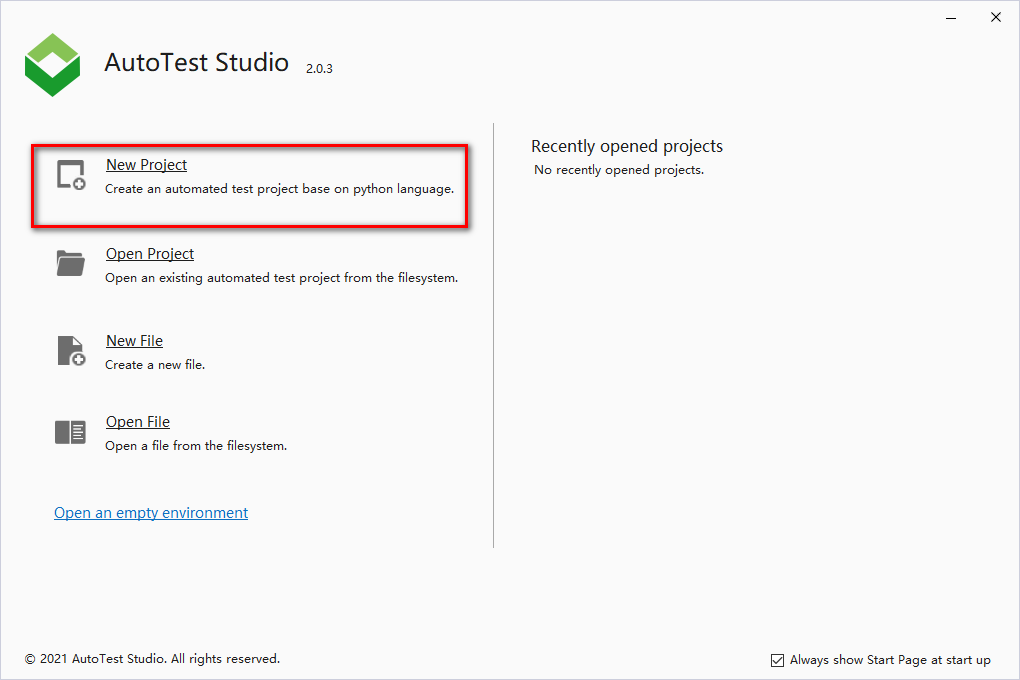
# AutoTest Studio Getting started tutorials 2: Start with a project

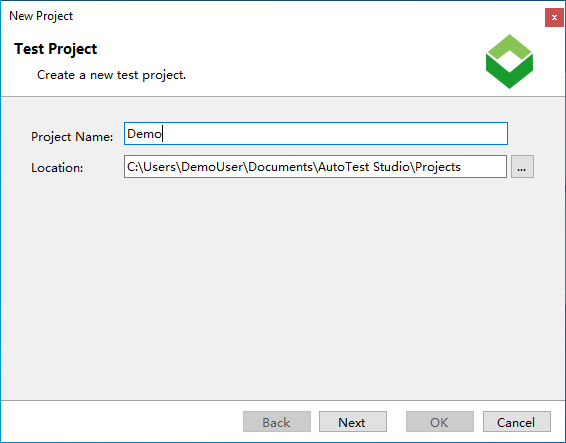
In this section, I will take a simple demo as an example to introduce how to start a test project. The scenario of the demo is to test a web service interface that queries ip address location information, and verify the interface by judging the status code after the interface is called. Is it normal? Interface prototype: http://ip-api.com/json/XXXX (XXXX is the ip address to be queried).

**New test project**

Open AutoTest Studio and select "**New Project**" on the start page to open the new project wizard.



Enter the project name and project save location in the New Project Wizard dialog box. This article takes "Demo" as an example, and the save location is the default path.



