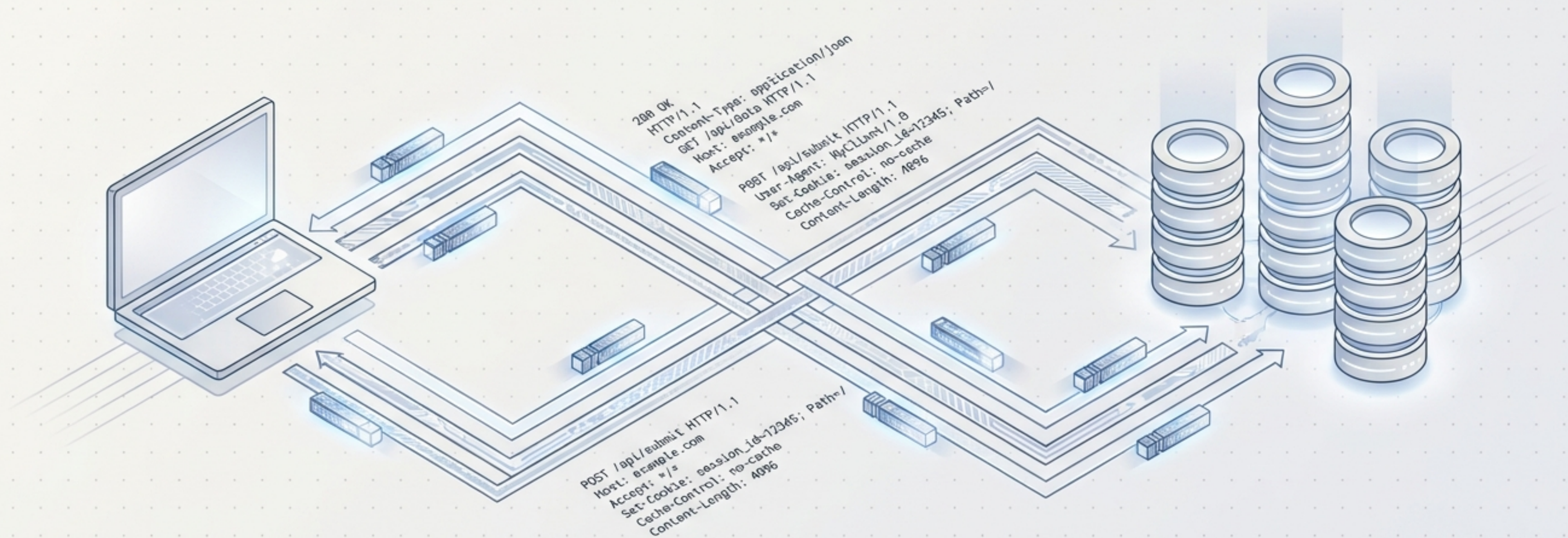


The Verbs of the Web

A visual guide to the actions browsers, apps, and servers use to communicate.



URLs are destinations, but methods are the actions.

[POST] https://api.store.com/orders

The **Verb** (What we want to do)

The **Noun** (Where)



Getting information
(Reading)



