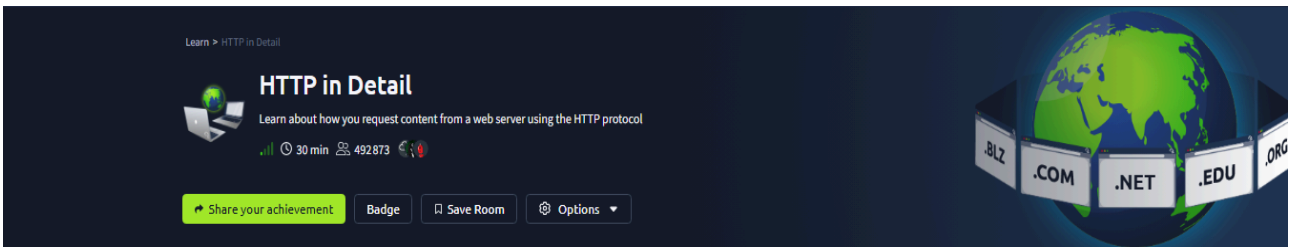


HTTP in Detail | 29 Septembre, 2025

Auteur : **GhostHunt3rx**
Email : **anonymousinconnu10@gmail.com**

Tâche 1 : Qu'est-ce que le protocole HTTP(S) ?

Qu'est-ce que le protocole HTTP ? (HyperText Transfer Protocol)

Le protocole **HTTP** est utilisé chaque fois que vous consultez un site web. Il a été développé par **Tim Berners-Lee** et son équipe entre **1989** et **1991**. Le protocole HTTP est un ensemble de règles utilisées pour communiquer avec les serveurs web afin de transmettre des données de pages web, qu'il s'agisse de HTML, d'images, de vidéos, etc.

Qu'est-ce que le HTTPS ? (HyperText Transfer Protocol Secure)

Le HTTPS est la version sécurisée du HTTP. Les données HTTPS sont cryptées, ce qui empêche non seulement les autres personnes de voir les données que vous recevez et envoyez, mais vous garantit également que vous communiquez avec le bon serveur web et non avec un imposteur.

Tâche 2 : Demandes et réponses

Lorsque nous accédons à un site web, votre navigateur doit envoyer des requêtes à un serveur web pour obtenir des ressources telles que des fichiers HTML, des images, puis télécharger les réponses. Avant cela, vous devez indiquer précisément au navigateur comment et où accéder à ces ressources. C'est là que les URL entrent en jeu.

Qu'est-ce qu'une URL ? (Uniform Resource Locator)

Si vous avez déjà utilisé Internet, vous avez déjà utilisé une URL. Une URL est principalement une instruction indiquant comment accéder à une ressource sur Internet. L'image ci-dessous montre à quoi ressemble une URL avec toutes ses fonctionnalités (elle n'utilise pas toutes les fonctionnalités dans chaque requête).

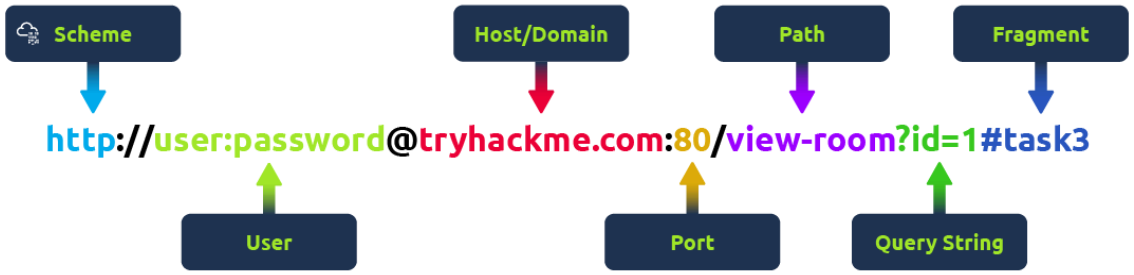


Schéma : indique le protocole à utiliser pour accéder à la ressource, tel que HTTP, HTTPS, FTP (File Transfer Protocol).

Utilisateur : certains services nécessitent une authentification pour se connecter. Vous pouvez saisir un nom d'utilisateur et un mot de passe dans l'URL pour vous connecter.

Hôte : nom de domaine ou adresse IP du serveur auquel vous souhaitez accéder.

Port : le port auquel vous allez vous connecter, généralement 80 pour HTTP et 443 pour HTTPS, mais il peut s'agir de n'importe quel port compris entre 1 et 65535.

Chemin d'accès : nom du fichier ou emplacement de la ressource à laquelle vous essayez d'accéder.