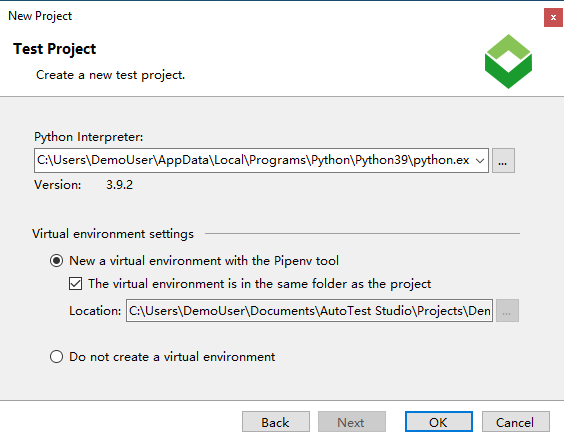
Configure the python environment of the project, "**Python Interpreter**" can select one from the pre-configured python list, or you can select a new python environment through the "**...**" button at the back.

AutoTest Studio supports running in a virtual environment. When you select "**New a virtual environment with the Pipenv Tool**", AutoTest Studio will create a virtual environment based on the selected "**Python Interpreter**". When there are multiple test projects on the same PC At the time, the virtual environment can isolate the python environment in which the project runs, so that each project has an independent python operating environment.

"**The virtual environment is in the same folder as the project**", save the configuration information of the virtual environment and the project in the same directory.

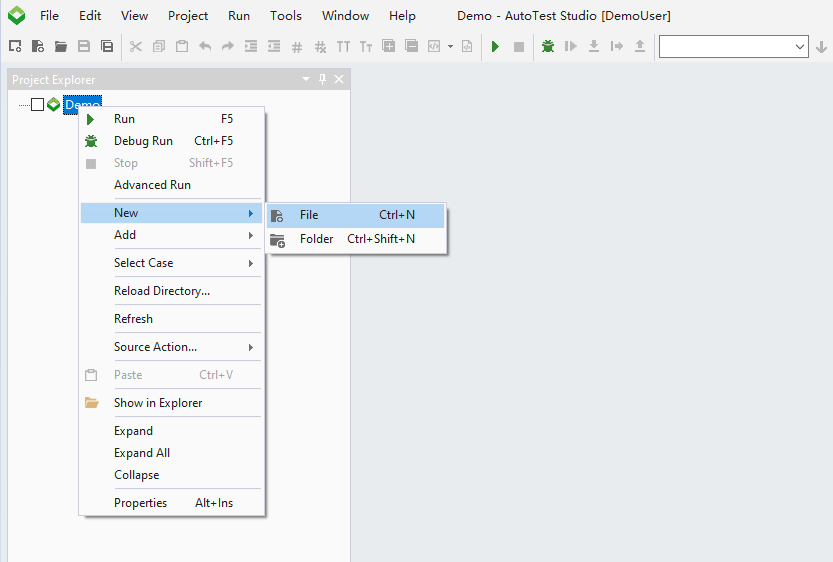
**Note:** After the Python environment configuration setting is completed, if you need to modify it later, you can also modify it in the project properties.

Click the "**OK**" button, AutoTest Studio will automatically create the project, including the initialization of the python environment.



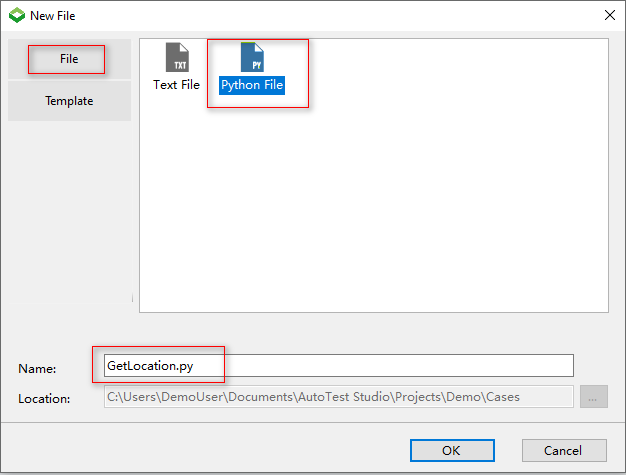
**New test case**

In the main interface of AutoTest Studio, in the right-click menu, click "**New**"-> "**File**" to pop up the test case creation wizard. It should be noted here that only test cases or files created through the right-click menu of the project directory can be associated with the current test project. And appear in the project's directory structure.



