Lundi 18 novembre 2019

Rappels sur les notions déjà vues

BDD : utilité : stocker des infos de manière structurée et persistante.

API JDBC : Bibliothèque qui permet de nous connecter à une BDD et de manipuler ses données.

Application web : on va utiliser le web pour utiliser l’application.

HTML et CSS sont des applis statiques cad que les données sont statiques.

Hypertexte : lien vers vers un autre document, un hyperdocument.

Hyperdocument : doc électronique contenant des images, son, vidéo, texte, petits morceaux de programme mais surtout des liens vers d’autres hyperdocuments.

Navigateur : outil pour interpréter les données HTML, CSS et JS. Cote client.

Le WEB est un service fourni par le réseau Internet. Tout comme l’e-mail.

http sert à transporter un document à un autre, c’est une couche de transport qui permet de transférer des données à une autre adresse.

Le réseau internet permet de lier deux ordinateurs entre eux via le http pour transporter les données.

WEB : système hypertexte qui permet de visiter des hyperdoc (pages) sur le réseau internet.

Serveur : ordi puissant qui va contenir l’application. Point (ordis) que d’autres points peuvent accéder. Stocke et délivre des sites web aux internautes(client)

Le client envoie des requêtes au serveur et le serveur envoie des réponses (des pages HTML) via le protocole http car nous utilisons internet pour ça.

Fonctionnement d’un site web : On tape l’url de la page, en faisant entrée, on fait une requete http vers le serveur Web (sous forme d’une demande) qui va chercher la page web et envoyer une requete http sous forme de réponse, plus précisément et donner une page HTML.

Client = ordinateur des internautes comme nous et serveur : ordi qui contient le site web.

Sites statiques : sites réalisés uniquement à l’aide des langages HTML et CSS. Il faut entrer dans les données pour changer les données si on veut les changer. Il faut le faire manipuler.

Sites dynamiques : données changent selon nos interactions (cas d’un formulaire). Dans ce cas, il y a connexion entre le serveur web et la bdd. Il faut ajouter JSP et MySQL comme langage de programmation.

Framework : code préparé pour des besoins pertinents et répétitifs. Il peut s’appliquer n’importe où pour éviter d’avoir à tout réecrire et que tout le monde utilise.

WAMP : container web qui contient un serveur web (serveur http apache) et une bdd mysql pour langage php.

JavaEE : Ensemble de standards, spécifications que nous pouvons utiliser pour faire des applis web.

JSE : pour les applis standards

Framework de la couche présentation : struts, JSF, Spring MVC.

Couche WEB : va traiter les demandes http du client. Le serveur web possède un serveur htpp pour interpréter la demande http. Il y a des pages JSP qui va correspondre au nom de la page demandée (pages web dynamiques qui contiennent code java, et html). Cette page va générer un rendu html.

La connexion a un cycle de vie court (requete http entrante et sortante)

Chaque site web doit etre incllus dans un serveur web (tomcat par ex).

Tomcat est le conteneur web.

Page jsp ne peut être interprétée que dans un serveur web pas comme html. Page java coté serveur.

Si on y met du code java, c’est le serveur qui va interpréter ce code.

Scriptlets : permet de mettre du code java dans une page jsp

Action : actions que l’on peut faire avec jsp. Ce sont des méthodes.

JSTL : librairie des tags jsp.

On va générer à partir de la page JSP la page HTML. La page JSP doit se transformer en servlet car le serveur n’interprète que les servlets.

Servlet : classe java qui permet d’envoyer au client une page jsp sous format html.

On peut créer directement notre servlet sans passer par une page JSP.

HttpServlet : Classe qui a des méthodes bien définies comme doGet() et doPost().

Une requete HTPP n’est qu’une chaine de carctère divisée en deux parties : la partie head et la partie body. Dans le head on a le statut d’une requete (chaque code va renvoyer un type d’erreur), le content type (par ex html), l’agent (navigateur), le type du statut.

<form action= « »  methode= « get ou post »></form>

Si méthode get : url : Localhost :3300/nomappl/nompage.jsp ?login=value&mdp=value, c’est pour la soumission de données dans le serveur

si on utilise Post() c’est dans le body.