



# Burp suite: Proxy

[Introduction](#)

[Intercept](#)

[HTTP history](#)

[Filter](#)

[Websocket history](#)

[Filter](#)

[Proxy options](#)

[Introduction](#)

[Proxy listeners](#)

[Adding proxy listeners](#)

[Editing/removing proxy listeners](#)

[Import/Export CA certificate/Regenerate CA certificate](#)

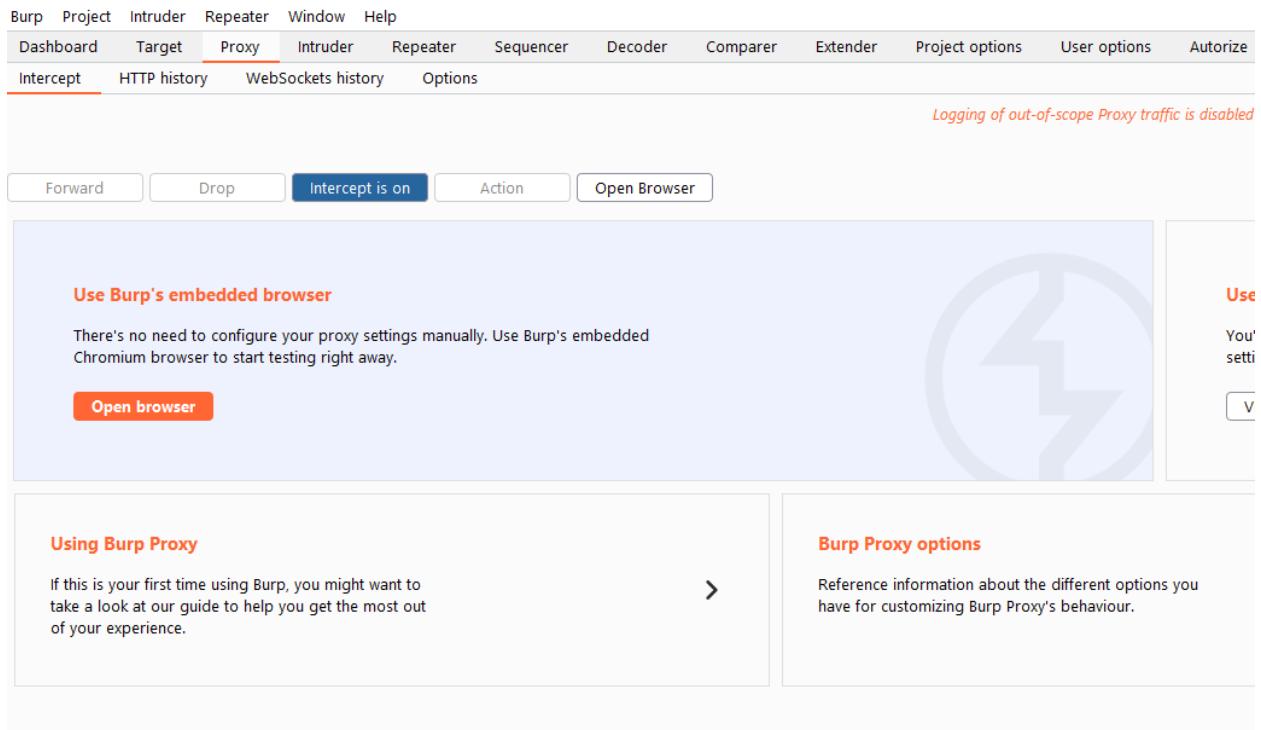
[Response modifications](#)

[Match and replace](#)

## Introduction

The proxy tab is going to allow us to interact with requests and responses in real time and investigate the proxy history. This is the tab i will use most as it contains all of my latests requests and can also handle websockets in the pro version of burp.

## Intercept



This is the basic landing tab of the proxy page. In here we have several options

- Put intercept on/off, if we enable intercept it will stop any request (even the ones not in scope) and allow you to edit, forward or drop requests
- Action, which allows us to perform a selection of actions on the current request that is being held by the interceptor
- Open browser, which allows us to open a builtin chromium browser that will automatically have the proxy set properly. One thing i don't like about this browser is that it won't remember any plugins you install or any passwords you save.

## HTTP history

## Filter

If we click the filter bar at the top of the screen, we can see some very useful filters.

1. Clicking the filter bar will open it for us
2. Showing only the parameterised requests will show us all the requests that allow us some kind of interaction with the server. These are the requests that we care about most as they talk to the API.

3. Filtering by mime type allows us to include or exclude certain types of files such as scripts or CSS files by checking or unchecking the checkbox respectively.
4. If we want to show or hide certain status codes, that's also possible like the 4xx status codes which are disabled by default
5. If we want to search for specific terms, that also possible. We can do this in negative search as well, which would mean that we would **exclude** requests with the search term in it.
6. If we want to hide or show certain file types, that's also possible but we can't use both together as they would cancel each other out
7. We can have multiple proxy listeners which listen for incoming traffic on different ports. We can also filter per port. More on this later on in the proxy options.

We can also annotate or comment certain requests by right clicking them in the site map, repeater or proxy history and then we can filter on those requests.

## Websocket history

Since recently, it's also possible for Burp suite to process websocket messages such that we can edit and resend them. This is a very useful feature as before we would have to write our own python proxies to translate these WS messages into HTTP messages and back again that would act inbetween our target and burp.

Burp Project Intruder Repeater Window Help  
 Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options Authorize AutoRepeater  
 Intercept HTTP history WebSockets history Options

Logging of out-of-scope Proxy traffic is disabled Re-enable

Filter: Showing all items

#	URL	Direction	Edited	Length	Comment	TLS	Time	Listener port	WebSocket ID
13199	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	21:01:40 12 ...	8080	98
13198	https://ferretshop.herokuapp.com/soc...	→ To server		1		✓	21:01:40 12 ...	8080	98
13197	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	21:01:15 12 ...	8080	98
13196	https://ferretshop.herokuapp.com/soc...	→ To server		1		✓	21:01:14 12 ...	8080	98
13195	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	21:00:48 12 ...	8080	98
13194	https://ferretshop.herokuapp.com/soc...	→ To server		1		✓	21:00:48 12 ...	8080	98
13193	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	21:00:22 12 ...	8080	98
13192	https://ferretshop.herokuapp.com/soc...	→ To server		1		✓	21:00:22 12 ...	8080	98
13191	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	20:59:56 12 ...	8080	98
13190	https://ferretshop.herokuapp.com/soc...	→ To server		1		✓	20:59:56 12 ...	8080	98
13189	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	20:59:30 12 ...	8080	98
13188	https://ferretshop.herokuapp.com/soc...	→ To server		1		✓	20:59:30 12 ...	8080	98
13187	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	20:59:04 12 ...	8080	98

Message

1 3

## Filter

Filter: Showing all items 1

Filter by request type

☐ Show only in-scope items

☒ Hide outgoing messages

☒ Hide incoming messages

2

Filter by search term 3

☐ Regex

☐ Case sensitive ☐ Negative search

Filter by annotation 4

☐ Show only commented items

☐ Show only highlighted items

Filter by listener 5

Port

Show all Revert changes

13191	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	20:59:56 12 ...	8080
13190	https://ferretshop.herokuapp.com/soc...	→ To server		1		✓	20:59:56 12 ...	8080
13189	https://ferretshop.herokuapp.com/soc...	← To client		1		✓	20:59:30 12 ...	8080

If we click the filter bar at the top of the screen, we can some very useful filters.

1. Clicking the filter bar will open it for us
2. In websockets, we can send messages to the server and it can respond but the server can also send messages to us and we can respond. To filter those messages we can hide outgoing or incoming messages

3. If we want to search for specific terms, that also possible. We can do this in negative search as well, which would mean that we would **exclude** requests with the search term in it.
4. We can also annotate or comment certain requests by right clicking them in the site map, repeater or proxy history and then we can filter on those requests.
5. We can have multiple proxy listeners which listen for incoming traffic on different ports. We can also filter per port. More on this later on in the proxy options.

## Proxy options

### Introduction

It may not seem like it but burp suite proxy options can have several useful things in store for us. It took me quite a while to figure this out but burp suite has even more features then it may seem at first.

### Proxy listeners

Several useful options exist in this section of the proxy options. We will be going through all of them.

The screenshot shows the 'Proxy Listeners' configuration window in Burp Suite. It includes a help icon, a title bar, and a description: 'Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.' Below this is a table with columns: Running, Interface, Invisible, Redirect, Certificate, and TLS Protocols. A single listener is listed with 'Running' checked, 'Interface' as '127.0.0.1:8080', 'Invisible' unchecked, 'Redirect' unchecked, 'Certificate' as 'Per-host', and 'TLS Protocols' as 'Default'. To the left of the table are 'Add', 'Edit', and 'Remove' buttons. Below the table, a note states: 'Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other tools or another installation of Burp.' At the bottom are 'Import / export CA certificate' and 'Regenerate CA certificate' buttons.

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
<input checked="" type="checkbox"/>	127.0.0.1:8080	<input type="checkbox"/>	<input type="checkbox"/>	Per-host	Default

### Adding proxy listeners

We can add another proxy listener. This is just another proxy that we can send traffic through and we can add one on another port or another ip address. This can be very interesting in the following situations.

- If you want to test a mobile application and a web application at the same time you might want to have them enter at a different port
- If you are testing multiple physical IoT devices, it might be useful to add another proxy listener to give all devices a separate listener.
- We might want to have burp suite listen on different ip addresses in network situations
- ...

## **Editing/removing proxy listeners**

If we made a mistake, we can edit or remove an existing proxy listener.

## **Import/Export CA certificate/Regenate CA certificate**

We might need the CA certificate to add to windows so we can decrypt the HTTPS traffic coming from our browser. We might also need this certificate if we want to test mobile applications that communicate over HTTPS. This certificate can be exported here and imported into our testing client. We can also regenerate the certificate here should we need to.

## **Response modifications**



## Response Modification



These settings are used to perform automatic modification of responses.

- ☒ Unhide hidden form fields
  - ☒ Prominently highlight unhidden fields
- ☒ Enable disabled form fields
- ☒ Remove input field length limits
- ☒ Remove JavaScript form validation
- ☐ Remove all JavaScript
- ☐ Remove <object> tags
- ☐ Convert HTTPS links to HTTP
- ☐ Remove secure flag from cookies

I usually unhide hidden form fields, that's just a simple client side protection and usually hidden fields will contain valuable information.

I will also prominently highlight them, doing this makes it so that any hidden form field will have a large red border around it, making it more easily visible.

I will also enable disabled form fields for the same reason as that I unhide hidden form fields.

I will also remove any field length limits, this might trigger some API errors if I send more characters than the API allows but sometimes developers don't even implement API checks and will only use front-end validation.

For the same reasons, I remove any javascript form validation. It's all just client side.

The other options I don't enable as they are there to protect against things like XSS and we can't just disable these fields on our victims browser.

## Match and replace

See the match and replace document.