

**VMLogin virtual multi-login**

**browser Instruction**

**Ver 1.0**

# **System Requirements**

## **Hardware requirements**

- RAM: 4GB recommended;
- 1GB available disk space
- Effective GPU

## **Supported operating systems**

Although VMLogin supports x86 (32-bit) systems, we recommend using VMLogin on x64 (64-bit) systems.

## **OS supported by VMLogin**

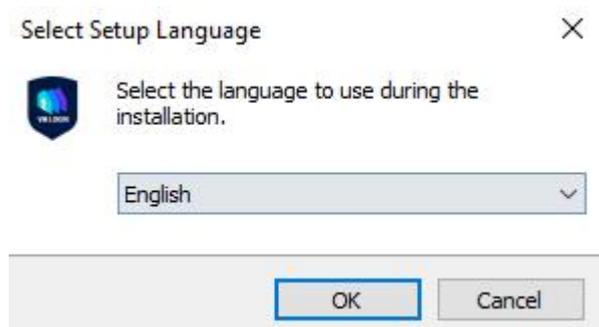
- Windows 10
- Windows Server 2016
- Windows 7
- Windows 8

## Software install

Users can download the latest software installation package from the official website: [us.vmlogin.com](http://us.vmlogin.com). The version used in this manual is 1.0.3.8.

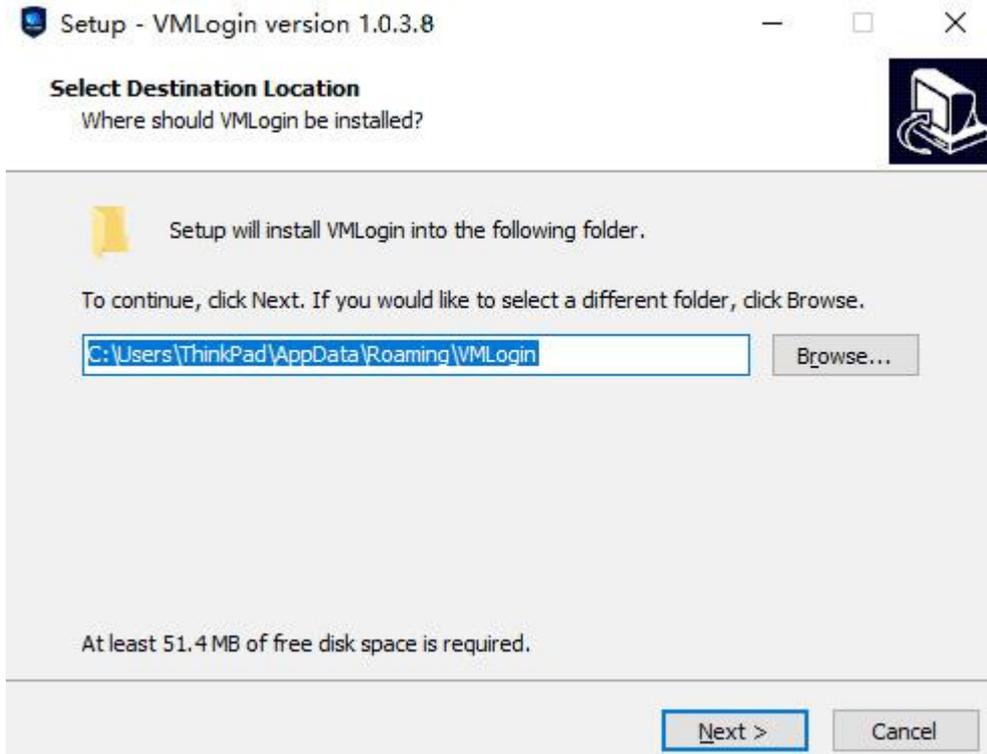


We can start the software installation package program, and the interface for selecting the installation language will be displayed:

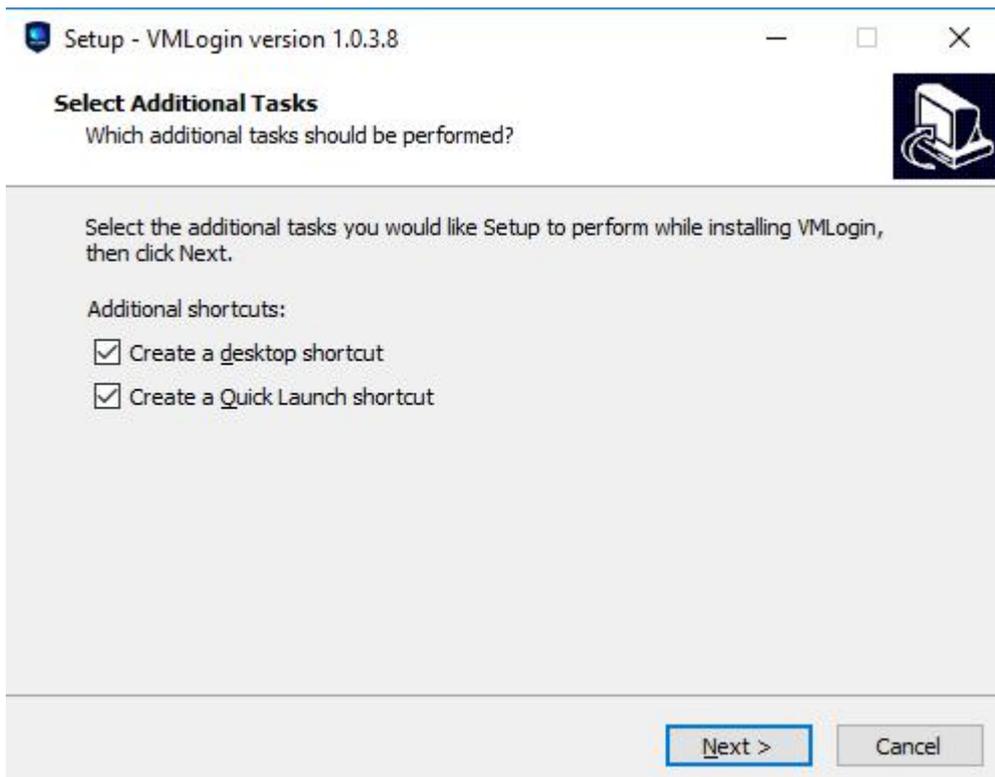


User can select the software language he likes, and can also change the language in the settings after installation. Currently supports Simplified Chinese and English.

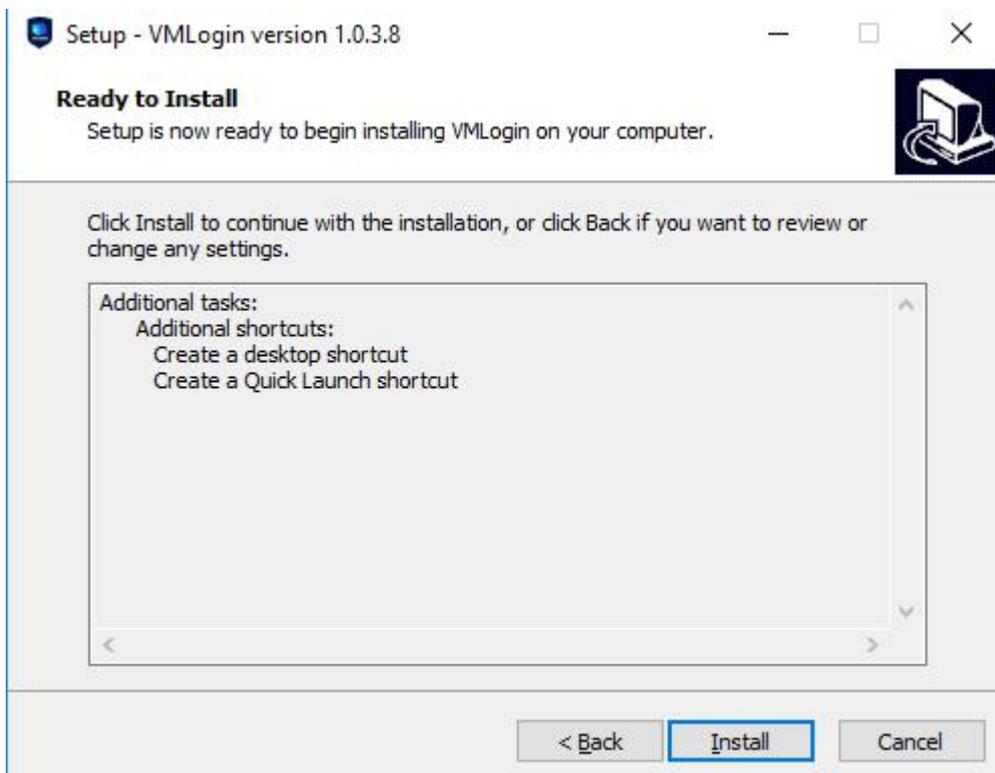
After the user select the language, enter the interface for selecting the installation directory:



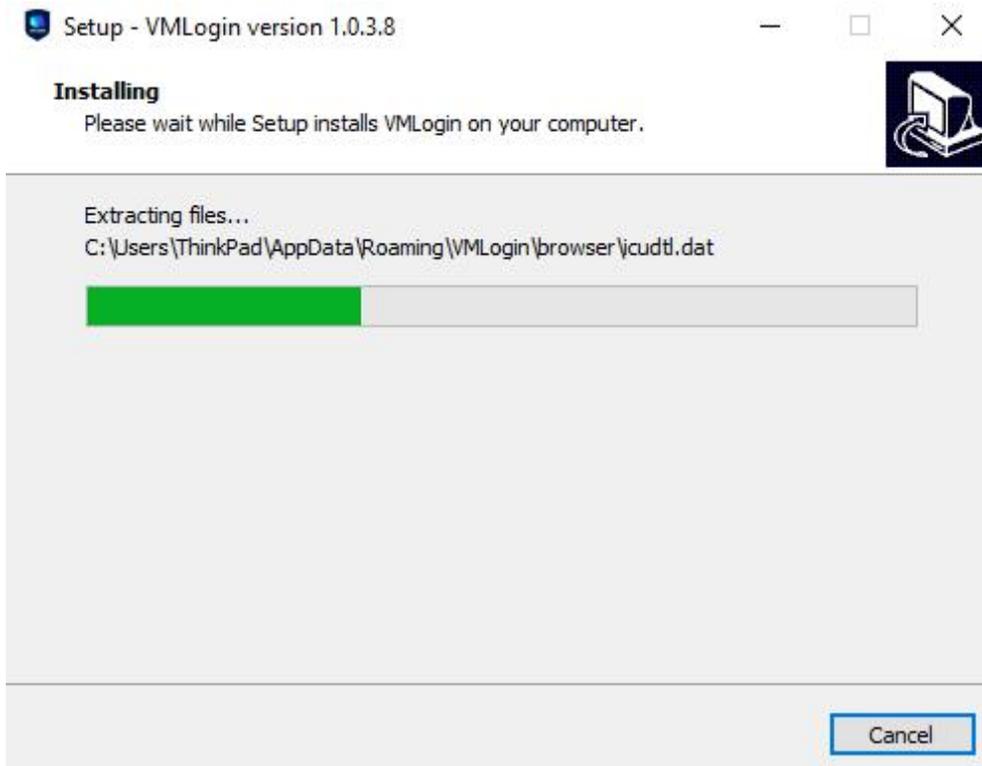
Here is generally installed by default, press the "Next" button, if the user customizes the installation directory, please pay attention to the directory permissions, the current user can edit.



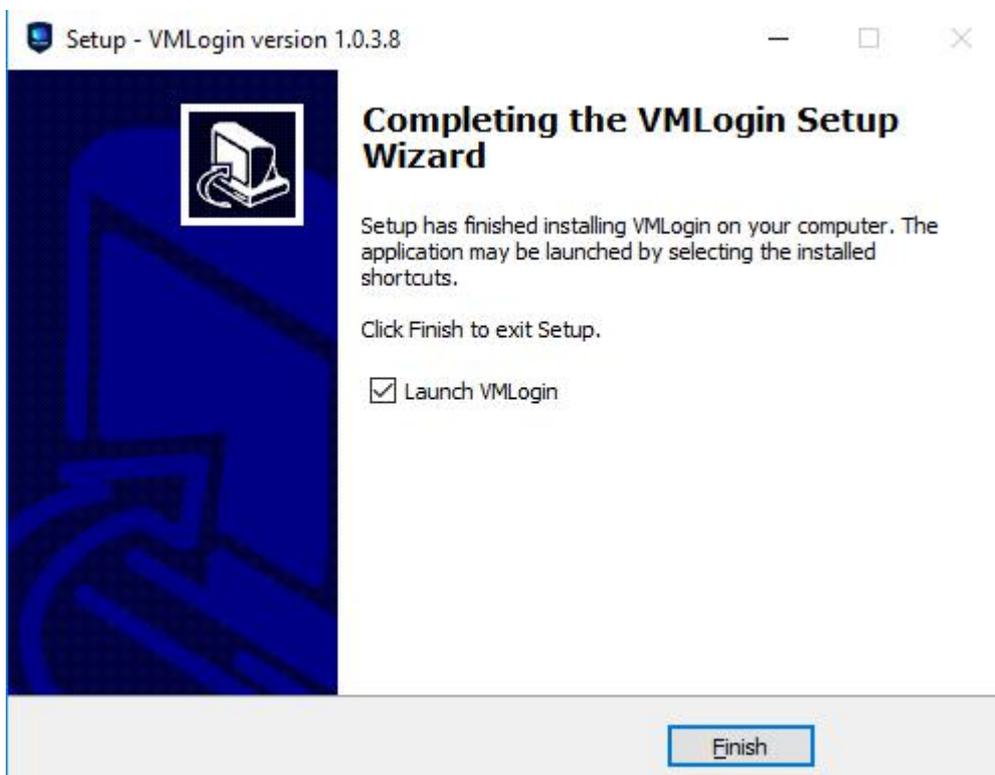
Generally, you need to create a desktop shortcut for convenient, we click the "Next" button to continue by default here.



At this point we can click the "install" button to install.



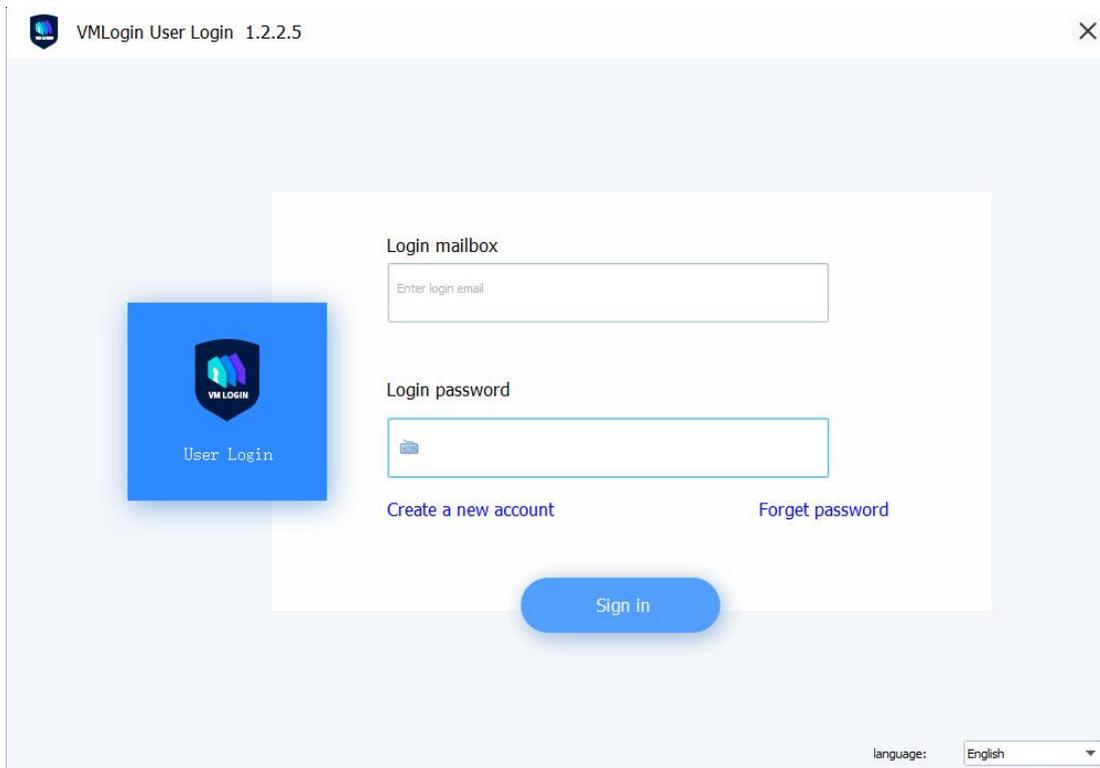
We need to wait for the installation progress to complete.