Through the creation wizard, select "**File**"->"**Python File**" to create a new test case file, fill in the name of the test case, and the file storage location cannot be changed.

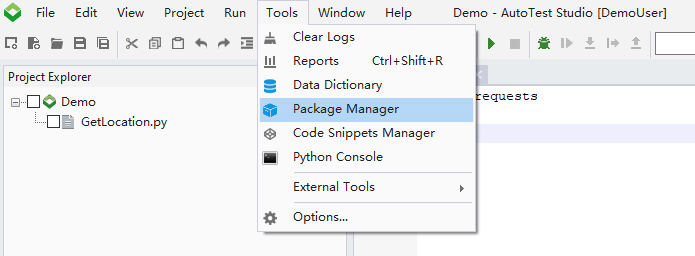
The creation of test cases can also be created through templates. AutoTest Stduio allows users to create different template files according to their needs and add them to the program.



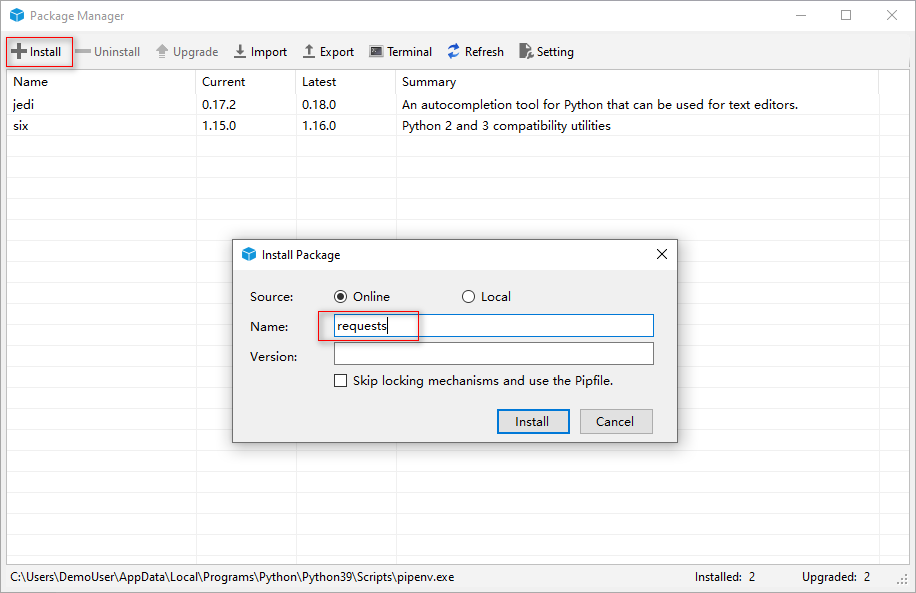
**Install the "requests" package**

Because the test cases in the demo will use the requests package of python, it is necessary to add the requests package through the python package manager tool. In the subsequent chapters, I will introduce the use of Package Manager in detail.

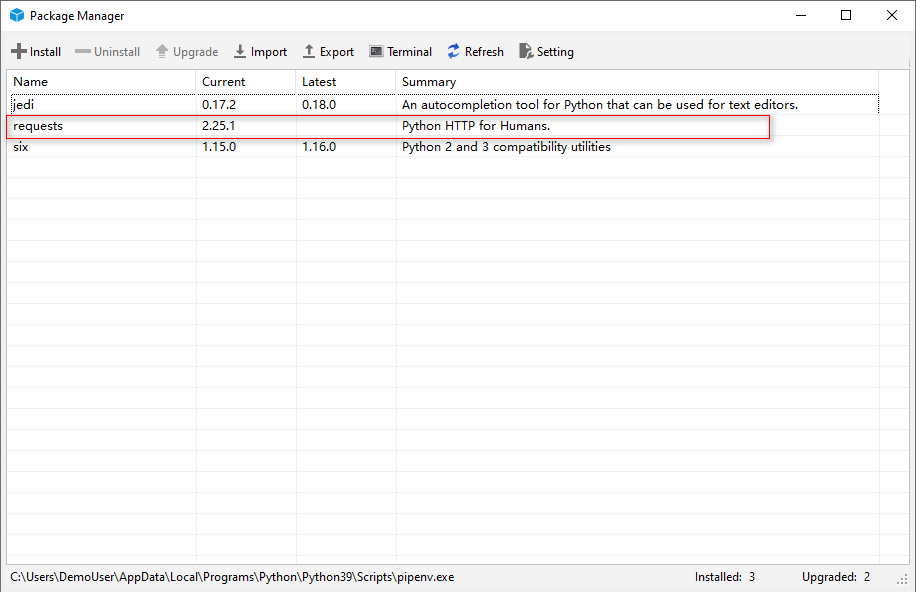
Click "**Tools**" -> "**Package Manager**" in the main menu of AutoTest Studio to open the python package manager.



