Table of Contents

Surge Overview

Surge is a web developer and proxy utility tool. This app is designed for developers and may need certain level of professional knowledge to use.

Features

- High Performance & Stability: With industrial-grade stability, Surge is capable of constantly running with high performance, yet it only occupies minimum system resources. It will perfectly handle all the traffic and leave you worry-free.
- Flexible Rule System: You may setup proxy forwarding rules based on domain, IP CIDR, GEOIP, etc. Surge will automatically send the requests to another proxy server. HTTP/HTTPS/SOCKS5/SOCKS5-TLS proxy protocols are all supported.
- HTTPS Decryption: HTTPS traffic can be decrypted by MitM. Certificate generator will help you generate CA certificate for debugging and make the certificate trusted by system.
- Local DNS Mapping: Surge supports local-customized DNS mapping. Its multiple functional modules, including wildcard, alias and custom DNS server, will be able to fulfill varied needs.
- Proxy Group: You may categorize several proxies as a group and a policy will be employed in accordance with the grouping. Proxy group can be configured as Auto Speed Test (select policy based on benchmarking URL access speed), SSID (select policy based on WiFi SSID), and manual-select.
- HTTP Rewrite: Rewrite the HTTP/HTTPS request to another URL based on customized rules, or simply block the request;
- External Controller: Surge can be managed by remote machine via surge-cli or Surge Dashboard.
- Full IPv6 Support: All functions work in IPv6 environment since version 2.0.0.

Special Features for iOS

- All functions work on cellular network.
- Capture all HTTP/HTTPS/TCP traffic from any applications on your device, and redirect the traffic to an HTTP/HTTPS/SOCKS5 proxy server following highly configurable rules, even when the application doesn't follow system proxy settings.
- Override system DNS settings even on cellular network. And boost the performance by sending DNS query to all DNS servers simultaneously.

- Surge Mac Dashboard is able to connect to Surge iOS through WiFi or USB, monitor and analyze network requests on the iOS devices. You can even examine Cellular network requests when Dashboard is connected through USB.

Known Issues

- The HTTP/HTTPS proxy server must support CONNECT method, even when accessing plain HTTP resources.
- HTTP pipelining is not supported.

Components

Surge includes several components.

Surge Proxy Server (Surge Core)

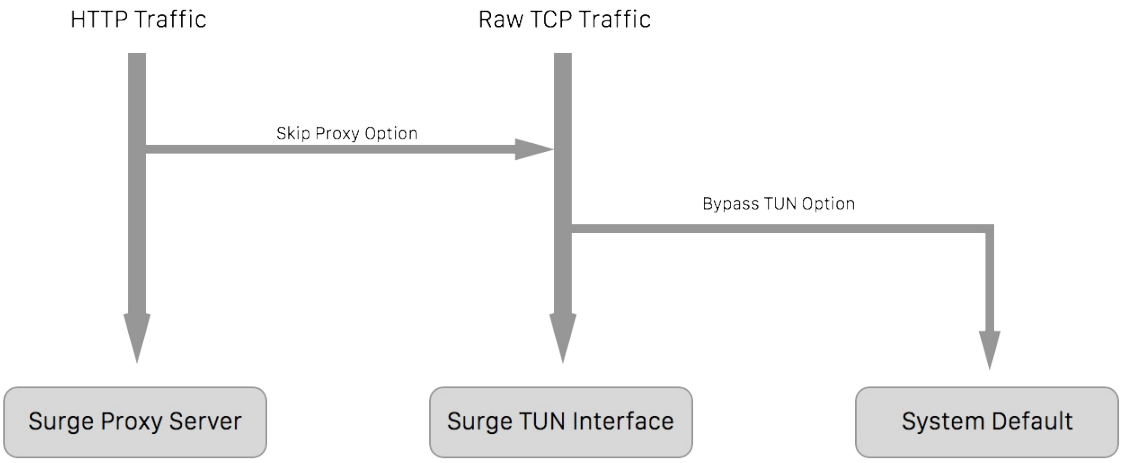
This is the core part of Surge. It's a full-function HTTP/SOCKS5 proxy server with high performance and stability, written in Objective-C and optimized for macOS and iOS.

