When do you use POST and when do you use GET?

Asked 16 years, 3 months ago Modified 2 years, 8 months ago

Viewed 277k times

417

From what I can gather, there are three categories:

1. Never use `GET` and use `POST`
2. Never use `POST` and use `GET`
3. It doesn't matter which one you use.

Am I correct in assuming those three cases? If so, what are some examples from each case?

http-post **http-get**

Share

Improve this question

Follow

edited Jan 17, 2015 at 20:23

NightFury

13.5k ● 6 ● 73 ● 122

asked Sep 5, 2008 at 19:05

Thomas Owens

116k ● 99 ● 317 ● 436

5 That is actually absolutely not true. GET and POST are both visible to the same extent, if you check out the headers sent

by your browser you'll see a list of the key-value pairs that you post – Velimir Tchatchevsky Sep 17, 2015 at 10:49

3 w3.org/2001/tag/doc/whenToUseGet.html – icc97 Oct 23, 2015 at 9:15

2 There is no standard way to encode more than name -> value pairs into query strings so unless your requests are very basic(i.e. no arrays or nested data structures) you should consider POST only which provides a body field that you can use with encoding formats (JSON, XML etc).
– themihai Jul 15, 2016 at 17:23

See this answer: stackoverflow.com/a/63170529/989468
– Chiwda Aug 26, 2020 at 8:07

27 Answers

Sorted by: Highest score (default)

458

Use `POST` for destructive actions such as creation (I'm aware of the irony), editing, and deletion, because you can't hit a `POST` action in the address bar of your browser. Use `GET` when it's safe to allow a person to call an action. So a URL like:

```
http://myblog.org/admin/posts/delete/357
```

Should bring you to a confirmation page, rather than simply deleting the item. It's far easier to avoid accidents this way.

`POST` is also more secure than `GET`, because you aren't sticking information into a URL. And so using `GET` as the

`method` for an HTML form that collects a password or other sensitive information is not the best idea.

One final note: `POST` can transmit a larger amount of information than `GET`. 'POST' has no size restrictions for transmitted data, whilst 'GET' is limited to 2048 characters.

Share Improve this answer

Follow

edited Jun 6, 2018 at 20:33

vs97

5,859 ● 3 ● 31 ● 43

answered Sep 5, 2008 at 19:12

Brian Warshaw

23k ● 9 ● 54 ● 72

97 Responses to GET requests might be cahched. Responses to POSTs must not. – mkoeller Dec 5, 2008 at 10:06

45 How does not sticking info in the URL make it more secure? (Btw, I am one of those who believes that a false sense of security is more dangerous, than not having security at all). – ePharaoh Apr 15, 2009 at 12:13

50 @ePharaoh: It stops people reading passwords by looking over the users shoulder at the address bar. – Quentin Jun 3, 2009 at 9:59

43 @ePharaoh: "Exposing slightly less data to a casual observer" would be probably a better formulation than "more secure" - URLs may end up many places, like logs, referers, caches. You are of course, right, this doesn't increase security - but it limits the worst insecure practices (see also: thedailywtf.com/Articles/The_Spider_of_Doom.aspx) – Piskvor left the building Jun 3, 2009 at 10:06

31 @David Dorward: Or by it's more common name: shoulder attack – Idan K Aug 29, 2010 at 11:40

253

In brief

- Use `GET` for `safe and` `idempotent` requests
- Use `POST` for `neither safe nor idempotent` requests

In details There is a proper place for each. Even if you don't follow RESTful principles, a lot can be gained from learning about REST and how a resource oriented approach works.

> A RESTful application will `use GETs` for operations which are both `safe and idempotent`.

A `safe` operation is an operation which does `not change the data` requested.

An `idempotent` operation is one in which the result will `be the same` no matter how many times you request it.

It stands to reason that, as GETs are used for *safe* operations they are automatically also *idempotent*. Typically a GET is used for retrieving a resource (a question and its associated answers on stack overflow for example) or collection of resources.

> A RESTful app will use `PUTs` for operations which are `not safe but idempotent`.

I know the question was about GET and POST, but I'll return to POST in a second.

Typically a PUT is used for editing a resource (editing a question or an answer on stack overflow for example).

> A `POST` would be used for any operation which is `neither safe or idempotent`.

Typically a POST would be used to create a new resource for example creating a NEW SO question (though in some designs a PUT would be used for this also).