In the "Package Manager", click "**Install**" in the pop-up dialog box, enter "requests", click "**Install**", "Package Manager" will automatically follow the requests package and related dependencies.



After the installation is successful, you can see the installed requests package in the list.



**Write use case code**

Enter the following code in the test case. The test code is relatively simple. You can query the ip address information of 8.8.8.8 through the interface and judge the result through AssertEqual.

import requests

from autotest import \*

SetCase("TEST-1","Get location by ip","1")

ip="8.8.8.8"

ipQueryUrl="http://ip-api.com/json/{0}".format(ip)

LogInfo("ipQueryUrl:{0}".format(ipQueryUrl))

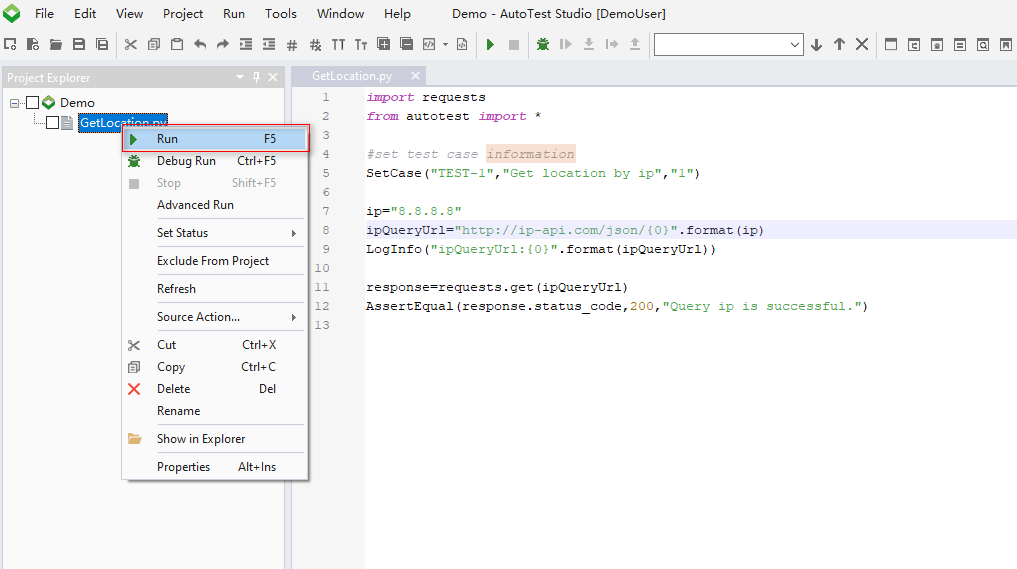
response=requests.get(ipQueryUrl)

AssertEqual(response.status\_code,200,"Query ip is successful.")