At this step, the software installation is complete.

# Software function

## 1.Start Software



When we start the software, we will see the login interface. New users need to create a new account to log in.

## 2.Create Account

VMLogin User registration

Registered mailbox

Enter email address (send verification email after registration)

Login password

Input password

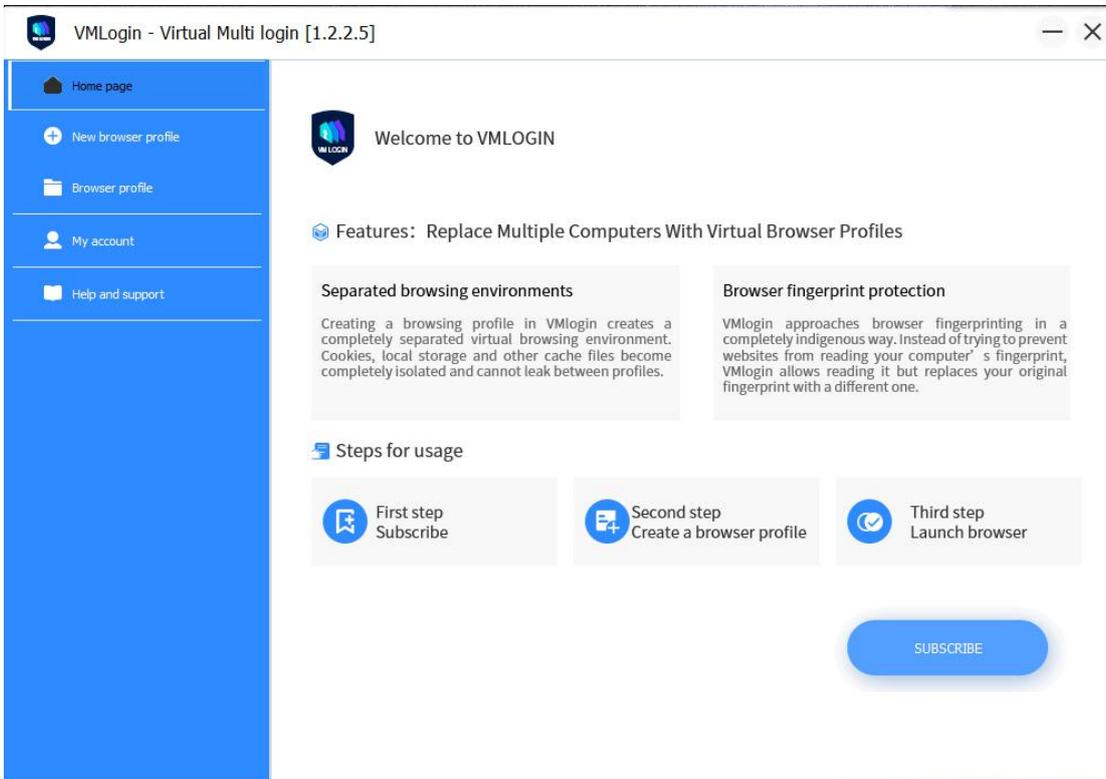
Confirm password

Confirm password

Registration

To register as a new user, you need to use email as your account, set a password, and register.

### 3.Function



After logging in, we will see the main interface of the software. The function bar on the left can be switched to enter the function interfaces related.

## A. Create browser profile

New browser profile

Basic configuration

Other configurations

Plugin fingerprint

Save profile

Close

Basic setup

Display name:

Note:

System:

WebRTC:

Public IP:   Auto detect IP

Lan IP:

Navigator parameters

User-Agent:

Resolution:

Language:

Platform:

Product:

Version:

appName:

Advanced fingerprint protection settings

Enable [font] fingerprint protection

Enable [canvas] fingerprint protection

Enable [audiocontext] fingerprint protection(noise mode)

Enable [webgl] fingerprint protection

WebGL vendor:

WebGL renderer:

Enable setting time zone based on IP

Set timezone:

Media device fingerprint settings

Number of custom media devices

Video input

Audio input

Audio output

Home page:

### Create browser profile-basic configuration

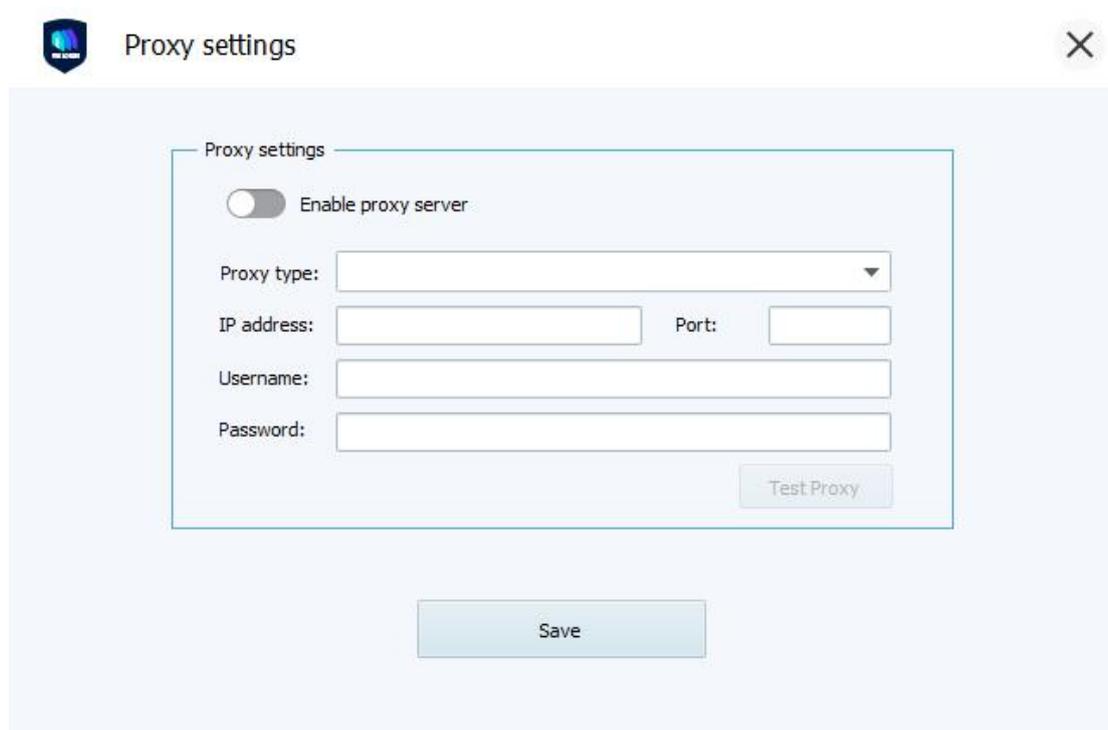
Name: This is the name of the configuration file

Remarks: You can customize the remarks of configuration files

Icon: You can choose an icon for easy identification

Operating system: Set the type of system that the browser is running

## Set up a proxy server



Proxy settings

Enable proxy server

Proxy type:

IP address:  Port:

Username:

Password:

Test Proxy

Save

Three proxy server types are supported here: http, socks4, socks5, and user authentication is supported.

## WebRtc

WebRTC is a browser plug-in that is commonly used by web applications that require fast and direct connections. WebRTC establishes the connection through the UDP protocol, so it will not be routed through the proxy server you use in the browser configuration file. Even if you use a proxy, the website can use this to obtain your real public and local IP addresses. The plugin can be used to reveal your local IP address or track media devices.

What information will the WebRTC plugin disclose?

- Public IP addresses
- Intranet IP addresses
- Number of media devices and their hash values

## **Different modes of WebRTC**

**Altered Mode-WebRTC**'s public and local IP addresses will be replaced

**Disabled Mode-WebRTC** plug-in will be disabled in the browser configuration file.

**Real mode-WebRTC** plugin will be enabled and will reveal your real IP address. If you are not connected via an agent (for example, you are connected via 4G / 5G or direct landline), you should use this mode.

Replacement mode settings

In the replacement mode, WebRTC parameters-public IP and local IP parameters, will be configured when the browser file starts, and match the IP address in the browser configuration file settings.

Effective intranet IP range

The intranet IP address must be within the valid address range, such as:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

Normally, the Group C (the third number) of residential users is 0, and the Group D (the fourth number) can be any number. Corporate online users may have different numbers in Group C and Group D, instead of 0.