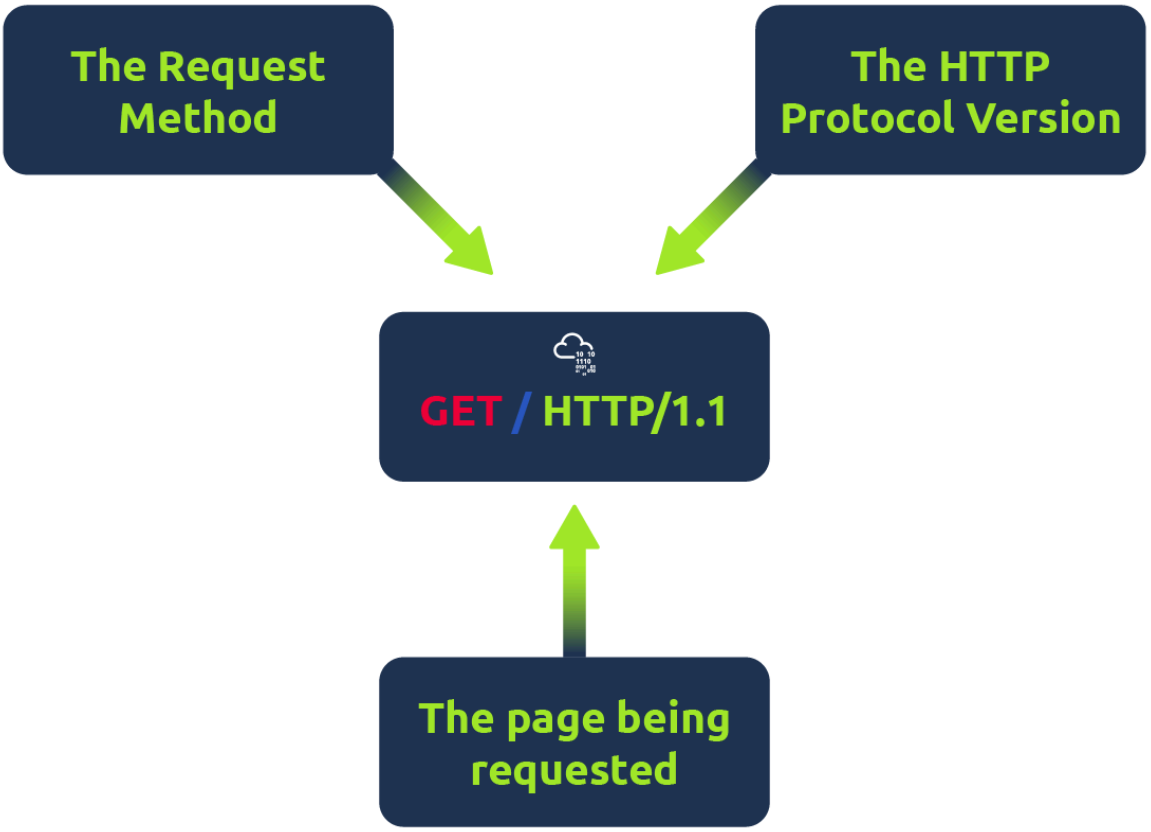
Chaîne de requête : informations supplémentaires pouvant être envoyées au chemin d'accès demandé. Par exemple, /blog?**id=1** indiquerait au chemin d'accès du blog que vous souhaitez recevoir l'article du blog dont l'identifiant est 1.

Fragment : il s'agit d'une référence à un emplacement sur la page réelle demandée. Il est couramment utilisé pour les pages dont le contenu est long

et peut être directement lié à une certaine partie de la page, de sorte qu'il est visible par l'utilisateur dès qu'il accède à la page.

Effectuer une requête

Il est possible d'effectuer une requête à un serveur web avec une seule ligne **GET / HTTP/1.1**



Mais pour une expérience Web beaucoup plus riche, vous devrez également envoyer d'autres données. Ces autres données sont envoyées dans ce qu'on appelle des en-têtes, qui contiennent des informations supplémentaires à transmettre au serveur Web avec lequel vous communiquez, mais nous y reviendrons plus en détail dans la tâche En-tête.

Exemple de requête :

```
GET / HTTP/1.1
```

```
Host: tryhackme.com
User-Agent: Mozilla/5.0 Firefox/87.0
Referer: https://tryhackme.com/
```

Pour décomposer chaque ligne de cette requête :

Ligne 1 : cette requête envoie la méthode GET (plus d'informations à ce sujet dans la tâche Méthodes HTTP), demande la page d'accueil avec / et indique au serveur web que nous utilisons le protocole HTTP version 1.1.

