|  |  |
| --- | --- |
| **Nom** | HAFSAOUI |
| **Prénom** | NADIA |
| **Groupe** | G 04 |

**Travaux Pratiques N° 1 : Protocole HTTP**

|  |  |
| --- | --- |
| d. Répondre aux questions suivantes : | |
| **Questions** | **Réponses** |
| - Quelle méthode http faisant l’objet de cette requête ? | GET |
| - Quelle est la requête URI de l’objet demandé ? | Cs.univ\_batna2.dz |
| - Prévoir le nom du fichier demandé | La page d’accueil de l’université de batne 2 |
| - Quelle est la version du protocole http ? | http 1.1 |
| - Quelle est la signification de « keep-alive » ? | Keep-Alive permet l’optimisation de la page web |
| - Que signifie « q=0.9 » ? | Signifie le langage utilisé. |
| - Quel type de donnée le client préfère recevoir le plus dans la réponse http ? | Document ; image ;video |
| f. Répondre aux questions suivantes : | |
| **Questions** | **Réponses** |
| - Que signifie la valeur 200 ? | indique la réussite d'une requête. |
| - Quel serveur est responsable de fournir la réponse http ? | http chunked response |
| - Comment le corps de la réponse http est-il séparé de son en-tête ? | Ligne blanche |
| - Quel est le type des données renvoyées par le serveur dans le corps de la réponse http ? | html |
| - Combien de lignes de code comporte le document retourné ? | 274 lignes |

Connection-specific DNS Suffix . :

Link-local IPv6 Address . . . . . : fe80::bcdd:ee1e:ab57:36e0%13

IPv4 Address. . . . . . . . . . . : 192.168.1.4

Subnet Mask . . . . . . . . . . . : 255.255.255.0

Default Gateway . . . . . . . . . : 192.168.1.1

Les requêtes HTTP expliquées simplement

Sur le Web, les clients, comme votre navigateur, communiquent avec les serveurs Web en utilisant le [protocole HTTP](https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/definition-protocole-http/). Ce protocole contrôle la façon dont le client formule ses demandes et la façon dont le serveur y répond. Le protocole HTTP connaît différentes méthodes de requête. Il est utile de connaître au moins les plus courantes d’entre elles. Dans cet article, vous apprendrez les détails des méthodes de requête HTTP.

Sommaire

1. [GET](https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/requete-http/#c242927)
2. [POST](https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/requete-http/#c242937)
3. [HEAD](https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/requete-http/#c242939)
4. [OPTIONS](https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/requete-http/#c242942)
5. [TRACE](https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/requete-http/#c242946)
6. [Méthodes spécifiques](https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/requete-http/#c242948)

GET

GET est l’ « ancêtre » des requêtes HTTP. Cette méthode de requête existe depuis le début du Web. Elle est utilisée pour **demander une ressource**, par exemple un fichier HTML, au **serveur Web.**

Si vous entrez l’URL *www.exemple.com* dans votre navigateur, celui-ci se connecte au serveur Web et lui envoie la requête GET :

GET /index.php

Le fichier *index.php* dans cet exemple est simplement la page d’accueil ; il est envoyé du serveur au navigateur.

De même, la demande de l’URL *www.exemple.com/test.html* serait formulée ainsi :

GET /test.html

Le serveur renvoie le fichier *test.html.*

Paramètres URL

Le serveur envoie le fichier test. La requête GET peut recevoir des **informations supplémentaires** que le serveur Web doit traiter. Ces paramètres d’URL sont simplement ajoutés à l’URL. La syntaxe est très simple :

* La chaîne de requête est introduite par un « ? » (point d’interrogation).
* Chaque paramètre est nommé, il se compose donc d’un nom et d’une valeur : « Nom=Valeur ».
* Si plusieurs paramètres doivent être inclus, ils sont reliés par un « & ».

Voici un exemple : sur le site Web d’une entreprise de logiciels, « Windows » est saisi comme plateforme et « Office » comme catégorie pour afficher les offres correspondantes du serveur :

GET /search?platform=Windows&category=office

Codage URL de la chaîne Query

Les chaînes Query nécessitent un encodage de caractères spéciaux car de **nombreux caractères spéciaux** ont une signification particulière. Par exemple, le texte « HTTP Overview » doit être codé comme suit pour être accepté comme chaîne d’interrogation

GET /search?thema=HTTP-%C3%9Cberblick%0A

 Conseil

Vous pouvez facilement créer des codages d’URL avec un outil en ligne par exemple avec un [encodeur URL](https://www.urlencoder.org/), et hors ligne avec la fonction Excel *URLENCODE()*.

