http :hyper text transfert protocol

de faire transiter les données entre le navigateur internet et le serveur se sonnées

https : Hyper Text Protocol SECURED

CR2ER UN PROTOCOLE DE TRANSFERT DE DONN2ES html un protocole de communication hypermédia : d’abord sur les fichiers de type media ,photo vidéos sons puis a été étendue au type fichier exécutable (.exe) et types de fichiers propriétaire avec l’apparition de l’attribut download de la balise<a> en HTML

1 requête http, réponse et sémantique

Méthode de la requête , par exemple :

-requête du téléchargement

-requête du téléversement

-requête récupérer les infos d’une vidéo ,image…

-requête de chargement de données texte

Cible de la requête :

* L’Url ou le chemin absolu qui amène a la ressource(http//www.mon exemple.com ou <http://www.mon> exemple.com/dossier1/fichier)

La version du http utilisé pour la requête qui permet de définir la version de la réponse et d’assurer une comptabilité entre les différents langages ou fichiers utilisés

Suite a la définition de la requête on défini les en-têtes de la requête, modifiables pour s’adapter aux informations nécessaires

Notamment le corps de la requête qui va permettre de stocker les informations POST qui seront envoyés au serveur .

Une fois stockés les données sont encodées en binaire /\*ce qui les rends illisibles pour les personnes non initié de la matrice\*/

2-Méthodologie d’envoi et de réceptions de données

GET : permet de récupère rune information(ressource) sans la modifier

POST :récupère une information dans un but de modification (cryptage, update)

PUT :ajouter ou remplacer une donnée sur le serveur

DELETE : permet de supprimer une donnée du serveur

Pour rappela vérification pour les formulaires ce fait via l’attribut méthode(<form method=’POST’>)

Si l’information est envoyé via la méthode POST en HTML elle doit être récupérer via la méthode $\_POST en PHP et sera transmise au serveur via une méthodologie POST

3- Code de réponse http

Code de réponse 404 fichier ou ressource non trouvée

Code réponse 200 qui indique une exécution parfaite de la requête

Code réponse 2XX : traitement avec succès de la requête http

Code réponse 3xx :requête qui n’a pas le bon chemin(qui doit être redirigée)

Code réponse 4XX : erreur coté client : erreur qui a eu lieu sur le navigateur de l’utilisateur.

Code réponse 5XX : erreur coté serveur :serveur n’as pas réussi a traiter la requête

Les codes réponse vont permettre de créer des condition de fonctionnement via des requêtes asynchrones (AJAX) afin de simplifier la vitesse de chargement de la page ( et ca peut également servir au débogage)

Compréhension et modification des requêtes afin de répondre a des besoins spécifiques

4- fonctionnement d’une session http

Une session http correspond a l’ensemble des requêtes effectuées pour réaliser le chargement du document web

On va diviser la session http en 3 parties :

-la connexion

-l’envoi de la requête

- la réception des données

La connexion se fait via protocole TCP/IP

TCP : Transmission Control Protocol

IP : Internet Protocol

Un protocol : établissement d’un pont entre un point A et un point B symbolisé par une requête

Le Cache : il s’agit des informations et des ressources qui seront stockés( souvent sous formes de cookies) pour pouvoir optimiser la navigation sur le document .les informations stockés dans le cache dès lors disponible hors connexion. Elle est stockée dans la mémoire du navigateur pour une page spécifique

Le plus souvent les informations seront temporaires, elles auront une propriété d’expiration qui sera défini soit par le créateur du site soit par le navigateur

Il existe deux formats de caches le cache privé et le cache partagé

Cache privé : qui va être spécifique a un navigateur et un utilisateur et a un utilisateur de ce navigateur(historique de recherche etc…)

Cache partagé : plus utilisé car stocke les données non personnels qui permettent d’optimiser le chargement quelque soit l’utilisateur

Réponse cachable ou non ?:

* Réponse cachable : le navigateur va extraire et stocker les données pour pouvoir les réutiliser
* Réponse non cachable : le navigateur ne pourra pas récupérer les données
* Réponse classique d’une information cachable est une réponse 200 a une requêtes GET
* IF GET=200 Cache

On va les stocker pour pouvoir créer de conditions de fonctionnement plus solide

* On va pouvoir forcer le stockage de certaines méthodologies du http

Une identification de cache va être définie par son URI (Uniform Ressource Id) qui va être déclarant précis de la ressource stockée

On va définir les informations de fonctionnements de la ressource caché via les en-tetes de la requête http

‘Expires’ : il s’agit d’un en tête HTML qui permet de définir la date d’expiration de la ressource caché

Dans le ca d’un cache partagé on va travailler avec la directive max-age : on l’écrira S MAXAGE ( le s signifiant SHARED)

Apres péremption la requête http va demander a vérifier si la données est toujours ,si

6 les redirections

Les redirections permettent d’informer le navigateur du changement d’URL d’une ressource spécifique, la création d’une redirection permet de définir la nouvelle URL afin de pouvoir faciliter la recherche en SEO( référencement)

Pour le référencement la page n’ a pas été perdue, une redirection lui permet de pouvoir changer une URL périmée par la nouvelle , on utilise un code 301 en réponse pour pouvoir appliquer la redirection dans les informations de la requête http

La commande qui permet de créer des redirections sera ‘redirect’

7. LE COOKIE

Pour pouvoir définir l’utilisation du cookie on va utiliser l’en-tête ‘Set-Cookie’, celui-ci va avoir besoin

Du bi-nome Nom/Valeur qu’il pourra récupérer pour pouvoir l’insérer dans la requête .

Setcookie (Nom , valeur , expiration…) (cf Cookie.php pour les informations des métadonnées)

8. négociation de données

La négociation se réalise a l’aide d’un en -tet ‘Vary’ , celui-ci va nous permettre de définir l’encodage des données et le type de données attendu par le serveur .

L’idée principale est d’aider le service