**Differences between HTTP version 1.0 and 1.1**

**HTTP 1.0:**

HTTP 1.0 could only define up to 16 status codes which was a reserved number. The main limitation of using the 16 status codes was that there was poor resolution reporting that was noticed and thus there was the need to come up with the HTTP 1.1.

HTTP 1.0 use comes with only allowance for the basic authentication with this facing a challenge of usernames and passwords that are used being unencrypted. This as you would rightly suppose brings forth the factor of risk of being snooped upon.

HTTP 1.0 does not have dependencies and thus the information collected by the activity of snooping can be used later in the future.

HTTP 1.0 design needed a new TCP connection for every request that was made through it. This caused a challenge as there was the cost and time of setting up a new TCP connection with every request, making the connection very slow.

HTTP 1.0 use the Expire header field to determine the resources of fresh or stale, and use condition request (conditional request) to determine the resource is still valid.

HTTP 1.0 also defines a Pragma:no-cache header, the client is using the head domain that requests resources can not be obtained from the cache, and must return to the source to obtain.

HTTP 1.0 client does not support the 100 response code. But let the client to join the Expect header field in the request message, and its value is set to 100-continue.

**HTTP 1.1:**

HTTP 1.1 came with 24 status codes that were able to solve the previous limitations that HTTP 1.1 faced. Error reporting was done faster and there was easy detection of errors when they occurred.

HTTP 1.1 was the warning header that had the capability of carrying out multiple number of secondary status alerts.

HTTP 1.1 did correct the issue, offering the use of Digest Access Authentication. This mirrors basic authentication and allows for servers top make use of a one time value which in effect make snooping quite difficult to achieve. A checksum of the password, username and one time value is made and these are all encrypted. You can thus rest assured that no snooping is possible when using HTTP 1.1.

HTTP 1.1 came up with the use of persistent connections and also the use of pipeline requests to work on the persistent connections.HTTP/1.1 on the basis of 1 joined the new features of cache, when the cache object Age over Expire to the stale object, cache does not need to abandon the stale object directly, but re activation and the source server(revalidation).

HTTP 1.1 on the basis of 1 joined the new features of cache, when the cache object Age over Expire to the stale object, cache does not need to abandon the stale object directly, but re activation and the source server(revalidation).

HTTP 1.1 adds a new status code 100(Continue). The client sent in advance a lead domain request, if the server because the permission refused the request, it sends back the response code 401 (Unauthorized); if the server receives the request echo response code 100, the client can continue to send the request entity integrity.

**How to pass user-data via GET? via POST? what is the diff?**

**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.

**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.

##### **Key difference:** Both get and post methods are used to send and receive information between web clients and servers. However, the difference between them lies in the visibility of their URL parameters, which appear clearly in the get method, but not in the post method.Key difference: Both get and post methods are used to send and receive information between web clients and servers. However, the difference between them lies in the visibility of their URL parameters, which appear clearly in the get method, but not in the post method.

* **Websites where POST method is used:**

<https://www.instagram.com/> - login&registation form;

<https://telegram.org/> - login&registation form.