

Programme officiel

Contenus	Capacités attendues	Commentaires	Durée
Formulaire d'une page Web	Analyser le fonctionnement d'un formulaire simple. Distinguer les transmissions de paramètres par les requêtes POST ou GET.	Discuter les deux types de requêtes selon le type des valeurs à transmettre et/ou leur confidentialité.	

Introduction

Comme vu précédemment, la communication entre client et serveur se fait à travers le protocole HTTP(S) sous forme de requêtes envoyées par le client auxquelles le serveur répond après les avoir traitées.

Afin de permettre à l'utilisateur d'un site web d'envoyer des données au serveur (pour se connecter par exemple), on peut utiliser un formulaire.

1.1 Le protocole HTTP

Le **protocole HTTP** (HyperText Transfert Protocol) a été mis au point en 1990 par Tim Berners-Lee au CERN afin de permettre le partage d'informations entre des physiciens travaillant dans des universités et instituts situés aux quatre coins du globe.

Avant cela, il était possible de correspondre par courrier électronique (mail) mais pas de stocker une information à un endroit donné pour qu'elle soit consultable par tous.

En 1994 naît la version sécurisée du protocole (HTTPS), portée par le navigateur Netscape.

Requêtes HTTP

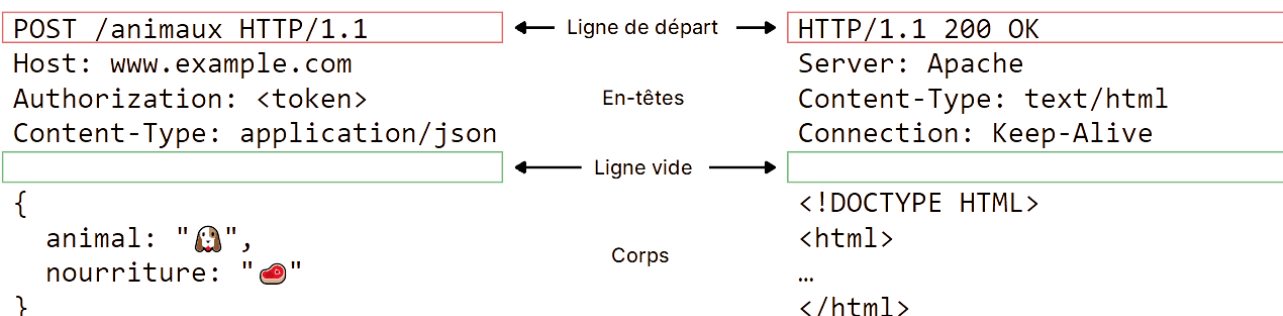
Clients et serveur communiquent à l'aide de requêtes HTTP qui sont envoyées par le client (un navigateur par exemple) et reçues puis interprétées par le serveur web qui y répond.

Requêtes et réponses suivent une structure composée de :

1. Une **première ligne** comportant des détails sur la requête (méthode, chemin, version du protocole) ou sur la réponse (version du protocole, code de réponse)
2. Des **en-têtes** optionnels apportant des détails sur la requête ou le contenu du corps.
3. Une **ligne vide** permettant de séparer les en-têtes du corps de la requête
4. Le **corps** du message pouvant contenir les données associées à une requête (ex : contenu d'un formulaire) ou dans le cas d'une réponse, un fichier HTML ou des données connues par le serveur.

Requête

Réponse



Méthodes HTTP

En fonction du type d'action que l'on souhaite effectuer, on peut spécifier la **méthode** (ou verbe) d'une requête HTTP. Celle-ci permet de communiquer au serveur l'action que l'on souhaite effectuer et celui-ci ne répondra pas de la même façon à deux méthodes différentes.

Il existe de nombreuses méthodes différentes mais les plus courantes sont **GET** et **POST**.

- **GET** permet de demander une ressource ou des données au serveur (ex : un fichier HTML ou une image).
- **POST** permet d'envoyer des données au serveur dans le corps de la requête. Elle provoque généralement un changement au niveau du serveur (ex : enregistrement d'un nouvel utilisateur dans la base de données).

Codes de réponse HTTP

La réponse d'un serveur contient toujours en première ligne le code de réponse, indiquant si la requête a été exécutée avec succès et le cas échéant, apportant des précisions sur l'erreur. Ils correspondent à des nombres allant de 100 à 599.

On peut relever quelques codes de réponse courants :

200 OK : La requête a réussi

404 Not Found : La ressource demandée est introuvable

403 Forbidden : Le client n'a pas le droit d'accéder à la ressource demandée

500 Internal Server Error : Le serveur a rencontré une erreur et ne peut pas compléter la requête

1.2 Les formulaires HTML

Afin de permettre à l'utilisateur d'un site web d'envoyer des données au serveur, on peut utiliser des formulaires.

Ceux-ci sont délimités par des balises **<form>** qui contiennent des balises **<input>** pouvant être de différents types.

Exemple de formulaire

```
<form method="POST">  
  <input type="text">  
  <input type="checkbox">  
  
  <input type="submit">  
</form>
```

Différents types d'inputs

L'apparence et le comportement des balises `<input>` peut être modifié avec l'attribut « **type** ».

Liste complète des différents types : <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>

Voici les principaux avec leur apparence sur Google Chrome :

<code><input type="text" placeholder="Identifiant"></code>	<input type="text" value="Identifiant"/>
<code><input type="submit" value="Envoyer"></code>	<input type="submit" value="Envoyer"/>
<code><input type="checkbox" checked></code>	<input checked="" type="checkbox"/>
<code><input type="checkbox"></code>	<input type="checkbox"/>

Différence entre GET et POST dans un formulaire

Il est possible de spécifier la méthode HTTP utilisée lorsque le formulaire est envoyé grâce à l'attribut « **method** » dans la balise `<form>` (voir l'exemple de formulaire ci-dessus).

La méthode permet de choisir comment les données sont transmises au serveur :

GET : Les données sont visibles à la fin de l'URL et la requête ne contient pas de corps. Cette méthode est adaptée pour une barre de recherche par exemple.

POST : Les données transmises ne sont pas visibles dans l'URL et sont contenues dans le corps de la requête. Cette méthode est la plus adaptée pour l'envoi de données sensibles comme des identifiants de connexion par exemple.

Encodage du contenu

Les données des formulaires sont encodées avant d'être envoyées au serveur, quelle que soit la méthode choisie. L'encodage utilisé est appelé Encodage-Pourcent.

Plusieurs caractères interdits dans les URLs sont remplacés par leur équivalent ASCII précédé d'un « % ».

Les espaces sont, quant à eux, remplacés par des signes « + ».

Nom d'utilisateur

Mot de passe

Requête HTTP :

GET /login?username=John+Smith&password=Canard27 HTTP/1.1