POST

Si vous souhaitez envoyer de grandes quantités de données, par exemple des images, ou des données confidentielles de formulaires au serveur, la méthode GET n’est pas idéale car toutes les données que vous envoyez sont écrites ouvertement dans la barre d’adresse du navigateur.

Dans ces cas, la méthode POST est la plus adaptée. Cette méthode n’écrit pas les paramètres de l’URL, mais les ajoute à l’[en-tête HTTP](https://www.ionos.fr/digitalguide/hebergement/aspects-techniques/http-header/).

Les requêtes POST sont principalement utilisées pour les **formulaires en ligne**. Voici un exemple de formulaire qui prend le nom et l’adresse électronique et les envoie au serveur en utilisant POST :

<html>

<body>

<form action="newsletter.php" method="post">

Name: <input type="text" name="name"><br>

E-mail: <input type="text" name="email"><br>

<input type="submit">

</form>

</body>

</html>

 Conseil

Vous trouverez aussi dans notre guide un article sur les différences entre GET et POST.

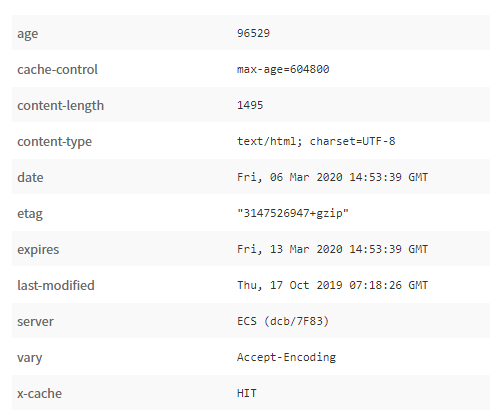
HEAD

La méthode HEAD est utilisée pour interroger l’en-tête de la réponse, sans que le fichier ne vous soit envoyé immédiatement. C’est utile, par exemple, si des fichiers volumineux doivent être transférés : grâce à la requête HEAD, le client peut d’abord être informé de la **taille du fichier** et seulement ensuite décider s’il veut recevoir le fichier.

Exemple :

HEAD /downloads/video1.mpeg HTTP/1.0

Dans l’en-tête de la réponse du serveur, le client trouvera l’information sur la taille du fichier dans le champ d’en-tête « content-length » :

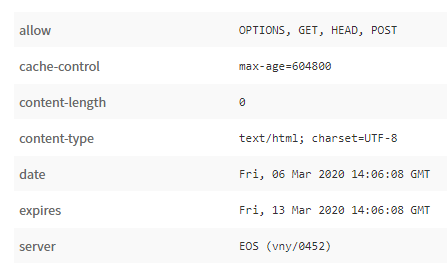
[](https://www.ionos.fr/digitalguide/fileadmin/DigitalGuide/Screenshots_2020/screenshot-of-a-http-response-to-the-request-head.png)Le serveur répond à la requête HEAD avec les données de base du fichier demandé.

OPTIONS

Avec la méthode OPTIONS, le client peut demander **quelles méthodes** le serveur supporte pour le fichier en question.

OPTIONS /download.php

La réponse peut aussi ressembler à ceci :

[](https://www.ionos.fr/digitalguide/fileadmin/DigitalGuide/Screenshots_2020/screenshot-of-http-response-to-options.png)Réponse du serveur à une requête OPTIONS

Dans le champ « allow », nous apprenons que le serveur prend en charge les méthodes OPTIONS, GET, HEAD et POST. Dans le champ « content-length », le chiffre 0 nous indique qu’aucun fichier n’a été transféré, seulement l’en-tête.