Some examples of intranet IP addresses:

- 192.168.0.21
- 192.168.0.23
- 172.16.0.168

Some typical local IP cases of enterprise users.

- 192.168.31.123
- 172.16.11.22
- 10.0.12.192

### **Navigator parameters**

JS.Navigator is a set of Javascript objects, which stores various parameters and their values, used to describe the details of the computer used. The browser can freely access all JS.Navigator object parameters. Because of their uniqueness, especially when the components are combined, the website can use these parameters to identify and track user fingerprints.

The website also analyzes the consistency of these settings to reveal fingerprint changes. Such analysis may expose the use of browser fingerprint randomizers.

## User-Agent

User-Agent is a native short string of browsers. By reading the user agent string, the website can recognize the version of your browser and operating system.

Below is an example of user agent value.

**Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.2785.8 Safari/537.36**

In this case, the website will speculate that the user is using Windows 8.1 and Chrome 62. "NT 6.3" is a different version of Windows. You can view other published versions in Wikipedia article.

When creating a browser configuration file, following your selection of operating system filters on the overview page, user agent values are obtained from the VMLogin fingerprint database. You can view the user agent value of the browser configuration file on the navigation bar page.

If you decide to manually set the user agent value in VMLogin, make sure that the value is consistent with the platform value (Platform). Differences in user agent values and platforms will be serious errors.

## **Platform**

The property of the platform is an object parameter of a Navigator, which can be used to indicate the compiler platform of the browser.

When creating a new browser file, the platform value and the user agent are obtained at the same time. Both values are affected by the operating system filter of the overview page. You can also manually set this value in the navigation bar (Navigator) page.

If you want to manually set the platform in VMLogin, make sure it matches the user agent value. The difference between the user agent value and the platform value is a very dangerous signal.

Available platform values for desktop browsers:

- Linux i686
- Linux armv7l
- MacIntel
- Win64
- Win32

Platform values available for mobile browsers:

- iPhone
- iPod
- iPad
- Android

You can get a complete list of available Navigator.Platform values from the Stackoverflow article.

## **Screen Resolution**

Analyzing the screen resolution value is a common method for browser fingerprint recognition. The website will also analyze the difference between the screen resolution reflected by the browser and the actual available screen area size. This difference can detect whether users are using online privacy tools, such as browser privacy add-ons.

In VMLogin, the screen resolution is extracted from the fingerprint database that generates the browser configuration file. You can also manually change this parameter by selecting the resolution you need in the list of commonly used resolution paths or by manually entering values. In addition, when re-acquiring the fingerprint, the screen resolution drop-down list and screen resolution you receive will be adjusted accordingly according to the operating system you selected on the overview page.

VMLogin will run the maximum browser resolution set in the browser configuration file. Maximizing the browser window is a typical behavior that is very common to most users. We do not recommend that you use it to maximize it. We also do not recommend using a resolution larger than your real screen in the browser configuration file, because the website will find that you are not using the maximized window.

When sharing a browser profile to others, we recommend that the resolution of the browser profile should not exceed the smallest screen used by your team. For example, your device is 4K, and your colleague uses a FullHD (1920x1080) display. At this time, we recommend that you keep the resolution of the browser configuration file at 1920x1080 or less. Otherwise, when your team opens the browser profile on different devices, the actual size of the window may be different.

## **Language settings**

The language setting of the Navigator object can help the website recognize your preferred language. Based on this value, the website will adjust the language in which the content is presented to you. Like any other Navigator object value, it can also be used for browser fingerprint recognition.

When you create a new browser file, it will default to the language most commonly used by people. VMLogin does not randomly obtain this value from the fingerprint database, because it will cause a serious mismatch between the browser language and the IP location. For example, you prefer Filipino when you are in Germany.

## **Enable [font] fingerprint protection**

Font fingerprint recognition is a method of identity verification. Based on the type of font used by the user and the way the font is drawn in the browser, the website can perform fingerprint tracking. Generally, there are two ways to use fonts in website fingerprint recognition

- Enumerate font list
- Fingerprint recognition based on font measurement

You can check how these methods are applied from [Browserleaks.com](https://www.browserleaks.com).

### Enumerate font list

The most common way to collect a list of words installed on an electronic device is CSS self-testing. In short, this method gets your font list by measuring the width of a phrase displayed in a specific font on the browser. If the widths match, it means that you have installed this font. If it does not match, it is presumed that this font is not installed.

By cyclically detecting the list and width of fonts that may be installed, the website can accurately snoop which words are installed on a device.

VMLogin uses a special algorithm to counter this detection method, allowing you to control the fonts that can be enumerated by the website for anti-tracking.

## **Enable hardware fingerprint [Canvas] protection**

Canvas is an HTML5 API for drawing 2D images and animations on web pages. In addition to the above functions, Canvas can also be used as an additional entropy for browser fingerprint recognition. According to a study by Englehardt and Narayanan at Princeton University (2016), more than 5% of websites use Canvas for fingerprint recognition.

In summary, Canvas implements fingerprint recognition by instructing the browser to draw a hidden Canvas image. The drawing result of this picture is slightly different on different machines; but if the machines are the same, the image is also the same. After the image is drawn, it is converted into a hash string, which is further used for additional entropy for authentication.

VMLogin provides three different operation modes to control the Canvas fingerprint of browser files: noise mode, off mode and block mode.

### **Noise mode**

When the website requests to read the Canvas function through the browser, the Canvas masking algorithm in the noise mode will intercept it halfway and add a random but consistent noise to the readout. In order to better understand its working principle, we can compare it to a "voice corrector".

When you use a voice modifier with a specific preset, it will change your voice, making it very different from the original voice, but this change will remain consistent over time.

Due to the random noise added to the reading, if the website uses data analysis technology, you will find that the fingerprint is 100% unique.

Your Fingerprint :

Signature	✓ 47327046
Uniqueness	100% (0 of 258561 user agents have the same signature)

## Off Mode

After setting Canvas to off mode, the website will get the real Canvas fingerprint of your device.

Setting Canvas to "off mode" is advantageous in certain situations, such as when the site has a poor response to 100% uniqueness or detects that Canvas is blocked.

caution! In the real environment, the hash value of the Canvas fingerprint is not unique, because there are the same copies of your device around the world. So if you show a real Canvas fingerprint, you will only be divided into users who use the same hardware. In addition, by changing other fingerprints, you can increase the entropy of the website's view of your browser profile as a separate identity.

## **Block mode**

Block mode completely prohibits websites from reading Canvas. When the website tries to read the Canvas that has been set to blocked mode from the browser configuration file, the returned value will be empty.

The way this situation is handled depends entirely on the site's own trade-offs. However, in the case of a browser error in the process of retrieving Canvas object data, such an event may also occur to users who have not concentrated on hiding their Canvas fingerprints.

Open the browser configuration file on multiple computers.

Caution! If you create a browser profile that sets Canvas to noise mode and open it on multiple devices with different hardware, the website will know that the Canvas hash value is not continuous when running on multiple platforms . Although the added noise is continuous, it only acts as a filter on the running equipment. Therefore, if the device is changed, the readout will also change.

The same browser configuration file is opened on two different devices.

Although the noise in this browser configuration file is persistent, the Canvas readout is still different.

If you need to get constant readout on multiple devices, you can try the following solutions:

With the hardware fingerprint set to noise mode, run VMLogin on the same virtual machine (VM) or virtual private server (VPS). Since these devices are

set in the same way, the Canvas fingerprint after adding noise will be consistent on multiple devices.

Run VMLogin on the same PC model with the same hardware, drivers, and operating system. Since these devices have the same hardware settings, the masked system fingerprint will remain the same on multiple devices.

### **Enable hardware fingerprint [AudioContext] protection**

The AudioContext fingerprint (also called "audio fingerprint") is a hash-derived value of the device's audio stack. It works as follows. Based on your audio settings and hardware, the website requires your browser to simulate the way of playing audio files as a sine function. This sine function is converted into a hash function and sent to the server as additional entropy in the fingerprint identification of the browser.

In VMLogin, you can control the reading of the AudioContext by adding random continuous noise, or allow the website to obtain the real audio fingerprint of your device.

### **Noise mode**

By turning on the noise mode in the AudioContext area, VMLogin will modify the audio heap at the browser level to generate a unique audio fingerprint. Since the audio stack is modified by random values, if data analysis is applied, the website will find that the fingerprint is 100% unique.

## **Off mode**

After setting the **Audio Context** shield to off mode, the website will get the real audio fingerprint of your device.