Creating new
things



Updating existing
things

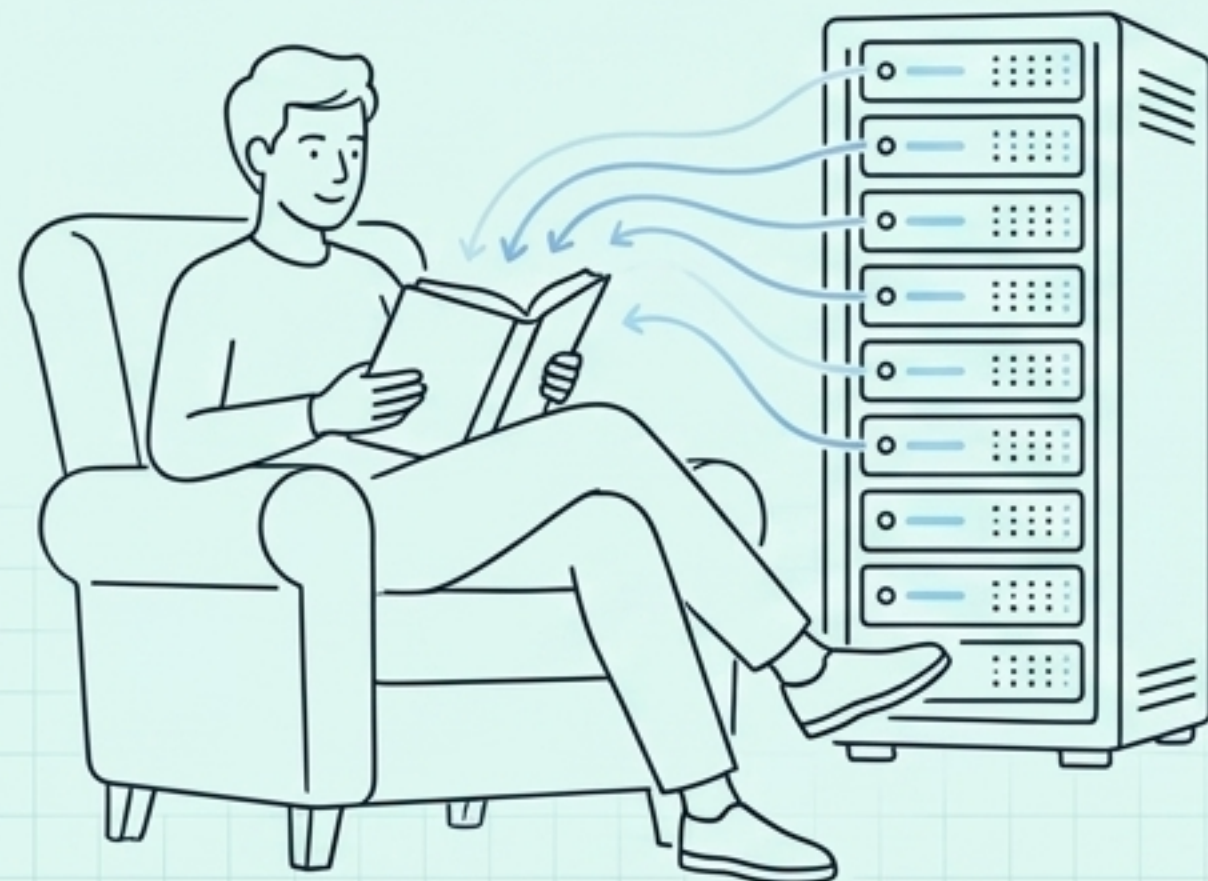


Deleting
things

The foundational rule of the web separates looking from touching.

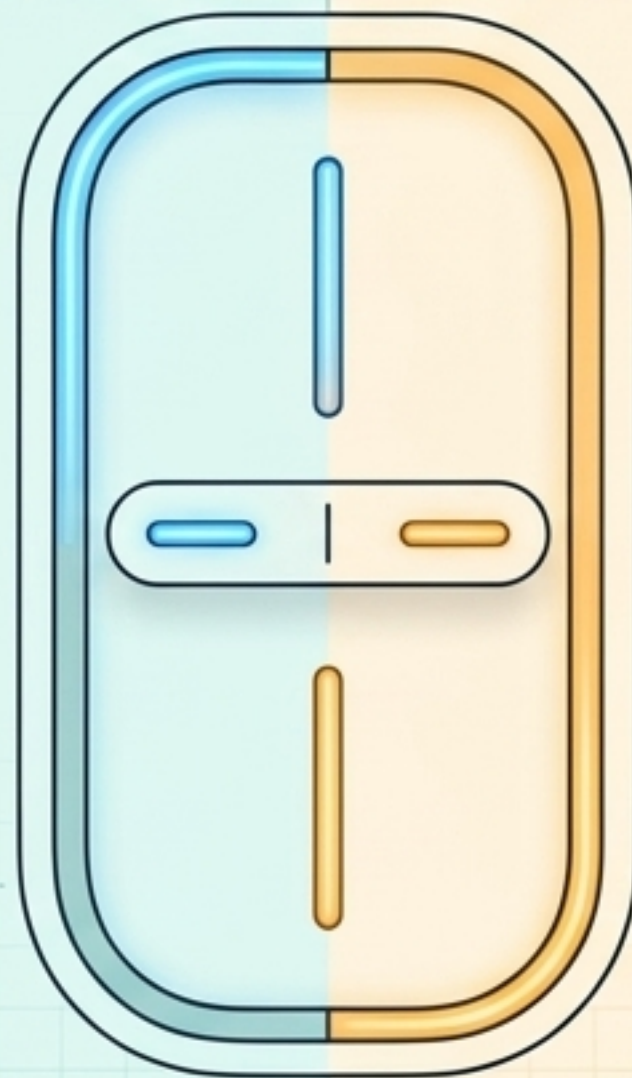
SAFE METHODS

Safe methods don't change anything. They only look at the information.