TRACE

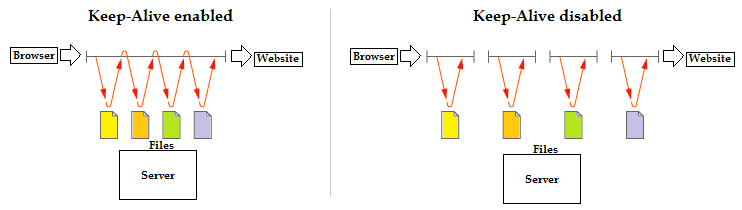
La méthode TRACE peut être utilisée pour tracer le **chemin** qu’une requête HTTP emprunte jusqu’au serveur puis jusqu’au client. Vous pouvez utiliser la commande [*tracert*](https://www.ionos.fr/digitalguide/serveur/outils/traceroute-et-les-chemins-de-paquets-de-donnees/) de Windows pour tracer vous-même cet itinéraire. Pour ce faire, entrez la commande suivante dans la ligne de commande (cmd.exe) :

# Optimiser son site Web rapidement grâce à KeepAlive

## ****Introduction****

### ****Comment fonctionne Keep-Alive?****

Dans ce didacticiel, vous apprendrez 4 méthodes différentes pour activer keep-alive. Keep-Alive permet au navigateur du visiteur de télécharger tout le contenu (comme JavaScript, CSS, images, vidéos, etc.) via une connexion TCP persistante au lieu de faire des demandes différentes pour chaque fichier. Cela fournira un gain de vitesse et de performance car le navigateur de votre visiteur sera en mesure d’obtenir tout via une seule connexion HTTP persistante. En bref, Keep-Alive est un modèle de communication entre un serveur Web et un navigateur qui améliorera vos performances. Voici une image qui vous aidera à comprendre les avantages de Keep-Alive:



### ****Avantages de l’activation de Keep-Alive:****

* Keep-Alive réduit l’utilisation du processeur et de la mémoire en raison de la moindre quantité de requêtes HTTP générées. Cela bénéficiera à tous les utilisateurs de la plateforme d’hébergement (hébergement gratuit, hébergement mutualisé, VPS)
* L’activation de Keep-Alive fournit le pipelining HTTP (livraison de requêtes via la même connexion TCP)
* Les demandes HTTPS nécessitent davantage de temps et de ressources. Keep-Alive sera grandement utile à votre site Web si vous utilisez HTTPS et SSL.
* Une latence réduite et une augmentation globale de la vitesse et des performances.
* Keep-Alive est pris en charge par tous les navigateurs modernes
* L’activation de Keep-Alive profitera également à votre site Web en termes de référencement  en raison de la meilleure performance du site.

En bref, Keep-Alive est un excellent moyen de réduire l’utilisation de vos ressources tout en augmentant la vitesse de votre site Web en même temps.

Sommaire

* [**Ce dont vous aurez besoin**](https://www.hostinger.fr/tutoriels/keep-alive/#Ce-dont-vous-aurez-besoin)
* [**Étape 1 – Analyse de votre site**](https://www.hostinger.fr/tutoriels/keep-alive/#Etape-1-8211-Analyse-de-votre-site)
* [**Étape 2 – Activation de Keep-Alive**](https://www.hostinger.fr/tutoriels/keep-alive/#Etape-2-8211-Activation-de-Keep-Alive)
* [**Étape 3 – Test des changements**](https://www.hostinger.fr/tutoriels/keep-alive/#Etape-3-8211-Test-des-changements)
* [**Conclusion**](https://www.hostinger.fr/tutoriels/keep-alive/#Conclusion)

## ****Ce dont vous aurez besoin****

Avant d’activer Keep-Alive, assurez vous d’avoir les éléments suivants:

* Accès au fichier **.htaccess**
* Accès à **httpd.conf** ( **facultatif** )
* Accès à **httpCoreModule** ( **optionnel** )

## ****Étape 1 – Analyse de votre site****

Tout d’abord, vous devez analyser vore site Web avec [**GTMetrix**](https://gtmetrix.com/) pour déterminer si Keep-Alive est activé ou désactivé. Voici les résultats après l’analyse d’une page de test:

gtmetrix scan keep alive 0

Sur certains serveurs ou fournisseurs d’hébergement, Keep-Alive est activé par défaut. Si votre analyse donne un score de 100%, il n’y a rien à faire.

## ****Étape 2 – Activation de Keep-Alive****

Il y a plusieurs façons d’activer Keep-Alive. Tout dépend de votre serveur ou fournisseur d’hébergement. Les voici:

### ****Option 1 – Modification du fichier .htaccess****

Pour activer Keep-Alive, le code suivant devrait faire l’affaire. Vous devez l’ajouter au fichier**.htaccess**. L’activation de Keep-Alive à l’aide de **.htaccess**annulera les paramètres du serveur, ce qui est une bonne chose.

