Le fonctionnement du Web

Table des matières

I Contexte

Durée : 2h

Pré-requis : Connaître le fonctionnement d'internet.

L'un des protocoles d'application les plus connus et utilisés sur Internet est HTTP. C'est le protocole qui est à la base du Web, ou *World Wide Web* (le fameux www), qui nous permet de naviguer sur différents sites. Nous allons voir que c'est un protocole au fonctionnement relativement simple, tout en étant très polyvalent.

II Serveurs web

Rappel

HTTP : HyperText Transfert Protocol (cf. p.45)

Serveur web

Définition

Un serveur web est un logiciel qui **traite les requêtes HTTP** de clients.

Il est installé sur une machine hébergeant des documents (HTML, CSS, JavaScript, etc.) et est accessible depuis Internet ou un réseau local (un intranet).

Les ressources servies par le serveur peuvent être **statiques**, c'est-à-dire sans qu'elles existent préalablement à la requête (un fichier HTML, une image PNG, etc.) ou **dynamiques**, c'est-à-dire construites à chaque requête faite au serveur.

Serveur Web et serveur HTTP

Remarque

Les serveurs web utilisent presque exclusivement le protocole HTTP, créé spécifiquement pour le Web. On parle par abus de langage de serveur web, mais le nom **serveur HTTP** est aussi possible.

Serveurs HTTP

Exemple

- **Apache** : le serveur web le plus utilisé avec 44,3% des parts de marché en février 2019.
- **Nginx** : le deuxième serveur web le plus utilisé mais est le premier parmi les 1000 sites les plus actifs.
- **Node.js** : un environnement d'exécution JavaScript en dehors du navigateur qui fait aussi office de serveur HTTP.

Installation d'un serveur Apache sous GNU/Linux

Méthode

Sur les distributions de famille Debian, `apt` est utilisé par défaut pour la gestion des logiciels. Il faut commencer par mettre à jour les dépôts de logiciels sur son serveur.

```
1 apt update
```

Puis installer la dernière version d'Apache ainsi :

```
1 apt install apache2
```

Après l'installation, le serveur est prêt à recevoir des requêtes HTTP sur son port 80.

Fondamental

Par défaut, Apache va placer ses fichiers de configuration dans le dossier `/etc/apache2` et sert les fichiers du répertoire `/var/www/html`.

Installation d'un serveur Nginx sous GNU/Linux

Complément

On pourra se référer aux documentations suivantes pour installer le serveur web Nginx :

- https://doc.ubuntu-fr.org/nginx
- Installer Nginx (nginx.com)[1]

À retenir

- Un serveur hébergeant des fichiers web a besoin d'un serveur web pour les rendre disponibles à travers le protocole HTTP.
- Apache et Nginx sont les serveurs web les plus utilisés.

1. https://docs.nginx.com/nginx/admin-guide/installing-nginx/installing-nginx-open-source/

III Exercice : Appliquer la notion

Question 1

[solution n°1 p. 34]

Installer `apache2` sur un VPS.

- *Louer un VPS avec système d'exploitation Linux* *(cf. p.47)*

Indice :

On utilisera l'utilitaire `apt` sur Debian/Ubuntu depuis un utilisateur possédant des droits administrateur.

Question 2

[solution n°2 p. 34]

Dans la barre d'adresse d'un navigateur web, entre l'adresse IP de votre VPS. Que voyez-vous ?

Question 3

[solution n°3 p. 34]

Quel est le chemin complet de la page par défaut service par Apache sur le serveur ?

Indice :

Par défaut, Apache sert les fichiers du répertoire `/var/www/html`.

Indice :

On peut lister le contenu d'un répertoire avec la commande `ls`.

IV Navigateurs web

Navigateurs web Az Définition

Un navigateur web est client HTTP qui a trois rôles.

1. Effectuer des requêtes HTTP (GET, POST, etc.).
2. Comprendre les réponses à ces requêtes.
3. Interpréter les fichiers web (HTML, CSS, JavaScript) retournés.

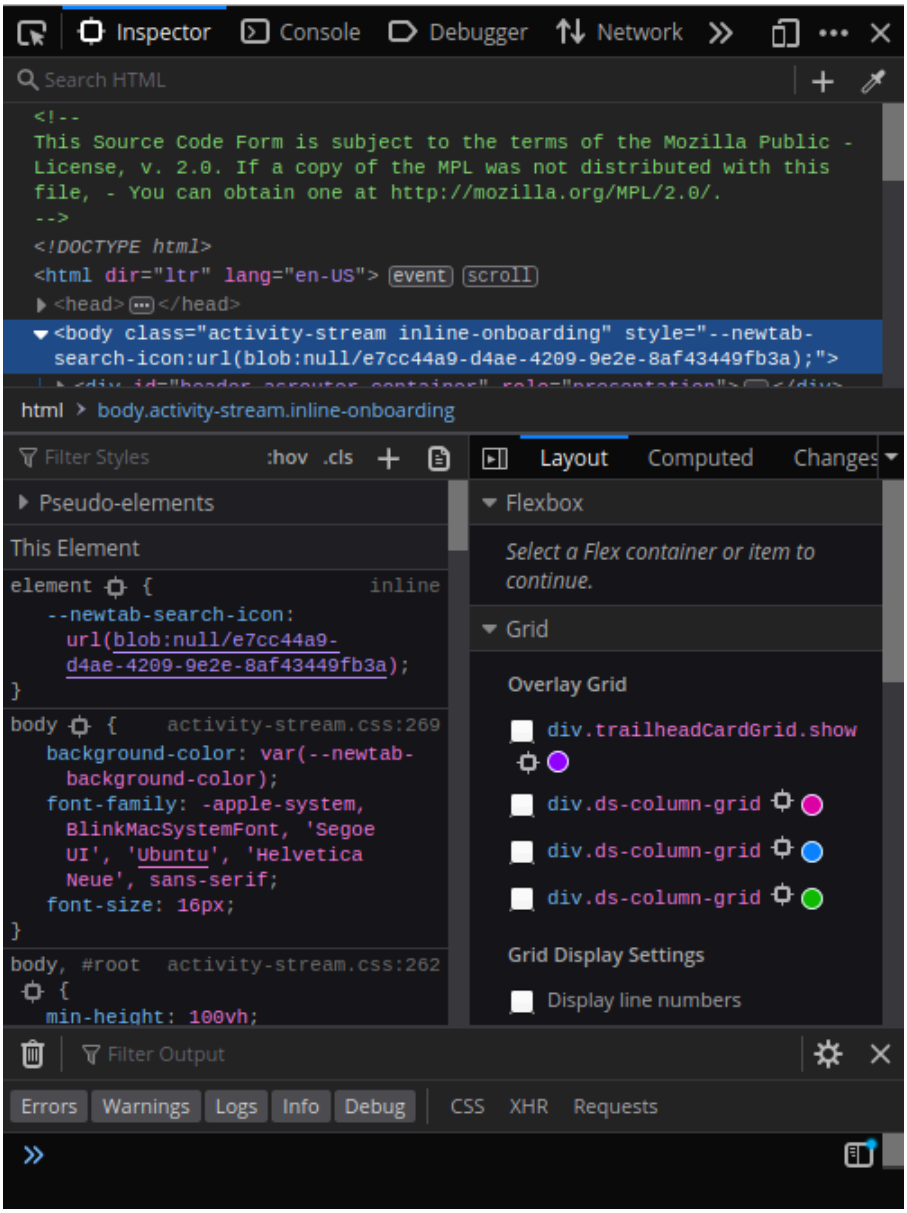
Remarque

Tous les navigateurs web possèdent un interpréteur JavaScript permettant d'exécuter ce langage sur l'ordinateur client.

Les outils de développement sous Firefox Fondamental

Les navigateurs possèdent aussi une partie « Outils de développement » destinée à aider les développeurs web dans leur activité.

Sous Firefox, ces outils sont disponibles en appuyant sur `F12` ou `clic droit > Inspecter l'élément`.



Outils de développement Firefox (F12)

Voilà à quoi ressemble l'inspecteur depuis le site `mozilla.org`.

On retrouve plusieurs onglets en haut de la fenêtre. En voici les plus importants :

- L'inspecteur :

  C'est celui que l'on voit sur l'image ci-dessus, on y retrouve l'HTML de la page et son CSS. Ils sont modifiables depuis cet onglet.

- La console :

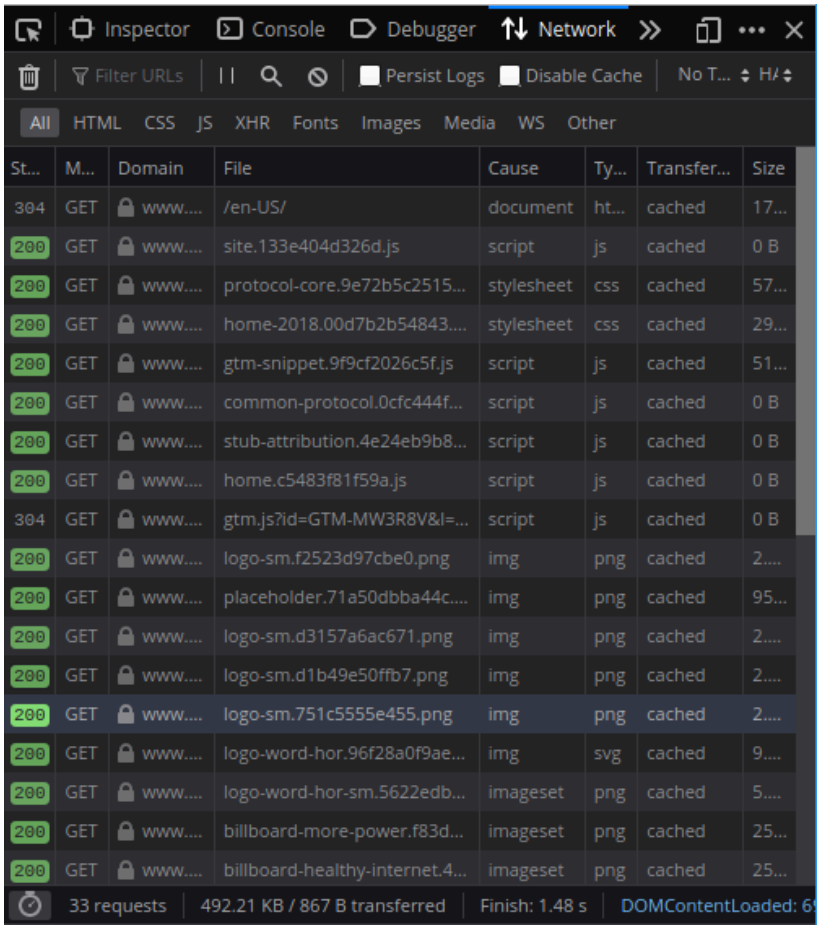
  C'est une console JavaScript. Un utilisateur peut donc y écrire des lignes de JavaScript qui seront interprétées à la volée.

