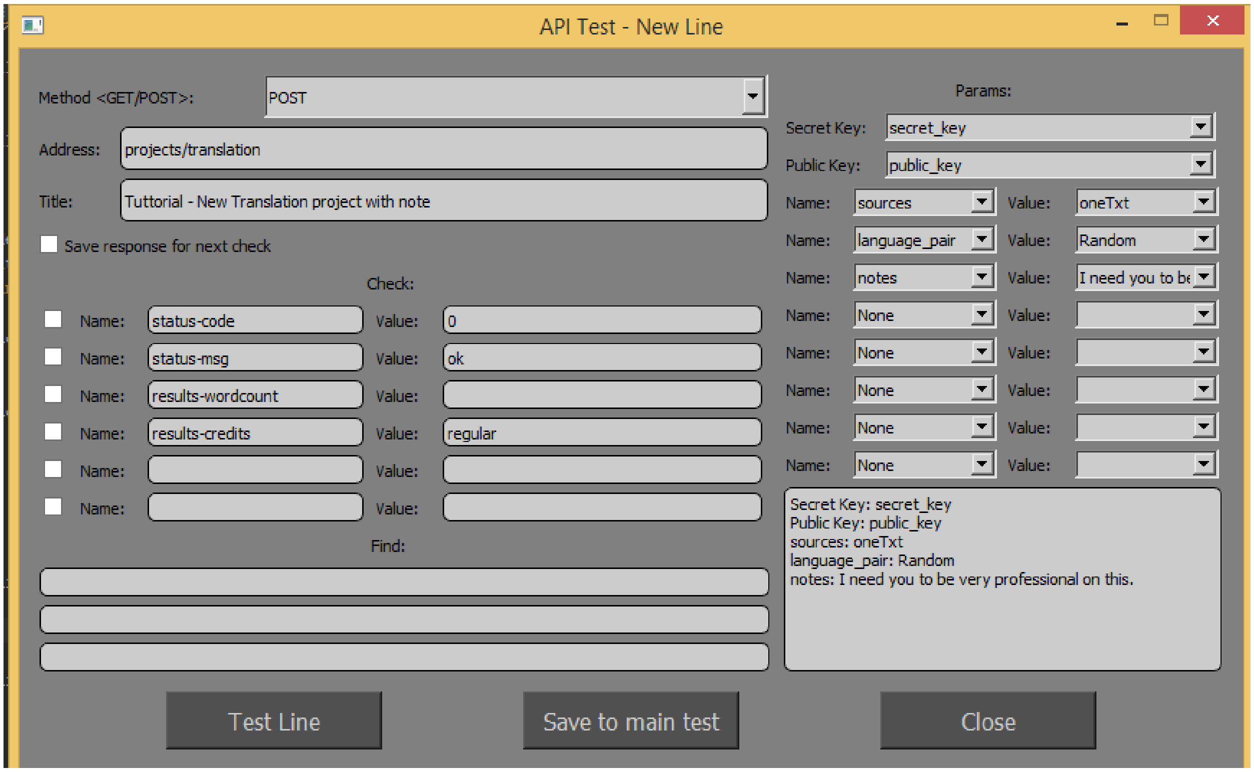
    By inputting one of those in the value field of a credits test, the program will automatically compare the response’s value to the calculated.

So if we want to do a credits check we need to input the path to the “credits:”

results-credits

And the value = “regular” (because it’s a new regular project).

17. The final line will look like that:



Method <GET/POST>: POST

Address: projects/translation

Title: Tuttorial - New Translation project with note

☐ Save response for next check

Check:

☐ Name: status-code Value: 0
☐ Name: status-msg Value: ok
☐ Name: results-wordcount Value:
☐ Name: results-credits Value: regular
☐ Name: Value:
☐ Name: Value:

Find:

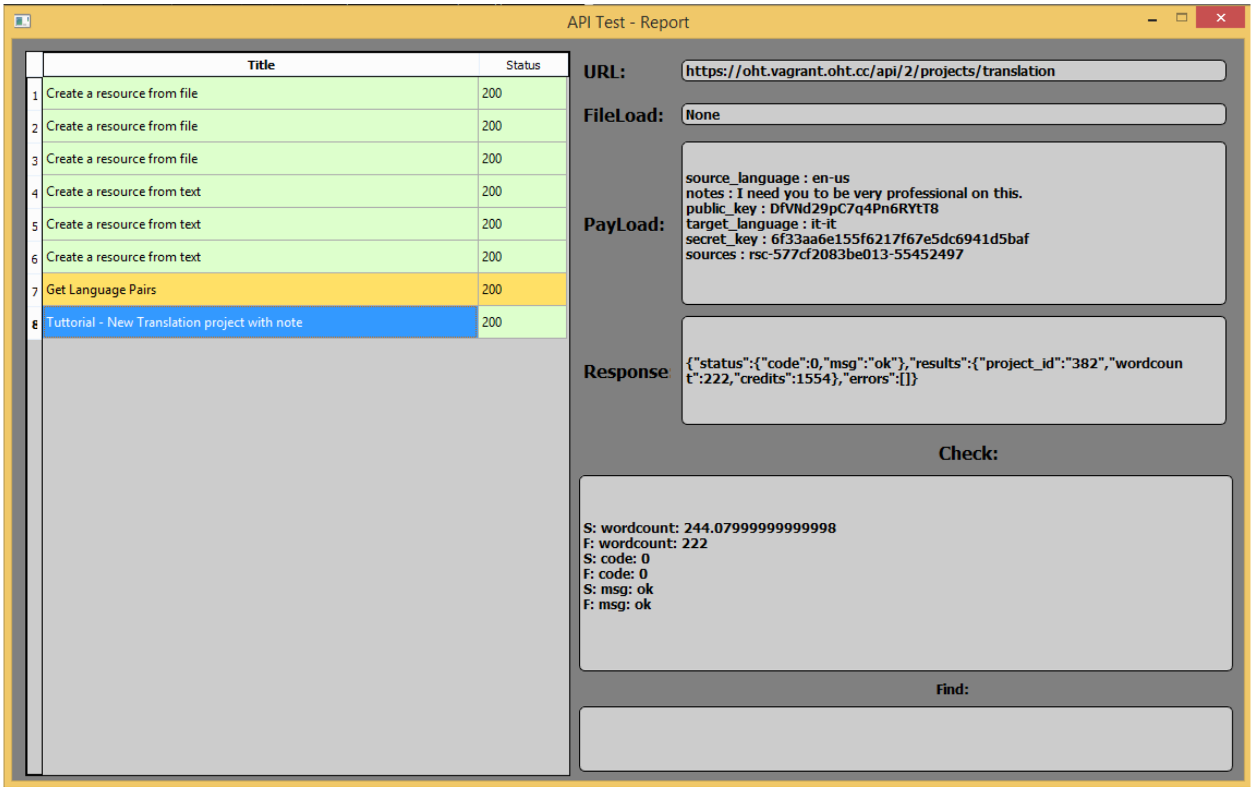
Params:

Secret Key: secret_key
Public Key: public_key
Name: sources Value: oneTxt
Name: language_pair Value: Random
Name: notes Value: I need you to be
Name: None Value:
Name: None Value:
Name: None Value:
Name: None Value:
Name: None Value:

```
Secret Key: secret_key
Public Key: public_key
sources: oneTxt
language_pair: Random
notes: I need you to be very professional on this.
```

18. Let’s try the line out:
    - Click on “Test Line”
    - Go to main window and click on “Start Test”
    - After test is finished click on “See Report”

And the report is:



	Title	Status
1	Create a resource from file	200
2	Create a resource from file	200
3	Create a resource from file	200
4	Create a resource from text	200
5	Create a resource from text	200
6	Create a resource from text	200
7	Get Language Pairs	200
8	Tuttorial - New Translation project with note	200

URL: https://oht.vagrant.oht.cc/api/2/projects/translation

FileLoad: None

PayLoad:

```
source_language : en-us
notes : I need you to be very professional on this.
public_key : DfVNd29pC7q4Pn6RYtT8
target_language : it-it
secret_key : 6f33aa6e155f6217f67e5dc6941d5baf
sources : rsc-577cf2083be013-55452497
```