In some cases, it may be advantageous to set the mode to off, especially when the website responds poorly to **100%** unique **Audio Context** readout.

Caution! In the real world, the hash of the audio fingerprint is not unique, because the same copy as your device and audio stack exists around the world. So if you show a true audio fingerprint, it will only be divided into user segments using the same audio hardware. In addition, by changing other fingerprints, you can increase the entropy of the website's view of your browser profile as a separate identity.

## **Enable hardware fingerprint [WebGL] protection**

WebGL is a JavaScript browser API for rendering 3D images on web pages.

The website can use WebGL to identify your device fingerprint. Generally, a website can do this in two ways:

WebGL Report-The complete WebGL browser report form is available and can be tested. In some cases, it will be converted into a hash value for faster analysis.

WebGL images-hidden 3D images rendered and converted to hashes.

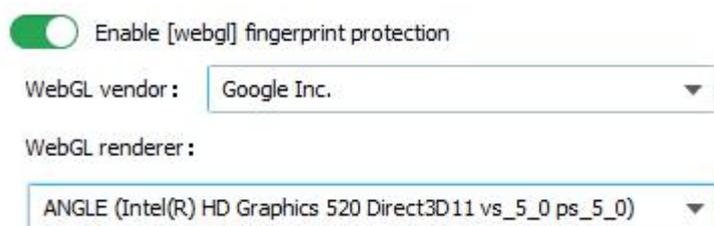
Because the final result depends on the hardware device that performs the calculation, this method generates unique values for different combinations of devices and their drivers. This method generates unique values for different device combinations and drivers.

You can check the website through Browserleaks test to see what information the website can get through this API.

### WebGL metadata masking

When you set WebGL to noise mode, WebGL metadata will be masked by VMLogin. This is an old mechanism, and we will improve this feature by hiding WebGL metadata and images separately.

When metadata masking is enabled, VMLogin will change WebGL vendor and renderer parameters based on the values obtained from the fingerprint database.



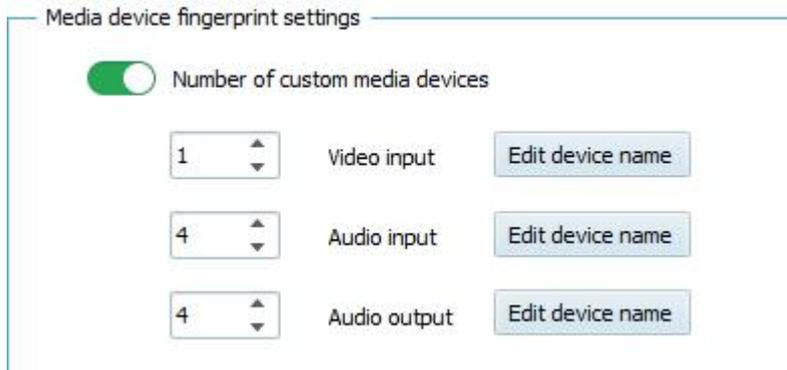
### Off mode

After setting WebGL shield to off mode, the website will get the real WebGL report and image hash value of your device.

Setting it to off mode is beneficial in certain situations, such as when the site has a 100% unique response or detects that WebGL readout is blocked.

These situations have a poor response.

## Media device fingerprint settings



The screenshot shows a settings panel titled "Media device fingerprint settings". At the top, there is a toggle switch labeled "Number of custom media devices" which is turned on. Below this, there are three rows of settings:

Device Type	Count	Action
Video input	1	Edit device name
Audio input	4	Edit device name
Audio output	4	Edit device name

WebRTC is a browser plug-in that, through a direct P2P connection, prompts audio and video calls within web pages, eliminating the need to install additional plug-ins or other local applications. To make this plugin work, WebRTC will connect to your media devices such as microphones, cameras, and headphones. Websites can use this tracking mechanism in two ways:

- Device enumeration
- Media device ID tracking

You can check these two authentication methods on the Browser leaks test website.

### Device enumeration

This method relies on retrieving the complete list of microphones, cameras and headphones that the user has installed to work. Although this number alone is not sufficient to target users clearly, it can still play a role.

In VMLogin, you can control the number of different media devices in the browser configuration file as needed.

You can change the parameters within the following range:

- Video input (the number of webcams): 0-1
- Audio input (number of microphones): 0-4
- Audio output (number of speakers): 0-4

In theory, the number of devices per user can exceed the above range.

However, because this situation is not common, we set these numbers within the range normally used by online users.

### Media device ID

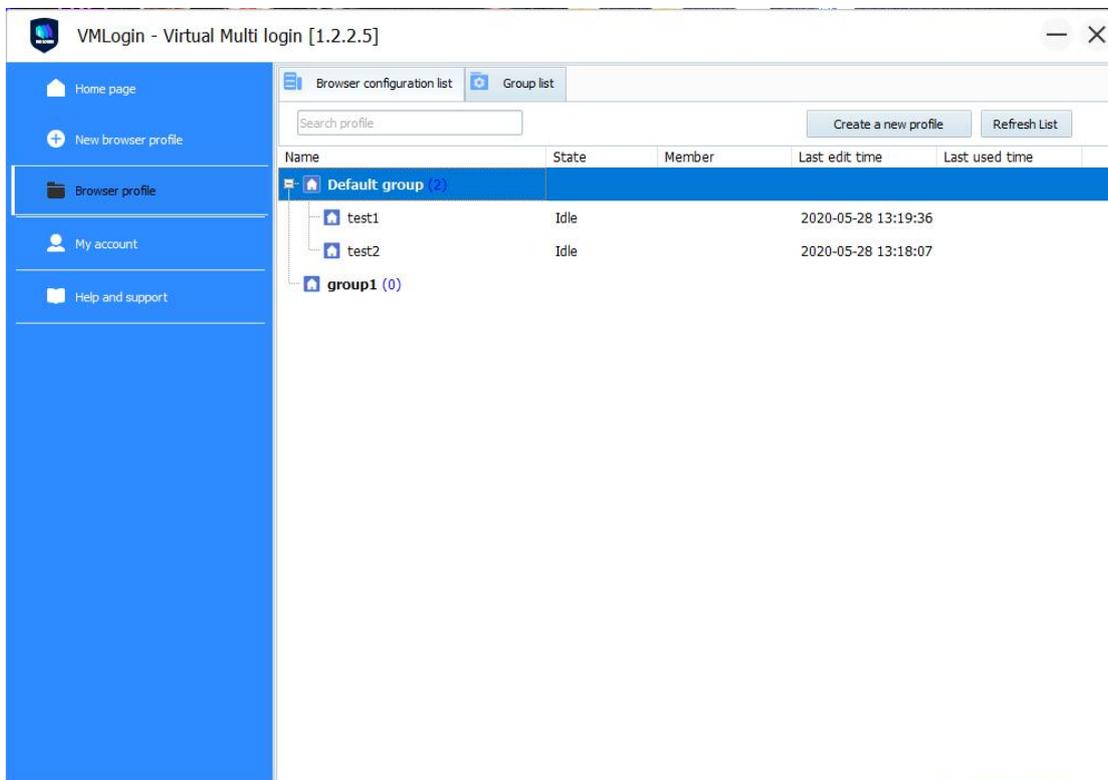
In order for WebRTC to work properly, the website not only needs to know the number and type of devices you have. In order to establish a perfect real-time communication, a unique device identifier is also necessary. You can think of your device address. Of course, the browser will not allow the website to know the full model name of your device, they will be replaced with a hash value, which is the device ID. At the same time, the website can also use these values for user identification.

Since the media device ID is unique to each user, it is a particularly effective technology in browser fingerprint recognition.