If you run the POST twice you would end up creating TWO new questions.

> There's also a DELETE operation, but I'm guessing I can leave that there :)

Discussion

In practical terms modern web browsers typically only support GET and POST reliably (you can perform all of these operations via javascript calls, but in terms of entering data in forms and pressing submit you've generally got the two options). In a RESTful application

the POST will often be overriden to provide the PUT and DELETE calls also.

But, even if you are not following RESTful principles, it can be useful to think in terms of using GET for retrieving / viewing information and POST for creating / editing information.

You should never use GET for an operation which alters data. If a search engine crawls a link to your evil op, or the client bookmarks it could spell big trouble.

Share Improve this answer

Follow

edited May 30, 2019 at 6:38

01000001

863 ● 3 ● 9 ● 22

answered Sep 5, 2008 at 19:18

reefnet_alex

9,745 ● 5 ● 34 ● 32

if you will create APIREST resource for login which you will choice, this is safe and is idempotent, i guest it.
– jhonny lopez Apr 28, 2016 at 3:51

2 A safe get is not automatically idempotent. The result set may be different with the same no destructive query. – RichieHH Oct 5, 2017 at 18:55

4 The way you wrote it, something like "GET currenttime" would be wrong because it is not idempotent (in the sense that repeated queries may produce different results); in fact *anything* queried for may change over time. So one should express idempotence rather in terms of *side effects* of the query *itself.* Since just asking for the current time has no *side effects*, this is - as one might expect - a perfect candidate for

GET and not POST. – Hagen von Eitzen Aug 4, 2018 at 11:00

what if I want to view data, but I need to pass ana array or a JSON as a a parameter, is still viable to stringify the array and send it as GET, or in this case is it okay to just use POST and send the array in the body? – A.J Alhorr Aug 25, 2020 at 12:23

Usually in a GET request, any parameters exist in the query string of the URL. Now, there are no restrictions within the HTTP spec that prevent you from having a non-empty GET request body, but some server configurations may not allow it. I think Elastic search's API allows info in the body of the GET request, for example. It's all preferential nowadays. – spencer741 Jun 23, 2021 at 13:07

87

Use GET if you don't mind the request being repeated (That is it doesn't change state).

Use POST if the operation does change the system's state.

Share Improve this answer

Follow

answered Sep 5, 2008 at 19:07

Douglas Leeder

53.3k ● 9 ● 99 ● 138

1 Since a form changes the system's state, why the default method of the HTML form tag is GET? – tirenweb Aug 24, 2011 at 17:33

5 @user248959 Does a search box change the visible state? The default was set a long time ago, probably almost by accident. I haven't delved into the history to even know if

POST was an option at the point forms were an option.
– Douglas Leeder Aug 25, 2011 at 7:23

@ziiweb Even if the majority of use cases of <form> is POST, it is better to define the dafault to be a "harmless" GET. This may seem absurd from a security standpoint when it leads to data exposed in log files etc., but it is fail-safe with regards to the server-side data (the serve should not modify data upon a GET). I suppose, one would set the focus differently today (preferably by dropping any default and making `method` mandatory) – Hagen von Eitzen Aug 4, 2018 at 11:09

Suppose I have an endpoint that accepts a file as input, does some processing on the file (example - extract data based on regex) and returns JSON data, then can I use GET request to upload a file to the server. Or should I use POST request?
– variable Dec 5, 2019 at 11:24

that was very concise – Yehia A.Salam Feb 8, 2023 at 21:28

70

Short Version

GET: Usually used for submitted search requests, or any request where you want the user to be able to pull up the exact page again.

Advantages of GET:

- URLs can be bookmarked safely.
- Pages can be reloaded safely.

Disadvantages of GET:

- Variables are passed through url as name-value pairs. (Security risk)

- Limited number of variables that can be passed. (Based upon browser. For example, Internet Explorer is limited to 2,048 characters.)

POST: Used for higher security requests where data may be used to alter a database, or a page that you don't want someone to bookmark.

Advantages of POST:

- Name-value pairs are not displayed in url. (Security += 1)
- Unlimited number of name-value pairs can be passed via POST. Reference.

Disadvantages of POST:

- Page that used POST data cannot be bookmark. (If you so desired.)