Surge TUN (iOS Version Only)

Some apps do not obey system proxy settings (such as Mail.app), because they need to use a raw TCP socket. This kind of traffic is handled by Surge TUN.

Currently, Surge TUN can only process TCP and UDP protocol. Other protocol packets (such as ICMP) will be dropped. You can use 'bypass-tun' option as a workaround.

This is the architecture for Surge iOS:



Surge Dashboard (Mac Version Only)

Surge Dashboard is a graphical user interface to review and inspect requests, list DNS cache, and modify configurations. It can connect to a local Surge instance, or a remote instance when the external-controller-access is set.

Surge CLI (Mac Version Only)

Surge CLI is a command line tool to control Surge. It's bundled in mac version. Just like Surge Dashboard, it can connect to a local Surge instance, or a remote instance when the external-controller-access is set.

Configuration

Most functions of Surge are controlled by the configuration file. Surge iOS and Surge Mac can share configuration files via iCloud Drive. You may edit the configuration file with your text editor, Surge iOS app, or Surge Dashboard.

Here is the configuration examples:

- English Version
- Chinese Version

Comment

Surge configuration file supports comment line, starts with '#', ';' and '//'. Comment inline will also be supported with separation of '//'. The comments for rules and hosts will be kept if editing the configuration with GUI, and the other comments will be deleted after saving.

Rule

Surge can forward requests to another proxy server or connect to the host directly, depending on customized rules.

Priority

Rules are matched from the first one to the last one, in the order they appear in the config file. In other words, rules at the top of the list have higher priority than latter ones.

Composition

Each rule consists 3 parts: rule type, a traffic matcher (except for FINAL rule), and a proxy policy: TYPE, VALUE, POLICY Example: DOMAIN-SUFFIX,apple.com, DIRECT IP-CIDR, 192.168.0.0/16,ProxyA

Surge supports 6 different types of rules: DOMAIN, DOMAIN-SUFFIX, DOMAIN-KEYWORD, GEOIP, IP-CIDR, or FINAL. Proxy policy must be named under one of a policy names, including ‘a proxy’, 'a policy group', 'DIRECT', or 'REJECT'. Rules must end with a FINAL rule to define the default behavior.

Example:

```
[Rule]
DOMAIN-SUFFIX,company.com,ProxyA
DOMAIN-KEYWORD,google,DIRECT
GEOIP,US,DIRECT
IP-CIDR,192.168.0.0/16,DIRECT
FINAL,ProxyB
```

DOMAIN, DOMAIN-SUFFIX and DOMAIN-KEYWORD are domain based rules. IP-CIDR and GEOIP are IP based rules.

Domain-based Rule

There are 3 domain-based rule types.

DOMAIN

```
DOMAIN,www.apple.com,Proxy
```

Rule matches if the domain of request matches exactly.

DOMAIN-SUFFIX

```
DOMAIN-SUFFIX,apple.com,Proxy
```

Rule matches if the domain of the request matches the suffix. For example: 'google.com' matches 'www.google.com', 'mail.google.com' and 'google.com', but does **not** match 'content-google.com'.

DOMAIN-KEYWORD

```
DOMAIN-KEYWORD,google,Proxy
```

Rule matches if the domain of the request contains the keyword.

Domain-based Rule Options

Option: force-remote-dns (Surge iOS Only)

```
DOMAIN,www.apple.com,Proxy,force-remote-dns
DOMAIN-SUFFIX,apple.com,Proxy,force-remote-dns
DOMAIN-KEYWORD,google,Proxy,force-remote-dns
```

When a raw TCP connection is handled by Surge TUN, the application will firstly try to resolve the domain, then send packets to the IP address directly. If the domain cannot be resolved locally, you can use this option to force the DNS resolution to happen remotely on the proxy.

Technically, when an application tries to resolve a domain which matches a rule with 'force-remote-dns' option, Surge will send a DNS answer with a fake IP address (240.1.x.x). When the application connects to the fake IP, Surge will remap it to a domain and send the request to the remote proxy.

> Notice: This option only works for Surge TUN (only exists in Surge iOS). Request handled by Surge Proxy will always be resolved remotely if the rule's policy is a proxy.

IP-based Rule

There are 2 IP-based rule types. A IP-based rule will trigger a DNS lookup if the hostname of the request is a domain. If the DNS lookup fails, Surge will abort the rule testing and report an error.

IP-CIDR

```
IP-CIDR,192.168.0.0/16,DIRECT
IP-CIDR,10.0.0.0/8,DIRECT
IP-CIDR,172.16.0.0/12,DIRECT
IP-CIDR,127.0.0.1/8,DIRECT
```

Rule matches if the IP address of the request matches a specified range.

GEOIP

```
GEOIP,US,DIRECT
```

Rule matches if the GeoIP test result matches a specified country code.

IP-based Rule Option

Option: no-resolve

```
GEOIP,US,DIRECT,no-resolve
IP-CIDR,172.16.0.0/12,DIRECT,no-resolve
```

When a GEOIP or IP-CIDR rule is encountered, Surge will send a DNS question to check if the hostname of request is a domain. You can select 'no-resolve' option to skip this rule for a request with domain.

> Notice: If some domains can't be resolved by local DNS server, please make sure there is no IP-based rule in front of the rule which matches that domain. Otherwise the rule testing will fail due to a DNS error. You can use 'no-resolve' to solve the issue too.

Final Rule

The FINAL rule must be written after all other rules. It defines the default policy for requests which are not matched by any other rules.

Example:

```
[Rule]
DOMAIN-SUFFIX,company.com,ProxyA
DOMAIN-KEYWORD,google,DIRECT
GEOIP,US,DIRECT
IP-CIDR,192.168.0.0/16,DIRECT
FINAL,ProxyB
```

Policy

Surge can forward requests to another proxy server or connect to the host directly, depending on customized rules. A policy indicates how Surge will deal with the requests.

There are 3 types of policies: proxy, policy group and bulit-in policy.

Proxy Policy

A proxy policy indicates forwarding the request to another proxy server. Surge supports HTTP/HTTPS/SOCKS5/SOCKS5-TLS proxy protocols.

Section [Proxy] declares proxy policies. You can create multiple proxies for different rules.

Example:

```
[Proxy]
ProxyHTTP = http, 1.2.3.4, 443, username, password
ProxyHTTPS = https, 1.2.3.4, 443, username, password
ProxySOCKS5 = socks5, 1.2.3.4, 443, username, password
ProxySOCKS5TLS = socks5-tls, 1.2.3.4, 443, username, password, skip-common-name-verify
=true
```

Parameters

Parameter for all proxy type

interface: Optional (Default: null).

Force to use a specified outgoing network interface or address (available in macOS only). Please make sure the interface has a valid route table to the destination address.

```
ProxyHTTP = http, 1.2.3.4, 443, username, password, interface = en2
```

Parameter for proxy with TLS (https and socks5-tls)

skip-common-name-verify: Optional, "true" or "false" (Default: false).

If this option is enabled, Surge will not verify whether the certificate common name field is matched.

Built-in Policy

There are two built-in policies: DIRECT and REJECT. DIRECT means the request should be sent to the host directly. REJECT means the request should be rejected.

DIRECT and REJECT can be used in rule and policy group directly. You can also define an alias in the proxy section.

Alias

```
[Proxy]
On = direct
Off = reject
```

Then you can use 'On' and 'Off' as a policy name in rule and policy group.

Interface Option

Direct policy alias supports 'interface' parameter like a proxy policy.

```
[Proxy]
Crop-VPN = direct, interface = utun0
WiFi = direct, interface = en2
```

Please make sure the interface has a valid route table to the destination address.

