

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole
HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures
HTTP

Proxy
Reverse Proxy

Web
Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

Langages Web 1



Connectivité Web

Florent Nicart

Université de Rouen

2016–2017

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

En 1989, Tim Berners-Lee, informaticien au CERN¹ propose un système hypertexte distribué et ses trois composants :

- le langage HTML permettant d'encoder des documents hypertextes,
- les URL, permettant d'identifier et de localiser (et lier) ces documents répartis à travers le monde,
- et le protocole HTTP permettant de naviguer dans le graphe ainsi créé.

Dans ce chapitre, nous allons nous intéresser à la connectivité du web : les URL et HTTP.

1. Organisation européenne pour la recherche nucléaire

Définitions : ressource

- La plupart du temps, une ressource correspond à un fichier. HTTP transporte alors un flux binaire correspondant à son contenu, (on parle de site web statique).
- Dans un fonctionnement dynamique, le serveur renvoie une « représentation » de la ressource. Ex : index d'un répertoire, article pris dans un SGBD et mis en forme à la volée en HTML (CMS²), image d'une webcam, l'heure, etc.
- On notera dans ce cas que les ressources web n'ont pas un comportement fonctionnel : deux requêtes identiques consécutives peuvent donner des résultats différents.

Représentations : texte ou binaire ?



Le monde s'encode en **10** catégories :

- le texte
- le binaire

Plus sérieusement un flux « texte » est un flux binaire où chaque entier est associé à un symbol implicitement ou explicitement.

→ le texte doit être encodé. (`char(65) = 'A'`)

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

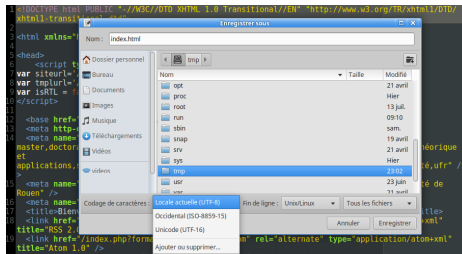
C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Encodage du texte



Encoder du texte consiste :

- Choisir une carte/méthode établissant une correspondance entre symboles et valeurs numériques,
- produire un flux où tous les caractères sont traduits selon cette carte,
- et enfin s'assurer que le destinataire ait connaissance de cette carte/méthode.

Encodage du texte

Types d'encodage

Encodages fixes :

- ASCII (*American Standard Code for Information Interchange*) : 127 symboles encodés sur 1 octet.
- UTF-32 (Unicode) : 4 octets pour encoder les symboles de tous les dialectes reconnus.

Encodages variables :

- ISO-8859 : encodage sur 1 octet reprenant l'ASCII pour la première partie et permettant de choisir des pages pour la seconde. La page est choisie une bonne fois pour toute et ne change pas en cours de flux (256 symboles adressables).
- UTF-8, UTF-16 : encodage adressant toute la carte Unicode (env 16 millions de symboles) utilisant dynamiquement au moins un (resp. deux) octet.

Encodage du texte

la carte ASCII

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]

Association des valeurs 0 à 127 de chaque octet.

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

Encodage du texte

Système de pages : iso8859

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

Code		Numéro de partie d'ISO 8859															
Déc	Hex	1	2	3	4	5	6	7	8	9	10	11	13	14	15	16	
160	A0	NBSP															
161	A1	ı	À	Ħ	À	Ë		'		i	À	ñ	"	ß	ı	À	
162	A2	ç	ˆ	ˆ	κ	ἥ		'	ç	ç	Ë	ı	ç	b	ç	ç	
163	A3	£	Ł	£	Ŕ	Í		£	£	£	Œ	ı	£	£	£	Ł	
164	A4	¤	¤	¤	¤	€	¤	€	¤	¤	İ	¤	¤	Č	€	€	
165	A5	¥	Ł		İ	S		Đ	¥	¥	İ	¤	"	č	¥	"	
166	A6	ı	Š	Ĥ	Ł	ı		ı	ı	ı	Ł	ı	ı	Đ	Š	Š	
167	A7	§	§	§	§	İ		§	§	§	§	ı	§	§	§	§	
168	A8	"	"	"	"	J		"	"	"	Ł	ı	Ø	W	š	š	
169	A9	©	Š	ı	Š	Ł		©	©	©	Đ	ı	©	©	©	©	
170	AA	ä	Ş	Ş	Ë	Ĥ		x	ä	Š	ı	Ŕ	W	ä	Ş	Ş	
171	AB	«	Ť	Ĝ	Œ	ἥ		«	«	«	Œ	ı	«	d	«	«	
172	AC	ı	Ž	ı	Ŧ	Ķ	,	ı	ı	ı	Ž	ı	ı	ı	ı	ı	

Affectation variable des valeurs 128 à 255 de chaque octet.

Encodage du texte

Unicode

- UTF-32 : chaque caractère est codé sur un mot de 32 bits (4 octets).
- UTF-16 : chaque caractère est codé sur 1 ou 2 mots de 16 bits (2 octets).
- UTF-8 : utilise dynamiquement 1 à 4 octets selon un schéma rappelant le système des adresses IPv4 :

0xxxxxxx	1 octet
10xxxxxx 10xxxxxx	2 octets
110xxxxx 10xxxxxx 10xxxxxx	3 octets
1110xxxx 10xxxxxx 10xxxxxx 10xxxxxx	4 octets

- Unicode supporte les deux boutismes (Little et big endian).
- UTF-8 est à l'heure actuelle l'encodage privilégié sur le web (il est utilisé par environ 82% des sites web).

Encodage du texte

Moralité

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

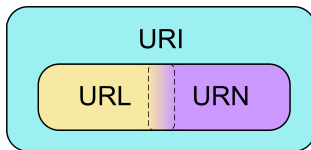
HTTPS

- Dans la suite de ce cours, nous allons manipuler des protocoles orientés texte, langages, des données textuels.

À chaque fois il conviendra de :

- savoir quels caractères on a le droit d'utiliser,
- s'assurer que, lors d'un échange, l'émetteur et le récepteur des données utilisent sur le même encodage.

Définitions : URL



La famille d'identifieurs **URI** (pour **Uniform Resource Identifier**) défini (RFC 2396 🖱️) un moyen de pouvoir identifier tout type de ressources à l'aide d'une chaîne caractères. Elle contient deux classes particulières :

- les **URLs (Uniform Resource Locator)** qui identifie une ressource et son emplacement ; ex :
`ftp://ftp.kernel.org/pub/`
- les **URNs (Uniform Resource Name)** qui identifie une ressource dans un espace de noms ; ex :
`URN urn:isbn:0-395-36341-1` pour un livre.

Définitions : URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

- Nous parlerons donc plutôt d'URL que d'URI ou d'URN.
- On emploie souvent le terme d'« adresse web », à tort puisque les URL peuvent porter sur d'autres protocoles que HTTP.
- Enfin, pour des références du type

`/accueil/index.html`

on parlera de *chemin relatif* plutôt que d'URL puisque l'URL correspondante n'existe que lorsqu'elle est construite par le navigateur à partir de l'emplacement du document la contenant.³

3. Essayez donc ce type de chemin dans la barre d'adresse de votre navigateur.

Syntaxe d'une URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

```
proto://[login[:passwd]@]host[:port] ↵  
/path[?req]+[#frag]
```

- *proto* : indique au navigateur le protocole à utiliser pour accéder à la ressource.
- Exemples : http, https, ftp, sftp, etc.
- Note : `file://...` dénote en général un système de fichiers local et ne fait appel à aucun protocole réseau.