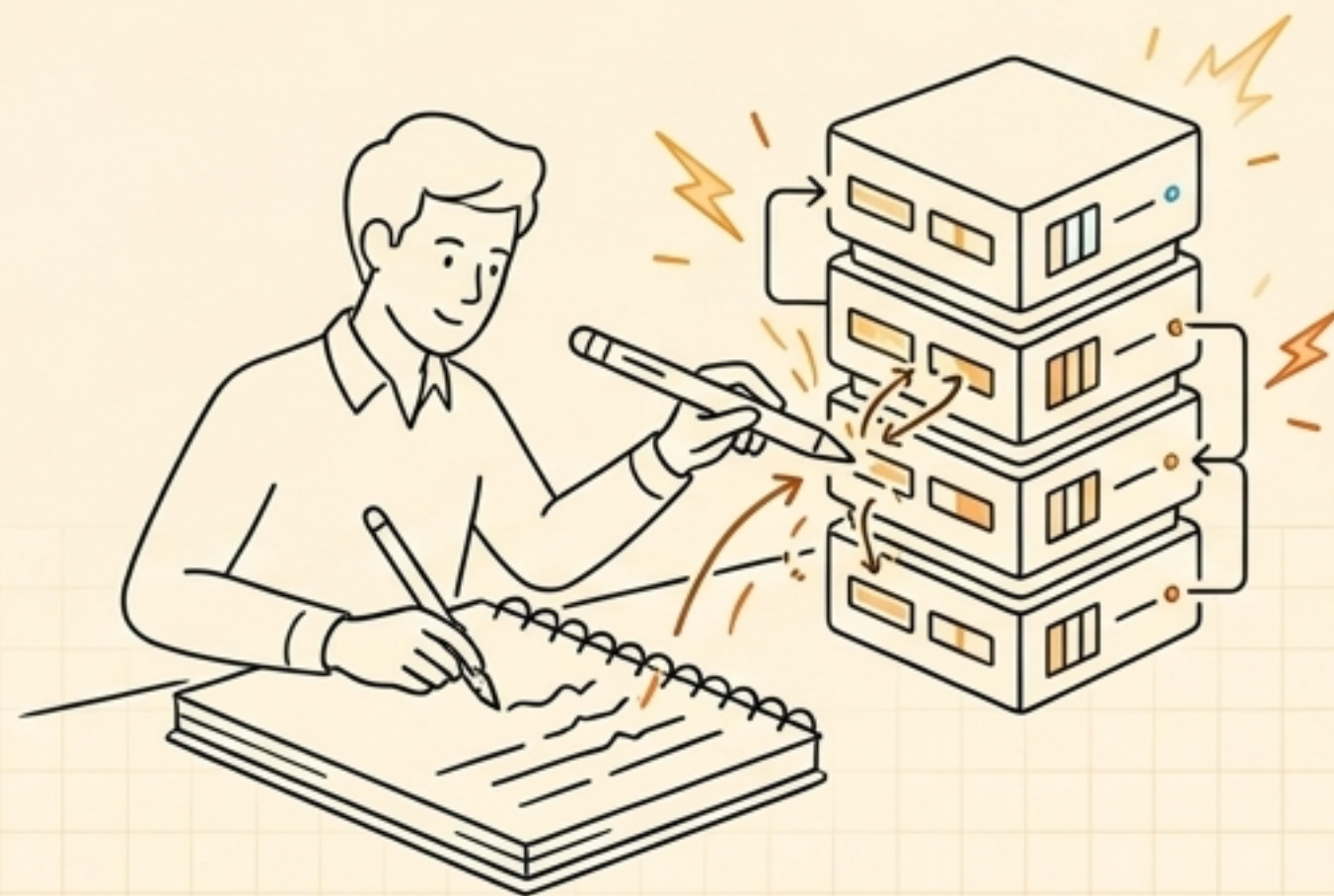
GET HEAD OPTIONS

SAFE (LOOKING) → UNSAFE (TOUCHING)



UNSAFE METHODS

Unsafe methods change data. They modify, create, or destroy server state.



POST PUT PATCH DELETE

GET securely retrieves information without changing the server.

Blueprint Action Card

Simple Meaning:

"Show me this information."

Purpose:

Read data from the server.

Body Included?:

Usually no.

Safe

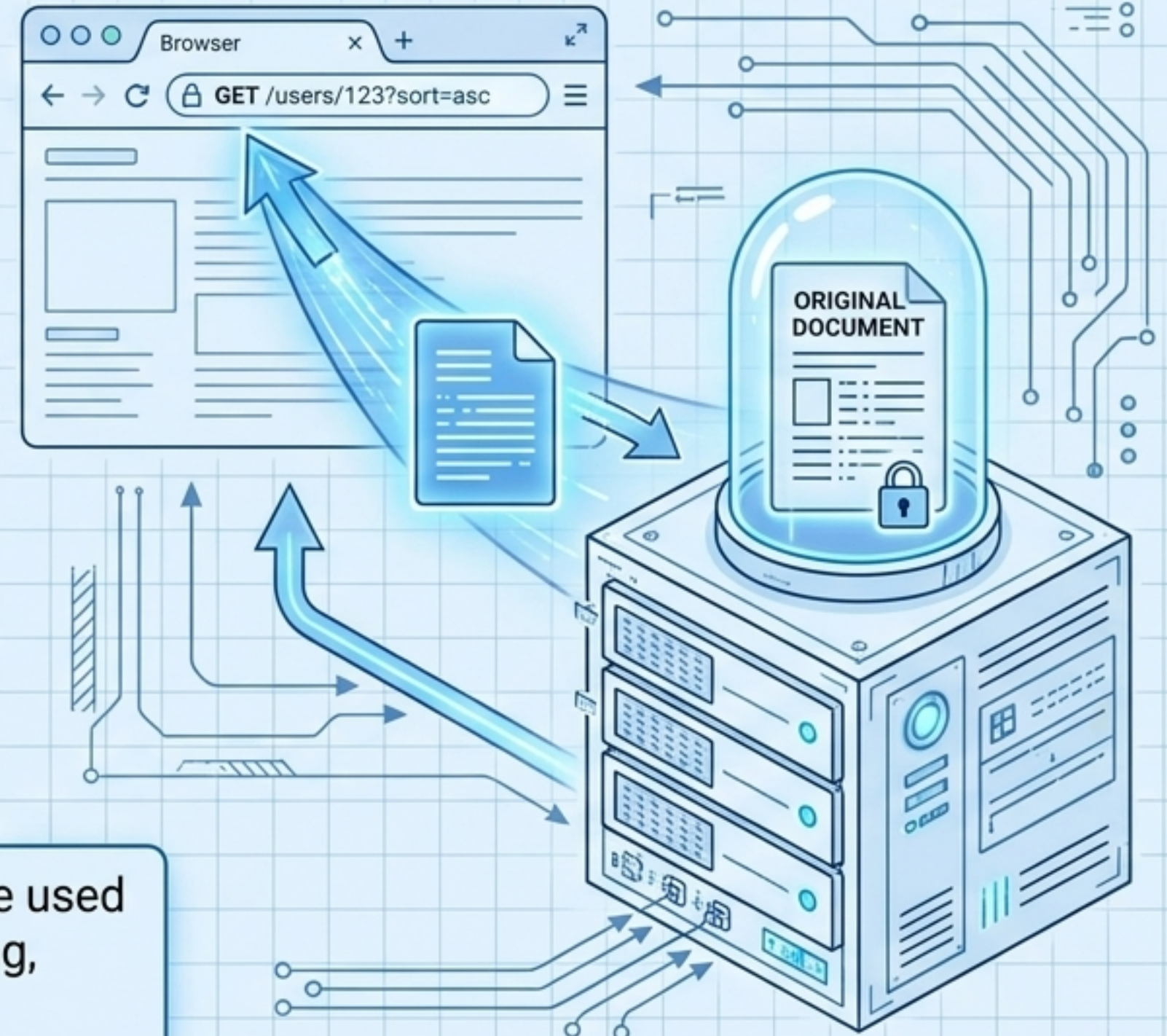
Cacheable

Bookmarkable

Linkable

```
GET /users/123?sort=asc
```

Query parameters are used for pagination, filtering, sorting, or searching.



POST submits payload data to create something entirely new.

Blueprint Action Card

Simple Meaning:

“Create something new with this data.”

Purpose:

Submissions (logins, signups), file uploads, creating records.