- Le déboguer :

  Cet outil permet un suivi très fin de ce qu'il se passe pendant l'interprétation de la page et l'exécution de JavaScript.

- Le moniteur réseau :

  Cet outil permet d'analyser avec une précision fine toutes les requêtes envoyées et leur réponse associée.

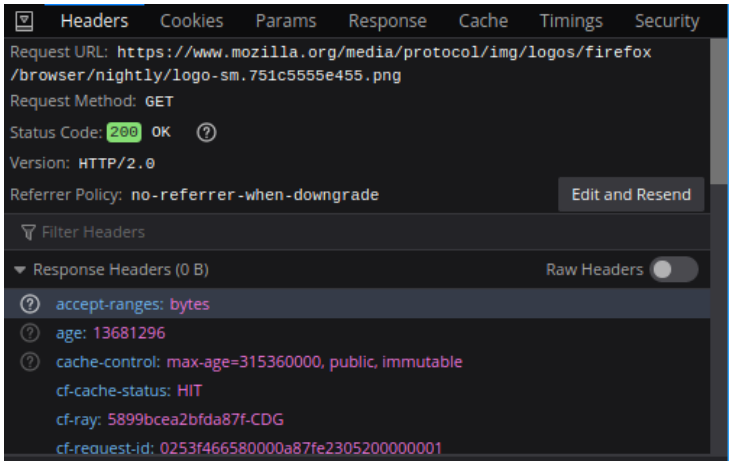


Extrait du moniteur réseau quand on accède à www.mozilla.org

Il y pour chaque requête :

- le code HTTP de la réponse (304 pour signifier que le document demandé n'a pas été modifié depuis la dernière demande et 200 pour dire que tout va bien),
- la requête complète, ici il s'agit d'accéder à la page d'accueil, il n'y a donc que des GET pour récupérer le HTML, le CSS, les script JavaScript et les images,
- l'adresse du serveur web (www.mozilla.org),
- la ressource demandée,
- le type du fichier (HTML, CSS, PNG, etc),
- la quantité transférée, cette valeur est à `cached` lorsque que le navigateur l'avait déjà en mémoire,
- la taille de la ressource.

En cliquant sur une requête, toutes ses informations apparaissent.



Outils de développement Firefox (onglet requête)

À retenir

- Les navigateurs web sont des clients HTTP.
- Ils interprètent les langages du Web (HTML, CSS, JS).
- Ils disposent d'outils de développement permettant d'inspecter les fichiers des pages web et les requêtes et réponses HTTP.

V Exercice : Appliquer la notion

Lancez une fenêtre Firefox, ouvrez la fenêtre des outils de développement (avec la touche F12) et allez sur la page d'URL https://fr.wikipedia.org/.

Question 1

[solution n°4 p. 34]

Quelle est la taille du fichier HTML ?

Indice :

Cette information peut être trouvée dans l'onglet Réseau, puis dans le sous-onglet HTML.

Indice :

Rafraîchissez la page avec F5.

Question 2

[solution n°5 p. 34]

Aller à l'URL : https://fr.wikipedia.org/wiki/Unefausseadresse.

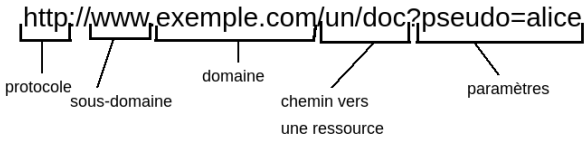
Quel code est renvoyé sur la ligne du fichier HTML ?

VI URL : Uniform Resource Locator

Les URL

Az Définition

Les URL sont nées en même temps que le World Wide Web et permettent d'identifier une ressource sur un serveur web.



Décomposition d'une URL

Elles contiennent :

- Le **protocole** à utiliser pour accéder à la ressource, ici HTTP,
- L'**adresse du serveur** à rejoindre, ici le nom de domaine `www.exemple.com`,
- Le **chemin de la ressource** dans le serveur,
- Éventuellement des paramètres que l'on veut envoyer au serveur (utilisés par les requêtes HTTP GET).

Exemple

Lorsque `https://www.wikipedia.org/wiki/URL` est rentrée dans un navigateur web, ce dernier fait une requête GET sur la ressource « `/wiki/URL` » du serveur pointé par `www.wikipedia.org`, en sécurisant la connexion grâce à HTTPS (que nous verrons par la suite).

Éléments de syntaxe d'une URL

Syntaxe

Certains caractères sont réservés pour un usage spécifique. Par exemple :

- le caractère / est réservé pour indiquer le chemin des fichiers à consulter,
- le caractère ? est réservé pour annoncer les paramètres à transmettre au serveur,
- le caractère = est réservé pour indiquer la valeur des paramètres,
- le caractère # est réservé pour indiquer un **fragment**, souvent une partie spécifique de la page web (par exemple, un paragraphe).

Lorsqu'un caractère réservé doit être utilisé pour autre chose dans l'URL, il est remplacé par un symbole pourcent suivi de son code ASCII au format hexadécimal ('/' devient %2F et '?' devient %3F).

À retenir

- Une URL permet d'identifier une ressource précise sur un serveur web donné.
- Les URL permettent de passer des paramètres aux serveurs web.

VII Exercice : Appliquer la notion

Question 1

[solution n°6 p. 34]

Lorsque l'on se rend sur la page Wikipédia suivante https://fr.wikipedia.org/wiki/Houmous#Origine, quel constat peut-on faire ? Pourquoi ?

Indice :

La navigateur semble nous amener au milieu de la page

Toujours sur Wikipédia, on souhaite visiter la page du double point d'interrogation "??", en se rend donc sur `https://fr.wikipedia.org/wiki/??`

Question 2

[solution n°7 p. 34]

Ça n'a pas l'air de fonctionner, pourquoi ?

Question 3

[solution n°8 p. 35]

Comment faire pour accéder à cette page ? Quelle serait l'URL correcte ?

Indice :

On peut s'aider de la table ASCII suivante

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

VIII Codes de retour HTTP

Mise en situation

Les réponses d'un serveur web à une requêtes HTTP peuvent être variées : renvoyer la ressource demandée, indiquer que la ressource envoyée par la requête a bien été reçue, refuser l'accès à une ressource, etc.

Pour permettre de déterminer le type d'une réponse, HTTP se base sur ce que l'on appelle les codes de retour (ou code d'état).

Code HTTP

Az Définition

Un code HTTP est un code à 3 chiffres qui est présent dans chaque réponse HTTP du serveur. Ce code est destiné au client HTTP pour lui indiquer le statut de sa requête.

Les codes HTTP sont séparés en 5 familles, qui se distinguent par le premier digit du code retour :

- `1xx` : Informations
- `2xx` : Succès
- `3xx` : Redirections
- `4xx` : Erreur du client
- `5xx` : Erreur du serveur

Les codes HTTP courants

Exemple

Il existe plusieurs dizaines de code HTTP, mais certains sont plus utilisés que d'autres, en voici quelques-uns :

- `200 OK` : « Tout est bon »,
- `201 Created`: « Ressource ajoutée avec succès »,
- `401 Unauthorized`: « Vous n'avez pas le droit d'accéder à cette ressource, il faut une authentification »,
- `404 Not Found`: « Ressource non trouvée »,
- `500 Internal Server Error`: « Erreur du serveur ».

Observer les codes HTTP

Exemple

La commande `curl` permet de montrer le code de retour HTTP reçu suite à une requête. Pour cela il suffit de rajouter l'option `-I`.

La première ligne affichée contient le code de retour HTTP.

```
1 $ curl -I https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol
2 HTTP/2 200
```

Si on demande une page qui n'existe pas on obtient le code 404.

```
$ curl -I https://fr.wikipedia.org/fausse_page
HTTP/2 404
```

À retenir

- Chaque réponse HTTP comporte un code qui permet d'indiquer au client le statut de la requête.
- Il existe 5 familles de code différents (informations, succès, redirections, erreurs client, erreurs serveur)

IX Exercice : Appliquer la notion

Question 1

[solution n°9 p. 35]

Quel code est renvoyé lorsque l'on effectue une requête HTTP GET sur la page https://wikipedia.org ? Qu'est-ce que cela signifie ?

Question 2

[solution n°10 p. 35]

Quel est le code HTTP que l'on s'attend à recevoir lorsque notre requête tente de créer une ressource sur le serveur ?

Question 3

[solution n°11 p. 35]

Que signifie le code HTTP 418 ?

Indice :

Ne soyez pas surpris de la réponse, c'est un poisson d'avril

X HTTPS : la version chiffrée du protocole HTTP

Mise en situation

HTTP est un protocole "en clair", c'est à dire que les requêtes et réponses qui sont échangées ne sont pas chiffrées. N'importe qui sur le réseau peut ainsi lire les données échangées. Cela inclut par exemple le mot de passe lorsque l'on se connecte à un compte en ligne, ou par exemple les relevés bancaires que l'on consulte sur le site de sa banque.

HTTPS est la version chiffrée de HTTP qui permet de résoudre ce problème. C'est un protocole qui vient s'ajouter par dessus HTTP pour s'assurer que seul le client et le serveur pourront avoir accès aux données échangées.

HTTPS et TLS Az Définition

HTTPS (pour *HTTP Secure*) est une extension de HTTP qui permet une communication sécurisée.

Pour cela il encapsule les requêtes HTTP dans un protocole appelé TLS (pour `Transport Layer Security`) qui permet de réaliser des échanges chiffrés.

TLS est simplement le protocole de mise en place et d'utilisation du canal de communication chiffré, l'algorithme de chiffrement à utiliser est laissé au choix des clients et serveurs HTTP . On peut parfois trouver le terme SSL, qui est l'ancêtre de TLS (et qu'il ne faut plus utiliser aujourd'hui).

Fonctionnement général de HTTPS

À chaque initialisation de connexion, avant d'envoyer la première requête HTTP, le client va ouvrir une communication TLS (on parle de session TLS) avec le serveur. Dans cette phase préalable, le navigateur et le serveur vont définir les algorithmes et clefs de chiffrement à utiliser et le serveur va envoyer un certificat.

Certificat

Le certificat est à la base du fonctionnement de HTTPS, il permet de certifier au client web que le serveur qui répond à la requête est bien le serveur souhaité. Par exemple si on se rend sur le site de sa banque (disons *www.mabanque.fr*), on veut s'assurer que c'est bien le serveur de *www.mabanque.fr* qui va nous répondre, et pas un serveur qui chercherait à usurper son identité. Pour cela le serveur va devoir présenter un certificat au client, permettant d'attester que l'on communique bien avec le serveur de *www.mabanque.fr*.

Fondamental

Tout site web qui veut rendre possible l'accès en HTTPS doit se procurer un certificat lié à son nom de domaine. Sans cela, HTTPS ne fonctionnera pas, et un message d'avertissement s'affichera dans le navigateur pour indiquer que le site n'est pas digne de confiance pour établir un échangé sécurisé.

Comment se procurer un certificat ? Méthode

Il suffit de le créer, c'est à la portée de n'importe quel administrateur système.

Comment s'assurer que quelqu'un ne crée pas un certificat à la place d'une autre personne ? Fondamental

Grâce à la chaîne de confiance.

Chaîne de confiance Exemple

Imaginons que je sois Ali et que, parce que j'ai un peu d'argent de côté, je suis d'accord pour en prêter à un ami dans le besoin. Il se trouve que je n'ai pas d'ami ayant besoin d'argent, mais un dénommé Baptiste se présente et me demande si je peux lui prêter un peu d'argent pour lancer son entreprise de cordonnerie. Je n'ai rien contre, mais je ne connaît pas vraiment Baptiste, je ne sais pas vraiment si je peux lui faire confiance. Il se trouve cependant que je connais très bien mon amie Charline, en qui j'ai une confiance totale. Charline connaît très bien Baptiste et m'assure qu'elle a complètement confiance en lui.

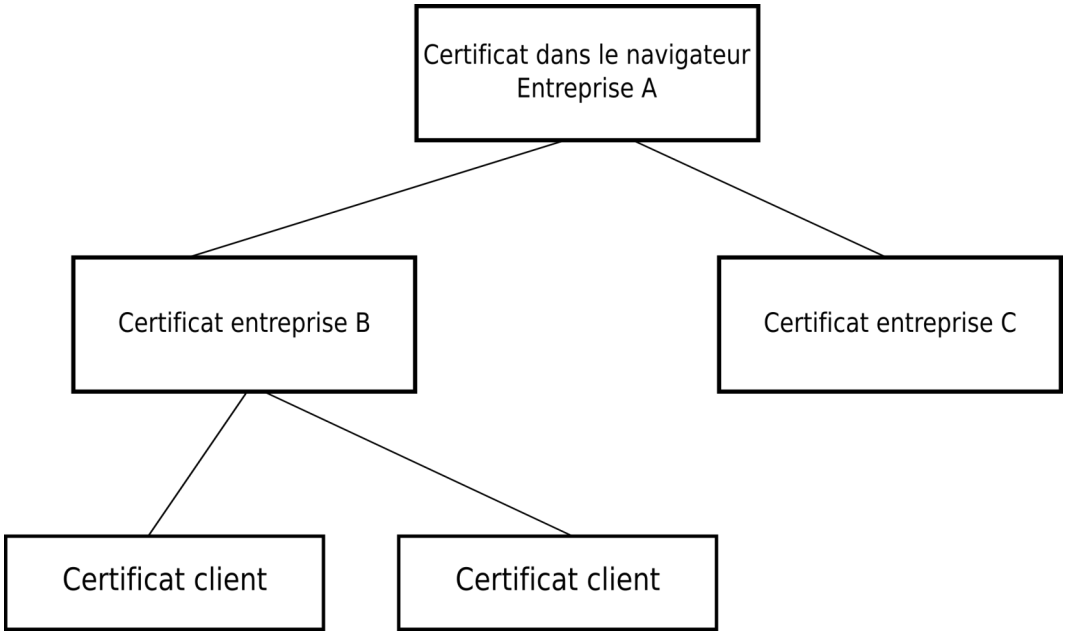
C'est ainsi que se forme une chaîne de confiance : j'ai confiance en Charline qui a confiance en Baptiste, j'ai donc confiance en Baptiste. Il pourrait y avoir d'autres personnes dans la chaîne, cela reviendrait au même. La seule condition **essentielle** est de n'accorder sa confiance qu'à des personnes dont on sait qu'elles ne vont pas accorder elles-mêmes leur confiance à la légère. Si toutes les personnes accordent leur confiance avec la même exigence, alors une chaîne de confiance peut fonctionner.

Autorité de certification et chaîne de confiance

Le fonctionnement est le même pour les navigateurs et les certificats. Chaque navigateur ne fera confiance, de base, qu'à une poignée de certificats (quelques dizaines). Ce sont généralement les certificats de très grandes entreprises, d'états, ou d'entreprises spécialisées dans la création de certificat. Pour intégrer cette liste de certificat "de confiance", les entités ont eu a passer des examens approfondis pour valider que le navigateur peut leur accorder confiance.

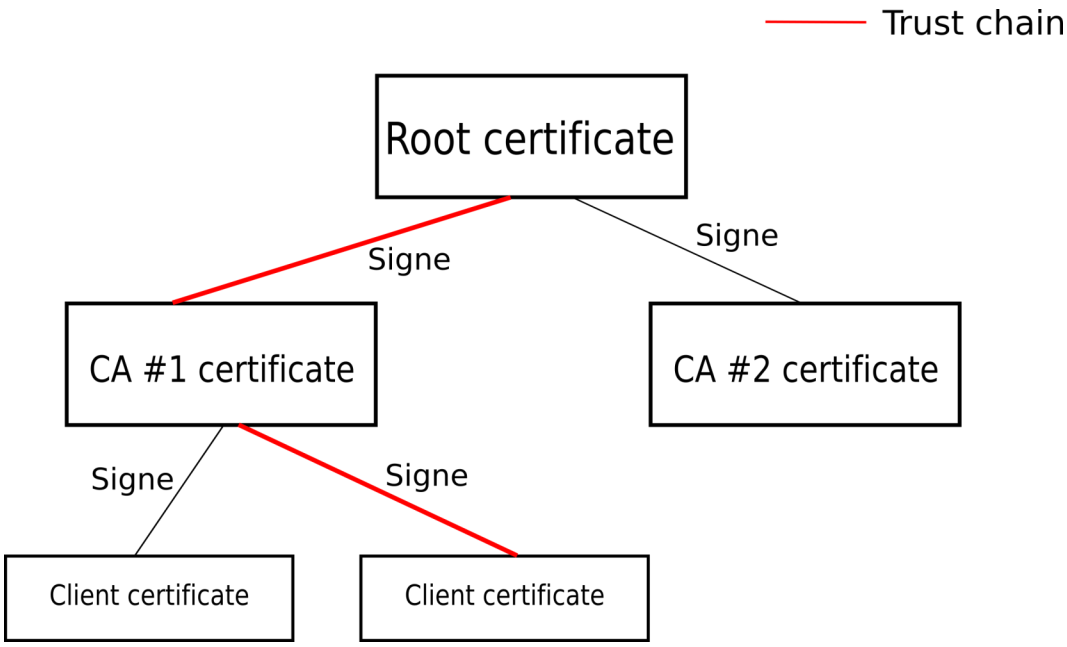
Les certificats fonctionnent ensuite comme une sorte de "preuve de confiance". Une personne qui possède un certificat peut créer un autre certificat pour une autre personne, et "lier" les certificats (on dit que le premier certificat sert à signer le second) entre eux.

Ainsi, lorsqu'un serveur présente un certificat qui ne fait pas partie de la liste connue (ce qui est généralement le cas), le navigateur va vérifier si il peut établir une chaîne de confiance entre le certificat reçu et un des certificats de confiance. Imaginons que le navigateur fasse confiance en un certificat de l'entreprise A. Ensuite l'entreprise A a créé un certificat pour l'entreprise B, qui est un hébergeur de site Web, en laquelle elle a confiance, et elle a signé le certificat de B avec son certificat. Enfin, l'hébergeur B va créer des certificats pour tout ses clients qui ont un site web, et signer les nouveaux certificats. Le navigateur aura confiance dans ces certificats grâce à la chaîne de confiance établi entre le certificat reçu et le certificat de l'entreprise A, en passant par celui de l'entreprise B.



Introduisons maintenant rapidement les termes techniques utilisés dans ce contexte.

- Une entreprise (ou toute autre entité en fait) qui produit des certificats pour d'autres entités s'appelle une **Autorité de Certification** (que l'on abrège AC, ou plus couramment CA en anglais).
- Lorsqu'une autorité de certification émet un certificat pour un tiers, on dit qu'elle **signe le certificat**, ce qui créé un lien dans la chaîne de confiance.
- La chaîne de liens entre des certificats s'appelle l**a chaîne de confiance**, le terme anglais *trust chain* étant le plus souvent utilisé.
- Le certificat étant "en haut" de la hiérarchie dans une chaine de confiance se nomme le **certificat racine** (ou *root certificate*). On parle aussi d'autorité de certification racine (*root CA*).



Exemple

Dans le cas du site `www.picasoft.net`, on peut constater que la chaîne de confiance est la suivante.

Picasoft a un certificat fourni par l'autorité de certification Let's Encrypt (qui a un certificat nommé `Let's Encrypt Authority X3`). Let's Encrypt est à son tour certifiée par Digital Signature Trust, qui a un certificat `DST Root CA X3`. Ce dernier certificat se trouve, lui, par défaut dans les navigateurs les plus courants (Firefox, Chrome, etc.).

À retenir

- HTTPS est une extension de HTTP qui permet de garantir la confidentialité des échanges entre le client et le serveur.
- Les certificats sont à la base du fonctionnement de HTTPS, et leur utilisation se base sur une chaîne de confiance.

XI Exercice : Appliquer la notion

L'objectif de l'exercice va être d'identifier la chaîne de confiance du certificat d'un site web. Pour cela nous utiliserons Firefox.

Question 1

[solution n°12 p. 36]

On cherche à afficher, dans Firefox, le certificat du site https://www.gandi.net

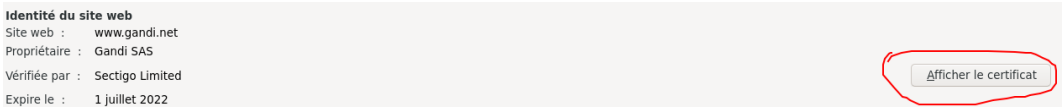
Vous pouvez chercher sur Internet comment afficher le certificat dans votre navigateur.

Indice :

Pour accéder au certificat d'un site web, cliquer sur le cadenas dans la barre d'adresse.

Ensuite en cliquant sur la flèche de droite.

Enfin en cliquant sur "Plus d'informations", une nouvelle fenêtre s'ouvre et permet de visualiser le certificat.



Question 2

[solution n°13 p. 36]

À l'aide du certificat, déterminez la chaîne de confiance.

Indice :

On peut voir que l'émetteur du certificat de Gandi est `Sectigo Limited`

XII Limites de HTTPS

Mise en situation

Bien qu'introduisant des mécanismes de chiffrement dans HTTP, HTTPS ne doit pas être vu comme une sécurité absolue sur le web. Il est important de bien comprendre que HTTPS permet de protéger les échanges, mais pas de protéger contre un site malveillant par exemple.

Nous nous attarderons aussi sur la manière dont les certificats sont obtenus, et les changements de politique à ce sujet ces dernières années.

Rappel

Le champ d'action de HTTPS se limite au chiffrement des communications.

Périmètre de protection de HTTPS

HTTPS garantit :

- que le site auquel on accède est bien celui que l'on a demandé (une personne malveillante ne peux pas usurper l'identité du serveur), grâce au certificat
- que les données échangées seront chiffrées entre le client et le serveur

À l'inverse **HTTPS ne garantit pas** :

- que le site consulté n'est pas un site malveillant ou de phishing
- que le site stocke les données de manières chiffrée/sécurisée

Attention

Il est souvent mis en avant que "un site avec le cadenas vert (donc un site utilisant HTTPS) est un site sécurisé". Comme vous pouvez le comprendre maintenant, cette formulation est un dangereux raccourci. Si tant est que l'on puisse définir ce qu'est un "site sécurisé", il est certain que **HTTPS serait une condition nécessaire mais non suffisante**.

Phishing : mabanque.fr et mabnaque.fr

Exemple

Un premier exemple serait un site malveillant, de phishing, qui tente de voler vos identifiants bancaires. Imaginons que votre banque ai pour adresse `www.mabanque.fr`.

Une personne malveillante pourrait créer une copie conforme (en apparence) de ce site, et l'héberger sur le site `www.mabnaque.fr`. L'objectif serait de jouer sur une faute de frappe ou d'inattention des personnes souhaitant se rendre sur le site de la banque. Cette personne malveillante n'aurait aucun mal à obtenir (puis à présenter) un certificat HTTPS, en effet il est bel et bien propriétaire de `www.mabnaque.fr` ! HTTPS (et le certificat) vous garantit simplement que vous aller envoyer vos informations sur le serveur du site malveillant, et pas que c'est bien le site de votre banque.

Authentification : résultat de laboratoire Exemple

Un second exemple, rapide, serait le site d'un laboratoire qui propose de récupérer vos résultats d'examens médicaux. Sur sa page d'accueil le site affiche, tout fier, être "entièrement sécurisé". Vous constatez que le site utilise bien HTTPS mais ne propose pas d'authentification : tous les résultats d'examens de tous les patients sont accessibles publiquement.

On voit ici aussi que HTTPS ne garantit en rien que la gestion du site soit réellement sécurisée.

Stockage : panier d'achat Exemple

Si vous communiquez avec un site de vente en ligne via HTTPS :

- les données sont bien chiffrées et déchiffrées par le site ;
- rien ne garanti que ses serveurs ne seront pas victimes d'intrusion par des tiers ;
- ou qu'il ne partage pas ces données volontairement avec des tiers.

Obtention des certificats

Les certificats peuvent être créés par n'importe qui, mais il est nécessaire de passer par une autorité de certification (AC) pour que celui-ci soit reconnu par les navigateurs.

Les AC ont donc une responsabilité importante : elles ne doivent attribuer des certificats qu'aux personnes ayant prouvé leur identité, à savoir qu'elles sont bien propriétaires du nom de domaine qu'elles souhaitent certifier.

Pendant des années, le processus pour obtenir un certificat était donc le suivant : le propriétaire d'un nom de domaine devait attester auprès d'une autorité de certification de son identité civile, et prouver que c'était bien lui qui était en possession du nom de domaine. Le processus était souvent manuel (et parfois même, par un canal hors-ligne), prenait quelques heures ou jours, et coûtait cher. Une autorité de certification classique facturait, en général, plusieurs centaines d'euros par an pour un certificat.

Du fait de la lourdeur et du coût de la procédure, HTTPS a ainsi longtemps été réservé à des professionnels et/ou des acteurs pour qui les enjeux de sécurité étaient important. Ceci a fortement contribué à l'image "cadenas vert/HTTPS = site sécurisé".

Let's Encrypt, la démocratisation de HTTPS

En 2014, l'Electronic Frontier Foundation (EFF) créé, en coopération avec l'université du Michigan et Mozilla, l'organisation à but non lucratif, **Internet Security Research Group** (ISRG). L'objectif de cette organisation est, entre autre, de lancer le projet **Let's Encrypt** : une autorité de certification permettant de rendre accessible l'obtention d'un certificat, et donc de généraliser l'adoption de HTTPS.

En 2015 l'autorité de certification **Let's Encrypt** est lancée et obtient rapidement les certifications nécessaires pour être présente dans tout les navigateurs. Let's Encrypt propose à n'importe quelle personne possédant un nom de domaine d'obtenir, de manière entièrement automatisée, un **certificat gratuit** pour son domaine. Le changement est majeur : on passe d'une procédure complexe et coûteuse à une procédure automatisée, simple, et gratuite.

L'adoption est massive, et HTTPS se généralise, ne restant plus cantonné à des sites professionnels. En 2020, Let's Encrypt annonce[2] fournir des certificats pour plus de 225 millions de nom de domaines, et les analyses du site Censys[3] estiment que plus de 50% des certificats TLS du monde sont fournit par Let's Encrypt.

Enjeux politique sur les autorités de certification

Les autorités de certification ont, comme nous l'avons vu, un rôle primordial et critique dans la mise en place d'une chaîne de confiance, et donc de HTTPS. La possibilité d'émettre des certificats qui seront acceptés par les navigateurs est un pouvoir très important : si une AC décide de créer un certificat sans vérifier l'identité du demandeur, ou tout simplement pour usurper l'identité d'un site, elle en a la possibilité.

Lorsque les autorités de certification sont des agences gouvernementales ou de grandes entreprises, on peut souligner le fait que ce pouvoir risque d'être utilisé à des fins non-légitimes (politique, espionnage industriel, etc.). De la même manière, lorsqu'une AC ne sécurise pas un minimum ses procédures et ses certificats, un attaquant peut les récupérer et s'en servir pour créer de faux certificats. Et, malheureusement, ce genre d'abus et d'incidents ont déjà été observé.

Exemple

En 2012, la société de transport publics de Ankara (EGO), en Turquie, a mis en place un certificat lui permettant d'usurper l'identité de tout les sites visités en HTTPS sur son réseau local. Le but étant d'espionner le contenu des sites visités par les employés. Ce faux certificat lui a été fournit par TurkTrust, une autorité de certification gouvernementale turque, qui, à l'époque, était présente dans la liste des certificats de confiance des principaux navigateurs.

Un article résume entièrement l'incident[4] et, même si il semble qu'il n'y ai pas eu de malveillance volontaire de la part de TurkTrust, l'autorité de certification a clairement faillit à son rôle ici.

À retenir

- HTTPS n'est pas une solution miracle, c'est un protocole qui permet de chiffrer les communication HTTP, rien de plus.
- Les autorités de certification ont un rôle crucial et un pouvoir conséquent dans la mise en place de ce protocole.

2. Let's Encrypt - https://letsencrypt.org/stats/#
3. Censys - https://censys.io/certificates/report?q=tags%3Atrusted&field=parsed.issuer.organization.raw&max_buckets=50
4. https://nakedsecurity.sophos.com/2013/01/08/the-turktrust-ssl-certificate-fiasco-what-happened-and-what-happens-next/

XIII Exercice : Appliquer la notion

L'objectif de l'exercice va être de trouver les certificats inclus dans votre navigateur, et comprendre comment le site "librecours.net" peut proposer du HTTPS à votre navigateur. On utilisera ici Firefox.

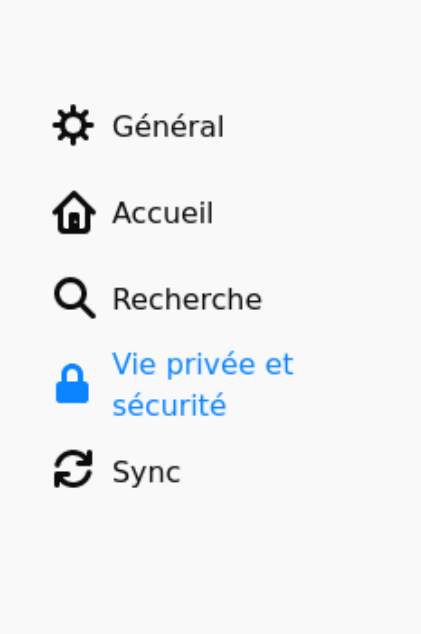
Question 1

[solution n°14 p. 37]

Dans Firefox, trouvez les certificats qui sont dits "de confiance".

Indice :

Ça se passe dans l'onglet *Vie privée et sécurité* des *Préférences*



Indice :

On peut écrire `about:preferences` dans la barre d'adresse puis choisir le menu vie privée et sécurité.

Indice :

On peut aussi écrire directement `about:preferences#privacy` dans la barre d'adresse.

Question 2

[solution n°15 p. 38]

En vous rendant sur le site https://librecours.net identifiez la chaîne de confiance du certificat.

Indice :

Pour accéder au certificat d'un site web, cliquer sur le cadenas dans la barre d'adresse.

Question 3

[solution n°16 p. 38]

Trouvez le certificat, inclus dans le navigateur, qui permet de valider la confiance dans le certificat du site Librecours.

XIV Auto-évaluation

1. Quiz

Exercice 1 : Quiz HTTP

[solution n°17 p. 38]

Exercice

À quoi sert une en-tête HTTP ?

- **A** À ajouter des informations supplémentaires, comme des méta-données, à la requête
- **B** C'est une requête envoyée juste avant la vrai requête (donc "en tête") pour préparer la connexion avec le serveur
- **C** C'est une valeur retournée par le serveur qui est stockée en mémoire du navigateur

Exercice

Que veut dire HTTP ?

- **A** High Transfert Technical Protocol
- **B** HyperText Transfert Protocol
- **C** HyperText Transport Parameter

Exercice

Quel est le port TCP utilisé par HTTP ?

Exercice

Dans le contexte de HTTP, que signifie le code 404 ?

- **A** Un code retourné par le serveur pour annoncer que la ressource demandée existe mais qu'elle nécessite de s'authentifier
- **B** Le numéro de code de la dernière version du protocole HTTP
- **C** Un code retourné par le serveur pour annoncer que la ressource demandée n'a pas été trouvée

- **D** Le nom de code du projet 404 qui a donné naissance au protocole HTTP

Exercice 6 : Quiz URL

[solution n°18 p. 39]

Exercice

Dans l'URL `https://www.picasoft.net/co/asso.html`, que représente `/co/asso.html` ?

- **A** Le nom de domaine
- **B** Le protocole
- **C** Le chemin de la ressource
- **D** Un paramètre de la requête

Exercice

Dans l'adresse `https://www.gandi.net/fr/simple-hosting`, quel est l'adresse du serveur web ?

- **A** `https://www.gandi.net`
- **B** `gandi.net`
- **C** `www.gandi.net`
- **D** `gandi.net/fr/simple-hosting`

Exercice

À quoi sert le caractère ? dans une URL ?

- **A** À indiquer un fragment (ou ancre) dans la page
- **B** À indiquer les paramètres de la requête
- **C** À indiquer la valeur des paramètres de la requête

Exercice 10 : Quizz HTTPS

[solution n°19 p. 40]

Exercice

Lorsque j'utilise HTTP, qui peut avoir accès aux données échangées ?