**Run test cases**

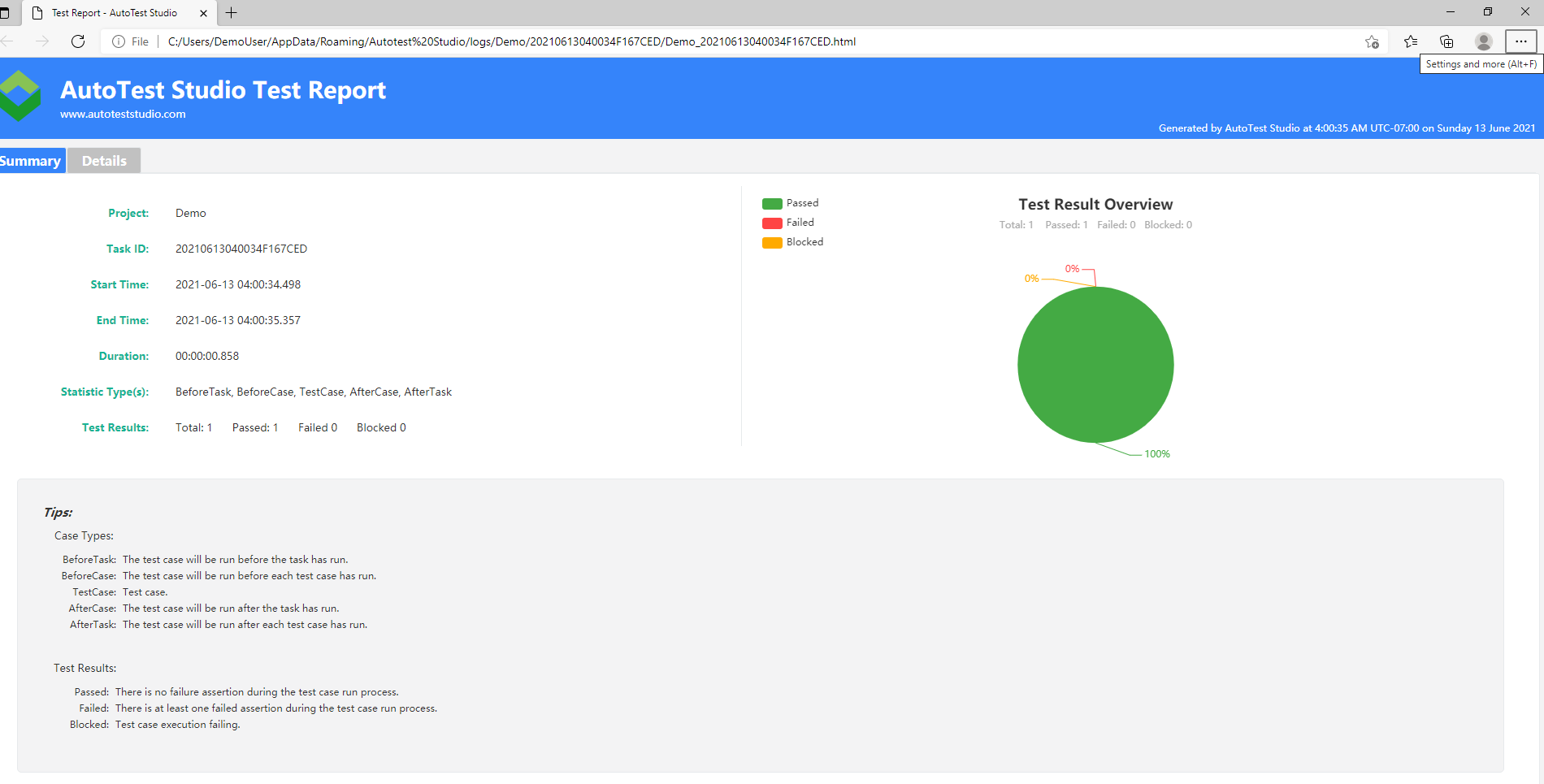
Select the use case in the project's directory structure, click "**Run**" in the right-click menu, and run the test case. After the run is completed, AutoTest Studio will automatically open the test report.

Note: To run a single test case, directly select the case and click "**Run**" in the right-click menu. If you are running multiple test cases, you need to select the test case you want to run, and click "Run" in the right-click menu of the parent directory .