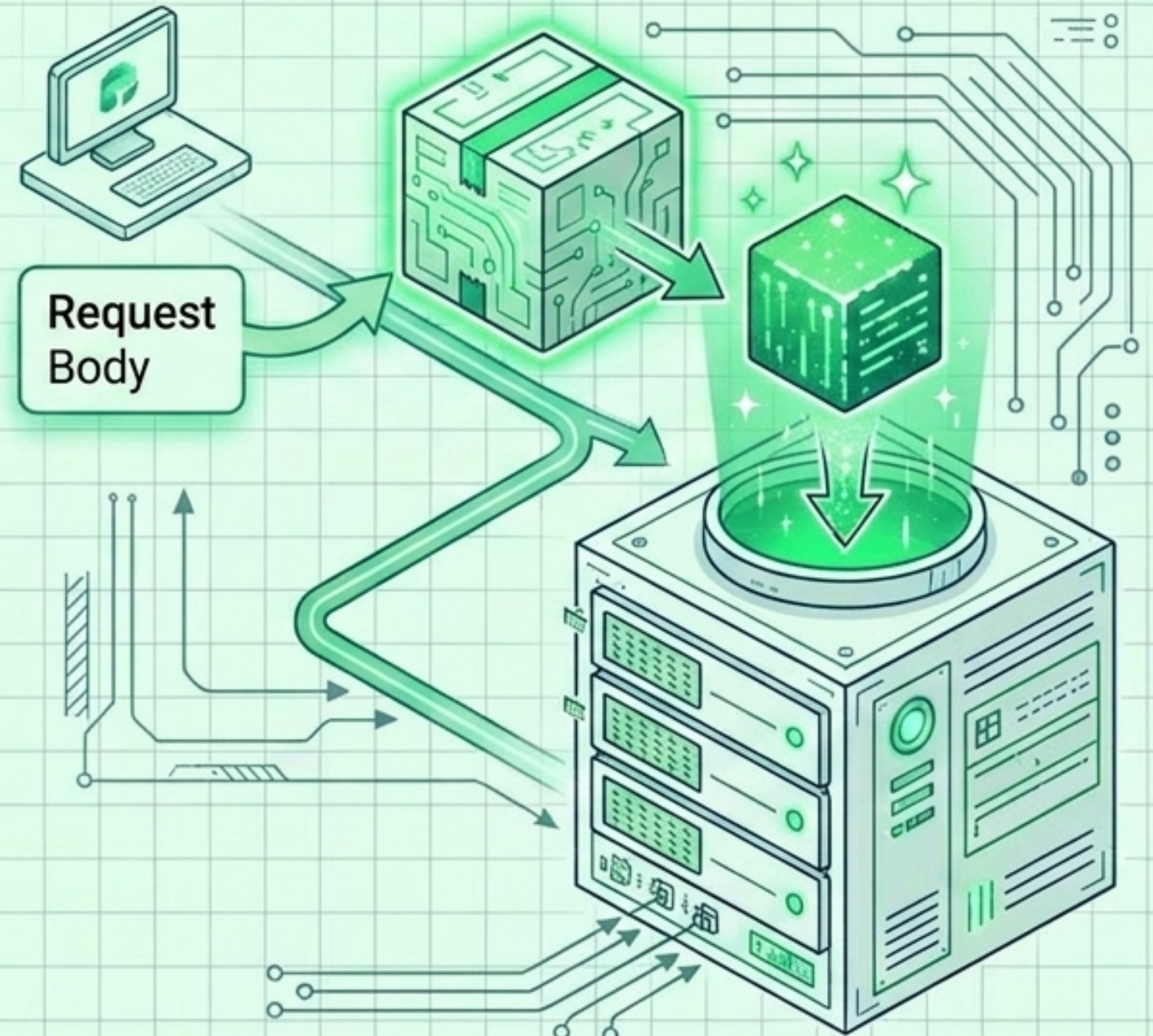
Body Included?:

Yes (Mandatory).

Unsafe

Not Cacheable

```
POST /users {  
  name: User,  
  email: new@email.com  
}
```

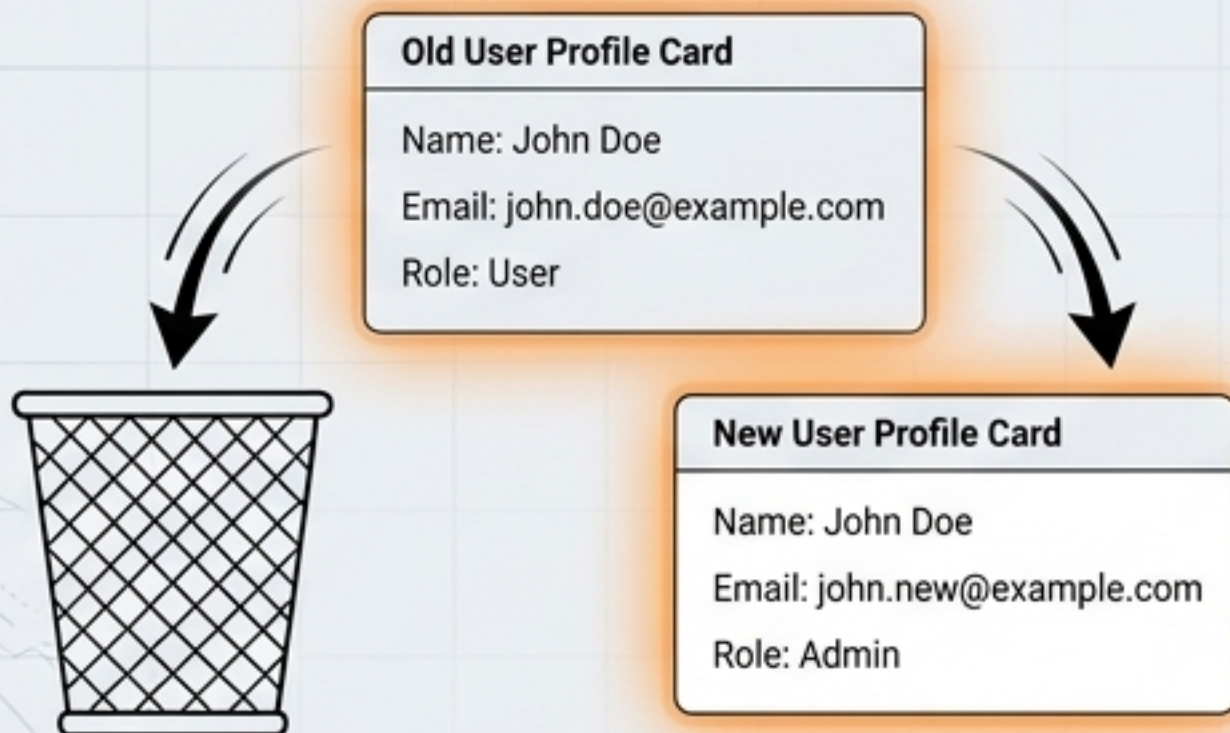


Updating data requires choosing between replacement and modification.

PUT

The Replacer

Rule: Replace the whole thing.

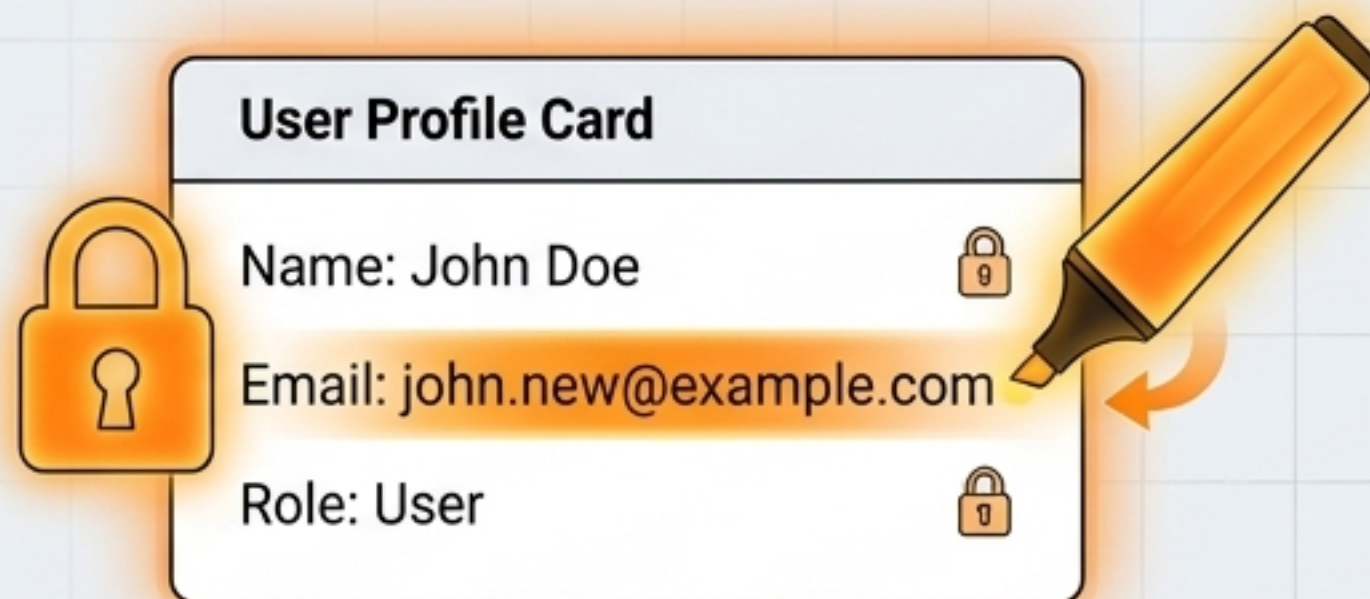


Note: You must send the complete updated version. If a field is left out, it gets erased.

PATCH

The Modifier

Rule: Update only certain fields.



Note: You only send the specific data you want to change.

DELETE permanently removes a resource from the server.

Blueprint Action Card

Simple Meaning:

“Remove this.”

Purpose:

Deleting database records, removing files, destroying relationships.

Body Included?:

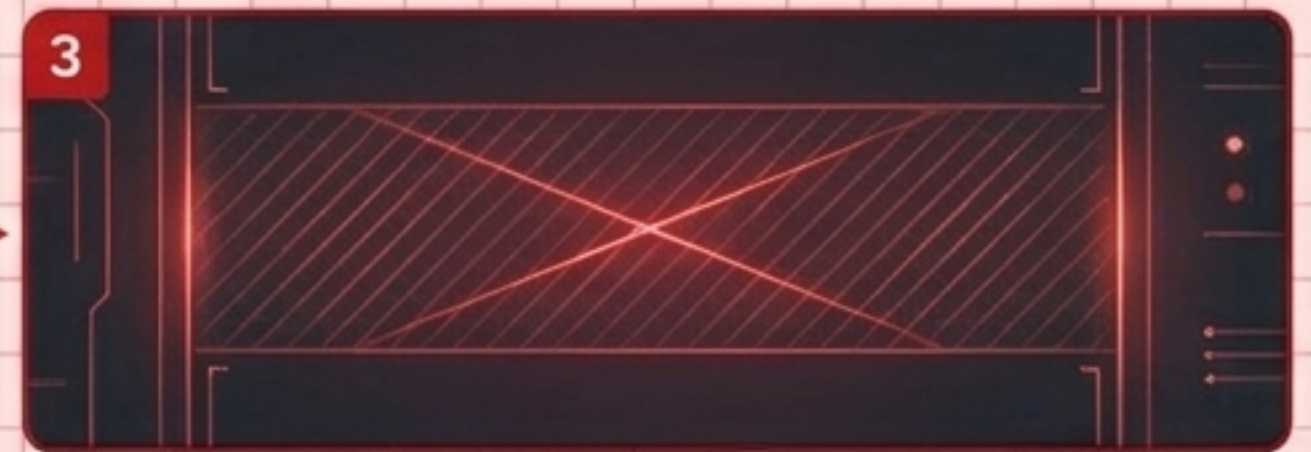
Sometimes (but rarely needed).

Unsafe

Not Cacheable

Not Bookmarkable

```
DELETE /comments/10
```



Utility methods handle metadata and communication permissions.

HEAD



Purpose: Get information about the file without downloading it (checking file size, cache validation).

OPTIONS



Purpose: Ask 'What's allowed here?' Used for API discovery and browser preflight requests (CORS checks) to establish communication capabilities before sending real data.

Safety rules dictate exactly how browsers handle your requests.



Bookmarks & Sharing

Only **GET** works.

If a user tries to share a URL, the browser will always attempt to **GET** it.



Caching

GET and **HEAD** only.

Browsers only save local copies of these requests to speed up loading.

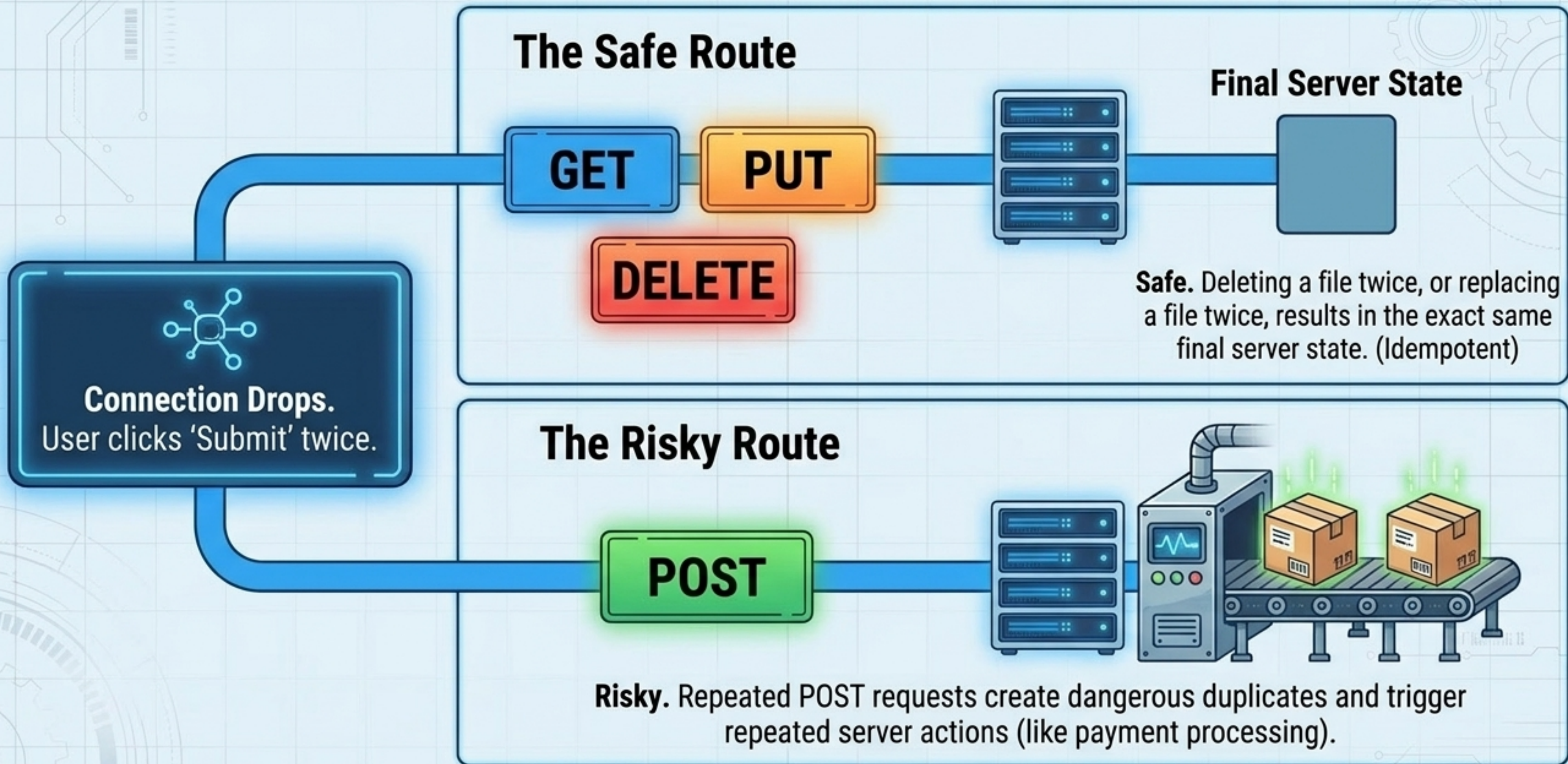
Username
Password
Submit →

HTML Forms

GET and **POST** only.

Standard built-in HTML forms cannot update or delete. Need to **PUT** or **DELETE**? You must use JavaScript or a specialized API tool.

Retrying failed requests carries hidden risks depending on the method.



Mismatched methods create dangerous architectural flaws.

Don't Do This



GET /delete-user/10

Because **GET** is safe, browsers and search engines might pre-fetch or crawl this URL, accidentally triggering massive data deletion without user input!



POST /get-products

Using **POST** just to fetch data breaks expectations. The browser refuses to cache it, causing slower load times and unnecessary server strain.

Do This

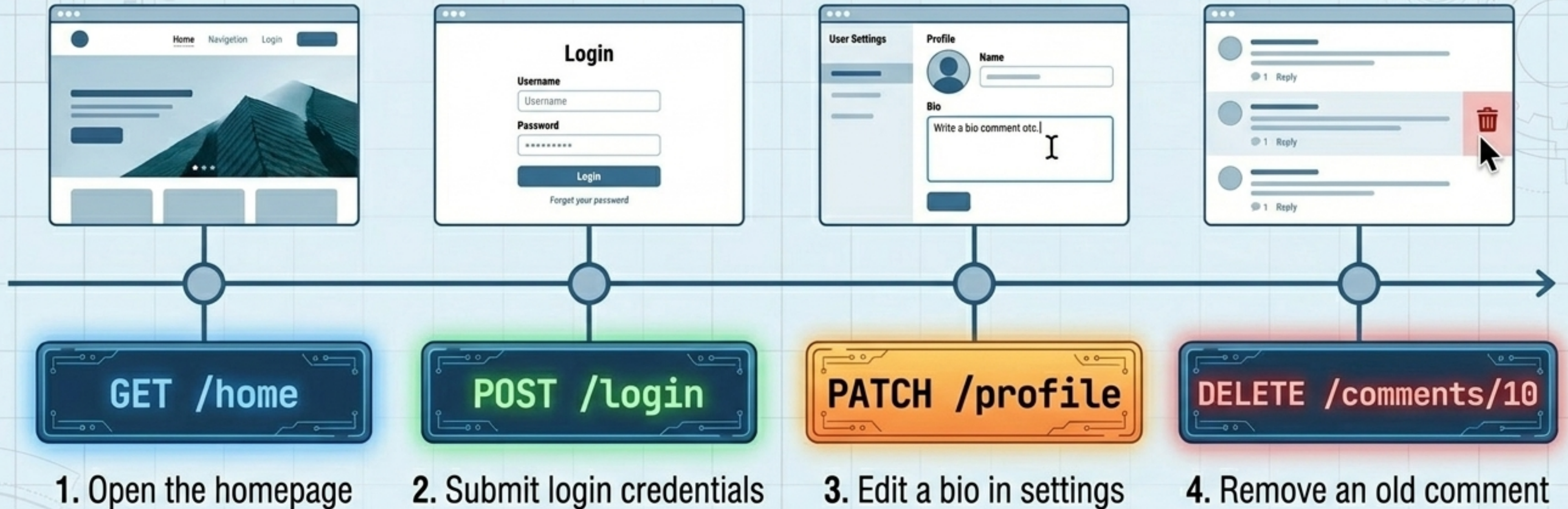


DELETE /users/10



GET /products

Every click in a user session triggers a specific structural verb.



Beautiful UI is simply a graphical wrapper around precise HTTP methods.

The Complete API Operator's Matrix

Method	Purpose	Safe?	Has Body?	Bookmarkable?	Safe to Retry?
GET	Read data	Yes	Usually No	Yes	Yes
POST	Create new resources	No	Yes	No	Risky
PUT	Replace whole resource	No	Yes	No	Yes
PATCH	Update parts of resource	No	Yes	No	Depends
DELETE	Remove a resource	No	Sometimes	No	Yes
HEAD	Get metadata only	Yes	No	No	Yes
OPTIONS	Ask what's allowed	Yes	Usually No	No	Yes

Think of them as the verbs of the web—use the right action for the right task.