Syntaxe d'une URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

```
proto://[login[:passwd]@]host[:port] ←  
/path[?req]+[#frag]
```

- *login* : si l'accès à la ressource nécessite une identification, indique le nom de l'utilisateur à utiliser (connu de l'hôte) ;
- Si une authentification est requise, un mot de passe sera demandé (boîte de dialogue du navigateur) pour une deuxième requête.
- Exemple :

```
http://florent@www.univ-rouen.fr
```

Syntaxe d'une URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

proto://[login[:passwd]@]host[:port] ↵
/path[?req]+[#frag]

- *passwd* : permet d'indiquer un mot de passe dès la première requête.
- implique de fournir le nom de l'utilisateur ;
- Exemple :

http://florent:clatu@www.univ-rouen.fr

Syntaxe d'une URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

`proto://[login[:passwd]@]host[:port] ↵
/path[?req]+[#frag]`

- *host* : un nom de domaine (complètement qualifié ou non) ou une adresse IP permettant d'établir une connexion avec l'hôte ;


- Exemples :

`http://www.univ-rouen.fr` (FQDN)

`http://www/` (FQDN complété par suffixe DNS)

`http://193.52.152.29/` (FQDN)

Syntaxe d'une URL

proto://[login[:passwd]@]host[:port] 
/path[?req]+[#frag]

- *port* : indique le port TCP sur lequel le service est joignable sur l'hôte.
- Par défaut, le navigateur détermine sa valeur en fonction du protocole (80 pour HTTP, 443, pour HTTPS, 21, pour FTP, etc.).

- Exemples :

http://www.univ-rouen.fr:8080

https://www.univ-rouen.fr:1111

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Syntaxe d'une URL

proto://[login[:passwd]@]host[:port] ↵
/path[?req]+[#frag]

- *path* : indique le chemin et le nom de la ressource sur l'hôte.
- Peut-être vide (racine du site).
- L'arborescence des chemins peut être virtuelle et ne pas correspondre à un sous-arbre du système de fichier de l'hôte.
- Exemples :

http://www.univ-rouen.fr

http://www.univ-rouen.fr/masters/

http://univ-rouen.fr/masters/gil.html

Syntaxe d'une URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web


Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

proto://[login[:passwd]@]host[:port] 
/path[?req]+[#frag]

- *req* : indique les paramètres de la requête.
- Souvent composé de couples cle=valeur séparés par un symbol &.
- Exemples :

http://www.univ-rouen.fr?lang=fr

http://www.google.fr/search?hl=fr&safe=off&client=ubuntu&hs=iUF&...

Syntaxe d'une URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

```
proto://[login[:passwd]@]host[:port] ←  
/path[?req]+[#frag]
```

- *frag* : indique l'identifiant d'un fragment d'une ressource.
- Interprété par le navigateur pour placer automatiquement la vue sur l'élément ayant l'id correspondant.
- Exemple :
`http://en.wikipedia.org/wiki/Fragment_`
`identifier#Exemples`

Syntaxe d'une URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

Une URL est un chaîne de caractères...**donc ?** ...

D'après la RFC :

- Le jeu de caractère utilisable est l'ASCII,
- Les espaces ou tout caractère non ASCII doivent être échappés à l'aide de caractère ASCII, ex :

ça farte ! → %C3%A7a%20farte%20!

- Les symboles appartenant à l'alphabet des URL doivent aussi être échappés : : / ? = # %

Syntaxe d'une URL

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

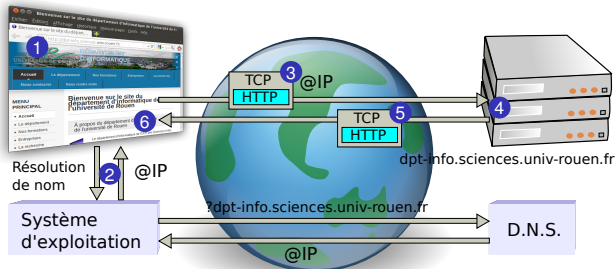
Une URL est un chaîne de caractères...**donc ?** ...
D'après la RFC :

- Le jeu de caractère utilisable est l'ASCII,
- Les espaces ou tout caractère non ASCII doivent être échappés à l'aide de caractère ASCII, ex :

`ça farte !` → `%C3%A7a%20farte%20!`

- Les symboles appartenant à l'alphabet des URL doivent aussi être échappés : `: / ? = # %`

Analyse et exploitation d'une URL



- ② Pour adresser la requête à l'hôte, une résolution de nom a lieu (sauf si l'URL contenait déjà une adresse IP) :

cache du navigateur > fichier hosts > requête DNS

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web

Dynamique

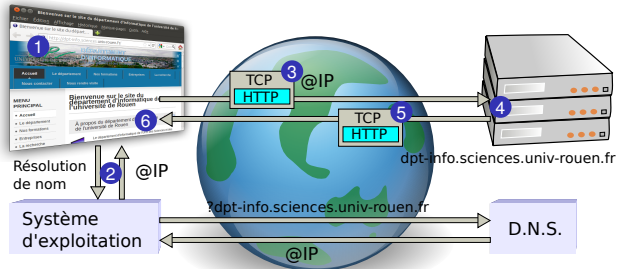
C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Analyse et exploitation d'une URL



- ③ Un paquet IP est forgé à l'attention de l'hôte qui contient la requête HTTP avec la partie *chemin* le l'URL. Le nom de l'hôte ne figure pas nécessairement dans la requête.

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web

Dynamique

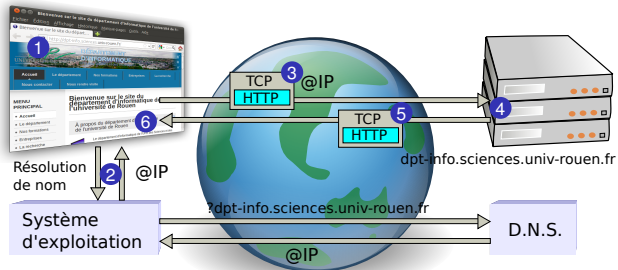
C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Analyse et exploitation d'une URL



- 4 Le service HTTP sur le serveur reçoit la requête, vérifie éventuellement les droits d'accès à la ressource, puis charge la ressource ou demande à un module de lui en fournir une représentation (calcul).

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

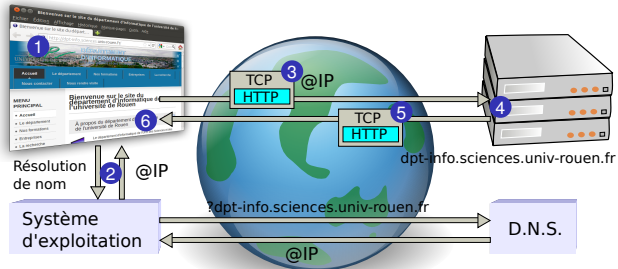
C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Analyse et exploitation d'une URL



- ⑤ Le service HTTP forge ensuite une réponse HTTP contenant le résultat de la requête ainsi qu'un code d'exécution qu'il envoie au client via la connexion TCP toujours ouverte.

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web

Dynamique

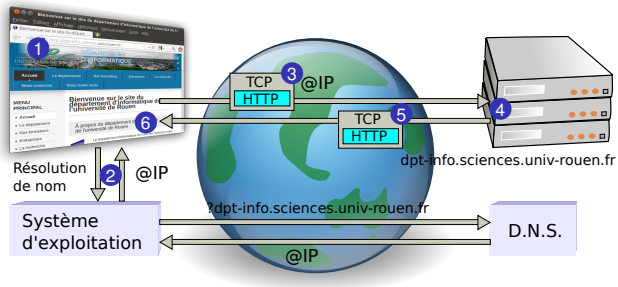
C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Analyse et exploitation d'une URL



- ⑥ Le client HTTP essaie de traiter le résultat de la réponse (affichage de la ressource, sauvegarde, ouverture avec une application tierce, ...) ou affiche un message en fonction du code d'exécution.
- ⑦ Le serveur ferme la connection en fonction de la version du protocole.

Définitions : HTTP

- **HTTP** (pour **H**yper**T**ext **T**ransfer **P**rotocol) est le protocole de transport réseau pour le web,
- son principe est simple : associer une *ressource* à une *URL* et permettre de lui appliquer un certain nombre d'opérations (lecture, enregistrement, suppression, ...),
- bien qu'orienté texte, HTTP transporte aussi bien des données binaires (pour preuve toutes les ressources associées à une page web : jpg, png, video, etc.).
- L'ensemble du système met en oeuvre un certain nombre de mémoires caches :
 - DNS : le navigateur, le système d'exploitation du client, le serveur DNS du domaine du client,...
 - données : le navigateur, le serveur mandataire (Proxy), etc ...

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Le protocole HTTP

Historique

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy



Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

Le protocole a eu très peu d'évolutions :

- La proposition de Tim Berners-Lee en 1989, avant sa standardisation est estampillée 0.9,
- En 1996, **HTTP 1.0** est standardisé par l'**IETF**⁴ (description dans la *RFC*⁵ 1945  ,
- En 1997, la version 1.1 est décrite dans la *RFC* 2068, puis en 1999 dans la *RFC* 2616  .

Dès sa standardisation, HTTP fournit toutes les opérations nécessaires pour l'implémentation des services web dits « Restful⁶ ».

-
4. Internet Engineering Task Force
 5. Request For Comments
 6. REpresentational State Transfer

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

Définitions : HTTP

Modèle OSI ^a

- HTTP est un protocole de la couche applicative (couche 7 du modèle OSI) reposant sur TCP,
- il utilise le port 80 (par défaut).

7	Application
6	<i>Présentation</i>
5	<i>Session</i>
4	Transport
3	Réseau
2	Liaison de données
1	Physique

a. Open System Interconnexion

Définitions : HTTP

Modèle OSI ^a

- HTTP est un protocole de la couche applicative (couche 7 du modèle OSI) reposant sur TCP,
- il utilise le port 80 (par défaut).

7	Application
6	<i>Présentation</i>
5	<i>Session</i>
4	Transport
3	Réseau
2	Liaison de données
1	Physique

a. Open System Interconnexion

Quelles sont les couches non implémentées ?

Définitions : HTTP

Modèle OSI ^a

7	Application
6	<i>Présentation</i>
5	<i>Session</i>
4	Transport
3	Réseau
2	Liaison de données
1	Physique

a. Open System Interconnexion

- HTTP est un protocole de la couche applicative (couche 7 du modèle OSI) reposant sur TCP,
- il utilise le port 80 (par défaut).

- pas de présentation : HTTP n'altère pas les données transportées,
- pas de session : HTTP est sans état et ne conserve aucune information entre chaque requête.

Structure du protocole

Requête :

Methode PATH protocol_version\r\n
Request_Header\r\n
\r\n
Optional_request_body

Réponse :

Version Status_Code Status_Text\r\n
Response_Header\r\n
\r\n
Ressource_Data

- Synchrones : 1 requête, 1 réponse sur la même connexion,
- Requêtes et réponse sont structurées en texte (ascii 7 bits)⁷,
- avec une structure symétrique

7. Ici \r représente le caractère *retour charriot* et \n le *saut de ligne*.

Syntaxe :

Methode PATH\r\n

- La commande **GET** est la seule méthode (commande) connue,
- la requête est terminée par le *retour à la ligne* de type <CR><LF> (\r\n), le serveur répond aussitôt.

Exemple :

GET /edt/mlgil.html\r\n

Syntaxe :

```
Methode PATH protocol_version\r\n
Request_Header\r\n
\r\n
Optional_request_body\r\n
```

- Pour assurer la compatibilité, la version du protocole est ajoutée à la fin de la commande,
- un en-tête comportant des méta-données sur la requête est ajouté,
- la requête est terminée par une ligne vide ou un corps de requête optionnel (voir POST).

Les requêtes

En version 1.0

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

- Trois méthodes sont supportées : **GET**, **HEAD**, et **POST** (voir plus loin).

Exemple :

```
GET /page.html HTTP/1.0\r\n
Host: example.com\r\n
Referer: http://example.com/\r\n
User-Agent:Mozilla/5.0 (X11; U; Linux i686;
fr; rv:1.9b5) Gecko/2008041514
Firefox/3.0b5\r\n
\r\n
```

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

Les champs de l'en-tête :

- **Authorization** : fournit un crédentiel permettant au client de s'authentifier ;
- **From** : fournit au serveur l'adresse email de contact de l'auteur de la requête ;
- **If-Modified-Since** : demande un ressource au serveur seulement si celle-ci a été modifiée après la date donnée (requête conditionnelle) ; exemple :

If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

Les champs de l'en-tête (suite) :

- **Host** : optionnel, il précise la partie nom de domaine (FQDN) de l'URL de la ressource demandée ; Permet d'avoir des hôtes virtuels partageant la même adresse IP ; Exemple :

Host: www.example.com

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

Les champs de l'en-tête (suite) :

- **Referer** : précise l'URL de la ressource ayant produit l'URL de la requête ; Exemple :

Referer: `http://www.exemple.org/accueil.html`

- **User-Agent** : permet d'identifier la nature du logiciel client (nom, version, système d'exploitation, plateforme, etc.) ; Exemple :

User-Agent:`Mozilla/5.0 (X11; U; Linux i686;
fr; rv:1.9b5) Gecko/2008041514
Firefox/3.0b5`

Les requêtes

En version 1.1

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

- La syntaxe reste la même en **HTTP 1.1** qu'en version 1.0.
- Le champs *Host* devient cependant obligatoire.
- Le serveur ne ferme plus systématiquement la connexion après traitement de la requête : les *connexions persistantes* permettent au client d'effectuer plusieurs requêtes via une seule connexion avec le serveur.

Les requêtes

En version 1.1

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

Nouveaux champs d'en-tête :

- **Connection** : indique si la connexion est persistante (**keep-alive** ou doit être fermée après la requête (**close**) ;
- **Accept** : types MIME acceptés par le client (voir plus loin) ;
- **Accept-Charset** : encodages de caractères acceptés.
- **Accept-Language** : les langues acceptées (concerne plutôt l'utilisateur),
- etc.

Méthodes de requête

- **GET** : demande une représentation de la ressource dont l'url est donnée. Ne modifie pas la ressource,
- **HEAD** : demande les métadonnées associées à la ressource dont l'URL est donnée,
- **OPTIONS** : permet d'obtenir la liste des options du serveur ou d'une ressource en particulier,
- **POST** : transmet des données à un processus associé à l'URL passée, ex : `/cgi-bin/subscribe.pl`. Les données sont transmises pour traitement, la ressource cible associée n'est pas modifiée.
- **PUT** : ajoute/remplace une ressource sur le serveur qui aura/a l'URL spécifiée ;
- **DELETE** : supprime la ressource sur le serveur qui a l'URL spécifiée ;

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Méthodes de requête

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

- **CONNECT** : réservé (usage futur) pour créer un tunnel avec un proxy,
- **TRACE** : demande au serveur de renvoyer ce qu'il a reçu (diagnostique) ;

Syntaxe d'une réponse HTTP

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

- En version *0.9*, la réponse n'est pas structurée et consiste uniquement en la représentation de la ressource (quid des ressources introuvables ?).
- À partir de *HTTP/1.0*, les réponses ont cette structure :

```
Version Status_Code Status_Text\r\n  
Response_Header\r\n  
\r\n  
Ressource_Data
```

Syntaxe d'une réponse HTTP

Exemple

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

HTTP/1.1 200 OK

Date: Wed, 14 Sep 2011 14:52:06 GMT

Server: Apache/2.2.3 (CentOS)

X-Powered-By: PHP/5.2.8

Set-Cookie: dd34e0d66751=d5j2l65rbv5; path=/

P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP"

Expires: Mon, 1 Jan 2001 00:00:00 GMT

Last-Modified: Wed, 14 Sep 2011 14:52:06 GMT

Cache-Control: no-store, no-cache, must-revalidate,
post-check=0, pre-check=0

Pragma: no-cache

Connection: close

Transfer-Encoding: chunked

Content-Type: text/html; charset=utf-8

<!DOCTYPE html PUBLIC ...

HTTP : les codes de retour

39 codes d'exécution répartis en 4 groupes :

- Groupe *Successful 2xx* (200 à 206) :
200 OK : requête satisfaite, contenu transmis dans le corps
204 No Content : requête satisfaite, aucun contenu transmis dans le corps
- Groupe *Redirection 3xx* (300 à 307) :
301 Moved Permanently : la ressource demandée a changé de façon permanente
304 Not Modified : (requête conditionnelle) la ressource n'a pas été modifiée
305 Use Proxy : demande effectuée impérativement au travers d'un proxy

HTTP : les codes de retour

- Groupe *Client Error* 4xx (400 à 417) :
 - 400 Bad Request** : requête non comprise par le serveur
 - 403 Forbidden**
modalités d'accès à la ressource ne permettant pas l'accès
 - 404 Not found**
ressource non trouvée
 - 408 Request Timeout**
le client n'a pas produit de requête dans le temps imparti
- Groupe des *Server Error* 5xx (500 à 505) :
 - 503 Service Unavailable**
le serveur est en maintenance ou surchargé
 - 505 HTTP Version Not Supported**
le serveur n'accepte pas la version HTTP demandée

En-tête d'une réponse HTTP

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

- L'en-tête d'une réponse HTTP donne des informations sur l'exécution de la requête,
- mais également des méta-informations sur la ressource concernée,
- des méta-informations sur le serveur lui-même, voire le système.
- On distingue plusieurs groupes d'attributs : généraux, de réponse, d'entité.

En-tête d'une réponse HTTP

Attributs généraux

- **Cache-Control** : indique comment la ressource doit être cachée (*no-cache, no-store, max-age, ...*) ;
- **Connection** : indique que la connection sera fermée après traitement ;
- **Date** : indique la date de traitement ; ex :
Date: Tue, 15 Nov 1994 08:12:31 GMT
- **Pragma** : directives d'implémentation optionnelle ;
- **Transfer-Encoding** : *chunked* indique un transfert par morceaux ; Utile sur les connexions persistantes pour les ressources volumineuses et les ressources dont la taille est indéterminée au moment du transfert ;
- ...

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

En-tête d'une réponse HTTP

Attributs de réponse

- **Accept-Ranges** : indique l'unité acceptée pour les transferts par portion ; ex : `Accept-Ranges: bytes`
- **Location** : utilisé pour rediriger le destinataire à une autre adresse ; ex :
`Location: http://www.w3.org/pub/WWW/People.html`
- **Retry-After** : accompagne une erreur *503 (Service Unavailable)* et indique une date une durée (en secondes) après laquelle une nouvelle tentative peut avoir lieu.
- **Server** : fournit⁸ une chaîne de caractère décrivant le serveur, voire le système ; ex :
`Server: Apache/2.2.3 (CentOS)`
- ...

8. Révèle également des vulnérabilités potentielles.

En-tête d'une réponse HTTP

Attributs de d'entité (ressource)

- **Allow** : indique les méthodes applicable à la ressource : *GET, HEAD, PUT* ;
- **Content-Encoding** : indique la méthode de décodage de la ressource ; ex : `Content-Encoding: gzip`
- **Content-Language** : la(les) langue(s) dans laquelle la ressource est exprimée ; ex :
`Content-Language: fr, mi, en`
- **Content-Length** : indique la taille de la ressource ;
- **Content-MD5** : fournit un hachage MD5 de la ressource ;
- **Content-Range** : indique la portion de la ressource qui a été transmise ; ex :
`Content-Range: bytes 500-999/1234`

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

En-tête d'une réponse HTTP

Attributs de d'entité (ressource) (suite)

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web

Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

- **Content-Type** : indique le type de la ressource (voir type MIME) ainsi que son encodage (jeu de caractères) ; ex :

`Content-Type: text/html; charset=ISO-8859-4`

- **Expires** : indique la date limite de validité de la représentation de l'entité ; ex :

`Expires: Thu, 01 Dec 1994 16:00:00 GMT`

- **Last-Modified** : indique la date de dernière modification ; ex :

`Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT`

- ...

- Les attributs d'entête peuvent être étendus.

Typage des ressources

MIME (Multipurpose Internet Mail Extensions)

Standard qui a été proposé par les laboratoires Bell Communications en 1991 afin de supporter, pour les pièces jointes des e-mails, d'autres jeux de caractères ainsi que des données binaires.

- Objectif : caractériser le contenu d'une ressource sans dépendre d'une extension dans son nom ;
- Un type MIME consiste en un type et un sous-type :
`Content-type: type_principal/sous_type`
- Par exemple pour une image PNG :
`Content-type: image/png`
- Pour un document LaTeX :
`Content-Type: text/x-latex; charset=utf-8`

RFC 2046 

Typage des ressources

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

- Le navigateur (user-agent) utilise le type MIME pour déterminer le traitement à appliquer aux données reçues : affichage (html, txt, images), sauvegarde ou ouverture avec une application tierce (ex : mp3, documents libre-office, etc).
- En général, le type MIME est déterminé par le serveur HTTP ; (Sous Linux : `file -i filename`);
- ou spécifié : balises **<!DOCTYPE>**


Encodages

Depuis quelque temps, les accents grognaient. Ils se sentaient mal aimÃ©s, dÃ©daignÃ©s, mÃ©prisÃ©s. A l'Ã©cole, les enfants ne les utilisaient presque plus. Chaque fois que je croisais un accent dans la rue, un aigu, un grave, un circonflexe, il me menaÃ§ait.

- Notre patience a des limites, grondait-il. Un jour, nous ferons la grÃ¢ve. Attention, notre nature n'est pas si douce qu'il y paraÃ®t. Nous pouvons causer de grands dÃ©sordres.

Je ne prenais pas les accents au sÃ©rieux. J'avais tort.

9

- La table ASCII ne satisfait pas tout le monde,
- d'autres alphabets existent dans le monde → d'autres pages de caractères
- i.e. un autre mapping entiers → lettre (pictogramme)
- pour afficher correctement un texte, le navigateur a besoin de connaître le jeu de caractères utilisé.
-  Un serveur statique ne détecte pas l'encodage. Il communique celui spécifié par sa configuration ou éventuellement fourni par le programme qui traite la requête.

Différents types de serveurs

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

- HTTP fonctionne un modèle client-serveur, mais la connexion n'est pas toujours point-à-point ;
- En effet, un serveur HTTP peut remplir différents rôles :
 - **serveur d'origine** : le serveur qui possède la ressource ;
 - **serveur mandataire (Proxy)** : serveur intermédiaire à la fois client et serveur ;
 - **proxy inverse (reverse proxy) ou passerelle** : agit comme un serveur d'origine en déléguant.
- Les requêtes conditionnelles constituent le mécanisme de base des serveurs mandataires.

Requête conditionnelle : If-modified-since

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

- Requête conditionnelle du client (date de modification)

```
GET / HTTP/1.1
Host: localhost
If-modified-since: Sat, 29 Oct 1994 19:43:31 GMT
```

- Réponse du serveur : transmission de la ressource

```
HTTP/1.1 200 OK
Date: Thu, 28 Sep 2006 13:13:54 GMT
Server: Apache/2.2.2 (Fedora)
Last-Modified: Sat, 23 Sep 2006 20:58:58 GMT
ETag: "1028b-77-39baa880"
Accept-Ranges: bytes
Content-Length: 119
Connection: close
Content-Type: text/html; charset=UTF-8
```

```
<html>
...
</html>
Connection closed by foreign host.
```

Comparaison de **Last-Modified** et de **If-modified-Since**
code de retour **200 OK** suivi de la **transmission de la ressource**

Requête conditionnelle : If-modified-since

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

- Requête conditionnelle du client (date de modification)

```
GET / HTTP/1.1
```

```
Host: localhost
```

```
If-modified-since: Sat, 23 Sep 2006 20:59:46 GMT
```

- Réponse du serveur : **NON**-transmission de la ressource

```
HTTP/1.1 304 Not Modified
```

```
Date: Thu, 28 Sep 2006 13:14:36 GMT
```

```
Server: Apache/2.2.2 (Fedora)
```

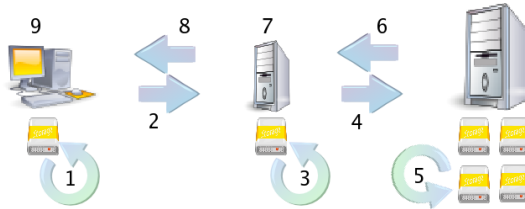
```
Connection: close
```

```
ETag: "1028b-77-39baa880"
```

```
Connection closed by foreign host.
```

Comparaison de **Last-Modified** et de **If-modified-Since**
code de retour **304 Not Modified** **sans** transmission de
ressource...

Proxy : principe d'une requête



- 1 interrogation des caches mémoire et disque
- 2 requête au proxy
- 3 recherche dans le (cache) disque du proxy
- 4 requête conditionnelle au serveur
- 5 localisation de la ressource sur les disques du serveur
- 6 réponse du serveur au proxy
- 7 stockage sur le disque du proxy
- 8 réponse du proxy au client
- 9 stockage sur le disque du client, en cache mémoire et affichage

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole

HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures

HTTP

Proxy

Reverse Proxy

Web

Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

Rappels sur les serveurs de proxy

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Pourquoi utiliser un serveur de proxy ?

- permet un accès aux ressources plus rapide ;
- économise de la bande passante (requêtes multiples + filtre bannières pub) ;
- permet de filtrer l'accès à certains sites ;
- permet accès aux ressources d'un serveur indisponible.

Caractéristiques techniques d'un serveur de proxy

- peu de sollicitations CPU ;
- relativement peu de mémoire ;
- beaucoup d'espace disque (nbre de clients + activité, entre plusieurs centaines de Go et quelques To) ;

Rôle du relai de proxy

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

Cheminement général (client-proxy-serveur)

- le client émet une requête ;
- le proxy (après éventuelle authentification) :
 - ➊ prend en compte la requête.
Étude basée sur la source (port et machine d'origine) et sur la destination (port et machine de destination), et éventuellement sur l'identification ou l'authentification du client ;
 - ➋ consulte sa base de règles pour prendre une décision :
 - accepte : relai de la requête vers le serveur ;
 - redirige : relai de la requête vers une autre machine ;
 - refuse : notifie au client le refus.
 - ➌ consigne sa décision dans un fichier de log ;

Utilisation du cache et Time-To-Live des ressources

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

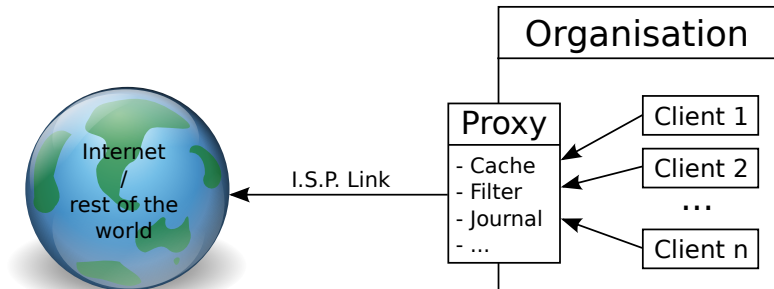
Sécurité

HTTPS

Fonctionnement simplifié

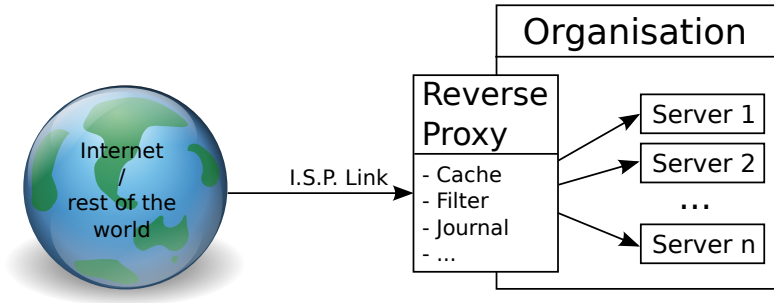
- 1 cherche dans le cache une copie de la ressource (table de hashage) ;
- 2 si la ressource n'est pas présente dans le cache :
 - récupération de la ressource auprès du serveur ;
 - détermination du TTL (*Time To Live*) ;
 - mise en cache avec indication du TTL pour requêtes ultérieures.
- 3 si la ressource est présente dans le cache :
 - si TTL non dépassé, fournir la ressource au client ;
 - si TTL dépassé, req. cond. (**If-modified-since**) au serveur. Suivant code de retour (304 ou 50x), fournir la copie cache.

Architecture de Proxy



- Isole les clients de l'organisation du reste du monde.
- Applique aux connections sortantes une politique de mise en cache, de filtrage, de journalisation, etc ...

Proxy inverse – passerelle



- Même rôle, mais en sens inverse,
- Rôles rempli vis-à-vis des serveurs de l'organisation.

Proxy inverse – passerelle

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

- Il contrôle les accès entrants de l'organisation.
- i.e. il est serveur mandataire pour le reste du monde.
- Il permet de
 - mettre en cache les ressources demandées,
 - appliquer une politique d'accès sur un ensemble de ressources sans modifier les serveurs web d'origine
 - évite d'exposer directement les serveurs d'origine (machines) sur Internet.

Comportement dynamique

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique


C.G.I.
Fonctionnement avec
états

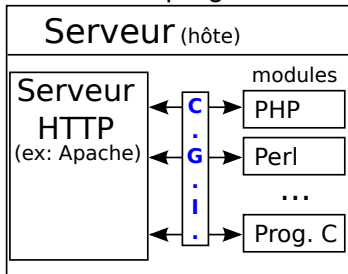
Sécurité

HTTPS

- Un site Web dynamique est un site dont certaines pages ont leur contenu généré dynamiquement (lors du traitement de la requête) ;
- \neq page dynamique ;
- \neq interactif ;
- Pour être dynamique, le serveur Web fait appel à un programme externe (ou un de ses modules) ;
- pour être interactif, ce programme doit être en mesure de recevoir des données en provenance du client, transmises dans la requête : l'interface **C.G.I.** ;
- pour transmettre des données, à partir d'une page web, le client utilise la structure de **formulaire**.

L'interface CGI

- **Common Gateway Interface (CGI)** est une interface de programmation standardisée (RFC 3875 ) située entre le serveur HTTP et le programme de traitement.



- En effet, le protocole HTTP suffit pour assurer l'envoi de données depuis le client vers le serveur (méthodes *GET* et *POST*).
- L'interface¹⁰ CGI permet au programme de recevoir ces données et de renvoyer sa réponse.

10. Notez qu'il s'agit d'un pléonasme !

La transmission des données

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

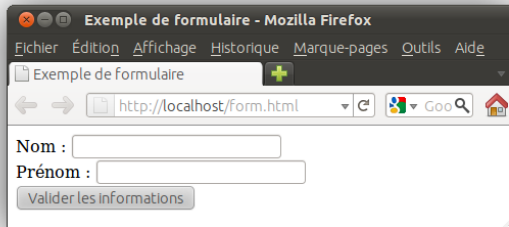
Sécurité

HTTPS

- Les données sont transmises à destination d'une ressource sur le serveur derrière laquelle se cache en fait un programme ; ex :
`http://example.com/subscribe.php`
- Les données sont transmises sous forme de couples `nom=valeur` ; ex :
`nom=jean-pierre`
- Le nom provient de l'attribut `name` de la balise dans le formulaire, la valeur est celle saisie par l'utilisateur dans ce champs.

La transmission des données

```
1 <html>
2 <head>
3 <title>Exemple de formulaire</title>
4 </head>
5 <body>
6 <form method="GET" action="cgi-bin/recup.pl" />
7   Nom :
8   <input type="text" name="nom" /><br/>
9   Prénom : <input type="text" name="prenom" /><br/>
10  <input type="submit" value="Valider les informations" />
11 </form>
12 </body>
13 </html>
```



Réception des données

Un exemple en Perl

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

```
1  #!/usr/bin/perl -wT
2  use strict;
3  use CGI;
4  my $q = new CGI;
5  my $nom = $q->param( "nom" );
6  my $prenom = $q->param( "prenom" );
7  print $q->header( "text/html" ),
8  $q->start_html( "Action_cgi-bin_recup.pl" ),
9  $q->p( "$prenom_$nom" ),
10 $q->end_html ;
```

- Un premier constat : le programme doit être en adéquation avec le formulaire !

Réception des données

Un exemple en Perl

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

```
1  #!/usr/bin/perl -wT
2  use strict;
3  use CGI;
4  my $q = new CGI;
5  my $nom = $q->param( "nom" );
6  my $prenom = $q->param( "prenom" );
7  print $q->header( "text/html" ),
8  $q->start_html( "Action_cgi-bin_recup.pl" ),
9  $q->p( "$prenom_$nom" ),
10 $q->end_html ;
```

- Un premier constat : le programme doit être en adéquation avec le formulaire !

Transfert des données

La méthode GET

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

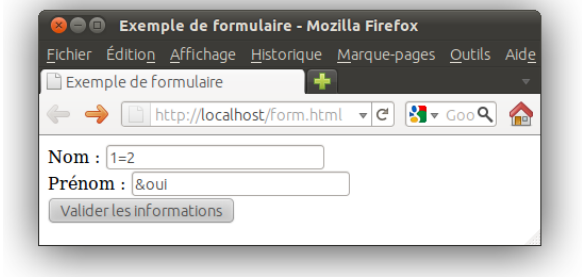
- La méthode GET classique du protocole HTTP est invoquée ;
- les données transmises sont concaténées à l'URL ;
- le symbole **?** sert à délimiter le nom de la ressource des données, les couples sont séparés par le symbole **&** ; ex :

`http://localhost/recup.pl?nom=marcel&prenom=proust`

Transfert des données

La méthode GET

- Les données sont protégées (ou échappées, *escaped* en anglais) pour éviter toute interférence avec la syntaxe des URLs ; par exemple, la saisie suivante :



donnera

`http://localhost/recup.pl?nom=1%3D2&prenom=%26oui`
au lieu du non sens :

`http://localhost/recup.pl?nom=1=2&prenom=&oui`

Transfert des données

La méthode POST

- La méthode GET expose les données dans l'URL.
- Problème de confidentialité (l'URL n'est pas cryptée).
- La taille est limitée (pas d'envoi de fichier).
- La méthode **POST** place les données dans le corps de la requête.
- Le formulaire précise la méthode à utiliser :

```
1 <html>
2 <head>
3 <title>Exemple de formulaire</title>
4 </head>
5 <body>
6 <form method="POST" action="http://localhost/cgi-bin/recup.pl"/>
7 Nom :
8 <input type="text" name="nom"/><br/>
9 Prenom : <input type="text" name="prenom"/><br/>
10 <input type="submit" value="Valider les informations"/>
11 </form>
12 </body>
13 </html>
```

Transfert des données

La méthode POST

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

La requête ressemble alors à :

```
POST /cgi-bin/recup.pl HTTP/1.1\n\r
Host: 127.0.0.1\n\r
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:6.0.2)
Gecko/20100101 Firefox/6.0.2\n\r
Accept: text/html,application/xhtml+xml,
application/xml;q=0.9,*/*;q=0.8\n\r
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3\n\r
Accept-Encoding: gzip, deflate\n\r
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\n\r
Connection: keep-alive\n\r
Content-Type: application/x-www-form-urlencoded\n\r
Content-Length: 23\n\r
\n\r
nom=1%3D2&prenom=%26oui
```

Fonctionnement avec états

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web

Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

- Le protocole HTTP ne maintient aucun état entre chaque requête.
- Les données transmises et le résultat de leur traitement sont « oubliés » à la fin de chaque requête.
- Pour développer des applications web, il est nécessaire de pouvoir maintenir un état entre chaque action de l'utilisateur (Panier, webmail, espace de travail numérique, etc.).
- Nous avons toutefois la possibilité de créer des variables persistantes, soit coté client, soit coté serveur

...

Fonctionnement avec états

Coté client

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

- Un serveur HTTP peut demander au client de sauvegarder de manière persistante un couple *nom=valeur* sur le poste de l'utilisateur.
- On appelle ses variables des **cookies** (*témoins de connexion* en bon français).
- Chaque cookie est associé à l'URL de la ressource qui a provoqué sa création.
- À chaque requête sur cette ressource, le navigateur transmet **systématiquement** les cookies associés dans l'en-tête !

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

```
netcat www.google.fr 80
```

```
GET / HTTP/1.1
```

```
Host: www.google.fr
```

```
HTTP/1.1 200 OK
```

```
Date: Thu, 15 Sep 2011 18:44:07 GMT
```

```
Expires: -1
```

```
Cache-Control: private, max-age=0
```

```
Content-Type: text/html; charset=ISO-8859-1
```

```
Set-Cookie: PREF=ID=3f95fe59802e375d:FF=0:TM=1316112247:  
LM=1316112247:S=Iys8lu6VpxlWrq8b; expires=Sat, 14-Sep-2013  
18:44:07 GMT; path=/; domain=.google.fr
```

```
Set-Cookie: NID=51=c0nKZX85_alnAdb4CXIwkIxaJNzuROLtr7pKKnR-  
hDQAAYklP7Vya5rPHMMxtLsOYK3WptMWLHsZ4wngeeEMR7tvJNx-  
esEltLcnopcaYfm1CXJlQKrBgyGHIKctEuF; expires=Fri, 16-Mar-2012  
18:44:07 GMT; path=/; domain=.google.fr; HttpOnly
```

```
Server: gws
```

```
X-XSS-Protection: 1; mode=block
```

```
Transfer-Encoding: chunked
```

```
...
```

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

Requête avec un navigateur :

```
GET / HTTP/1.1
Host: www.google.fr
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:6.0.2)
Gecko/20100101 Firefox/6.0.2
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Cookie: PREF=ID=PREF=ID=3f95fe59802e375d:FF=0:TM=1316112247:
LM=1316112247:S=Iys8lu6VpxlWrq8b;NID=51=c0nKZX85_alnAdb4CXI
wkIxaJNzuROLtr7pKKnR-hDQAAyKlP7Vya5rPHMMxtLsOYK3WptMWLHsZ4w
ngeeEMR7tvJNx-esElLcnopcaYfm1CXJlQKrBgyGHIXKctEuF
```


Cookies

Attributs des cookies

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

```
Set-Cookie: NID=51=c0nKZX85_alnAdb4CXIwkIxaJNzuROLtr7pKKnR-  
hDQAAYklP7Vya5rPHMMxtLsOYK3WptMWLHsZ4wngeeEMR7tvJNx-  
esEltLcnopcaYfmlCXJlQKrBgyGHIXKctEuF; expires=Fri, 16-Mar-2012  
18:44:07 GMT; path=/; domain=.google.fr; HttpOnly
```

- **domain** et **path** : limite l'accès au cookie au domaine et/ou chemin,
- **Expires** et **Max-Age** : fixe une date d'expiration/durée de validité en secondes,
- **Secure** : restreint l'envoi du cookie aux connections cryptées (https),
- **HttpOnly** : exclut tout autre méthode d'accès que HTTP, par exemple les javascripts `document.cookie`.

Fonctionnement avec états

Coté serveur

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec
états

Sécurité

HTTPS

- Il peut être souhaitable que les données ne soient pas conservées du coté du client.
- les **sessions** permettent d'associer à un client donné, un ensemble de variables coté serveur.
- C'est le programme coté serveur qui manipule les variables, le client n'en a pas connaissance et ne peut y accéder directement.
- Pour pouvoir maintenir le lien entre un utilisateur et ses variables, on utilise en général ... des **cookies** !

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

Exemple de cookie de session pour un programme en Java :

HTTP/1.1 200 OK

Date: Thu, 15 Sep 2011 20:33:35 GMT

Set-Cookie: JSESSIONID=1D8280D6F79DC463C4129A35E0ED2C99; Path=/
Content-Type: text/html; charset=ISO-8859-1

Transfer-Encoding: chunked

...

- La gestion des variables sur le serveur dépend du langage de programmation (*Java, PHP, etc*),
- nous y reviendrons plus tard.

Éléments de sécurité

... d'abord

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

- La sécurité du web est un enjeu majeur aujourd'hui !
- De nombreuses applications, autrefois sur le bureau, sont en ligne (bureautique, sauvegarde, etc.)
- ainsi que de nombreux services quotidiens : banque, achats par correspondance, etc.
- La cryptographie est une des pierres angulaires de la sécurité sur les réseaux,
- une autre étant une programmation propre et rigoureuse.

Éléments de sécurité

Chiffrement

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

Deux grandes méthodes de chiffrement (à clé) :

- cryptographie symétrique (à clé secrète) : une seule clé pour chiffrer et déchiffrer ;
- cryptographie asymétrique (à clé publique et privée) : une clé pour chiffrer et une clé pour déchiffrer ;

Deux usage pour l'asymétrie :

- clé de chiffrement publique : protection des données ;
- clé de déchiffrement publique : authentification, certificats ;

Éléments de sécurité

Chiffrement

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

Deux grandes méthodes de chiffrement (à clé) :

- cryptographie symétrique (à clé secrète) : une seule clé pour chiffrer et déchiffrer ;
- cryptographie asymétrique (à clé publique et privée) : une clé pour chiffrer et une clé pour déchiffrer ;

Deux usage pour l'asymétrie :

- clé de chiffrement publique : protection des données ;
- clé de déchiffrement publique : authentification, certificats ;

Éléments de sécurité

HTTPS

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

- **HTTPS** (HTTP Secured) fait fonctionner le protocole HTTP classique sur une connexion **TLS**
- *TLS* authentifie le serveur et chiffre les communications
- HTTPS utilise le port TCP 43 (par défaut).
- Réduit les risques d'attaque du type *man-in-the-middle*.

Il est possible d'alterner HTTP et HTTPS sur un même site.
HTTPS doit être utilisé pour toute donnée sensible :

- identifiant et mot de passe,
- cookie (de session),
- etc.

Éléments de sécurité

Contrôle d'accès

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec états

Sécurité

HTTPS

Le contrôle d'accès à un site peut se faire :

- par l'application : formulaire d'authentification, gestion de la session, etc. ; Requier une implémentation ou l'utilisation d'un module existant ; Plus flexible (ex : captcha¹¹, cas¹², annuaire LDAP, etc.) ;
- par le protocole HTTP : le fichier .htaccess ; Ne nécessite pas d'implémentation, orienté utilisateur.

11. Completely Automated Public Turing test to tell Computers and Humans Apart.

12. Central Authentication Service

Éléments de sécurité

Contrôle d'accès – fichier .htaccess

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

```
# Procédé de protection (Apache 1.2 ne connaît que "basic")
authType      basic

# Nom de la section autorisée
authName      Section_WEB_du_bureau_des_comptes

# Fichier pour description utilisateur (ne pas placer parmi les documents!)
authUserFile  /home/ms/acceshttp/utilisateurs.txt

# Fichier pour description groupes (ne pas placer parmi les documents!)
authGroupFile /home/ms/acceshttp/groupes.txt

# Dans ce cas particulier, les autorisations (allow) prévalent
# sur les interdictions (deny)...
# (order allow,deny provoquerait le contraire)
order         deny,allow

# ... mais d'abord, on interdit tout
deny from     all

# seules l'adresse de l'ordinateur ET la reconnaissance de
# l'utilisateur autorisent l'accès!
satisfy       all

# seuls les ordinateurs de ce groupe d'adresses IP ont droit d'accès
allow from    153.46.90.

# seuls le "chef" et sa secrétaire du groupe "developp" ont droit d'accès
require group developp
require user  chef secretaire
```

Éléments de sécurité

Contrôle d'accès – fichier .htaccess

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

- Lors de l'accès à une ressource, le serveur vérifie l'existence de ce fichier dans le répertoire contenant la ressource et ses parents (contrôle toute la sous-arborescence) ;
- Le serveur renvoie une erreur *401 Unauthorized* ainsi qu'un *défit* dans l'en-tête de la réponse ;
- Le client ouvre une boîte de dialogue demandant identifiant et mot de passe à l'utilisateur ;
- Le client calcul un hachage du couple (id,mdp) en fonction du déficit (ex : MD5) et le met dans l'en-tête de la requête qu'il réitère ; ex :
`Authorization: Basic OmhhcmxvZ6s=`

Éléments de sécurité

Contrôle d'accès – fichier .htaccess

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

- Le serveur compare le hachage avec celui qu'il a calculé de son côté à partir du couple (id,mdp) qu'il connaît ;
- S'ils sont identiques, la requête est traitée, sinon, l'erreur *401* est à nouveau déclenchée ;
- Pour toute les requêtes suivantes, le client enverra le hachage dans l'en-tête !

Conséquences :

- Le mot de passe ne circule pas en clair sur le réseau (sur les implémentations récentes) ; **MAIS**
- Le token d'identification qui en résulte **OUI** (sauf sur HTTPS bien sûr).

Éléments de sécurité

Indexation

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

- Pour pouvoir fonctionner, les moteurs de recherche ont besoin d'être alimentés.
- Pour cela, des programmes autonomes parcourent le web et indexent ses données. Ces **robots** tendent à indexer tout ce qui est accessible même sans démarche de référencement : suivi de liens, index de répertoires, etc.
- La mise en ligne, puis l'indexation, d'information est souvent un processus irréversible.
- Un fichier *robots.txt* placé à la racine du site permet de contrôler l'indexation par les robots

Éléments de sécurité

Indexation - fichier robots.txt

Ce fichier lève des interdictions.

Syntaxe :

User-agent: expression

Disallow: path

expression liste les clients concernés (* = tous)

path chemin d'un sous-arbre à exclure de l'indexation

Exemple :

User-agent: *

Disallow: /forum

User-agent: googlebot

Disallow:

Exclut tous les robots du sous répertoire *forum*.

Autorise googlebot.

Éléments de sécurité

Indexation

- Les robots suivent les instructions du fichier robot.txt sur la base du volontariat !
- Si c'est accessible, alors c'est indexable.
- L'absence de lien interne vers une ressource ne protège pas celle-ci contre l'indexation. (Ex : lien externe depuis un forum).
- La protection minimale contre l'indexation est le contrôle d'accès, par application (une URL externe pouvant contenir un login et mot de passe HTTP).
- Une meilleure protection consiste en un proxy de contrôle, voire un tunnel d'accès.
- La meilleure protection reste la mise hors ligne.

Eric Schmidt, PDG de Google Inc.

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Éléments de sécurité

Indexation

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP

Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

« S'il y a des choses que vous souhaitez cacher à tout le monde, vous ne devriez peut-être pas les faire. Si vous avez vraiment besoin de ce niveau de vie privée, la réalité c'est que les outils de recherche Google y compris conservent effectivement ces informations pendant un temps donné et il est important de noter, par exemple, que nous sommes tous soumis au Patriot Act aux Etats-Unis et qu'il est possible que toutes ces informations soient mises à disposition des autorités. »

Eric Schmidt, PDG de Google Inc.

Conclusion

De l'utilité de ce chapitre

Connaître le fonctionnement d'HTTP nous permet de comprendre :

- que HTTP est le lien entre la vue déportée et le reste (le contrôleur) de l'application,
- protocole privilégié pour la communication de machine à machine (les services web),
- \Rightarrow c'est un bus de communication applicatif.
- Son fonctionnement contraint la manière dont les éléments du système interagissent,
- On peut le voir comme un *RPC* des applications web (applications distribuées !).
- Environnement sensible (le web) : sessions, cryptographie, ...

Introduction

Ressources web

Encodage

Les URL

Analyse URL

Le protocole HTTP

Principe

Structure

Les requêtes

Les méthodes

Les réponses

Syntaxe

Codes de retour

En-tête

Types MIME

Encodage

Infrastructures HTTP

Proxy

Reverse Proxy

Web Dynamique

C.G.I.

Fonctionnement avec états

Sécurité

HTTPS

Quelques références

Introduction

Ressources web
Encodage
Les URL
Analyse URL

Le protocole HTTP

Principe
Structure
Les requêtes
Les méthodes
Les réponses
Syntaxe
Codes de retour
En-tête
Types MIME
Encodage

Infrastructures HTTP


Proxy
Reverse Proxy

Web Dynamique

C.G.I.
Fonctionnement avec
états

Sécurité

HTTPS

- HTTP 1.0 : RFC 1945  (en-têtes : RFC 822 )
- HTTP 1.1 : RFC 2616 
- URI : RFC 2396 
- Types MIME : RFC 2046 
- C.G.I. 1.1 : RFC 3875 

(RFCs cliquables)