Policy Group

A policy group may contain multiple policies. It can be a proxy policy, another policy group or a built-in policy (DIRECT and REJECT).

There are three group types: ‘select‘, ’url-test‘ and ’ssid‘. Section [Proxy Group] declares policy group.

Manaully Select Group

Select which policy will be used on the user interface.

```
SelectGroup = select, ProxyHTTP, ProxyHTTPS, DIRECT, REJECT
```

> In iOS version. You can use Today Widget to quickly select policy for the first 'select' group. You can enable/disable this feature in 'More' tab.

Auto URL Test Group

Select which policy will be used by benchmarking speed to a URL.

```
AutoTestGroup = url-test, ProxySOCKS5, ProxySOCKS5TLS, url = http://www.google.com/generate_204
```

Parameters

url: Required

Specify which URL will be tested.

interval: Optional, s (Default: 600s).

Determine how long the benchmark result will be discarded.

tolerance: Optional, ms (Default: 100ms).

Policy will be changed only when the new winner has a higher score than the old winner's score plus the tolerance.

timeout: Optional, s (Default: 5s).

Abandon a policy if it is not finished until timeout.

SSID Group

Select which policy will be used by Wi-FI SSID.

```
SSIDGroup = ssid, default = ProxyHTTP, cellular = ProxyHTTP, SSIDName = ProxySOCKS5
```

Parameters

default: Required.

The policy when no matched SSID option has been found.

cellular: Optional.

The policy under cellular network. If not provided, the default policy will be used.

DNS

To implement complex DNS features, Surge has its own DNS client implementation. It may perform different behaviors to system DNS implementation.

Upstream DNS Server

By default, Surge uses system's DNS server address setting. You may override it using 'dns-server' option.

Detail

To boost performance Surge will send DNS query to all DNS servers simultaneously, just like dnsmasq with '--all-servers' parameter. The first answer from servers will be used. Surge iOS app and Surge Dashboard will show which server responds first.

If Surge has not received answer from any server in 2 seconds, it will resend the question to all servers again. After 4 times retries, Surge will stop lookup and report a DNS error.

Some domain may have a poor NS server, so that the DNS server may return an empty answer due to server-side timeout or other connectivity issues. Surge will report an empty DNS error, if **all** servers return explicit empty answers, or some servers return empty answers and the others have not responded in 2 seconds.

When IPv6 is available and enabled, Surge DNS client will send A and AAAA questions to upstream DNS Server. In current version, which answer will be used depends on which answer returns first.

DNS Override

You can use this option to override system's DNS setting.

```
[General]
dns-server = 8.8.8.8, 8.8.4.4
```

Use keyword 'system' to append additional DNS servers to system's setting. (Duplicate servers will be ignored)

```
[General]
dns-server = system, 8.8.8.8, 8.8.4.4
```

Local DNS Mapping

Surge supports local-customized DNS mapping. It's equivalent to /etc/hosts, but with more powerful features, including wildcard, alias and assigning DNS server.

```
[Host]
abc.com = 1.2.3.4
*.dev = 6.7.8.9
foo.com = bar.com
bar.com = server:8.8.8.8
```

Wildcard

You can use *prefix to wildcard all sub-domains. Please note that Surge uses a simple string match. For example,* google.com will match google.com, foo.google.com and bargoogle.com. And *.google.com will *not* match google.com.

```
[Host]
*.dev = 6.7.8.9
```

Alias

It's equivalent to CNAME record.

```
[Host]
foo.com = bar.com
```

Assigning DNS Server

You can assign a specified DNS server to one or more domains.

```
[Host]
bar.com = server:8.8.8.8
```

Since Surge has its own DNS client implementation, some hostnames may fail to resolve. You can use 'server:system' to let system handle the lookup.

```
[Host]
Macbook = server:system
```

By default, all hostnames with suffix '.local' will be resolved by the system.

Combined Usage

All features can be used together. For example:

```
[Host]
*.dev = foo.com
*.bar.com = server:system
```

URL Rewrite

Surge can rewrite the request's URL with 2 different methods, or reject certain requests by URL.

Example:

```
[URL Rewrite]
^http://www.google.cn http://www.google.com header
^http://yachen.com https://yach.me 302
^http://ad.com/ad.png _ reject
```

The rewrite rule consists 3 parts: regular expression, replacement and type.

Header Mode

Surge will modify the request header and redirect the request to another host if necessary. The client will not notice this rewrite action.

The "Host" field in request header will be modified to match the new URL.

```
[URL Rewrite]
^http://www.google.cn http://www.google.com header
```

> You can't redirect to an URL with HTTPS scheme. And you can't redirect a HTTPS request.

302 Mode

Surge will simply return a 302 redirect response. HTTPS requests can be redirected if MitM for the hostname is enabled.

```
[URL Rewrite]
^http://yachen.com https://yach.me 302
```

Reject Mode

Reject the request if the pattern is matched. The replacement parameter will be ignored. HTTPS requests will be rejected if MitM for the hostname is enabled.

```
[URL Rewrite]
^http://ad.com/ad.png _ reject
```

HTTPS Decryption (Man-in-the-Middle Attack, MitM)

Surge may decrypt HTTPS traffic by MitM. Please see Wikipedia article for more information.

Certificate generator will help you generate a new CA certificate for debugging and make the certificate trusted by system. It's available in Surge Dashboard (Mac version) and Surge iOS Config Editor. This certificate is generated locally and will only be saved in your config file and system Keychain. The key of certificate is generated randomly using OpenSSL.

You can also use an existed CA certificate. Export the certificate to PKCS#12 format (.p12) with passphrase. Please note that the passphrase cannot be empty due to system limitation. Use "base64" command to encode in base64 string and append these settings below to your config file.

```
[MITM]
enable = true
ca-p12 = MIIJtQ.........
ca-passphrase = password
hostname = *google.com
```

Surge will only decrypt traffic to hosts which are declared in "hostname". Prefix wildcard *is allowed. It uses the same wildcard rule as Local DNS Mapping. Although you can use single* to enable MitM to all requests, it's not recommended.

> Some applications has strict security policy to use pinned certificates or CA. Enabling decryption to these hosts may casue problems.

External Controller Access

Surge can be managed from remote machine by tools like Surge Dashbaord or Surge CLI. You need to manually enable this function.

```
[General]
external-controller-access = apassword@127.0.0.1:8888
```

This parameter consists three parts: password, listen address and port number. No part can be omitted.

When you use this function in Surge iOS, listening on 127.0.0.1 means only connections from USB cable will be allowed. Listening on 0.0.0.0 means connections from local Wi-Fi network will also be allowed. Connection from cellular network will always be restricted due to a security consideration.

Misc Options

```
[General]
ipv6 = false
loglevel = notify

interface = 0.0.0.0
port = 6152
socks-interface = 0.0.0.0
socks-port = 6153

skip-proxy = 127.0.0.1, 192.168.0.0/16, 10.0.0.0/8, 172.16.0.0/12, 100.64.0.0/10, loca
lhost, *.local
bypass-system = true
bypass-tun = 192.168.0.0/16, 10.0.0.0/8, 172.16.0.0/12
```

Common Options

Enable full IPv6 support (Default: false)

`ipv6 = false`

loglevel (Default: notify)

`loglevel = notify`

One of verbose, info, notify or warning. It's not recommended to enable verbose in daily use because this will slow down the performance significantly.

skip-proxy

`skip-proxy = 127.0.0.1, 192.168.0.0/16, 10.0.0.0/8, 172.16.0.0/12, 100.64.0.0/10, localhost, *.local`

In iOS version, this option forces connections to these domain/IP ranges to be handled by Surge TUN, instead of Surge proxy. In macOS version, these settings will be applied to system when "Set as System Proxy" is enabled. This option is used to fix compatibility problems with some apps.

- To specify a single domain, enter the domain name - for example, apple.com.
- To specify all websites on a domain, use an asterisk before the domain name - for example, *apple.com.
- To specify a specific part of a domain, specify each part - for example, store.apple.com.

- To specify hosts or networks by IP addresses, enter a specific IP address such as 192.168.2.11 or an address range, such as 192.168.2.* or 192.168.2.0/24.

> Notice: If you enter an IP address or address range, you will only be able to bypass the proxy when you connect to that host using that address, not when you connect to the host by a domain name that resolves to that address.

Surge Mac Special Options

Server listen interface (Default: 127.0.0.1)

```
interface = 0.0.0.0
```

HTTP server port (Default: 6152)

```
port = 6152
```

SOCKS5 server listen interface (Default: 127.0.0.1)

```
socks-interface = 0.0.0.0
```

SOCKS5 server port (Default: 6153)

```
socks-port = 6153
```

Surge iOS Special Options

Bypass System Related Request (Default: true)

```
bypass-system = true
```

This option will add some special rules. First, these rules below are added to allow domains to bypass Surge proxy server and use raw TCP:

```
api.smoot.apple.com
configuration.apple.com
xp.apple.com
smp-device-content.apple.com
guzzoni.apple.com
captive.apple.com
*.ess.apple.com
*.push.apple.com
*.push-apple.com.akadns.net
```

Second, this rule is added with the highest priority:

```
IP-CIDR, 17.0.0.0/8, DIRECT, no-resolve
```

If you disable this option, it may lead to some system problems, such as delays in push notifications.

> Notice: Entire 17.0.0.0/8 address block is assigned to Apple.

Excluded Routes

```
bypass-tun = 192.168.0.0/16, 10.0.0.0/8, 172.16.0.0/12
```

Surge TUN can only process TCP and UDP protocols. Use this option to bypass specific IP ranges to allow all traffic to pass through.

> Notice: This option only works for Surge TUN. Requests handled by Surge Proxy Server will not be affected. Combine 'skip-proxy' and 'bypass-tun' to make sure that certain HTTP traffic bypasses Surge.

Managed Configuration

Only Surge iOS supports this feature, from version 2.2.2

Surge can update a config file from an URL automatically. If the config starts with

```
#!MANAGED-CONFIG http://test.com/surge.conf interval=60 strict=true
```

The config can only be updated when Surge main application is running.

> Note: The new config in remote should also contain #!MANAGED-CONFIG line. Otherwise this config will become a regular one.

Parameters

interval: Optional, s (Default: 86400s).

Determine how long the config will be updated.

strict: true or false (Default: false).

If strict is true, Surge will require a force update after the interval arrives. Otherwise if the update fails the user may still use the outdated config.

> Note: Even when strict is true, the user still can start Surge by widget or VPN switch in Settings.

FAQ for Surge iOS

Q: How does Surge iOS work?

> There are two main components in Surge: Surge proxy server and Surge TUN interface. After being started, Surge sets itself as the default HTTP/HTTPS proxy server to handle all HTTP/HTTPS traffic, which allows Surge to boost performance by using HTTP connections' keep-alive mechanism globally. But some apps do not obey system proxy settings (such as Mail.app), because they need to use a raw TCP socket. This kind of traffic is handled by Surge TUN interface.

Q: Why does Surge stop unexpectedly sometimes? The VPN icon suddenly disappears.

> There are two reasons that may lead to this problem: Surge has reached the system memory limit for network extension apps, and/or you have triggered some bugs in Surge.
>
> iOS system limits network extension apps to use about 6MB memory at most. Surge may use a little more memory under some circumstances and get killed by the system. We will keep working on improving the stability by reducing memory usage and fixing bugs.

Q: In the system's battery usage panel, it says that Surge consumes a large portion of power. Why?

> Surge handles all network traffic on your device. So the system counts all network power consumption to Surge. In fact, Surge does not use much power on top of the system network power consumption and does not drain your battery.