Longer Version

Directly from the Hypertext Transfer Protocol -- HTTP/1.1:

> ### 9.3 GET
>
> The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI. If the Request-URI refers to a data-producing process, it is the produced data which shall be returned as the entity in the

> response and not the source text of the process, unless that text happens to be the output of the process.
>
> The semantics of the GET method change to a "conditional GET" if the request message includes an If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match, or If-Range header field. A conditional GET method requests that the entity be transferred only under the circumstances described by the conditional header field(s). The conditional GET method is intended to reduce unnecessary network usage by allowing cached entities to be refreshed without requiring multiple requests or transferring data already held by the client.
>
> The semantics of the GET method change to a "partial GET" if the request message includes a Range header field. A partial GET requests that only part of the entity be transferred, as described in section 14.35. The partial GET method is intended to reduce unnecessary network usage by allowing partially-retrieved entities to be completed without transferring data already held by the client.
>
> The response to a GET request is cacheable if and only if it meets the requirements for HTTP caching described in section 13.

See section 15.1.3 for security considerations when used for forms.

9.5 POST

The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. POST is designed to allow a uniform method to cover the following functions:

- Annotation of existing resources;
- Posting a message to a bulletin board, newsgroup, mailing list, or similar group of articles;
- Providing a block of data, such as the result of submitting a form, to a data-handling process;
- Extending a database through an append operation.

The actual function performed by the POST method is determined by the server and is usually dependent on the Request-URI. The posted entity is subordinate to that URI in the same way that a file is subordinate to a directory containing it, a news article is subordinate to a newsgroup to which it is posted, or a record is subordinate to a database.

> The action performed by the POST method might not result in a resource that can be identified by a URI. In this case, either 200 (OK) or 204 (No Content) is the appropriate response status, depending on whether or not the response includes an entity that describes the result.

Share Improve this answer

Follow

edited Jul 28, 2012 at 5:12

icktoofay

129k ● 23 ● 257 ● 236

answered Jun 3, 2009 at 9:42

Cimplicity

2,732 ● 2 ● 26 ● 26

2 "Page that used post data cannot be bookmarked": well, that's an advantage, no? You probably don't want your data-altering query to be bookmarked. – Piskvor left the building Jun 3, 2009 at 10:11

I suppose if every time post was used you were using it for a security driven purpose then this would be an advantage. Usually it is, but there is that length limit on GET. Maybe, somebody is just passing a ton of non-security related data and would like the page to be bookmarked? Who knows... – Cimplicity Jun 3, 2009 at 10:43

Regarding a disadvantage of GET, namely that "Variables are pased through url as name-value pairs", would MVC eliminate that issue because of routing and the resultant friendly URLs? – MrBoJangles Jun 29, 2011 at 22:06

@MrBoJangles: Using nice URLs does not prevent the 'person looking over shoulder' risk referred to. Side note:

MVC does not require routing with nice URLs and routing with nice URLs does not require MVC; they are sometimes used together, but can also be used separately. – icktoofay Jul 28, 2012 at 5:15

In the .NET world, for all practical purposes, nice url capability = MVC. I suppose you could do some IIS rewrites or some weird code-based ones but they're even less pleasant. MVC, needless to say, for the win. – MrBoJangles Jul 30, 2012 at 14:29

32

The first important thing is the *meaning* of GET versus POST :

- GET should be used to... get... some information **from** the server,
- while POST should be used to send some information **to** the server.

After that, a couple of things that can be noted :

- Using GET, your users can use the "back" button in their browser, and they can bookmark pages
- There is a limit in the size of the parameters you can pass as GET *(2KB for some versions of Internet Explorer, if I'm not mistaken)* ; the limit is much more for POST, and generally depends on the server's configuration.

Anyway, I don't think we could "live" without GET : think of

how many URLs you are using with parameters in the query string, every day -- without GET, all those wouldn't work ;-)

Share Improve this answer

Follow

edited Sep 12, 2016 at 13:01

Vucko

20.8k ● 12 ● 59 ● 108

answered Feb 15, 2010 at 17:49

Pascal MARTIN

401k ● 81 ● 663 ● 667

Well, if everyone used pretty-urls in a GET style: `http://example.com/var1/value1/var2/value2/var3/value3` we could 'technically' not have GET anymore... – Tyler Carter Feb 15, 2010 at 18:07