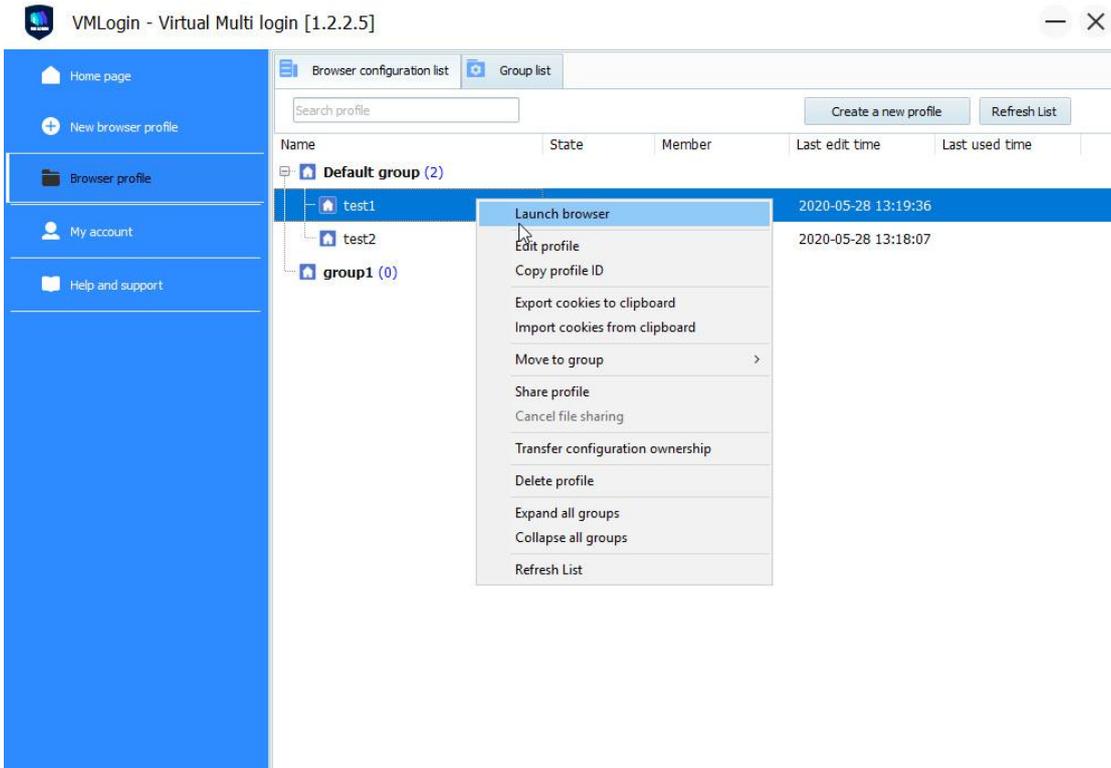
In VMLogin, when this function is enabled, the real device ID of each device will be masked.

If you turn off media device masking, your real device ID will be obtained by the website. Even if you turn on WebRTC's IP masking, this will still happen. Default homepage: Every time you start the configured browser, it will automatically open this URL.

## B. Browser configuration file management

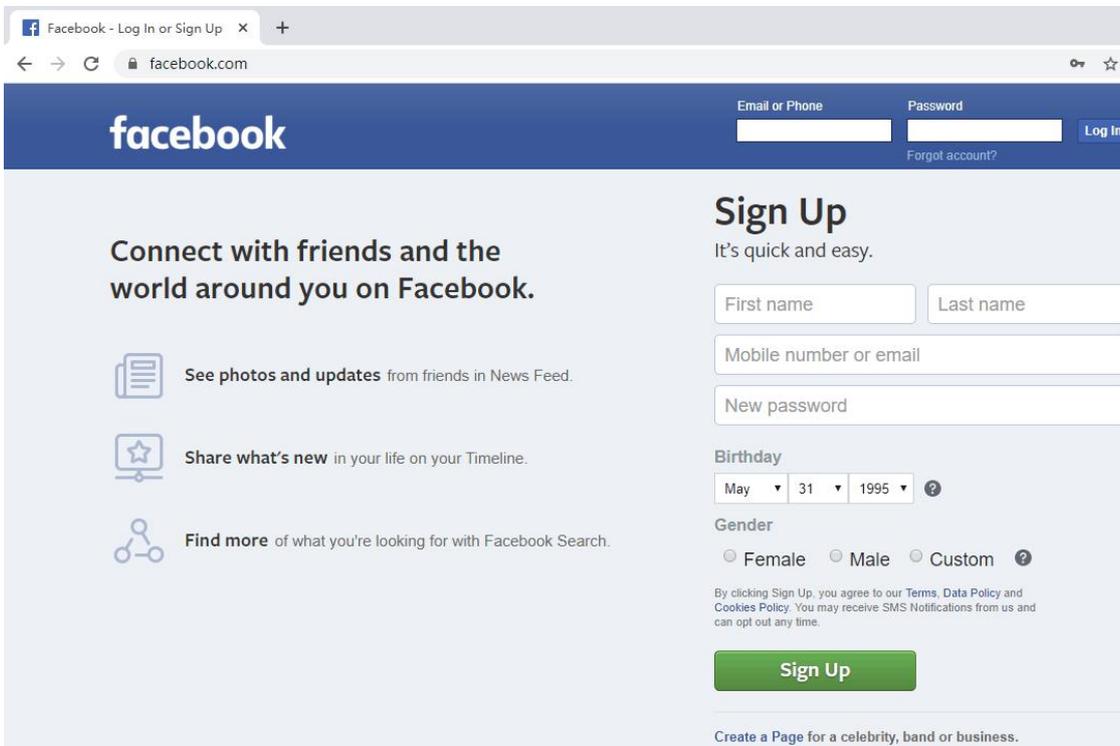


For the list of browser configuration files, you can use the right-click menu to perform related operations.



In the right-click menu, you can:

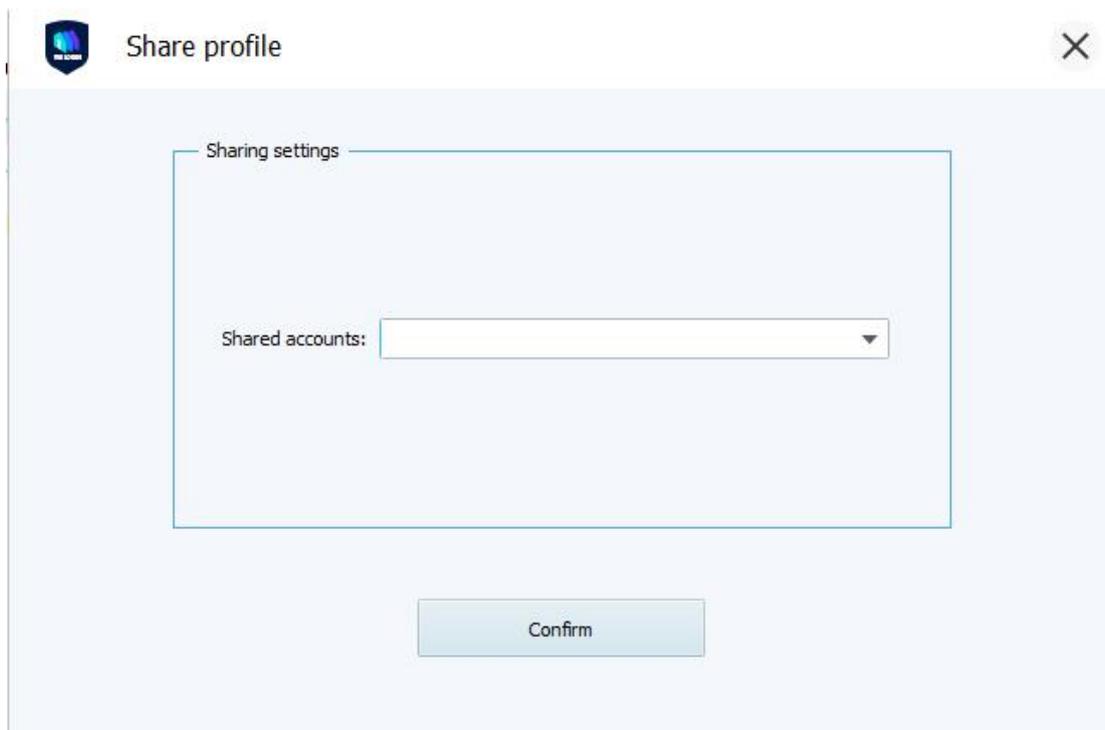
Start browser: start the browser corresponding to the configuration file



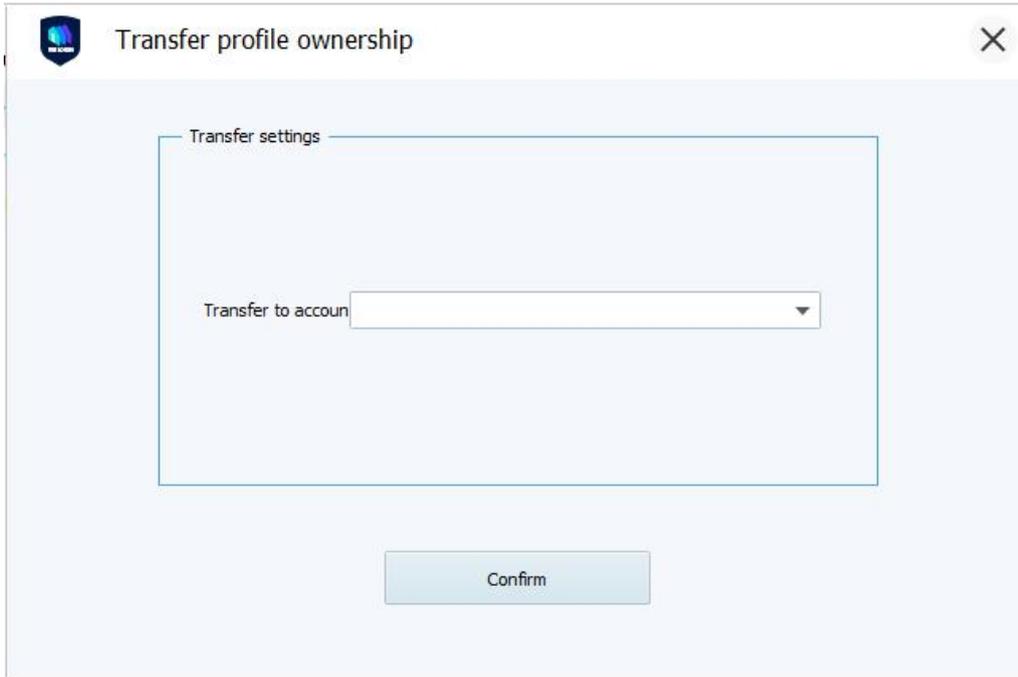
Edit configuration file: modify the browser configuration.

Move configuration file group: You can move the configuration file to the specified group.

Share configuration file: You can share the configuration file to other users.

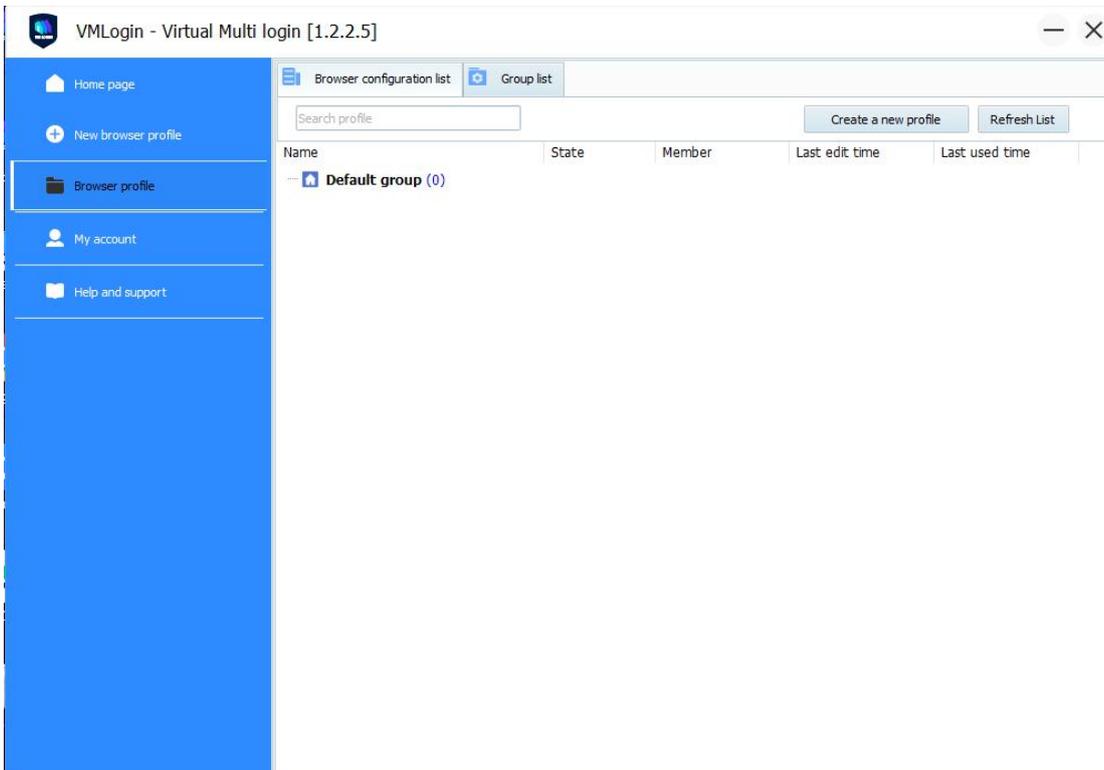


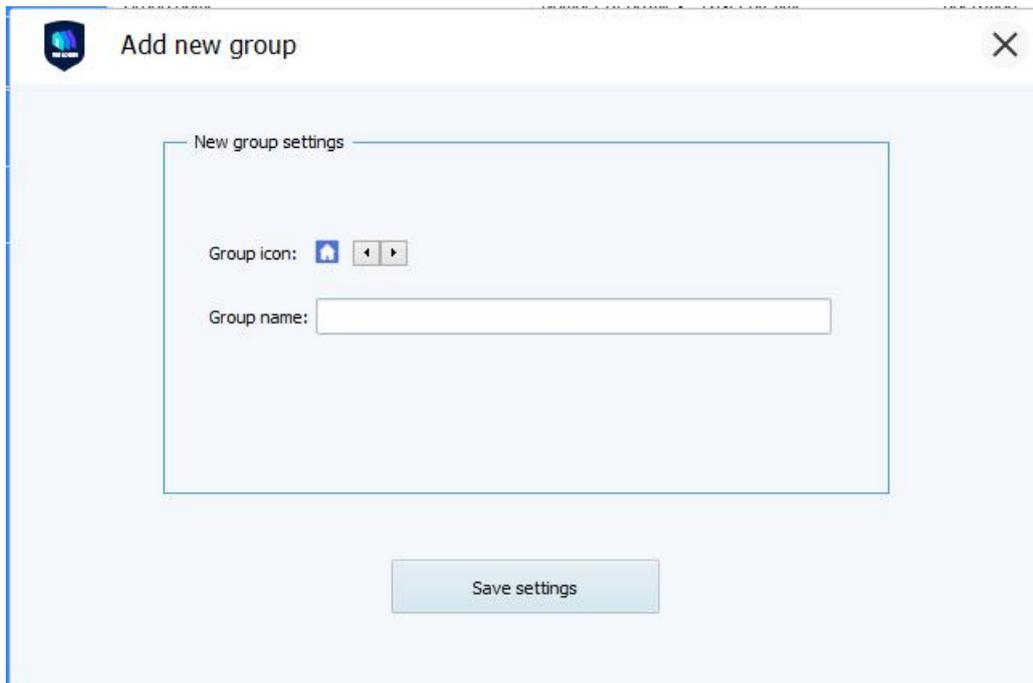
Transfer configuration file ownership: you can transfer the configuration file to other users



Delete configuration file: Delete the browser configuration file from your account

## Add group





The group list can be managed, and new group names can be added at the same time.

## Cookie import

Browser cookies, also known as HTTP cookies, are used to store user data and data stored in the user's local terminal. One of the main functions of cookies is user authentication. By using the cookie import function in VMLogin, you can import the cookies you exported from the regular browser in the launched browser.