Q: Why I always encounter error "Cannot allocate memory"?

> This error has been confirmed to be an iOS system bug. The only way to fix it is to reboot your device. We are keeping working with Apple to fix this bug.

Q: What does "Bypass System Related" option do internally?

> When this option is enabled, Surge adds some special rules to allow some domains to use raw TCP instead of proxy.
>
> First, add these rules below to allow domains to bypass Surge proxy server and use raw TCP:
>
> ```
> api.smoot.apple.com
> api.smoot.apple.com.cn
> configuration.apple.com
> xp.apple.com
> smp-device-content.apple.com
> guzzoni.apple.com
> captive.apple.com
> *.ess.apple.com
> *.push.apple.com
> *.push-apple.com.akadns.net
> ```
>
> Second, add this rule with the highest priority:
>
> `IP-CIDR, 17.0.0.0/8, DIRECT, no-resolve`
>
> If you disable this option, it may lead to some system problems, such as delays in push notifications.

Q: What does "Skip proxy" option do?

> This option forces connections to these domain/IP ranges to be handled by Surge TUN, instead of Surge Proxy Server. This option is used to fix compatibility problems with some apps.
>
> - To specify a single domain, enter the domain name - for example, apple.com.
> - To specify all websites on a domain, use an asterisk before the domain name - for example, *apple.com.
> - To specify a specific part of a domain, specify each part - for example, store.apple.com.
> - To specify hosts or networks by IP addresses, enter a specific IP address such as 192.168.2.11 or an address range, such as 192.168.2.* or 192.168.2.0/24.
>
> Notice: If you enter an IP address or address range, you will only be able to bypass the proxy when you connect to that host using that address, not when you connect to the host by a domain name that resolves to that address.

Q: What does “Force Remote DNS” option do?

> Surge always tries to resolve these domains in remote proxy server. This option is useful if some domains cannot be resolved by local DNS.
>
> Notice: This option is only for Surge TUN interface. When a request is sent to Surge proxy server, Surge always tries to resolve the domain remotely if it matches a rule without DIRECT policy.

FAQ for Surge Mac

Q: Why does 1Password browser extension alert signature error?

> Please add 127.0.0.1 and localhost to skip-proxy option. `skip-proxy = 127.0.0.1, localhost`

Q: How can I transfer my license to another computer?

> You may deactivate your machine in 'Preferences' and then activate another computer.
>
> If you cannot access your previous computer, you can reset your license here: http://nssurge.com/account. Please note you can only reset your license every 90 days.

Surge Mac Release Note

Version 2.0.10

- Surge talks to HTTP proxies with a plain HTTP method for non-HTTPS requests now, instead of CONNECT.
- Improved compatibility with some HTTP server.
- Improved compatibility with some DNS server.

Version 2.0.9

- Dashborad: The height of the detail panel will not change now while switching pages.
- A notification will show when proxy client access from other machine.
- Used SF Mono as monospaced font for header and body data display.
- Supported TCP half-open mechanism.

Version 2.0.8

- Add a new option 'exclude-simple-hostnames' in the gereral section.
- Dashborad: Selected row will not be lost while the filter or sort column changed.
- Dashborad: Fixes some issues in the active panel.

Version 2.0.5

- Bug fixes.

Version 2.0.3

- New feature: Show connectivity quality in menu.

  Surge will send a DNS question to all DNS servers concurrently to test physical network connectivity while opening the menu.

- Fixes a problem that Surge may freeze while opening the menu.
- Fixes a problem that if a policy group contains duplicate policies, Surge may crash.

Version 2.0.2

- Dashboard will no longer display process icon in remote mode.

- Fixes a bug: "Set as System Proxy" option does not work properly if only SOCKS service is enabled.
- Fixes a bug: Dashboard can't add a rule with no-resolve option on and comment not empty.
- Minor bug fixes.

Version 2.0.1

- Bug fixes