- **A** Le client web (navigateur)

- **B** Le serveur Web distant
- **C** Les personnes qui inspectent le trafic sur le réseau
- **D** Mon fournisseur d'accès à Internet

Exercice

Lorsque j'utilise HTTPS, qui peut avoir accès aux données échangées ?

- **A** Le client web (navigateur)
- **B** Le serveur Web distant
- **C** Les personnes qui inspectent le trafic sur le réseau
- **D** Mon fournisseur d'accès à Internet

Exercice

TLS est :

- **A** Une alternative au protocole HTTP
- **B** Un protocole d'échange chiffré utilisé par HTTPS
- **C** Un outil pour créer les certificats des sites que l'on héberge

Exercice

Pour activer HTTPS sur mon site il faut

- **A** Que j'obtienne un certificat
- **B** Que je devienne une autorité de certification
- **C** Rien, il suffit d'activer TLS sur son serveur Web

Exercice 15 : Quiz enjeux de HTTPS

[solution n°20 p. 41]

Exercice

Quelles sont les garanties que HTTPS apporte ?

- **A** Le site que l'on visite stocke les données de manière sécurisée

- **B** Les communications avec le site sont chiffrées
- **C** Le serveur web qui répond à nos requêtes est bien celui du site que l'on souhaite visiter
- **D** Le site visité n'est pas un site malveillant

Exercice

Let's Encrypt est :

- **A** Un protocole de chiffrement pour HTTPS
- **B** Un outil pour configurer son serveur web en HTTPS
- **C** Une autorité de certification gratuite et accessible au public

Exercice

Quels usages non légitime les autorités de certification pourraient avoir de leur pouvoir ?

- **A** Créer des certificats usurpant l'identité d'autres sites
- **B** Désactiver l'accès à un site web dans un navigateur
- **C** Permettre, par manque de sécurisation de ses certificats, à des pirates d'obtenir la possibilité de créer des certificats eux-mêmes

2. Exercice : Enquête MDN

[solution n°21 p. 42]

Voici les informations d'une requête HTTP effectuée à l'aide de `curl` (en utilisant l'option `-v`).

```
POST / HTTP/2

Host: www.example.com
user-agent: curl/7.72.0
accept: */*
authorization: Basic YWxhZGRpbjpzZXNhbWVPdXZyZVRvaQ==
content-length: 14
content-type: application/x-www-form-urlencoded
```

Voici les informations liées à la réponse du serveur.

```
HTTP/2 200
accept-ranges: bytes
cache-control: max-age=604800
content-type: text/html; charset=UTF-8
date: Fri, 30 Oct 2020 13:58:45 GMT
etag: "3147526947"
expires: Fri, 06 Nov 2020 13:58:45 GMT
last-modified: Thu, 17 Oct 2019 07:18:26 GMT
server: EOS (vny/0454)
content-length: 1256
```

On s'aidera de la section HTTP sur MDN pour répondre aux questions : developer.mozilla.org/en-US/docs/Web/HTTP

Exercice

Quel est le type de requête HTTP envoyée ?

developer.mozilla.org/fr/docs/Web/HTTP/Méthode[5]

- **A** GET
- **B** HEAD
- **C** POST
- **D** PUT
- **E** DELETE

Exercice

Quel est l'adresse du serveur que l'on a contacté ?

developer.mozilla.org/fr/docs/Web/HTTP/Headers/Host

Exercice

Que nous indique l'en-tête `Authorization` ?

developer.mozilla.org/fr/docs/Web/HTTP/Headers/Authorization

- **A** La requête s'est authentifiée avec l'utilisateur `Basic` et le mot de passe `YWxhZGRpbjpzZXNhbWVPdXZyZVRvaQ`
- **B** La requête s'est authentifiée en utilisant la méthode `Basic`
- **C** La requête s'est authentifiée en utilisant la chaîne d'authentification `YWxhZGRpbjpzZXNhbWVPdXZyZVRvaQ==`

Exercice

Quel est le code de retour de cette requête ?

developer.mozilla.org/fr/docs/Web/HTTP/Status

5. https://developer.mozilla.org/fr/docs/Web/HTTP/M%C3%A9thode

Solutions des exercices

Solution n°1

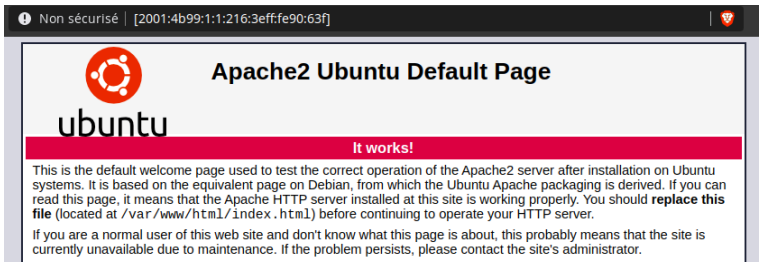
[exercice p. 6]

```
1 apt update && apt install apache2
```

Solution n°2

[exercice p. 6]

Une page s'affiche :



Cette page est servie automatiquement par Apache lors d'une nouvelle installation.

Solution n°3

[exercice p. 6]

```
1 ls /var/www/html
```

La commande retourne `index.html`, ce qui indique que le fichier servi par Apache est situé à l'emplacement `/var/www/html/index.html`.

Solution n°4

[exercice p. 11]

La taille est de 135 Ko environ.

Solution n°5

[exercice p. 11]

Le code 404 est renvoyé car la ressource est introuvable.

Solution n°6

[exercice p. 14]

On constate que le navigateur ouvre la page Houmous de Wikipédia, et se rend directement au niveau de la partie "Origine".

C'est grâce au fragment dans l'URL (on parle aussi d'ancre) qui est spécifié par `#Origine`.

Solution n°7

[exercice p. 14]

Le point d'interrogation dans l'URL sert à indiquer le début des paramètres. Il n'est donc pas valide pour indiquer le chemin d'un fichier (ici la page `/??`)

Solution n°8

[exercice p. 14]

Pour les caractères invalides dans une URL, on utilise un symbole pourcent suivi de la notation ASCII en hexadécimal. ? en hexadécimal ASCII donne 3F.

L'URL correcte est donc https://fr.wikipedia.org/wiki/%3F%3F

Solution n°9

[exercice p. 17]

Le code renvoyé est `301` ce qui signifie que la page est redirigée vers une autre page, de manière définitive.

Solution n°10

[exercice p. 17]

Le code `201` pour "Created"

Solution n°11

[exercice p. 17]

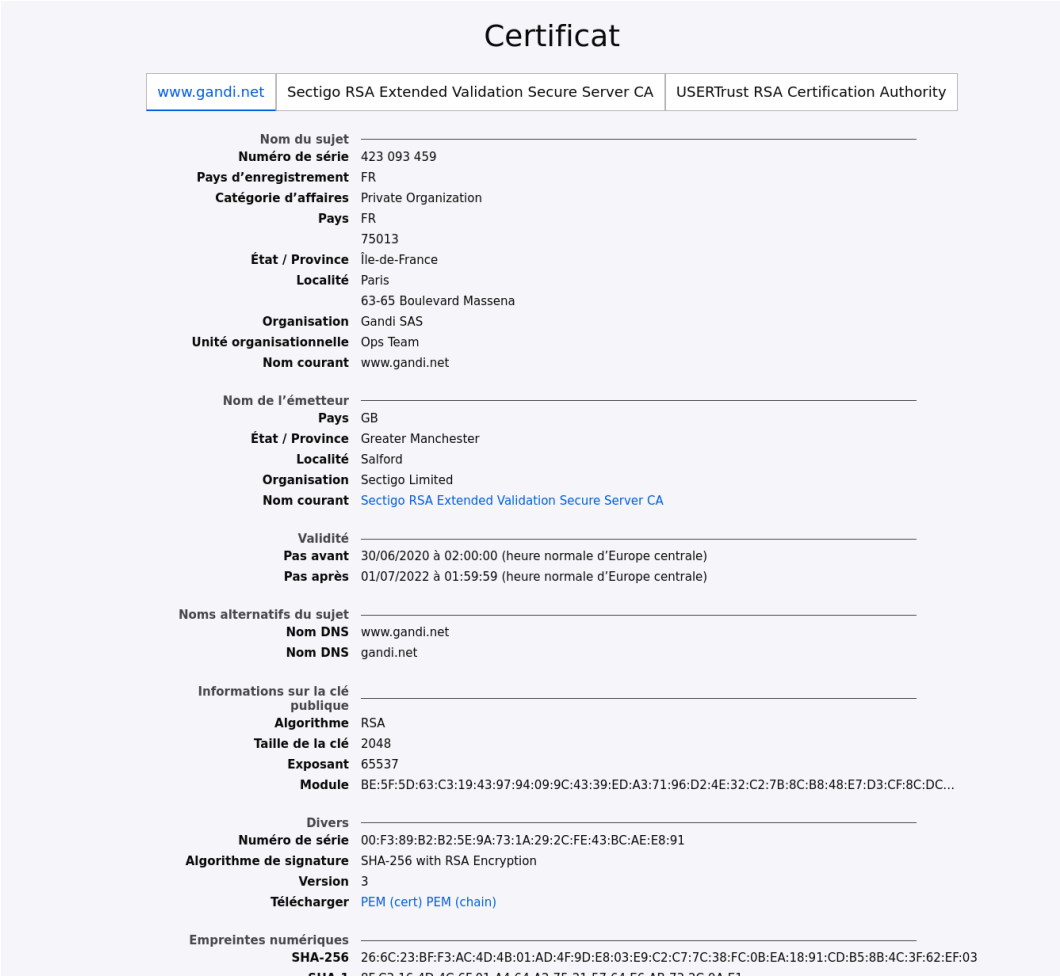
Le code 418 signifie "I'm a Teapot", c'est à dire "Je suis une théière".

C'est une vraie farce introduite dans le protocole HTTP le 1er avril 1998. Bien évidemment, il est très peu utilisé.

Solution n°12

[exercice p. 22]

En cliquant sur le cadenas, puis sur "Afficher le certificat", on obtient les informations du certificat dans une interface graphique. On y retrouve par exemple le nom de domaine "www.gandi.net".



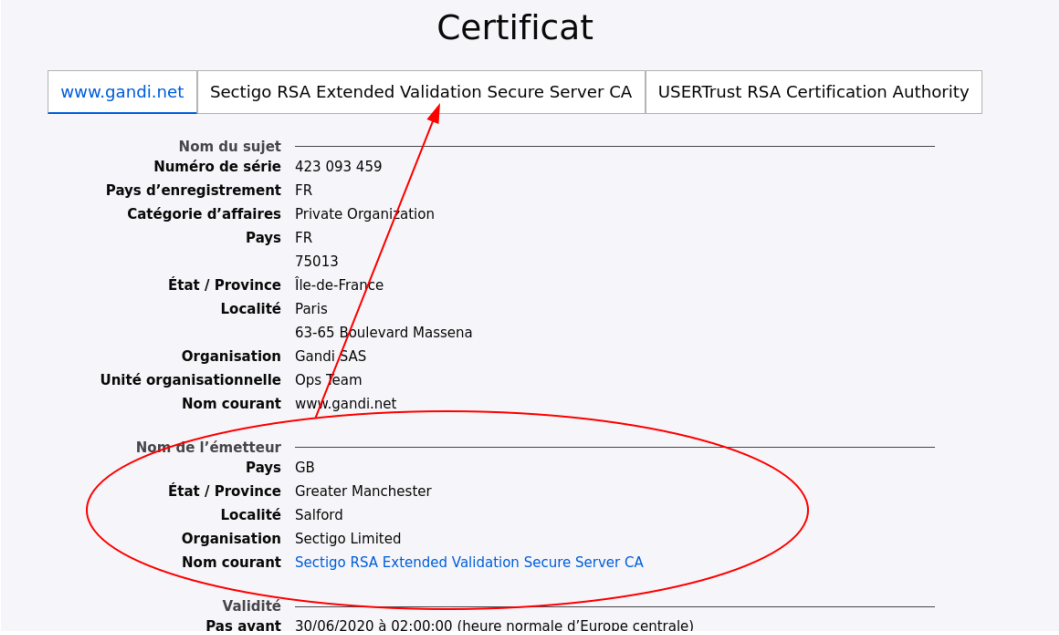
Solution n°13

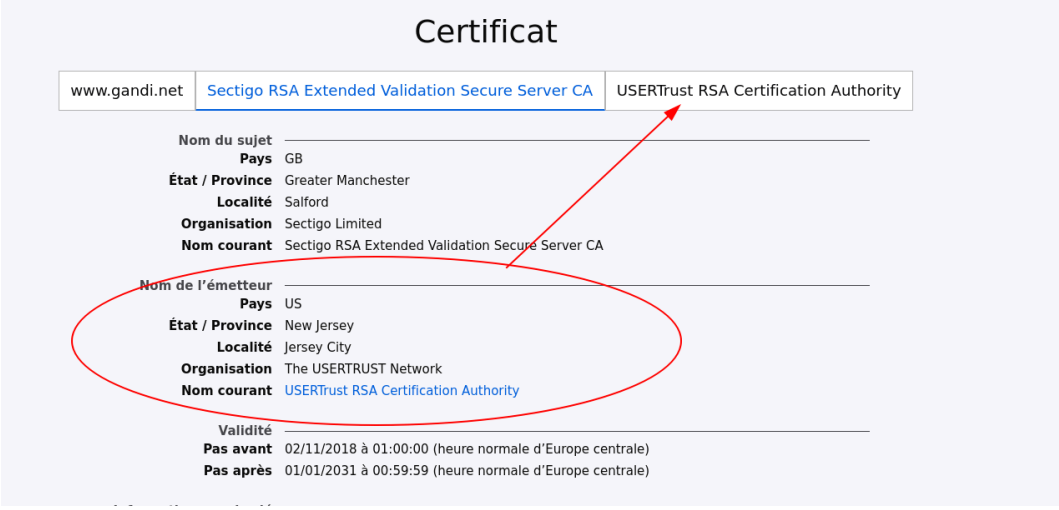
[exercice p. 23]

On constate la chaîne de confiance suivante :

USERTrust -> Sectigo -> Gandi

Pour cela il suffit de regarder le partie "Nom de l'émetteur" du certificat, puis de faire cela en remontant la chaîne. L'interface affiche cela très simplement à l'aide des onglets pour chaque certificat.



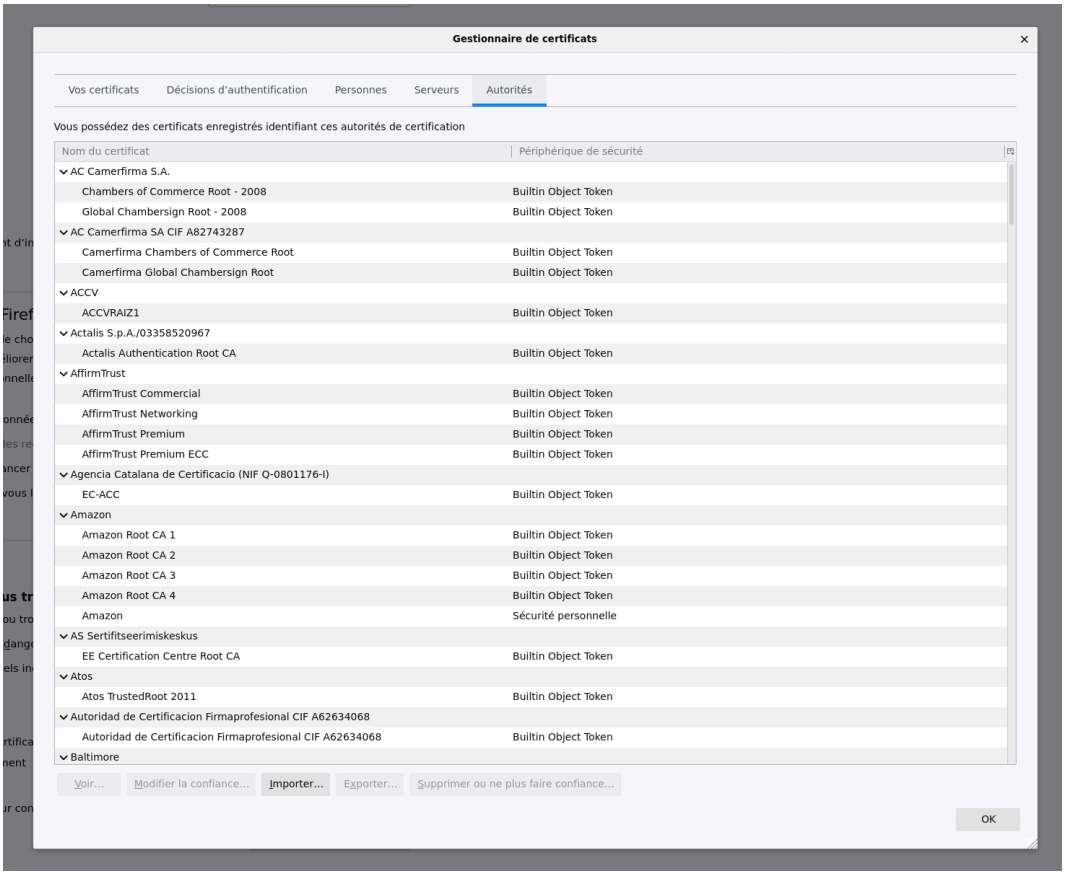


Solution n°14

[exercice p. 27]

La section sécurité propose d'afficher les certificats.

Ce qui ouvre une fenêtre avec la liste de tous les certificats intégrés dans le navigateur.



Solution n°15

[exercice p. 28]

librecours.net → Let's Encrypt → Internet Security Research Group

Complément

Vous pouvez lire le certificat du site directement en entrant l'adresse suivante dans votre navigateur Firefox

```
about:certificate?
cert=MIIEdDCCA1ygAwIBAgISBD0fp4fOUBcJOYSRuwevPppZMA0GCSqGSIb3DQEBCwUAMDIxCzAJBgNVBAYTAlVTMRYwFA
```

Solution n°16

[exercice p. 28]

On constate que le certificat `Let's Encrypt`, qui a signé le certificat de Librecours, est présent dans notre navigateur.

Solution n°17

[exercice p. 29]

Exercice

À quoi sert une en-tête HTTP ?

A À ajouter des informations supplémentaires, comme des méta-données, à la requête

B
C'est une requête envoyée juste avant la vrai requête (donc "en tête") pour préparer la connexion avec le serveur

C C'est une valeur retournée par le serveur qui est stockée en mémoire du navigateur

Exercice

Que veut dire HTTP ?

A High Transfert Technical Protocol

B HyperText Transfert Protocol

C HyperText Transport Parameter

Exercice

Quel est le port TCP utilisé par HTTP ?

Le port 80

Exercice

Dans le contexte de HTTP, que signifie le code 404 ?

A
Un code retourné par le serveur pour annoncer que la ressource demandée existe mais qu'elle nécessite de s'authentifier

B Le numéro de code de la dernière version du protocole HTTP

C
Un code retourné par le serveur pour annoncer que la ressource demandée n'a pas été trouvée

D Le nom de code du projet 404 qui a donné naissance au protocole HTTP

Solution n°18

[exercice p. 30]

Exercice

Dans l'URL `https://www.picasoft.net/co/asso.html`, que représente `/co/asso.html` ?

A Le nom de domaine

B Le protocole

- (C) Le chemin de la ressource
- **D** Un paramètre de la requête

Exercice

Dans l'adresse `https://www.gandi.net/fr/simple-hosting`, quel est l'adresse du serveur web ?

- **A** `https://www.gandi.net`
- **B** `gandi.net`
- (C) `www.gandi.net`
- **D** `gandi.net/fr/simple-hosting`

Exercice

À quoi sert le caractère ? dans une URL ?

- **A** À indiquer un fragment (ou ancre) dans la page
- (B) À indiquer les paramètres de la requête
- **C** À indiquer la valeur des paramètres de la requête

Solution n°19

[exercice p. 30]

Exercice

Lorsque j'utilise HTTP, qui peut avoir accès aux données échangées ?

- [A] Le client web (navigateur)
- [B] Le serveur Web distant
- [C] Les personnes qui inspectent le trafic sur le réseau
- [D] Mon fournisseur d'accès à Internet

HTTP étant un protocole "en clair", tout ce qui est échangé est lisible pour toute personne sur le trajet des échanges.

Exercice

Lorsque j'utilise HTTPS, qui peut avoir accès aux données échangées ?

- **A** Le client web (navigateur)
- **B** Le serveur Web distant
- **C** Les personnes qui inspectent le trafic sur le réseau
- **D** Mon fournisseur d'accès à Internet

Grâce à HTTPS, les communications sont chiffrées entre le client et le serveur

Exercice

TLS est :

- **A** Une alternative au protocole HTTP
- **B** Un protocole d'échange chiffré utilisé par HTTPS
- **C** Un outil pour créer les certificats des sites que l'on héberge

TLS est le protocole qui permet d'encapsuler HTTP pour mettre en place HTTPS

Exercice

Pour activer HTTPS sur mon site il faut

- **A** Que j'obtienne un certificat
- **B** Que je devienne une autorité de certification
- **C** Rien, il suffit d'activer TLS sur son serveur Web

Solution n°20

[exercice p. 31]

Exercice

Quelles sont les garanties que HTTPS apporte ?

- **A** Le site que l'on visite stocke les données de manière sécurisée

- [ ] **B** Les communications avec le site sont chiffrées
- [x] **C** Le serveur web qui répond à nos requêtes est bien celui du site que l'on souhaite visiter
- [ ] **D** Le site visité n'est pas un site malveillant

Exercice

Let's Encrypt est :

- [ ] **A** Un protocole de chiffrement pour HTTPS
- [ ] **B** Un outil pour configurer son serveur web en HTTPS
- [x] **C** Une autorité de certification gratuite et accessible au public

Exercice

Quels usages non légitime les autorités de certification pourraient avoir de leur pouvoir ?

- [ ] **A** Créer des certificats usurpant l'identité d'autres sites
- [ ] **B** Désactiver l'accès à un site web dans un navigateur
- [x] **C** Permettre, par manque de sécurisation de ses certificats, à des pirates d'obtenir la possibilité de créer des certificats eux-mêmes

Solution n°21

[exercice p. 32]

Exercice

Quel est le type de requête HTTP envoyée ?

developer.mozilla.org/fr/docs/Web/HTTP/Méthode[6]

- [ ] **A** GET
- [ ] **B** HEAD
- [x] **C** POST
- [ ] **D** PUT

6. https://developer.mozilla.org/fr/docs/Web/HTTP/M%C3%A9thode

E DELETE

C'est une requête POST, comme indiqué dans la première ligne de la requête.

Exercice

Quel est l'adresse du serveur que l'on a contacté ?

developer.mozilla.org/fr/docs/Web/HTTP/Headers/Host

www.example.com

L'en-tête `Host` de la requête indique que l'adresse est `www.example.com`.

Exercice

Que nous indique l'en-tête `Authorization` ?

developer.mozilla.org/fr/docs/Web/HTTP/Headers/Authorization

A

La requête s'est authentifiée avec l'utilisateur `Basic` et le mot de passe `YWxhZGRpbjpzZXNhbWVPdXZyZVRvaQ`

B La requête s'est authentifiée en utilisant la méthode `Basic`

C

La requête s'est authentifiée en utilisant la chaîne d'authentification `YWxhZGRpbjpzZXNhbWVPdXZyZVRvaQ==`

« Si c'est le type d'authentification "Basic" qui est utilisé, les identifiants sont construits de la manière suivante :

- L'identifiant de l'utilisateur et le mot de passe sont combinés avec deux-points : `aladdin:sesameOuvreToi`
- Cette chaîne de caractères est ensuite encodée en base64 : `YWxhZGRpbjpzZXNhbWVPdXZyZVRvaQ==` »

Exercice

Quel est le code de retour de cette requête ?

developer.mozilla.org/fr/docs/Web/HTTP/Status

200 OK

Le code HTTP 200, qui correspond simplement à "OK".

Crédits des ressources

Contenu d'une requête HTTP et de sa réponse p. 47
Attribution - Partage dans les Mêmes Conditions - framasoft.org

Outils de développement Firefox (onglet requête) p. 10
Licence : Domaine Public

Décomposition d'une URL p. 12
Licence : Domaine Public

Contenus annexes

1. HTTP : HyperText Transfert Protocol

Objectif

- Se familiariser avec HTTP.

Mise en situation

HTTP est le protocole du Web : il permet à un client (le navigateur) d'obtenir des pages et des ressources depuis un serveur web (le site).

On connaît ce protocole car les adresses web commencent par `http://`.

Par exemple, `http://en.wikipedia.org/wiki/Tim_Berners-Lee` signifie : je souhaite communiquer avec le serveur `en.wikipedia.org` en utilisant le protocole HTTP, je lui demande de me donner la page qui s'appelle `Tim_Berners-Lee`. On appelle cela une requête HTTP.

Une courte histoire du Web — Rappel

HTTP est conçu au début des années 90 par Tim Berners-Lee[7] dans le cadre de ses travaux au CERN. L'objectif est de mettre en place un système qui permette l'échange d'informations et de ressources, entre les chercheurs du CERN, en s'appuyant sur le réseau interne. Le projet prend rapidement le nom de *World Wide Web* et abouti à la création du protocole HTTP et du premier navigateur.

HTTP — Définition

HTTP (pour *HyperText Transfert Protocol*) est le protocole utilisé par le Web pour le transfert de ressources.

Ce protocole fonctionne sous la forme de **requête-réponse** dans une architecture **client-serveur** : le client adresse une requête HTTP à un serveur qui lui enverra une réponse appropriée.

Ce protocole repose généralement sur un protocole de transport fiable, tel que TCP.

Port 80 — Fondamental

Le port **80** est le port réservé par les serveurs Web pour HTTP.

Les requêtes HTTP

Il existe différent types de requêtes HTTP, voici les plus importantes :

- GET : c'est la plus commune et elle demande à recevoir une certaine ressource disponible sur un serveur web. Quand un client fait une requête GET, c'est qu'il veut uniquement recevoir une ressource et qu'il ne cherche pas à la modifier.

7. Tim Berners-Lee - https://fr.wikipedia.org/wiki/Tim_Berners-Lee

- POST: une requête POST est utilisée dans l'intention de modifier ou ajouter une ressource sur le serveur web.
- DELETE : elle est utilisée par le client lorsqu'il veut supprimer une ressource d'un serveur web.

Format des requêtes

Les requêtes HTTP suivent le format suivant :

- le type de la requête (GET, POST, etc.) avec la référence de la ressource en question sur le serveur,
- des en-têtes donnant diverses informations au serveur,
- un corps optionnel contenant des données utiles à la requête ; dans le cas de l'ajout d'une ressource, le corps contiendra la ressource en elle-même.

Un message HTTP

Exemple

```
GET /wiki/Hypertext_Transfer_Protocol HTTP/1.1
Host: www.wikipedia.org
```

On demande à recevoir (GET) la ressource /wiki/Hypertext_Tansfert_Protocol se trouvant sur le serveur www.wikipedia.org.

La directive `Host:` est une en-tête.

Réaliser une requête GET

Méthode

- Les navigateurs web exécute une requête GET à chaque fois qu'un utilisateur demande à consulter une page web.
- Il existe des programmes tels que `wget` ou `curl` pour effectuer des requêtes HTTP en ligne de commande.

```
curl https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol
```

```
wget https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol
```

Cette commande enregistre le contenu de la page `Hypertext_Transfer_Protocol` de Wikipédia. Contrairement à un navigateur web, la page n'est pas affichée ni mise en page.

Les en-têtes HTTP

Les en-têtes (ou *headers*) HTTP permettent d'ajouter de nombreuses informations aux requêtes, ce qui facile l'ajout de fonctionnalités au protocole : gestion du cache, de l'authentification, des formats de données à échanger, etc.

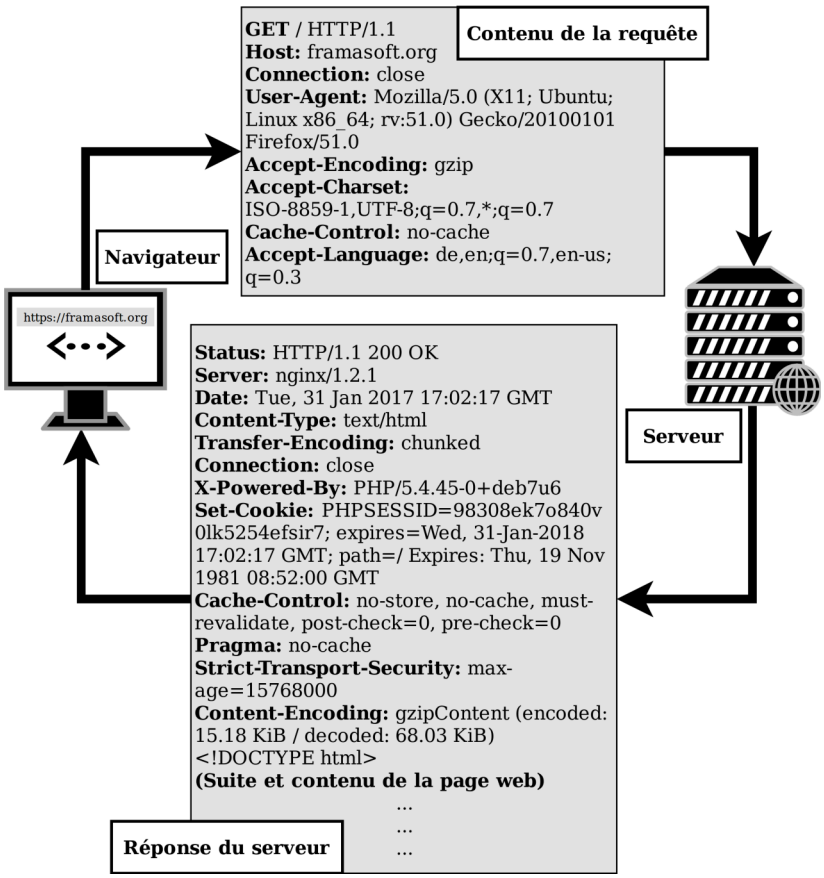
Exemple

Nous avons vu plus haut une requête utilisant le *header* `Host`. Un autre très utilisé est le *header* `Authorization` qui est utilisé pour gérer l'accès avec des identifiants à une ressource. Sans rentrer dans le détail, voici un exemple :

```
GET /internal/secret_file.txt HTTP/1.1
Authorization: Basic a3lhbmU6
```

Ici on demande l'accès à la ressource /internal/secret_file.txt en utilisant de l'authentification. L'en-tête `Authorization` permet de spécifier que l'on utilisera la méthode d'authentification "Basic" (il en existe plusieurs) et on passe comme valeur d'identifiant "a3lhbmU6".

Récapitulatif Fondamental



Contenu d'une requête HTTP et de sa réponse

À retenir

- HTTP est le protocole du Web qui permet d'échanger des ressources, le plus souvent des fichiers HTML.
- Il permet de faire des requêtes pour demander à consulter, ajouter, modifier ou supprimer une ressource sur un serveur web.

Complément

- *Serveurs web* (cf. p.4)
- *Navigateurs web* (cf. p.7)

2. VPS : serveur dédié virtuel

Objectifs

- Savoir ce qu'est un VPS
- Savoir créer un VPS chez un hébergeur
- Savoir se connecter à distance sur un VPS avec SSH

Serveur

Rappel

Un **serveur** est un ordinateur accessible depuis Internet, qui rend des **services** aux utilisateurs.

Il se distingue des **ordinateurs personnels** que l'on ne peut pas contacter directement aussi simplement depuis Internet.

Utilisation quotidienne des serveurs

Exemple

- Lorsque je me rends sur le site wikipedia.org[8], je demande en réalité aux **serveurs** de Wikipédia de m'envoyer le contenu de la page que je veux afficher.
- Un ami ne peut pas accéder aux fichiers de mon ordinateur personnel : pour les partager, je dois les téléverser sur un **serveur** (envoyer un mail, utiliser un service partage de fichiers, etc.).

Remarque

Tout ordinateur personnel peut être transformé temporairement en serveur, mais on ne traite pas ce cas ici.

VPS

Définition

Un VPS (serveur dédié virtuel, ou *Virtual Private Server*) peut s'envisager comme un serveur réservé à son usage personnel. En réalité, il s'agit d'une partie d'un serveur physique isolée du reste du système : un serveur **virtuel**.

À quoi sert un VPS ?

Exemple

Un VPS peut servir :

- à mettre en ligne un site web : serveur Apache, Nginx.
- à travailler à plusieurs sur une même machine : partage de fichier avec SFTP, serveur NextCloud, serveur GitLab.
- à tester et installer d'autres applications web : Etherpad, Mattermost.
- à tester et installer des applications d'Internet : mail.

8. Wikipédia - https://fr.wikipedia.org

Créer un VPS chez un hébergeur

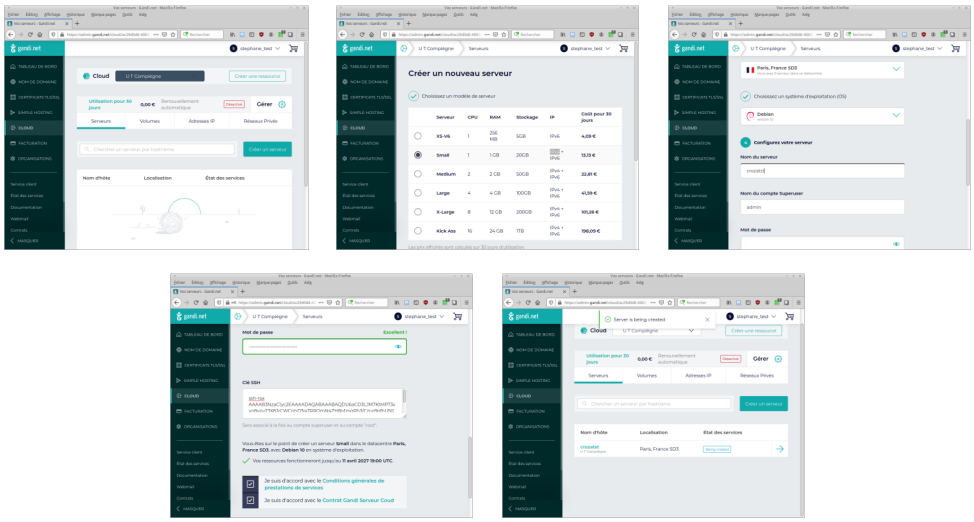
Méthode

Il existe plusieurs **hébergeurs** professionnels qui proposent la location de VPS, on retrouvera en général les étapes suivantes :

1. Se rendre sur le site de l'hébergeur (exemple : gandi.net[9])
2. Choisir une offre (à noter que pour disposer d'un serveur réellement accessible sur Internet par tout le monde, il faut que le VPS soit doté d'une adresse IPv4)
3. Choisir le système d'exploitation souhaité, sa version (par exemple : Debian 10)
4. Choisir un nom pour identifier le VPS, créer un mot de passe **robuste** pour le compte administrateur et éventuellement associer une clé SSH

Créer un VPS chez Gandi

Exemple



Créer un VPS chez Gandi

Remarque

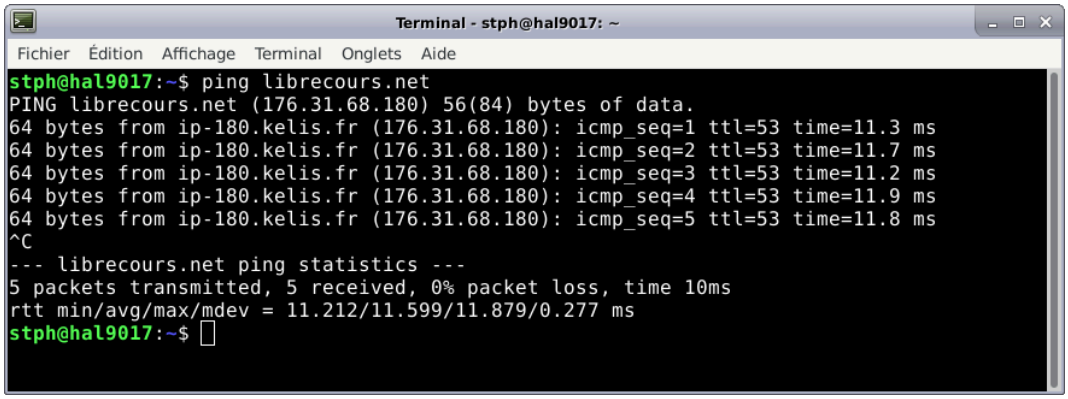
Le VPS est contactable par son **adresse IP**, qui est unique sur Internet et est l'équivalent d'une adresse postale.

On peut utiliser la commande `ping` pour vérifier qu'un serveur répond bien.

9. https://www.gandi.net/fr/cloud

Ping

Exemple



Accéder à un VPS avec SSH

Méthode

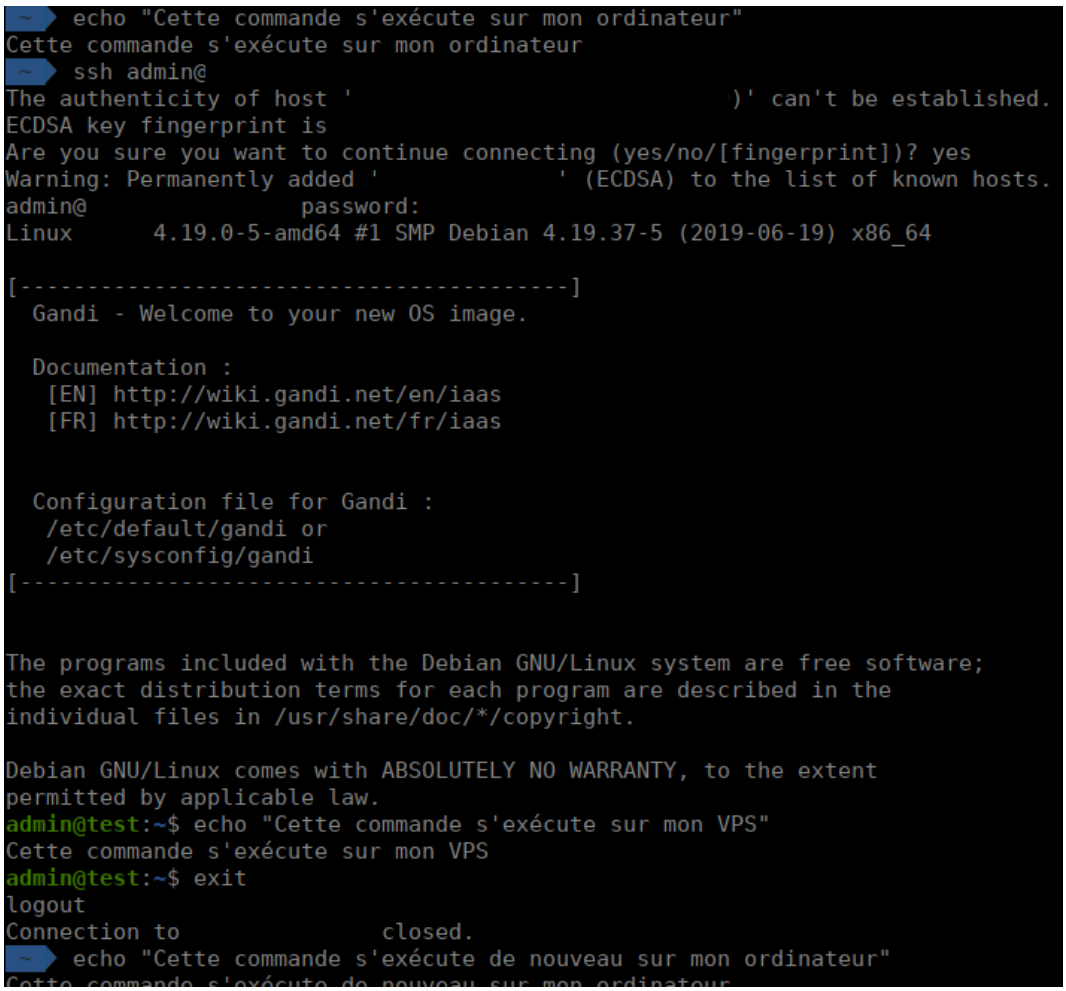
Pour travailler sur un VPS, il faut un moyen de s'y connecter et d'y ouvrir un shell. SSH (*Secure SHell*) est un outil standard qui remplit cette fonction : une fois la connexion établie, on travaille sur un VPS comme on travaille sur un shell local.

Dans un shell local, copier la commande reçue par mail pour ouvrir un shell distant sur le VPS.

```
1 ssh <super-utilisateur>@<adresse-IP>
```

Accéder à un VPS avec SSH

Exemple



Cette image montre une session SSH classique :

- La première commande s'exécute sur l'ordinateur local.
- Après la connexion SSH, les commandes s'exécutent automatiquement sur le VPS distant.
- La commande `exit` ferme la connexion SSH, les commandes s'exécutent de nouveau sur l'ordinateur local.

Autres fournisseurs de VPS français

Complément

Il existe d'autres fournisseurs de VPS français, comme OVH[10] et Scaleway[11].

10. OVH - https://www.ovh.com/fr/
11. Scaleway - https://www.scaleway.com/fr/

SSH et Windows 10

Complément

Windows n'intègre pas SSH par défaut. Il y plusieurs possibilités pour l'installer :

- Suivre le tutoriel de Microsoft[12] pour activer l'utilisation de SSH dans `powershell`.
- Installer un logiciel tiers, comme PuTTY[13].
- Utiliser SSH dans un shell Bash, en installant le sous système Linux[14].

Pourquoi louer un VPS et pas un serveur physique ?

Complément

Les VPS répondent à un problème classique : louer un serveur physique impose de choisir des composants adaptés à la puissance voulue. Si les besoins augmentent, il faut changer de machine, ce qui peut être très coûteux.

Les fournisseurs de serveurs ont trouvé une astuce : séparer un serveur physique en plusieurs serveurs **virtuels**, dont la puissance peut être adaptée en fonction des besoins. Pour les utilisateurs, le coût est moindre, et pour les fournisseurs, l'utilisation d'un serveur physique est optimisée.

À retenir

- Un VPS est l'équivalent d'un serveur que l'on peut louer pour son usage personnel. Il est accessible depuis Internet.
- SSH permet de se connecter à distance sur son VPS, et d'y exécuter des commandes.
- Il existe plusieurs fournisseurs de VPS français, comme Gandi, OVH ou Scaleway.

12. Installation d'OpenSSH sous Windows 10 - https://docs.microsoft.com/fr-fr/windows-server/administration/openssh/openssh_install_firstuse
13. PuTTY - https://putty.org/
14. https://docs.microsoft.com/fr-fr/windows/wsl/install-win10