Response:

```
{"status":{"code":0,"msg":"ok"},"results":{"project_id":"382","wordcoun
t":222,"credits":1554},"errors":[]}
```

Check:

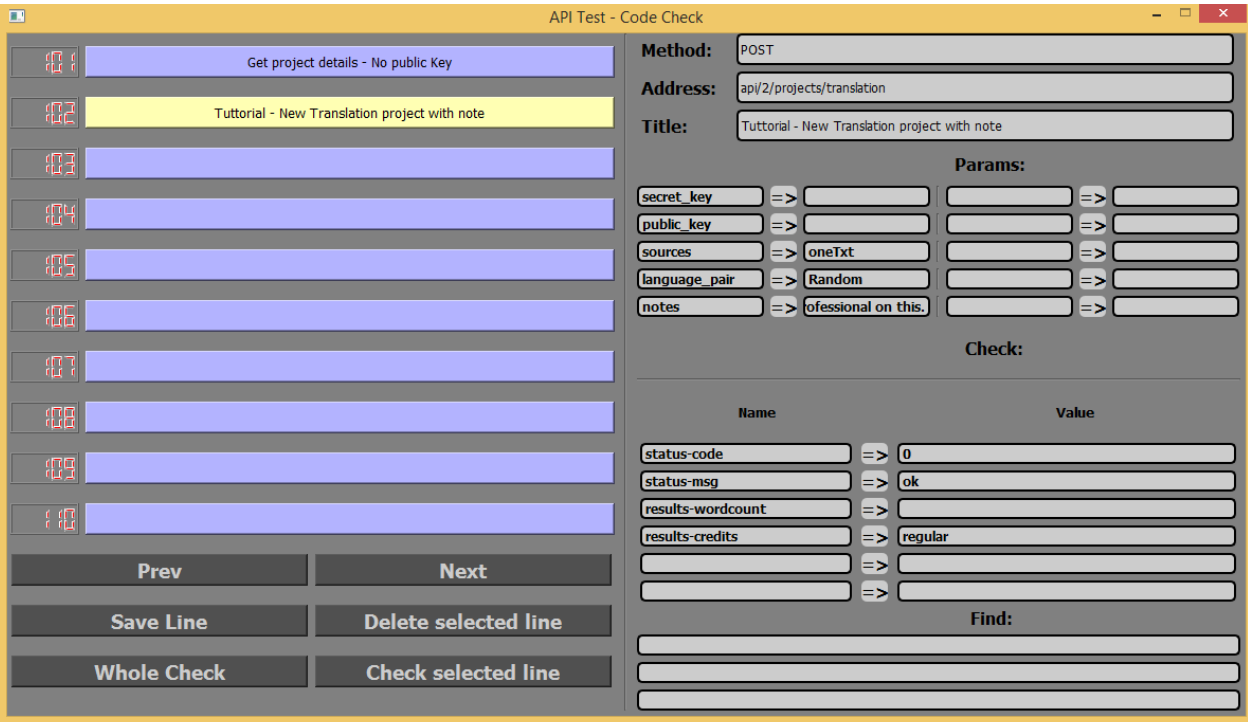
```
S: wordcount: 244.07999999999998
F: wordcount: 222
S: code: 0
F: code: 0
S: msg: ok
F: msg: ok
```

Find:

You can see all checks we made are there with the correct values. Line’s color is green and if you want you can open the project in your browser and see everything is correct there.

19. Close report window.
20. In New Line window click on “Save to main test” and close the window.
21. In main window click on “Whole check display”. In Code window click on “Next” till you reach the end of the test file.

22. And here you have it, our new line is at the end of the test file and will be executed once we start the whole test:



01 Get project details - No public Key

02 Tuttorial - New Translation project with note

Method: POST

Address: api/2/projects/translation

Title: Tuttorial - New Translation project with note

secret_key	=>			=>	
public_key	=>			=>	
sources	=>	oneTxt		=>	
language_pair	=>	Random		=>	
notes	=>	ofessional on this.		=>	

Name		Value
status-code	=>	0
status-msg	=>	ok
results-wordcount	=>	
results-credits	=>	regular
	=>	
	=>	