## How to import cookies in VMLogin

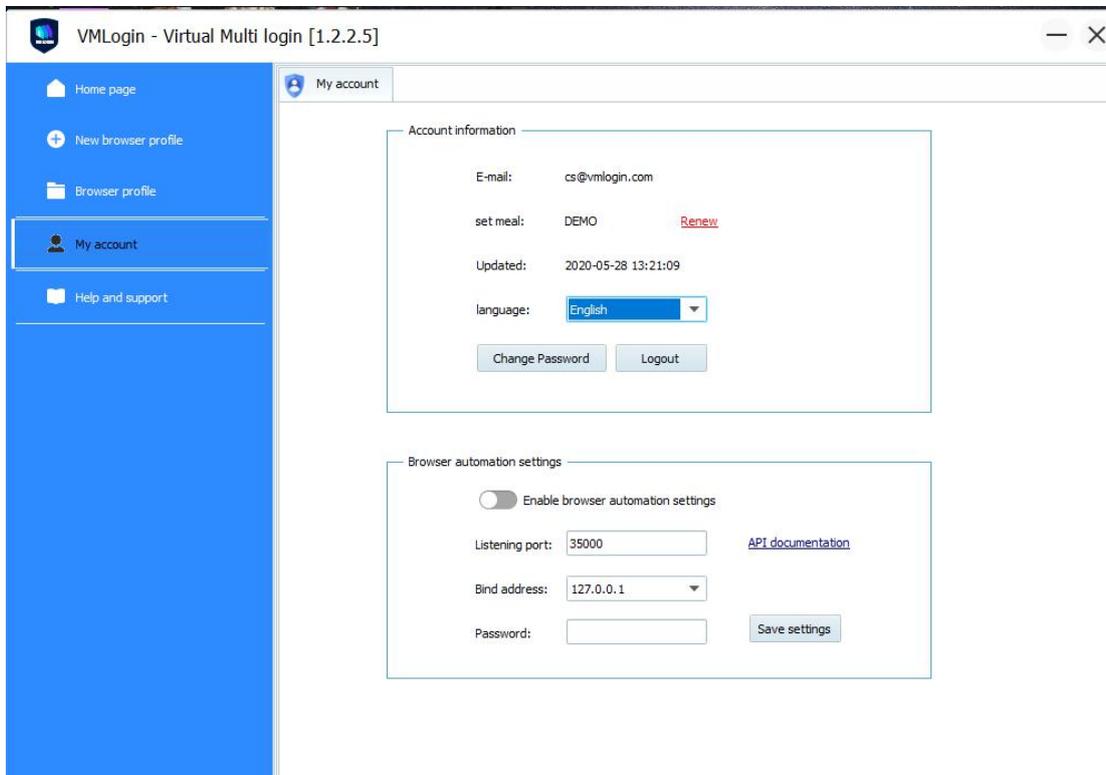
Cookies in JSON and Netscape formats can be imported in the launched browser.

Before importing cookies in JSON format, you need to use a JSON validator to check whether the file format is correct.

Sample cookie format:

```
[{"name": "xs", "path": "/", "value": "2:cDMG:2:1582872010:-1:-1", "domain": ".vmlogin.com", "secure": true, "expires": 1589094326, "session": false, "hostOnly": false, "httpOnly": false}, {"name": "fr", "path": "/", "value": "3v1si8FPVJV.AAWWjnhtd", "domain": ".vmlogin.com", "secure": true, "expires": 1589094326, "session": false, "hostOnly": false, "httpOnly": false}]
```

## C. My account



You can view your account information in the My Account section, and you can configure browser automation settings.

Browser automation allows you to automate tasks in VMLogin's browser configuration file. From creating simple automated scripts to complex web crawlers, you can search, collect and interact with web data.

VMLogin browser automation is based on Selenium WebDriver.

Normally, if you run Selenium code, you will first connect to the Chrome driver and then set up the functions you need. When using VMLogin with Selenium code, you don't need to do this. You will use the Remote Web Driver program to connect to the Vmlogin application or a browser

configuration file through a local port, set the required functions, and execute Selenium commands in a predefined browser configuration file.

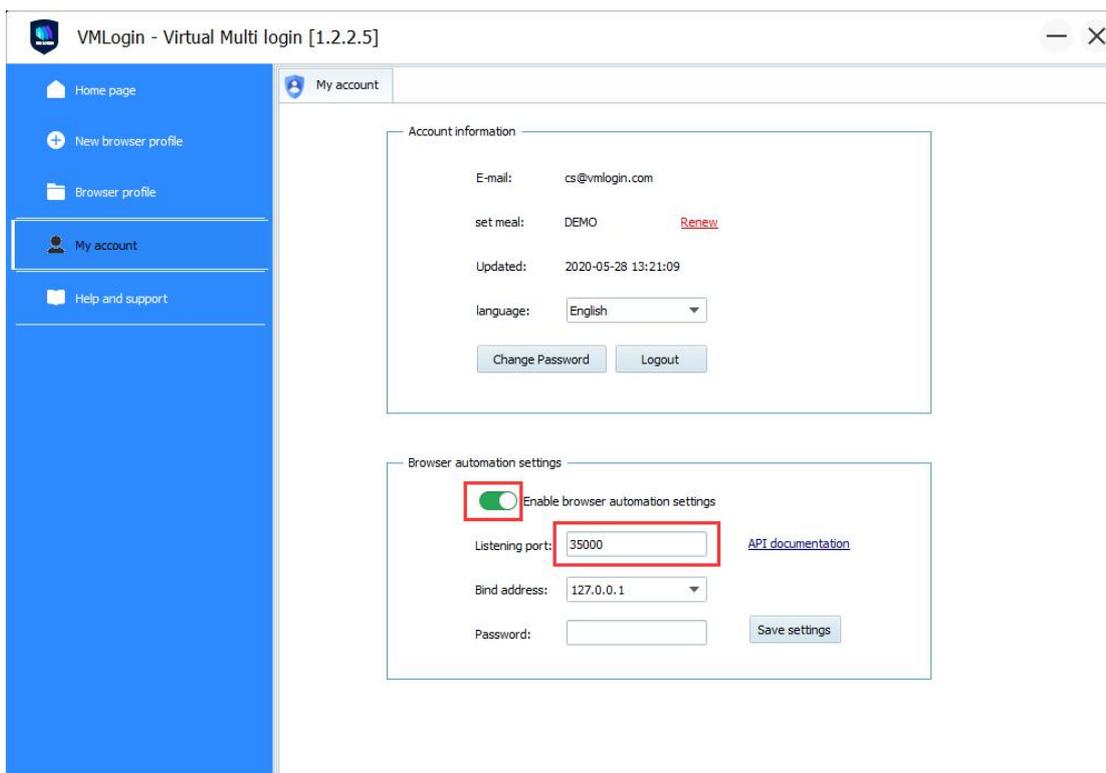
## Supported languages

The Selenium framework provides multiple languages that can be used together, so VMLogin automation can also run on multiple coding languages. But at present, we only provide technical support for Java and Python.

## Use Selenium in Vmlogin

Define VMLogin port

You need to define the software port in advance to use Selenium automation. Here is how to define the port:



Open the browser automation setting in "My Account", and set the usable port in the listening port. The default is 35000 here, and you can also set an access password.

Then, you can connect to VMLogin through the defined port.

## Java case:

```
import org.junit.Assert;
import org.openqa.selenium.firefox.FirefoxOptions;
import org.openqa.selenium.remote.RemoteWebDriver;

import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

public class BrowserProfile {
    public static void main(String[] args) throws Exception {
        BrowserProfile bp = new BrowserProfile();
        //TODO replace with existing profile ID. Define the ID of the browser profile, where the code will
        be executed.
        String profileId = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx";

        //Define DesiredCapabilities
        DesiredCapabilities dc = new DesiredCapabilities();

        //Instantiate the Remote Web Driver to connect to the browser profile launched by startProfile
        method
        RemoteWebDriver driver = new RemoteWebDriver(new URL(bp.startProfile(profileId)), dc);

        //Perform automation
        driver.get("https://vmlogin.com/");
        Assert.assertEquals("VMLOGIN", driver.getTitle());
        driver.quit();
    }

    private String startProfile(String profileId) throws Exception {
        /*Send GET request to start the browser profile by profileId. Returns response in the following
        format:
        {"status":"OK","value":"http://127.0.0.1:XXXXX"}', where XXXXX is the localhost port on which
        browser profile is
        launched. Please make sure that you have Vmlogin listening port set to 35000. Otherwise
        please change the port
        value in the url string */
        String url = "http://127.0.0.1:35000/api/v1/profile/start?automation=true&profileId=" + profileId;

        URL obj = new URL(url);
```

```

URLConnection con = (URLConnection) obj.openConnection();

con.setRequestMethod("GET");

BufferedReader in = new BufferedReader(
    new InputStreamReader(con.getInputStream()));
String inputLine;
StringBuffer response = new StringBuffer();

while ((inputLine = in.readLine()) != null) {
    response.append(inputLine);
}
in.close();

//Get JSON text from the response and return the value by key "value"
JSONObject jsonResponse = new JSONObject(response.toString());
return jsonResponse.getString("value");
}
}

```

The interface can also pass in proxy server information. If the incoming proxy information will overwrite the proxy information in the configuration file, this overwriting is temporary and will not really modify the configuration file. It is only valid for the automation interface:

**`http://127.0.0.1:35000/api/v1/profile/start?automation=true&profileId=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx&proxytype=socks5&proxyserver=ip&proxyport=1080&proxyusername=&proxypassword=`**

Agent types may be these three: proxytype=socks5 proxytype=socks4 proxytype=http

The proxy username and password can be left blank.