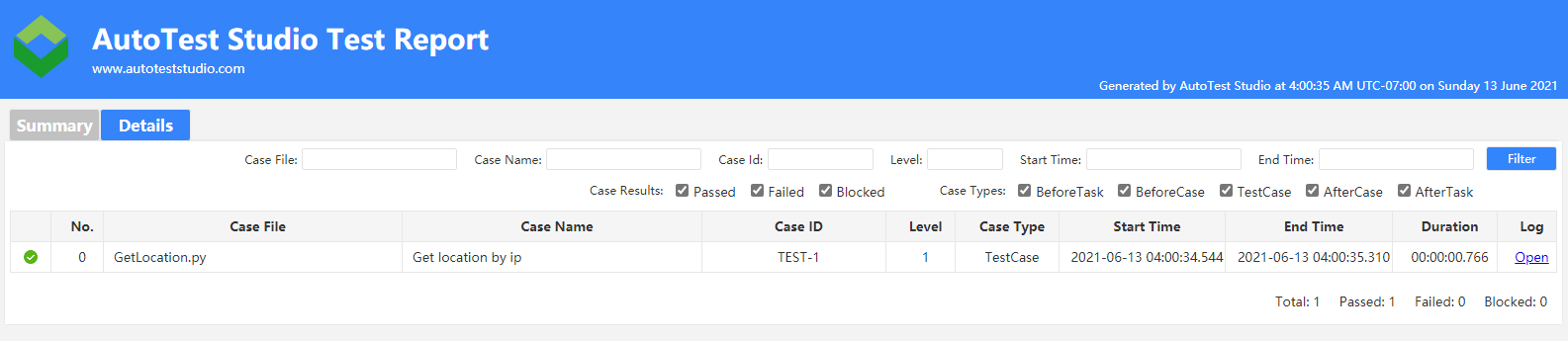


**View test reports and logs**

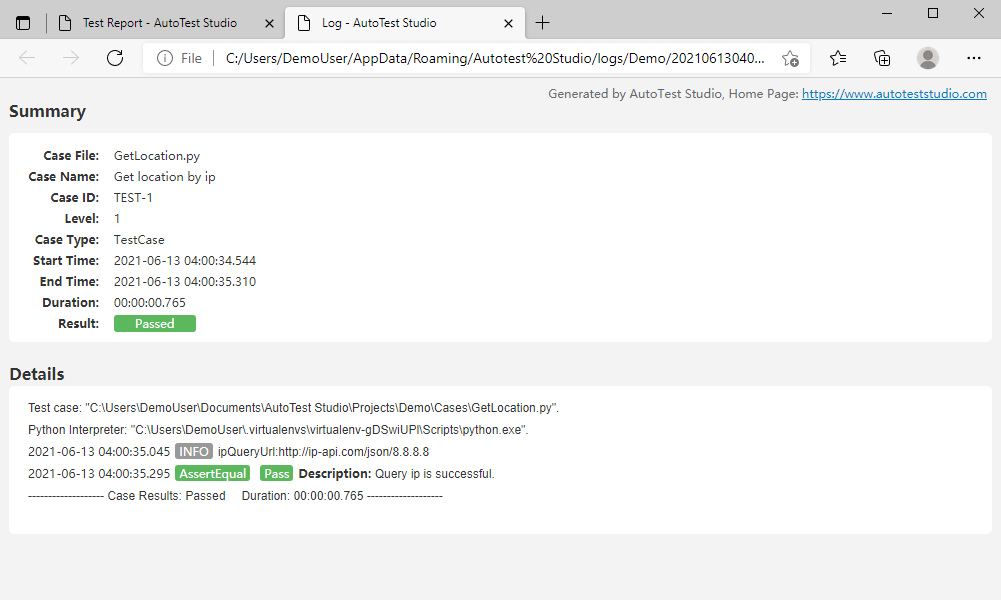
The test report includes the basic information and statistical information of the current task running.



Click "**Details**" of the test report to view the running information of the test case.



On the right side of the test case list item, click "Open" to open the log file of the test case operation.



Since then, it has been introduced how to create test tasks and test cases through AutoTest Studio, and run the test tasks.