6 @Chacha102 Except that you still have to GET that resource. Nearly all pages, images, scripts, etc. are loaded in web browsers using GET. – Ryan Feb 15, 2010 at 18:10

3 @Chacha102 - Even the `www.mypage.com/contact/` uses GET internally to something like `index.php?url=/contact/` – Thiago Belem Feb 15, 2010 at 18:17

1 Emphasis on the size limit of GET! Also, GET parameters are included in bookmarks, while POST aren't. And, the user can refresh a GET-requested page but not a POST-requested one (without a warning about resending the info). – Ricket Feb 15, 2010 at 18:18

15

Apart from the length constraints difference in many web browsers, there is also a semantic difference. GETs are supposed to be "safe" in that they are read-only

operations that don't change the server state. POSTs will typically change state and will give warnings on resubmission. Search engines' web crawlers may make GETs but should never make POSTs.

Use GET if you want to read data without changing state, and use POST if you want to update state on the server.

Share Improve this answer

Follow

edited Feb 15, 2010 at 21:03

answered Feb 15, 2010 at 17:50

Mark Byers

837k ● 199 ● 1.6k ● 1.5k

1 +1. This is the key conceptual difference from the rfc from which everything else follows. w3.org/Protocols/rfc2616/rfc2616-sec9.html#sec9.3 – Frank Farmer Feb 15, 2010 at 18:01

8

My general rule of thumb is to use Get when you are making requests to the server that aren't going to alter state. Posts are reserved for requests to the server that alter state.

Share Improve this answer

Follow

answered Sep 5, 2008 at 19:08

TonyLa

734 ● 4 ● 6

8

One practical difference is that browsers and webservers have a limit on the number of characters that can exist in a URL. It's different from application to application, but it's certainly possible to hit it if you've got `textarea`s in your forms.

Another gotcha with GETs - they get indexed by search engines and other automatic systems. Google once had a product that would pre-fetch links on the page you were viewing, so they'd be faster to load if you clicked those links. It caused **major** havoc on sites that had links like `delete.php?id=1` - people lost their entire sites.

Share Improve this answer

Follow

answered Feb 15, 2010 at 17:48

ceejayoz

180k ● 41 ● 308 ● 380

1 Your webserver probably also has limits on this. – Billy ONeal Feb 15, 2010 at 17:49

Well, there's a limit to POST as well. – chelmertz Feb 15, 2010 at 17:50

1 Great point, @BillyONeal, I've added that in. @chelmertz Yes, but I can change that if I want, and it's much higher. I've POSTed 1 gigabyte files to Apache instances, for example. – ceejayoz Feb 15, 2010 at 17:51

I understand URLs getting indexed by search engines. I don't understand what does that have to do with GET. I mean isn't a URL just a URL? – mfaani Nov 1, 2016 at 18:04

2 @Honey Search engines follow links. Links make GET requests. Search engines don't submit forms (if they did, you'd see Google signing up for an account on your site

every few days) and thus make no POST requests.
– ceejayoz Nov 1, 2016 at 18:07

7

Use GET when you want the URL to reflect the state of the page. This is useful for viewing dynamically generated pages, such as those seen here. A POST should be used in a form to submit data, like when I click the "Post Your Answer" button. It also produces a cleaner URL since it doesn't generate a parameter string after the path.

Share Improve this answer

Follow

answered Sep 5, 2008 at 19:07

Kyle Cronin
79k ● 45 ● 151 ● 167

6

Because GETs are purely URLs, they can be cached by the web browser and may be better used for things like consistently generated images. (Set an Expiry time)

One example from the gravatar page: http://www.gravatar.com/avatar/4c3be63a4c2f539b013787725dfce802?d=monsterid

GET may yeild marginally better performance, some webservers write POST contents to a temporary file before invoking the handler.

Another thing to consider is the size limit. GETs are capped by the size of the URL, 1024 bytes by the standard, though browsers may support more.

Transferring more data than that should use a POST to get better browser compatibility.

Even less than that limit is a problem, as another poster wrote, anything in the URL could end up in other parts of the brower's UI, like history.

Share Improve this answer

Follow

answered Sep 6, 2008 at 8:46

davenpcj
12.7k 5 42 38

6

1.3 Quick Checklist for Choosing HTTP `GET` or `POST`

Use GET if:

```
    The interaction is more like a question (i.e.,
it is a safe operation such as a query, read
operation, or lookup).
```

Use POST if:

```
    The interaction is more like an order, or
    The interaction changes the state of the
resource in a way that the user would perceive
(e.g., a subscription to a service), or
    The user be held accountable for the results
of the interaction.
```

Source.

Share Improve this answer

Follow

edited Jun 20, 2020 at 9:12

Community Bot

1 ● 1

answered Aug 29, 2013 at 12:41

Anagha

760 ● 1 ● 10 ● 16

4

> i dont see a problem using get though, i use it for simple things where it makes sense to keep things on the query string.

Using it to update state - like a GET of `delete.php?id=5` to delete a page - is very risky. People found that out when Google's web accelerator started prefetching URLs on pages - it hit all the 'delete' links and wiped out peoples' data. Same thing can happen with search engine spiders.

Share Improve this answer

Follow

answered Sep 5, 2008 at 19:13

ceejayoz

180k ● 41 ● 308 ● 380

4

There is nothing you can't do per-se. The point is that you're not *supposed* to modify the server state on an HTTP GET. HTTP proxies assume that since HTTP GET does not modify the state then whether a user invokes HTTP GET one time or 1000 times makes no difference. Using this information they assume it is safe to return a

cached version of the first HTTP GET. If you break the HTTP specification you risk breaking HTTP client and proxies in the wild. Don't do it :)

Share Improve this answer

Follow

answered Feb 15, 2010 at 17:53

Gili

89.7k ● 104 ● 410 ● 720

It's not just browsers that count on GET being safe and idempotent: search engine spiders and prefetching browsers (like fasterfox) also rely on this. – Frank Farmer Feb 15, 2010 at 18:05

@gili, finally someone with correct answer. I'm really surprised how many people got this one wrong. thumbs up! – lubos hasko Feb 15, 2010 at 18:34

4

This traverses into the concept of REST and how the web was kinda intended on being used. There is an excellent podcast on Software Engineering radio that gives an in depth talk about the use of Get and Post.

Get is used to pull data from the server, where an update action shouldn't be needed. The idea being is that you should be able to use the same GET request over and over and have the same information returned. The URL has the get information in the query string, because it was meant to be able to be easily sent to other systems and people like a address on where to find something.

Post is supposed to be used (at least by the REST architecture which the web is kinda based on) for pushing

information to the server/telling the server to perform an action. Examples like: Update this data, Create this record.

Share Improve this answer

Follow

edited Jun 29, 2011 at 22:28

MrBoJangles

12.2k ● 17 ● 64 ● 82

answered Sep 5, 2008 at 19:23

kemiller2002

115k ● 28 ● 199 ● 253

"There is an excellent podcast on Software Engineering radio that gives an in depth talk about the use of Get and Post." Where is it? – yeeen Feb 16, 2010 at 9:04

Here's that linkage: se-radio.net/2008/05/episode-98-stefan-tilkov-on-rest I also edited the link above, although I don't have edit rights and it's gotta be peer-reviewed, etc. – MrBoJangles Jun 29, 2011 at 22:09

2 Suppose I have an endpoint that accepts a file as input, does some processing on the file (example - extract data based on regex) and returns JSON data, then can I use GET request to upload a file to the server. Or should I use POST request? – variable Dec 5, 2019 at 11:33

3

POST can move large data while GET cannot.

But generally it's not about a shortcomming of GET, rather a convention if you want your website/webapp to be behaving nicely.

Have a look at
http://www.w3.org/2001/tag/doc/whenToUseGet.html

Share Improve this answer

Follow

answered Feb 15, 2010 at 17:49

cherouvim

31.9k ● 15 ● 105 ● 155

3

From RFC 2616:

> 9.3 **GET**
> The GET method means retrieve whatever information (in the form of an entity) is identified by the Request-URI. If the Request-URI refers to a data-producing process, it is the produced data which shall be returned as the entity in the response and not the source text of the process, unless that text happens to be the output of the process.

> 9.5 **POST**
> The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. POST is designed to allow a uniform method to cover the following functions:
>
> - Annotation of existing resources;

> - Posting a message to a bulletin board, newsgroup, mailing list, or similar group of articles;
> - Providing a block of data, such as the result of submitting a form, to a data-handling process;
> - Extending a database through an append operation.
>
> The actual function performed by the POST method is determined by the server and is usually dependent on the Request-URI. The posted entity is subordinate to that URI in the same way that a file is subordinate to a directory containing it, a news article is subordinate to a newsgroup to which it is posted, or a record is subordinate to a database.
>
> The action performed by the POST method might not result in a resource that can be identified by a URI. In this case, either 200 (OK) or 204 (No Content) is the appropriate response status, depending on whether or not the response includes an entity that describes the result.

edited Feb 16, 2010 at 8:47

answered Feb 15, 2010 at 17:53

Dmytro

2,227 ● 3 ● 21 ● 32

2

I use POST when I don't want people to see the QueryString or when the QueryString gets large. Also, POST is needed for file uploads.

I don't see a problem using GET though, I use it for simple things where it makes sense to keep things on the QueryString.

Using GET will allow linking to a particular page possible too where POST would not work.

Share Improve this answer

Follow

edited Mar 25, 2015 at 7:39

Gajendrasinh Chauhan

3,397 ● 8 ● 41 ● 56

answered Sep 5, 2008 at 19:09

John Boker

83.7k ● 17 ● 100 ● 130

Why cant we use GET for file upload? – variable Dec 5, 2019 at 11:33

1

The original intent was that GET was used for getting data back and POST was to be anything. The rule of thumb that I use is that if I'm sending anything back to the server, I use POST. If I'm just calling an URL to get back data, I use GET.

Share Improve this answer

Follow

answered Sep 5, 2008 at 19:08

Chris Miller

4,899 ● 4 ● 34 ● 50

1

Read the article about HTTP in the Wikipedia. It will explain what the protocol is and what it does:

> **GET**
>
> Requests a representation of the specified resource. Note that GET should not be used for operations that cause side-effects, such as using it for taking actions in web applications. One reason for this is that GET may be used arbitrarily by robots or crawlers, which should not need to consider the side effects that a request should cause.

and

> **POST** Submits data to be processed (e.g., from an HTML form) to the identified resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.

The W3C has a document named URIs, Addressability, and the use of HTTP GET and POST that explains when

to use what. Citing

> 1.3 Quick Checklist for Choosing HTTP GET or POST
>
> - Use GET if:
>   - The interaction is more like a question (i.e., it is a safe operation such as a query, read operation, or lookup).

and

> - Use POST if:
>   - The interaction is more like an order, or
>   - The interaction changes the state of the resource in a way that the user would perceive (e.g., a subscription to a service), or o The user be held accountable for the results of the interaction.
>
> However, before the final decision to use HTTP GET or POST, please also consider considerations for sensitive data and practical considerations.

A practial example would be whenever you submit an HTML form. You specify either *post* or *get* for the form

action. PHP will populate $_GET and $_POST accordingly.

Share Improve this answer

Follow

answered Feb 15, 2010 at 17:54

Gordon

317k ● 76 ● 545 ● 565

1

In PHP, `POST` data limit is usually set by your `php.ini`. `GET` is limited by server/browser settings I believe - usually around `255` bytes.

Share Improve this answer

Follow

edited Jan 17, 2015 at 20:38

atiquratik

1,300 ● 3 ● 28 ● 35

answered Feb 15, 2010 at 19:13

jellyfishtree

1,851 ● 1 ● 10 ● 11

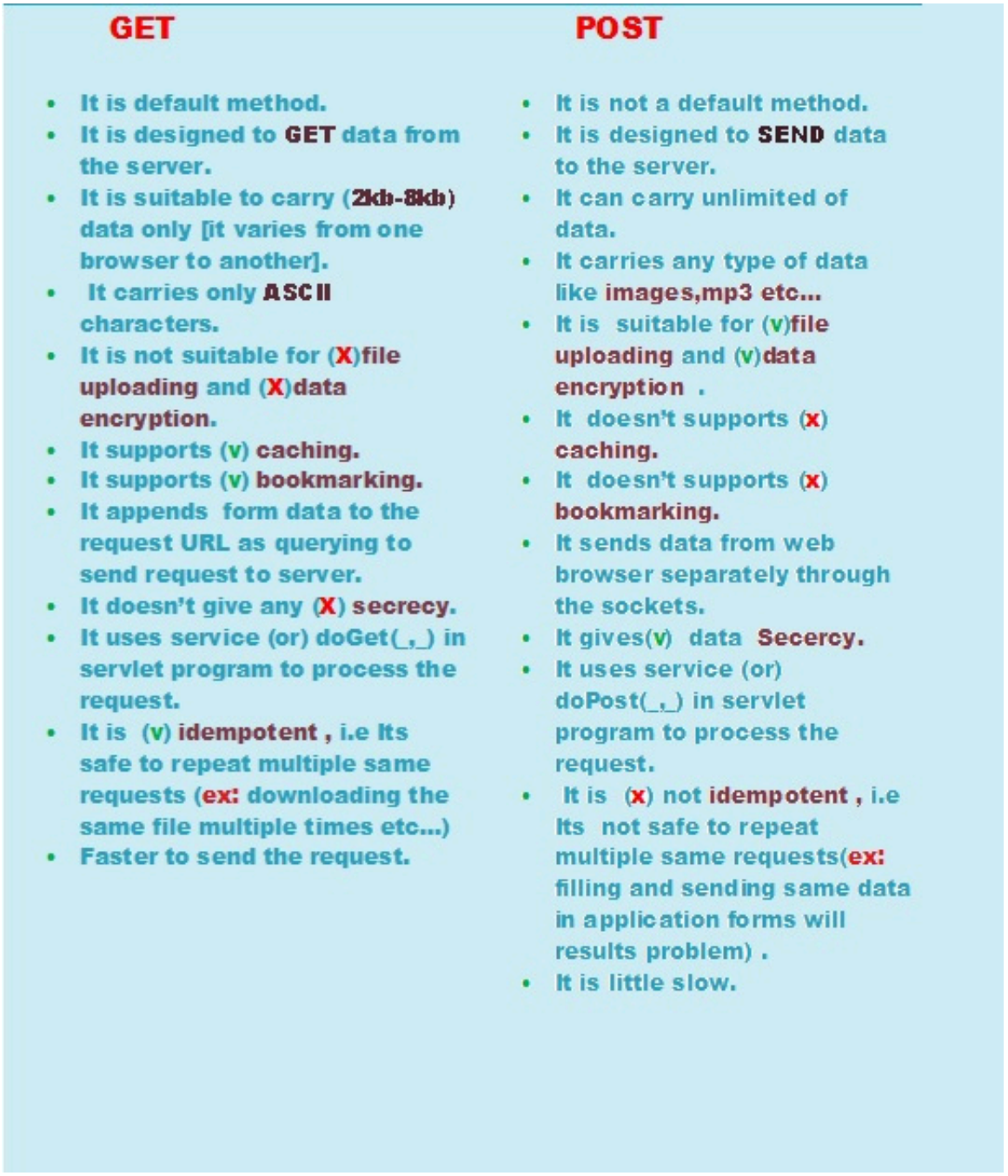
1

From w3schools.com:

> What is HTTP?
>
> The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.
>
> HTTP works as a request-response protocol between a client and server.

> A web browser may be the client, and an application on a computer that hosts a web site may be the server.
>
> Example: A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.
>
> Two HTTP Request Methods: GET and POST
>
> Two commonly used methods for a request-response between a client and server are: GET and POST.
>
> GET – Requests data from a specified resource
> POST – Submits data to be processed to a specified resource

Here we distinguish the major differences:



Share Improve this answer

Follow

edited Mar 15, 2016 at 0:14

Benjamin W.

51.6k ● 19 ● 125 ● 131

answered Mar 13, 2016 at 12:27

Madhusudhan Reddy

25 ● 1 ● 3

1 It would be much better for searchers and readers to enter the content of the image into the answer. Also, the first part of

the answer doesn't really help in answering the question.
– Dave Schweisguth Mar 13, 2016 at 14:30

Copy paste from here - you must properly cite your source and the licence of the source must allow reuse, which I don't think w3schools does. Apart from that, do you really think this adds something that wasn't covered in the other 25 answers?
– Benjamin W. Mar 15, 2016 at 0:07

0

Well one major thing is anything you submit over `GET` is going to be exposed via the URL. Secondly as Ceejayoz says, there is a limit on characters for a URL.

Share Improve this answer

Follow

answered Feb 15, 2010 at 17:50

prodigitalson
60.4k ● 10 ● 104 ● 115

0

Another difference is that POST generally requires two HTTP operations, whereas GET only requires one.

Edit: I should clarify--for common programming patterns. Generally responding to a POST with a straight up HTML web page is a questionable design for a variety of reasons, one of which is the annoying "you must resubmit this form, do you wish to do so?" on pressing the back button.

Share Improve this answer Follow

edited Feb 15, 2010 at 18:14

answered Feb 15, 2010 at 17:49

Plynx

11.5k ● 3 ● 33 ● 33

3 POST does not require 2 http operations. – Billy ONeal Feb 15, 2010 at 17:56

3 post-redirect-get requires 2 operations: en.wikipedia.org/wiki/Post/Redirect/Get – cherouvim Feb 15, 2010 at 17:58

1 POST may require 2 round trips to the server -- a common pattern is to POST with an `expect: 100-continue` header, and then only send data once the server responds with a `100 CONTINUE`. – Frank Farmer Feb 15, 2010 at 18:09

Nice article cherouvim, I never knew the pattern had a name. – Plynx Feb 15, 2010 at 18:16

@cherouvim: Post redirect get does, but plain post does not. You could just as simply have get redirect get with the same results. It has nothing to do with the protocol your form uses for submission. – Billy ONeal Feb 15, 2010 at 19:44

0

As answered by others, there's a limit on url size with get, and files can be submitted with post only.

I'd like to add that one **can** add things to a database with a get and perform actions with a post. When a script receives a post or a get, it can do whatever the author

wants it to do. I believe the lack of understanding comes from the wording the book chose or how you read it.

A script author **should** use posts to change the database and use get only for retrieval of information.

Scripting languages provided many means with which to access the request. For example, PHP allows the use of `$_REQUEST` to retrieve either a post or a get. One should avoid this in favor of the more specific `$_GET` or `$_POST`.

In web programming, there's a lot more room for interpretation. There's what one **should** and what one **can** do, but which one is better is often up for debate. Luckily, in this case, there is no ambiguity. You **should** use posts to change data, and you **should** use get to retrieve information.

Share Improve this answer

Follow

answered Feb 15, 2010 at 18:22

Elizabeth Buckwalter

968 ● 8 ● 20

-1

HTTP Post data doesn't have a specified limit on the amount of data, where as different browsers have different limits for GET's. The RFC 2068 states:

> Servers should be cautious about depending on URI lengths above 255 bytes, because some older client or proxy implementations may not properly support these lengths

Specifically you should the right HTTP constructs for what they're used for. HTTP GET's shouldn't have side-effects and can be safely refreshed and stored by HTTP Proxies, etc.

HTTP POST's are used when you want to submit data against a url resource.

A typical example for using HTTP GET is on a Search, i.e. Search?Query=my+query A typical example for using a HTTP POST is submitting feedback to an online form.

Share Improve this answer

Follow

answered Feb 15, 2010 at 17:52

mythz

143k ● 30 ● 252 ● 396

-1

Gorgapor, `mod_rewrite` still often utilizes `GET`. It just allows to translate a friendlier URL into a URL with a `GET` query string.

Share Improve this answer

Follow

edited Jan 17, 2015 at 21:03

atiquratik

1,300 ● 3 ● 28 ● 35

answered Sep 5, 2008 at 19:48

Brian Warshaw

23k ● 9 ● 54 ● 72

Please add some explanation to your answer such that others can learn from it. How does this help to decide whether to use GET or POST? – Nico Haase Jan 21, 2022 at 12:58

-1

Simple version of POST GET PUT DELETE

- use GET - when you want to get any resource like List of data based on any Id or Name
- use POST - when you want to send any data to server. keep in mind POST is heavy weight operation because for updation we should use PUT instead of POST internally POST will create new resource
- use PUT - when you

Share Improve this answer

Follow

edited Nov 13, 2018 at 7:26

Pang

10.1k ● 146 ● 85 ● 124

answered Nov 27, 2015 at 17:28

Rajesh

662 ● 2 ● 7 ● 14

8 *"use PUT - when you"* Where's the rest of the sentence? – Pang Nov 13, 2018 at 7:27

1 It's great that someone liked the first two bullets of this answer so much apparently that they upvoted it sans the final bullet haha :'-) – pooley1994 Jun 6, 2020 at 15:16

"POST is heavy weight operation" - what does that mean? By which terms is a POST request more "heavy weight" than a

GET request that uses the same set of parameters?
– Nico Haase Jan 21, 2022 at 12:57

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.