
> Introduction

> Get started

✓ Sending requests

Building requests

> Authorizing requests

Receiving responses

Grouping requests in collections

Using variables

Managing environments

Visualizing responses

Specifying examples

Using cookies

Working with certificates

Generating client code

Troubleshooting requests

> Capturing request data

> Using GraphQL requests

> Using gRPC requests

Using WebSocket requests

Using SOAP requests

> Writing scripts

> Using collections

> Using Postman Flows

> Using the Postman CLI

> Collaborating in Postman

> Designing and developing your API

> Documenting your API

> Monitoring your API

> API Governance

> Administration

> Using reports

> Developer resources

> Integrations

Home / Sending requests

Using cookies

Postman's cookie manager enables you to view and edit cookies that are associated with different domains. You can manually create cookies for a domain, or you can **capture cookies** using the Postman proxy or Postman Interceptor. You can then use the cookies stored in the cookie jar when sending requests in Postman.

To turn off sending cookies with a request, select the request's **Settings** tab, then select **Disable cookie jar**.

Contents

- [What are cookies?](#)
- [Using the cookie manager](#)
- [Accessing cookies in scripts](#)
- [Creating cookies programmatically](#)
- [Syncing cookies](#)

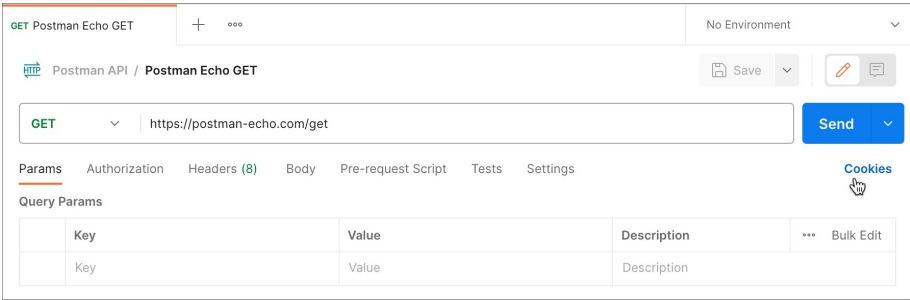
What are cookies?

A computer *cookie* is more formally known as an HTTP cookie, a web cookie, an Internet cookie, or a browser cookie. The name is a shorter version of “magic cookie,” which is a term for a packet of data that a computer receives and then sends back without changing or altering it.

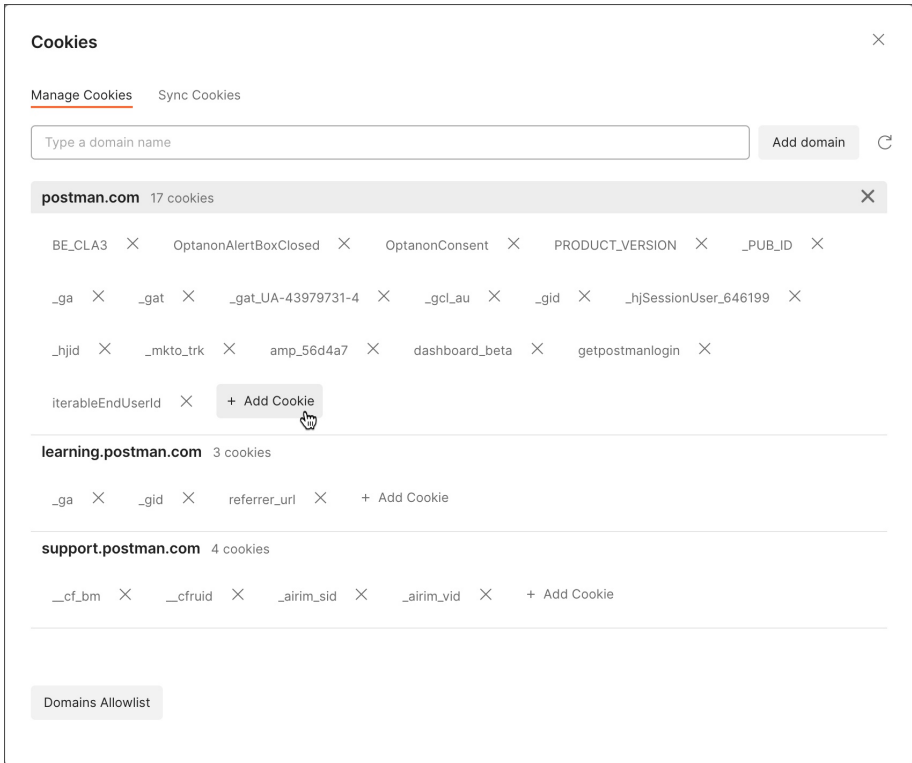
A cookie typically has two pieces of data: a unique ID for each user and a site name. Cookies enable websites to retrieve this information when you revisit them, so that they can remember you and your preferences and tailor page content for you based on this information. Without cookies, you’d have to sign in again after you leave a site or rebuild your shopping cart if you closed a web page. This makes cookies an important a part of the internet experience.

Using the cookie manager

To manage cookies in Postman, open a request, then select **Cookies** (under **Send**).



The **Manage Cookies** window displays a list of domains and the cookies associated with each one. If you want to view cookies for a domain that isn't present in the list, you can **add a domain**.



Creating cookies

To add a new cookie for a domain, select **+ Add Cookie**. A pre-generated cookie string compliant with **HTTP State Management standards** <<https://datatracker.ietf.org/doc/html/rfc6265#section-4.1>> is created.

```
<cookieName>=<cookieValue>; path=/; domain=.domain.com; HttpOnly; Secure; Expires=Tue, 19 Jan 2038 03:14:07 GMT;
```

Postman supports the following attributes:

cookieName, **cookieValue** - The name of the cookie and the value stored in it.

Domain - The domain Postman will send the cookie to.

Path - The URL path that the cookie is restricted to. If the path is `/`, the cookie will be sent to all requests in the specified domain.

HttpOnly - If present, the cookie won't be accessible to the client-side scripts run on the page (for example, with `document.cookie` in JavaScript). The cookie will only be added to the cookie header in requests that are made. This field doesn't have an effect on Postman's behavior.

Secure - If present, the cookie is only sent when the URL begins with `https://` and won't be sent over an insecure connection.

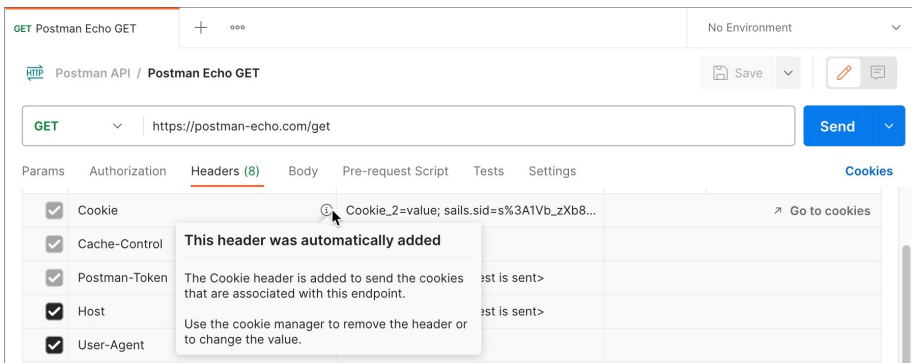
Expires - The time after which the cookie will expire and not be sent by Postman.

Select **Save** to save the cookie to the cookie store under the relevant domain.

You can also add or edit the cookies in a response with the **Set-Cookie header** <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie>>.

Sending cookies with a request

When you make a request to a domain you have added a cookie to, the cookie will automatically appear in your request **Headers** tab. If the cookie isn't visible, select **hidden** to show autogenerated headers.



You can't override cookie headers directly in the **Headers** tab. Edit the cookie in the cookie manager, or delete the cookie and set your request headers manually. You can add cookies in the cookie manager as well as on the **Headers** tab, and Postman will merge the cookies before sending the request.

Adding a domain

If you want to view or add cookies for a domain that isn't present in the cookie manager, you can add a domain. To add a new domain, enter the domain name in the box (don't include a port number or `http://`) and select **Add domain**. You can then add cookies for the new domain.

Updating a cookie

To update an existing cookie for a domain, select the cookie you want to edit. You can edit any property, and select **Save** to update.

Deleting a cookie

To delete a domain and all cookies associated with it, select ✕ next to the domain. To delete an individual cookie, select ✕ next to the cookie.

Accessing cookies in scripts

When you add a domain to the allowlist, cookies for that domain can be accessed in scripts.

To add a domain to the allowlist, do the following:

- Open a request, then select **Cookies** (under **Send**).
- In the **Manage Cookies** window, select **Domains Allowlist**.
- Enter the domain to be allowed and select **Add**.

Creating cookies programmatically

You can programmatically create and delete cookies, instead of relying on the graphical interface. This gives you a greater degree of control over cookies.

The first step to perform in any kind of operation on cookies is to create a *cookie jar*, an object that will contain the cookies and the methods that will be used to operate on cookies.

Creating a cookie jar

To create a cookie jar, use the `pm.cookies.jar()` method. This creates an object to contain the cookies and methods for accessing them.

```
const cookieJar = pm.cookies.jar();
```

Creating a cookie

After a cookie jar is created, you can place cookies into it. Set a cookie using the `.set()` function. This function takes a URL, a cookie name, and a cookie value:

```
// create a cookie
cookieJar.set(URL, cookie name, cookie value, callback(error, cookie));
```

You can also set a PostmanCookie or its compatible cookie object using the `.set()` function:

```
// create a PostmanCookie
cookieJar.set(URL, { name: cookie name, value: cookie value, httpOnly: true }, callback (error, cookie));
```

Get a cookie

To retrieve a cookie, use the `.get()` function. This function takes a URL and the name of the required cookie. It returns the value of cookie.

```
// get the created cookie
cookieJar.get(URL, cookie name, callback(error, cookie));
```

Get all the cookies

To get all the cookies for a particular URL that are in the cookie jar, use the `.getAll()` function. This function takes a URL and returns all the cookies for that URL:

```
// get the created cookies
cookieJar.getAll(URL, callback(error, cookies));
```

Delete a cookie

To delete a cookie, use the `.unset()` function. This function takes a URL and the name of the cookie to be removed:

```
// Delete the created cookie
cookieJar.unset(URL , cookie name, callback (error));
```

Delete all the cookies

To clear all cookies for a URL, use the `.clear()` function. This function takes the URL you want to remove cookies for. The `.clear()` function removes all cookies for a particular URL but doesn't remove all the cookies in the jar, as there may be cookies for more than one URL in the cookie jar.

```
// delete the set cookies
cookieJar.clear(URL, callback (error));
```

Deleting and then setting cookies in sequence

To clear all cookies for a URL `.clear()` and then place a cookie into a cookie jar `.set()`, use a callback function.

Function calls execute asynchronously. Use a callback function to ensure functions execute in sequence:

```
cookieJar.clear(URL, (error) => {
  jar.set(URL, cookie name, cookie value, callback(error, cookie));
});
```

Properties not supported

The following properties aren't supported by Postman:

SameSite

Cookie Prefixes:

__Secure-

__Host-

Syncing cookies

Postman can capture cookies for a browser or client application using the Postman proxy or Postman Interceptor. For the domains you select, captured cookies are continuously synced with the Postman cookie jar. This enables you to use any authentication sessions in your browser or client application to make API calls in Postman.

Learn more about **[capturing cookies with the Postman proxy or Postman Interceptor](#)**

Last modified: 2023/08/23

[← Specifying examples](#)

[Working with certificates →](#)

Product

- [What is Postman?](https://www.postman.com/product/what-is-postman/)
<https://www.postman.com/product/what-is-postman/>
- [API repository](https://www.postman.com/product/api-repository/)
<https://www.postman.com/product/api-repository/>
- [Tools](https://www.postman.com/product/tools/)
<https://www.postman.com/product/tools/>
- [Governance](https://www.postman.com/product/governance/)
<https://www.postman.com/product/governance/>
- [Workspaces](https://www.postman.com/product/workspaces/)
<https://www.postman.com/product/workspaces/>
- [Integrations](https://www.postman.com/product/integrations/)
<https://www.postman.com/product/integrations/>
- [Enterprise](https://www.postman.com/postman-enterprise/)
<https://www.postman.com/postman-enterprise/>
- [Plans and pricing](https://www.postman.com/pricing/)
<https://www.postman.com/pricing/>
- [Download the app](https://www.postman.com/downloads/)
<https://www.postman.com/downloads/>
- [Support Center](https://support.postman.com/hc/en-us)
<https://support.postman.com/hc/en-us>

API Categories

- [App Security](https://www.postman.com/category/app-security)
<https://www.postman.com/category/app-security>
- [Payments](https://www.postman.com/category/payments)
<https://www.postman.com/category/payments>



Company

- [About](https://www.postman.com/company/about-postman/)
<https://www.postman.com/company/about-postman/>
- [Careers and culture](https://www.postman.com/company/careers/)
<https://www.postman.com/company/careers/>
- [Press and media](https://www.postman.com/company/press-media/)
<https://www.postman.com/company/press-media/>
- [Contact us](https://www.postman.com/company/contact-us/)
<https://www.postman.com/company/contact-us/>
- [Partner program](https://www.postman.com/partner-program/)
<https://www.postman.com/partner-program/>

Legal and Security

- [Terms of Service](https://www.postman.com/legal/terms/)
<https://www.postman.com/legal/terms/>
- [Trust and Safety](https://www.postman.com/trust/)
<https://www.postman.com/trust/>
- [Privacy policy](https://www.postman.com/legal/privacy-policy/)
<https://www.postman.com/legal/privacy-policy/>
- [Cookie notice](https://www.postman.com/legal/cookies/)
<https://www.postman.com/legal/cookies/>

Social

-  Twitter <https://twitter.com/getpostman>
-  LinkedIn
<https://www.linkedin.com/company/postman-platform>

Financial Services

<<https://www.postman.com/category/financial-services>>

DevOps

<<https://www.postman.com/category/devops>>

Developer Productivity

<<https://www.postman.com/category/developer-productivity>>

Data Analytics

<<https://www.postman.com/category/data-analytics>>

Communication

<<https://www.postman.com/category/communication>>

Artificial Intelligence

<<https://www.postman.com/category/artificial-intelligence>>



GitHub <<https://github.com/postmanlabs>>



YouTube

<<https://www.youtube.com/c/Postman>>



Twitch <<https://www.twitch.tv/getpostman>>