Ligne 2 : nous indiquons au serveur web que nous voulons le site web tryhackme.com

Ligne 3 : nous indiquons au serveur web que nous utilisons le navigateur Firefox version 87.

Ligne 4 : nous indiquons au serveur web que la page web qui nous a redirigés vers celle-ci est https://tryhackme.com.

Ligne 5 **:** les requêtes HTTP se terminent toujours par une ligne vide afin d'informer le serveur web que la requête est terminée.

Exemple de réponse :

```
HTTP/1.1 200 OK

Server: nginx/1.15.8
Date: Fri, 09 Apr 2021 13:34:03 GMT
Content-Type: text/html
Content-Length: 98


<html>
<head>
    <title>TryHackMe</title>
</head>
<body>
```

```
    Welcome To TryHackMe.com
</body>
</html>
```

Pour décomposer chaque ligne de la réponse :

Ligne 1 : HTTP 1.1 est la version du protocole HTTP utilisée par le serveur, suivie du code d'état HTTP, dans ce cas « 200 OK », qui nous indique que la requête s'est déroulée avec succès.

Ligne 2 : cette ligne nous indique le logiciel du serveur web et son numéro de version.

Ligne 3 : la date, l'heure et le fuseau horaire actuels du serveur web.

Ligne 4 : l'en-tête Content-Type indique au client le type d'informations qui va être envoyé, par exemple HTML, images, vidéos, pdf, XML.

Ligne 5 : Content-Length indique au client la longueur de la réponse, ce qui nous permet de vérifier qu'aucune donnée ne manque.

Ligne 6 : la réponse HTTP contient une ligne vide pour confirmer la fin de la réponse HTTP.

Lignes 7 à 14 : les informations qui ont été demandées, dans ce cas la page d'accueil.

Tâche 3 : Méthodes HTTP

Les méthodes HTTP permettent au client d'indiquer l'action qu'il souhaite effectuer lorsqu'il envoie une requête HTTP. Il existe de nombreuses méthodes HTTP, mais nous aborderons ici les plus courantes, même si vous utiliserez principalement les méthodes GET et POST.

Demande GET

Elle est utilisée pour obtenir des informations à partir d'un serveur web.

Demande POST

Elle est utilisée pour envoyer des données au serveur web et éventuellement créer de nouveaux enregistrements.

Demande PUT

Elle est utilisée pour envoyer des données à un serveur web afin de mettre à jour des informations.

Demande DELETE

Elle est utilisée pour supprimer des informations/enregistrements d'un serveur web.

Tâche 4 : Codes d'état HTTP

Codes d'état HTTP :

Dans la tâche précédente, vous avez appris que lorsqu'un serveur HTTP répond, la première ligne contient toujours un code d'état informant le client du résultat de sa requête et éventuellement de la manière de la traiter. Ces codes d'état peuvent être classés en 5 catégories différentes :

100-199 - Information Response	These are sent to tell the client the first part of their request has been accepted and they should continue sending the rest of their request. These codes are no longer very common.
200-299 - Success	This range of status codes is used to tell the client their request was successful.

300-399 - Redirection	These are used to redirect the client's request to another resource. This can be either to a different webpage or a different website altogether.
400-499 - Client Errors	Used to inform the client that there was an error with their request.
500-599 - Server Errors	This is reserved for errors happening on the server-side and usually indicate quite a major problem with the server handling the request.

Codes d'état HTTP courants :

Il existe de nombreux codes d'état HTTP différents, sans compter que les applications peuvent même définir les leurs. Nous allons passer en revue les réponses HTTP les plus courantes que vous êtes susceptible de rencontrer :

200 - OK	The request was completed successfully.
201 - Created	A resource has been created (for example a new user or new blog post).

301 - Moved Permanently	This redirects the client's browser to a new webpage or tells search engines that the page has moved somewhere else and to look there instead.
302 - Found	Similar to the above permanent redirect, but as the name suggests, this is only a temporary change and it may change again in the near future.
400 - Bad Request	This tells the browser that something was either wrong or missing in their request. This could sometimes be used if the web server resource that is being requested expected a certain parameter that the client didn't send.
401 - Not Authorised	You are not currently allowed to view this resource until you have authorised with the web application, most commonly with a username and password.
403 - Forbidden	You do not have permission to view this resource whether you are logged in or not.
405 - Method Not Allowed	The resource does not allow this method request, for example, you send a GET request to the resource /create-account when it was expecting a POST request instead.

404 - Page Not Found	The page/resource you requested does not exist.
500 - Internal Service Error	The server has encountered some kind of error with your request that it doesn't know how to handle properly.
503 - Service Unavailable	This server cannot handle your request as it's either overloaded or down for maintenance.

Si vous êtes plutôt visuel, consultez également l'excellente ressource http.cat pour étudier les codes d'état. Cliquez maintenant sur le bouton « Afficher le site » à droite pour voir à quoi ressemblent certains de ces messages d'état HTTP dans un navigateur.

Tâche 5 : En-têtes

Les en-têtes sont des bits de données supplémentaires que vous pouvez envoyer au serveur web lorsque vous effectuez des requêtes.

Bien qu'aucun en-tête ne soit strictement requis lors d'une requête HTTP, vous aurez des difficultés à afficher correctement un site web.

En-têtes de requête courants

Il s'agit d'en-têtes envoyés par le client (généralement votre navigateur) au serveur.

Hôte : certains serveurs Web hébergent plusieurs sites Web. En fournissant les en-têtes d'hôte, vous pouvez indiquer celui dont vous avez besoin, sinon vous recevrez simplement le site Web par défaut du serveur.

Agent utilisateur : il s'agit du nom et du numéro de version de votre navigateur. Il indique au serveur Web que votre navigateur l'aide à formater correctement le site Web pour votre navigateur. De plus, certains éléments HTML, JavaScript et CSS ne sont disponibles que dans certains navigateurs.

Content-Length : lorsque vous envoyez des données à un serveur web, par exemple dans un formulaire, la longueur du contenu indique au serveur web la quantité de données à attendre dans la requête web. De cette façon, le serveur peut s'assurer qu'il ne manque aucune donnée.

Accept-Encoding : indique au serveur web les types de méthodes de compression pris en charge par le navigateur afin que les données puissent être réduites pour être transmises sur Internet.

Cookie : données envoyées au serveur pour l'aider à mémoriser vos informations (voir la tâche cookies pour plus d'informations).

En-têtes de réponse courants

Il s'agit des en-têtes renvoyés au client par le serveur après une requête.

Set-Cookie : informations à stocker qui sont renvoyées au serveur web à chaque requête (voir la tâche cookies pour plus d'informations).

Cache-Control : durée de stockage du contenu de la réponse dans le cache du navigateur avant qu'il ne le demande à nouveau.

Content-Type : indique au client le type de données renvoyées, c'est-à-dire HTML, CSS, JavaScript, images, PDF, vidéo, etc. Grâce à l'en-tête content-type, le navigateur sait alors comment traiter les données.

Content-Encoding : méthode utilisée pour compresser les données afin de réduire leur taille lors de leur envoi sur Internet.

Tâche 6 : Cookies

Vous avez probablement déjà entendu parler des cookies.

Il s'agit simplement de petits fichiers de données stockés sur votre ordinateur.

Les cookies sont enregistrés lorsque vous recevez un en-tête « Set-Cookie » provenant d'un serveur web. Ensuite, à chaque nouvelle requête que vous effectuez, vous renvoyez les données du cookie au serveur web.

Comme le protocole HTTP est sans état (il ne garde pas trace de vos requêtes précédentes), les cookies peuvent être utilisés pour rappeler au serveur web qui vous êtes, certains paramètres personnels pour le site web ou si vous avez déjà visité le site web auparavant.

Prenons l'exemple d'une requête HTTP :

Les **cookies** peuvent être utilisés à de nombreuses fins, mais ils servent le plus souvent à l'authentification sur les sites web. La valeur du cookie n'est généralement pas une chaîne de texte en clair où vous pouvez voir le mot de passe, mais un jeton (code secret unique difficile à deviner).

Afficher vos cookies

Vous pouvez facilement voir quels cookies votre navigateur envoie à un site web en utilisant les outils de développement de votre navigateur. Si vous ne savez pas comment accéder aux outils de développement de votre navigateur, cliquez sur le bouton « Afficher le site » en haut de cette tâche pour obtenir un guide pratique.

Une fois les outils de développement ouverts, cliquez sur l'onglet « Réseau ». Cet onglet vous montrera une liste de toutes les ressources demandées par votre navigateur. Vous pouvez cliquer sur chacune d'elles pour obtenir une analyse détaillée de la demande et de la réponse. Si votre navigateur a envoyé un cookie, vous le verrez dans l'onglet « Cookies » de la demande.

Tâche 7 : Formuler des demandes

Cliquez sur le bouton « *Afficher le site* » à droite.

Il s'agit d'un émulateur permettant d'effectuer des requêtes HTTP de démonstration. À l'aide des connaissances acquises lors des tâches précédentes, vous pouvez l'utiliser pour répondre aux questions ci-dessous.

Effectuez une requête POST vers la page /login avec le nom d'utilisateur thm et le mot de passe letmein.
Remarque : utilisez le bouton en forme d'engrenage à droite pour gérer les paramètres du corps.