<ifModule mod\_headers.c>

Header set Connection keep-alive

</ifModule>

Cette méthode devrait fonctionner sur la plupart des fournisseurs d’hébergement sous Linux. Si vous ne savez pas où trouver **.htaccess** , lisez [**ce tutoriel**](https://www.hostinger.fr/tutoriels/trouver-creer-htaccess/)

**Option 2 – Activation de Keep-Alive dans Apache via le fichier httpd.conf**

Si vous avez accès au fichier de configuration Apache, vous pouvez activer l’extension à partir de là. Voici ce que vous devriez modifier:

#

# KeepAlive: Whether or not to allow persistent connections (more than

# one request per connection). Set to "Off" to deactivate.

#

KeepAlive On

#

# MaxKeepAliveRequests: The maximum number of requests to allow

# during a persistent connection. Set to 0 to allow an unlimited amount.

# We recommend you leave this number high, for maximum performance.

#

MaxKeepAliveRequests 50

#

# KeepAliveTimeout: Number of seconds to wait for the next request from the

# same client on the same connection.

#

KeepAliveTimeout 10

* **KeepAlive On** active l’extension.
* **MaxKeepAliveRequests** définit le nombre maximal de requêtes autorisées pour une connexion unique. 50 est un bon nombre pour commencer
* **KeepAliveTimeout**spécifie la durée pendant laquelle le serveur attend de nouvelles demandes d’un client. Il est recommandé de commencer par une valeur petite. Par exemple 5 ou 10. Et de l’augmenter ultérieurement si nécessaire. Une valeur trop élevée peut entraîner une surcharge du serveur.

Si vous ne trouvez pas le fichier **httpd.conf** , exécutez la commande suivante en ligne de commande:

find / -name httpd.conf

### ****Option 3 – Activation de Keep-Alive dans NGINX****

Keep-Alive est activé par défaut sur NGINX, mais dans certains cas, il peut être désactivé. Vous pouvez l’activer en [**utilisant HttpCoreModule**](http://nginx.org/en/docs/http/ngx_http_core_module.html) . Cherchez la valeur keepalive\_disable. C’est souvent à cause d’elle que KeepAlive est désactivé. Avant de l’activer, assurez-vous de connaître la raison pour laquelle il est désactivé.

### ****Option 4 – Serveur Windows (IIS)****

Si vous utilisez un serveur sous Windows, vous pouvez facilement activer l’extension Keep-Alive à l’aide d’une commande.

La commande suivante le permet:

appcmd set config /section:httpProtocol /allowKeepAlive:true

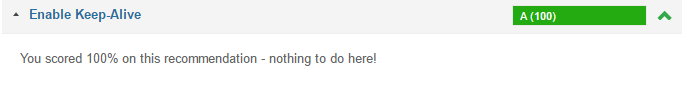
Et si vous souhaitez le désactiver, utilisez:

appcmd set config /section:httpProtocol /allowKeepAlive:false

Vous pouvez également consulter le [**tutoriel officiel**](https://technet.microsoft.com/en-us/library/cc772183(v%3Dws.10).aspx) de Microsoft pour obtenir quelques informations supplémentaires.

## ****Étape 3 – Test des changements****

Une fois que Keep-Alive est activé, exécutez une autre analyse avec [**GTMetrix**](https://gtmetrix.com/) pour voir si tout fonctionne. Voici les résultats après l’activation de Keep-Alive:



Il est également possible de vérifier si Keep-Alive fonctionne en vérifiant votre en-tête HTTP. Cela peut se faire via le terminal en utilisant la commande suivante:

curl -I http://votresite.com/exemple.php

Voici un exemple:

curl -i http://hostinger.com/index.php

Qui nous donne quelque chose comme:

HTTP/1.1 301 Moved Permanently

Connection: keep-alive

Server: nginx

Date: Fri, 23 Dec 2016 18:58:14 GMT

Content-Type: text/html

Content-Length: 178

Location: https://www.hostinger.com/index.php

La partie Connection: keep-alive signifie que Keep-Alive est fonctionnel.

## ****Conclusion****

Pour résumer, l’activation de Keep-Alive est un excellent moyen d’améliorer la vitesse et les performances de votre site. La connexion TCP persistante assurera des temps de chargement plus rapides et une efficacité accrue, ce qui vous permettra d’avoir des visiteurs satisfaits.

Si vous souhaitez optimiser encore plus votre site Web, ces tutoriels vous aideront également: