

Développer **une** **application** **web**

Développer une application web

Alain Cazes

Ancien maître de conférences en informatique au Cnam

Joëlle Delacroix

Maître de conférences au Cnam
et chef de département pour le DUT Informatique

DUNOD

Toutes les marques citées dans cet ouvrage
sont des marques déposées par leurs propriétaires respectifs.

Illustration de couverture :
Communication
© Julien Eichinger - fotolia.com

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.

Le Code de la propriété intellectuelle du 1^{er} juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements

d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour

les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée. Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du

droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).



© Dunod, 2016

11 rue Paul Bert, 92240 Malakoff
www.dunod.com

ISBN 978-2-10-074375-9

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2^o et 3^o a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

TABLE DES MATIÈRES

PARTIE 1 • LES APPLICATIONS WEB – UNE PREMIÈRE APPROCHE

CHAPITRE 1 • INTRODUCTION AUX APPLICATIONS WEB	3
1.1 Une brève histoire de l'internet et du web	3
1.2 Un exemple introductif d'application web	4
Vision externe d'une application web	4
Vision interne d'une application web : requêtes, protocole, pages, réponses	6
CHAPITRE 2 • STRUCTURE D'UNE APPLICATION WEB	13
2.1 Le réseau	14
2.2 Le serveur	16
Organisation générale	16
2.3 Le client	19
2.4 Fonctionnement général	20
Étape 0	20
Étape 1	21
Étape 2	21
Étape 3	22
Étape 4	22
2.5 Serveur statique et dynamique	23
Serveur statique	23

Serveur dynamique	24
2.6 Dynamique de fonctionnement	25
2.7 Un peu de pratique : les outils de développement	26
Exemple : MAMP	27
Un QCM pour s'évaluer	

PARTIE 2 • LES ASPECTS CLIENTS

CHAPITRE 3 • ARCHITECTURE ET ÉLÉMENTS DE BASE D'UN CLIENT WEB	35
3.1 Structure générale du client	35
3.2 Les bases du langage HTML	37
Un langage de balises	37
Structure d'une page HTML	38
3.3 Page HTML et CSS	40
3.4 Page HTML et DOM	41
DOM-HTML définition	41
Javascript et DOM	42
Première approche du langage Javascript	43
CHAPITRE 4 • HTML	51
4.1 La structure du langage HTML	51
Balises (Tag) et attributs	52
Format d'une page HTML	53
4.2 L'élément <head>	55
Les propriétés liées aux textes	56
Les propriétés liées aux couleurs de fond	57
Les propriétés liées aux boîtes, bords et enrobages	57
Les propriétés liées au positionnement et à l'affichage	58
Diverses autres propriétés	59
4.3 L'élément <body>	59
Élément de groupement	59
Élément texte	61
Les contenus embarqués	63
Les liens	76
Les sections et titres	77
Les listes	79
Les données tabulaires	82
Les éléments formulaires <form></form>	85
Les éléments de scripting	89

CHAPITRE 5 • CSS	105
5.1 Syntaxe du langage CSS	91
5.2 Insertion des CSS dans une page HTML	93
Insertion à partir d'un fichier externe	93
Insertion à partir d'un code interne	93
Insertion directement dans les éléments HTML	94
 CHAPITRE 6 • JAVASCRIPT	 95
6.1 Les éléments de base du Javascript	97
Variables simples	97
Instructions	97
6.2 Javascript : langage orienté objet	106
Généralités sur les objets	106
Les objets prédéfinis	107
Les tableaux	107
La manipulation du temps	115
La manipulation des chaînes : type String	116
Les manipulations mathématiques : objet Math	118
6.3 Javascript : un langage orienté objet lié au web	119
Généralités	119
Objets liés au web	121
6.4 Production de pages dynamiques avec Javascript	128
La méthode Write()	128
Modification de l'arborescence du DOM à l'aide des méthodes Javascript	130
6.5 Le webworker	133
Processus et thread	133
Fonctionnement du Web Worker	134
Un QCM pour s'évaluer	

PARTIE 3 • LES ASPECTS RÉSEAU

CHAPITRE 7 • LES BASES DU MODÈLE CLIENT-SERVEUR	141
7.1 Le modèle client-serveur	141
7.2 Internet : principes de la communication réseau	142
Internet ou l'interconnexion de réseaux	142
L'adressage des applications sur l'interconnexion de réseau	143
La communication sur l'interconnexion de réseau	145
La couche réseau	146
La couche transport	146
La couche application	147

CHAPITRE 8 • LE PROTOCOLE HTTP	149
8.1 La notion d'URL	149
Définition	150
Exemples	150
8.2 Protocole HTTP	151
Du client au serveur : les méthodes	152
Les en-têtes HTTP	155
Amélioration des performances	157
Le protocole HTTPS	158
Un QCM pour s'évaluer	

PARTIE 4 • LES ASPECTS SERVEURS ET LES FLUX DE DONNÉES

CHAPITRE 9 • LES ASPECTS SERVEUR ET LE LANGAGE PHP	163
9.1 Les aspects serveurs	163
9.2 PHP et la production de pages HTML	165
Éléments de base	165
Variables simples	166
Les instructions	168
Fonctions : définitions et appels	172
Les variables complexes	173
9.3 PHP : fonctions utiles	175
Fonctions de manipulation des variables	175
Fonction de manipulation des tableaux	176
Fonctions pour la manipulation des éléments d'un tableau	176
Fonctions de manipulation des chaînes de caractères	178
Fonctions de manipulation des dates	180
Fonction d'inclusion de fichiers	180
9.4 PHP avancé	180
Le stockage des données	180
La gestion des communications et le partage des données entre pages	180
La redirection	181
CHAPITRE 10 • LES FLUX DE DONNÉES	183
10.1 Les échanges de données	184
Envoi de données d'un client vers un serveur	184
Récupération des données par le serveur	190
Un exemple complet	191
Les échanges de données : synthèse	194
Les scripts CGI	197

10.2	Le partage de données	198
10.3	Les redirections	201
	En utilisant Javascript	201
	En utilisant PHP	202
10.4	La synchronisation	203
10.5	Les échanges de données asynchrones : Ajax	204
	L'objet XMLHttpRequest : méthodes, propriétés, événements	205
10.6	Concepts avancés	206
	Les Websockets	206
	Les Serveur-Sent Event (SSE)	210
	Un QCM pour s'évaluer	

PARTIE 5 • STOCKAGE ET APPLICATIONS DÉCONNECTÉES

CHAPITRE 11 • LES ASPECTS STOCKAGE	217
11.1 Les cookies	217
Cookies et navigateur	218
Cookies et serveur distant	219
11.2 Le WebStorage	220
11.3 Les fichiers PHP	223
Fonctions importantes	223
Upload de fichiers	229
11.4 Notions liées aux bases de données	230
Architecture de base d'un site web avec une base de données	230
Un exemple pour illustrer le fonctionnement	233
CHAPITRE 12 • LES APPLICATIONS DÉCONNECTÉES	241
12.1 Les applications purement locales	241
12.2 Les applications temporairement déconnectées	244
Indicateurs d'état et de changement d'état	244
Image locale de l'application : le fichier cache « manifest »	244
API ApplicationCache	245
Exemple d'une structure d'application déconnectée	246
Un QCM pour s'évaluer	

PARTIE 6 • ANNEXES

Les couleurs en CSS	251
Les balises HTML	253
Les attributs	260
Les propriétés	266
INDEX	273

Partie 1

Les applications web

- une première approche

PLAN

Chapitre 1 : Introduction aux applications web

Chapitre 2 : Structure d'une application web

OBJECTIFS

- Définir ce qu'est une application web.
- Présenter l'organisation générale d'une application web et ses principaux composants : le client, le serveur, le protocole de communication.

L'utilisation du web et des applications qu'il héberge est aujourd'hui une chose courante. Qui ne surfe pas sur le web afin de rechercher des informations, réaliser des achats, jouer, écouter de la musique ou regarder une vidéo ?

Une application web est un programme de type client-serveur qui s'exécute sur le web et rend un service. La nature et la complexité de ces applications peuvent être très différentes.

Dans cette première partie, nous nous proposons d'aborder le fonctionnement de ces applications web à partir de quelques exemples concrets qui vont nous permettre de définir leurs composants. Nous aborderons également les notions système, réseaux et algorithmiques nécessaires à leur compréhension.

INTRODUCTION AUX APPLICATIONS WEB

1

PLAN

- 1.1 Une brève histoire du web
- 1.2 Un exemple introductif d'application web

OBJECTIFS

- Définir ce qu'est une application web.
- Identifier ses principaux composants.
- Présenter le vocabulaire.

1.1 UNE BRÈVE HISTOIRE DE L'INTERNET ET DU WEB

Jusqu'aux années 1950, mettre en réseau des ordinateurs consistait simplement à relier une poignée d'ordinateurs entre eux, par des lignes physiques en point à point ou par l'intermédiaire d'un ordinateur qui jouait le rôle de pont.

L'idée d'interconnecter l'ensemble des machines mondiales dans un seul réseau progresse et suscite en 1960 des travaux sur l'aiguillage des paquets de données qui aboutissent notamment en 1968 à la définition de l'adressage IP. En 1969, le réseau expérimental **Arpanet** est créé par l'ARPA (*Advanced Research Projects Agency* dépendant du DoD, *Department of Defense*) afin de relier quatre instituts universitaires : le Stanford Institute, l'université de Californie à Los Angeles, l'université de Californie à Santa Barbara et l'université d'Utah. Cette première interconnexion de quatre nœuds constitue la naissance d'Internet. Dans les années 1980, les connexions jusque-là restreintes à l'armée et aux universités concernent d'autres organismes éducatifs et des sociétés d'équipements informatiques qui participent aux travaux de recherche. Dans le même temps, un autre réseau basé sur le protocole TCP/IP et interconnectant lui aussi des universités s'est développé. En 1984, il prend le pas sur Arpanet et devient Internet. De 1984 à 1989, le nombre d'ordinateurs connectés est multiplié par 100 et atteint les 100 000. Ils sont 1 000 000 en 1992 et 36 000 000 en 1996.

Parallèlement, des applications apparaissent et se développent :

- 1971 : création du courrier électronique ;
- 1973 : création du transfert de fichiers à l'aide du protocole FTP ;

- 1978 : création de TCP/IP ;
- 1979 : apparition des premiers forums de discussion ;
- 1990 : apparition du web (www) ;
- 1993 : apparition du premier navigateur graphique appelé Mosaic ;
- 1994 : apparition de Netscape Navigator ;
- 1995 : apparition du langage Java, Netscape 2.0 peut lire des programmes Java ce qui ouvre la voie à l'animation des pages web ;
- 1996 : avec un peu de retard, Microsoft conçoit son premier navigateur : Internet Explorer 1.0 ;
- 1998 : fondation de Google ;
- 2001 : création de Wikipédia, encyclopédie au sein de la fondation Wikimedia.

1.2 UN EXEMPLE INTRODUCTIF D'APPLICATION WEB

Au moyen d'un exemple simple, commençons par introduire le vocabulaire et les composants d'une application web.

Nous présentons à cet effet une double vision, externe puis interne, d'une application web.

Vision externe d'une application web

Lorsqu'un utilisateur souhaite « surfer » sur le web afin d'accéder à un site web, il utilise un outil appelé *navigateur*.

Définition 1.1 : Navigateur

Un navigateur est un outil permettant d'accéder à des ressources sur le web. Les plus utilisés à l'heure actuelle sont Google Chrome, Mozilla Firefox, Internet Explorer, Safari et Opera.

Ce navigateur affiche un document d'accueil (cf. fig. 1.1) dans lequel une fenêtre de pilotage permet la saisie de l'**adresse web** du serveur. Le document affiché est appelé « page ».

Définition 1.2 : Adresse web du serveur

L'adresse web du serveur identifie le serveur web de façon unique sur le web. Elle est aussi appelée URL (*Uniform Resource Locator*).

La figure 1.1 illustre l'affichage de ce document d'accueil à l'aide d'un navigateur Firefox. Dans la fenêtre de pilotage du navigateur, le lien saisi établit une correspondance avec le serveur et permet d'accéder à la première page du service souhaité (ici « pages jaunes »). La page correspondant au service demandé s'affiche dans le navigateur (figure 1.2).

1.2 • Un exemple introductif d'application web



Figure 1.1 – Page d'accueil du navigateur avec fenêtre de pilotage.

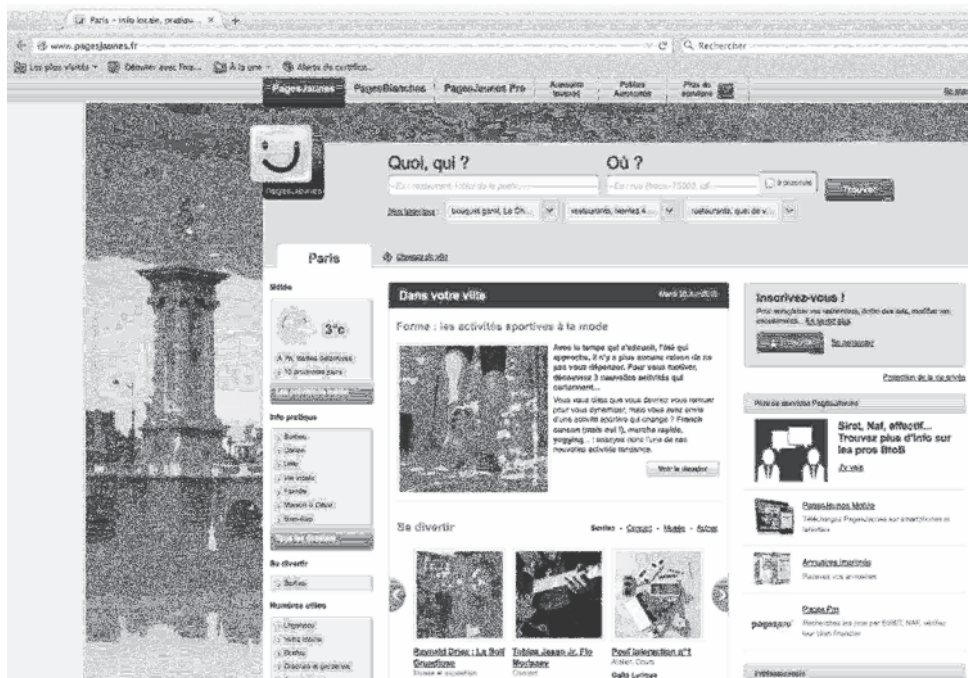


Figure 1.2 – Page affichée en retour de la requête, <http://www.pagesjaunes.fr>

Une **page** est composée d'une grande variété d'éléments :

- des onglets cliquables : Pages Jaunes, Pages Blanches, Annuaire Inversés, Trouver, etc. Ils permettent de changer de pages ;
- des images ;
- des informations textuelles ;
- des zones de saisie « Quoi, Qui, Où » qui permettent de saisir des textes et d'appeler une nouvelle page en lui envoyant comme paramètres les valeurs saisies dans les champs ;
- des couleurs et des éléments de mise en forme (gras, souligné, taille police, etc.

L'affichage de cette page résulte d'un ensemble d'actions :

- dans la barre de pilotage du navigateur, nous avons saisi une « **requête** » qui a été émise par le **client** (le navigateur) à destination d'un **serveur** (ordinateur distant connecté à Internet) disposant de logiciels capables d'interpréter et de répondre à la requête.
- le navigateur reçoit une **réponse** du serveur sous forme d'un fichier qu'il **interprète**. Cette interprétation par le client du fichier produit l'**affichage** correspondant.
- l'affichage des pages est le résultat d'un dialogue entre le client et le serveur au travers d'un réseau. Ce dialogue est régi par un **protocole** définissant les règles du dialogue. Dans le cas des services web, les échanges se font selon un modèle « **client/serveur** » par échange de messages. Les messages sont les requêtes (du client vers le serveur) et les réponses (du serveur vers le client). Le protocole qui gère les échanges est le protocole **HTTP** (*HyperText Transfer Protocol*). Les « pages » sont des fichiers exprimés dans un format compris par le navigateur qui affiche le résultat de l'interprétation du contenu de la page. Les pages sont localisées sur le serveur, acheminées par le réseau vers le client, interprétées et affichées par le client.

Dans notre exemple, la requête `http://www.pagesjaunes.fr` demande d'accéder à la page d'accueil du serveur dont l'adresse est `www.pagesjaunes.fr` et la réponse est l'affichage de l'interprétation du fichier correspondant.

Le modèle client/serveur est un modèle de communication entre deux entités : un client émet des requêtes et un serveur qui rend un service en réponse à la requête. Ce modèle est détaillé dans la partie 3 de cet ouvrage.



Un protocole de communication spécifie les règles permettant à deux entités de communiquer. Le protocole FTP par exemple spécifie les règles permettant le transfert de fichiers sur un réseau. Le protocole HTTP est le protocole permettant à un client navigateur d'accéder aux ressources d'un serveur web. Ce protocole est présenté dans la partie 3 de cet ouvrage.



Un fichier est un document stocké sur le support de masse d'un ordinateur.



Vision interne d'une application web : requêtes, protocole, pages, réponses

L'exécution d'une application web met donc en œuvre un certain nombre d'actions internes qui font intervenir :

- des **requêtes** émises par le client ;
- des **réponses** fournies par le serveur ;
- un protocole de communication ;
- un ensemble de **pages**.

Nous illustrons ces notions à l'aide de trois exemples différents qui présentent trois types de serveurs différents :

- le serveur statique ;
- le serveur statique interactif ;
- le serveur dynamique.

Premier cas : le serveur statique

Une requête est envoyée au serveur afin de demander l'affichage d'une page. Aucune donnée n'est transmise avec cette demande.

La réception de la requête provoque l'envoi par le serveur d'une nouvelle page.

• *Les requêtes*

La fenêtre de pilotage du navigateur contient l'adresse de la page recherchée. Ainsi, en cliquant sur l'onglet « PagesBlanches », le navigateur génère la requête : `http://www.pagesjaunes.fr/pagesblanches` (figure 1.3)

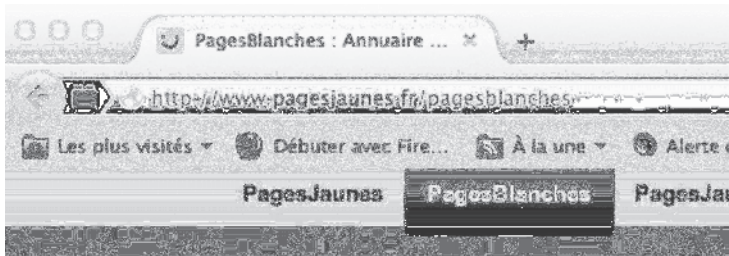


Figure 1.3 – Illustration de la requête générée par clic sur l'onglet PagesBlanches.

Cet exemple donne la forme générale d'une requête dans laquelle sont spécifiés :

- le **protocole** : HTTP. Il gère la communication entre le client (le navigateur) et le serveur (l'application httpd) et précise les règles des échanges entre ces deux entités. Apache est le serveur « **httpd** » le plus populaire. C'est un programme qui s'exécute en tâche de fond sur le serveur (un processus démon dans la terminologie système, le « d » suffixant son nom indiquant sa nature) et qui, notamment, extrait le nom de la page à renvoyer (dans notre exemple : « pagesblanches ») ;
- le nom du **serveur ciblé** : `www.pagesjaunes.fr` ;
- le nom de la **page recherchée** sur le serveur : `pagesblanches`

• *Les pages*

Les pages sont stockées sur le serveur et sont « nommées » de façon à ce que l'application « **httpd** » puisse les retrouver.

Ces pages sont des « fichiers » qui peuvent avoir des formats variables. Ce sont par exemple :

- des programmes écrits dans des langages de programmation tels que HTML, Java, PHP, etc. ;
- des fichiers texte au format pdf, doc, etc. ;
- des fichiers audio, vidéo, des images, etc.

• *Les réponses*

Dans cet exemple, le client émet une requête à laquelle le serveur répond en renvoyant le fichier « pagesblanches ». Le navigateur reçoit la réponse et interprète

le fichier reçu. Le résultat de cette interprétation entraîne **l’affichage** de la page (figure 1.4).

Les pages sont des fichiers qui peuvent avoir des formats variés. À la réception de la réponse, le navigateur doit connaître le type de fichier afin de choisir l’interpréteur qui convient au format du fichier reçu. Pour cela, la réponse du serveur contient un en-tête qui indique divers éléments caractérisant le fichier et notamment le type du fichier, défini au travers du format Mime.

Définition 1.3 : Format MIME

Un format MIME est un format de données sur Internet composé de deux parties : un *type* et un *sous-type* et d’un ou plusieurs autres champs si besoin.

Exemple :

- type audio, sous-type audio/mpeg
- type image, sous-type image/gif



Figure 1.4 - Illustration de la réponse affichée en réponse à la requête générée par clic sur l’onglet PagesBlanches.

Définition 1.4 : Serveur statique

L’exemple précédent illustre une situation très fréquente. Une requête émise par le navigateur est reçue par le serveur qui envoie une nouvelle page vers le client. Il n’y a pas d’échange de données entre la page appelante et la page appelée : les contenus des pages sont autonomes. La page appelée ne contient pas de variables susceptibles de modifier son contenu. Le serveur est « statique ».

Deuxième cas : le serveur statique interactif

Pour ce second cas, un bouton à l'intérieur de la page est cliqué ; il n'y a pas de requête générée par le navigateur. Il y a uniquement une modification de la page qui était affichée dans le navigateur.

La modification est réalisée au niveau du client par le navigateur, à travers un code écrit dans un langage de programmation tel que Javascript, dont l'exécution est déclenchée par le clic du bouton. Ce langage permet d'accéder et de modifier tout ou partie des éléments de la page affichée. Javascript est présenté dans la partie 2 de cet ouvrage.

Nous avons donc une sorte d'affichage « interactif » car certains éléments de la page sont « sensibles » à des « événements » externes (ex. : « clic souris ») qui entraînent l'exécution de code modifiant la page. Par exemple, figure 1.5 un clic sur l'onglet « Annuaire Inversé » entraîne la modification affichée.

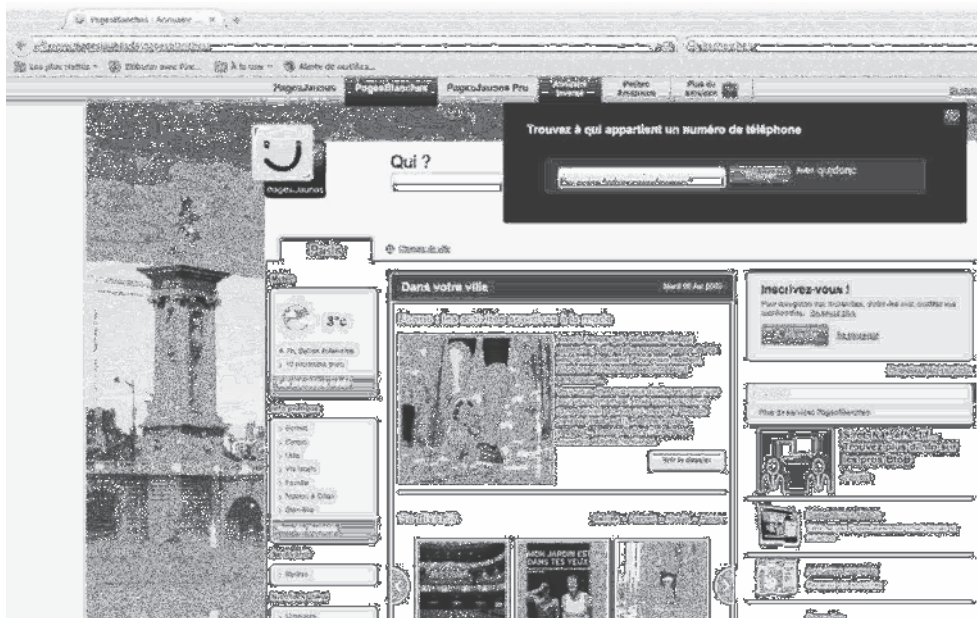


Figure 1.5 - Illustration de la réponse affichée en réponse au clic sur l'onglet AnnuaireInversé.

La page n'a pas changé, mais un bandeau noir contenant un champ de saisie est apparu. Il n'y a pas eu une nouvelle page chargée ; seule une modification locale a été réalisée par le navigateur. Cette modification locale est réalisée sans remplacement de la page qui était présente dans le navigateur. Ceci est différent de l'exemple précédent où chaque action provoquait un remplacement de la page affichée sur le navigateur. Dans ce dernier cas, le navigateur ne génère pas une requête et n'attend donc pas de réponse.

Définition 1.5 : Serveur statique interactif

Cet exemple illustre une situation dans laquelle un programme modifie les caractéristiques d’affichage de la page. Ce programme, inséré dans la page, est écrit dans un langage tel que Javascript pour lequel le navigateur dispose d’un interpréteur. Nous sommes ici en présence d’un serveur statique interactif.

Troisième cas : le serveur dynamique

Pour ce troisième exemple, la page affichée contient des champs de saisie permettant à l’utilisateur de saisir des informations qui doivent être communiquées à une autre page : Par exemple, il saisit un identifiant et un mot de passe qui doivent être communiqués à une application pour que l’utilisateur soit authentifié. Dans une telle situation, le navigateur génère une requête intégrant les données contenues dans les champs de saisie afin de les transmettre au serveur.

- *Les requêtes*

Dans notre exemple, la page « pagesblanches » contient le bouton « Trouver » associé aux champs de saisie (« Qui ? », « Où ? ») (figure 1.6).

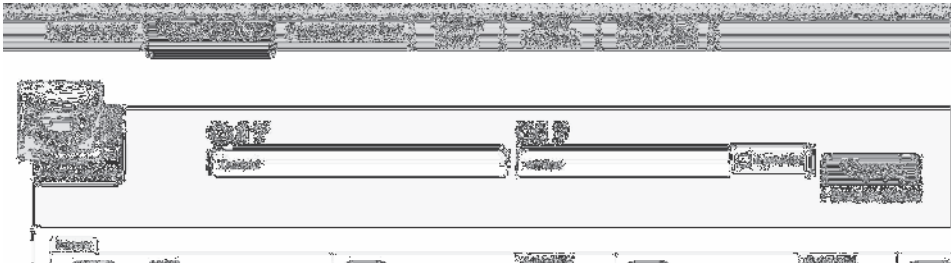


Figure 1.6 - Illustration d’une page avec des champs de saisie.

La requête générée doit désigner la page à afficher et intégrer les données saisies. La manière dont les données sont transmises est variable.

- *Les réponses*

La requête produite par le navigateur fait référence à une nouvelle page « **listeReponseClassique.jsf** » qui a récupéré les champs de saisie et fournit les résultats trouvés (figure 1.7).

Le contenu des champs de saisie a été communiqué à la page ciblée par la requête. Ceci correspond à une première façon de transmettre les données saisies. Elle met en œuvre un programme Java « **listeReponseClassique.jsf** » capable de récupérer les données.

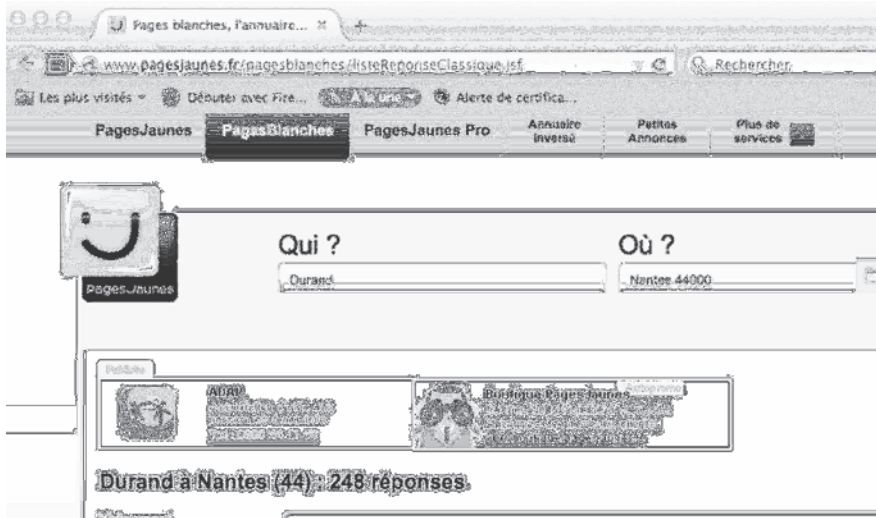


Figure 1.7 – Page «listeReponseClassique.jsf».

Une autre façon de transmettre le contenu des champs de saisie est illustrée par la figure 1.8. Dans ce cas, la requête contient :

- le nom du serveur `www.laredoute.fr`,
- le chemin d'accès à la page qui doit s'afficher « `ppdp/prod-324494543.aspx` »,
- et le contenu du champ de saisie à transmettre « `docid=0000000000001` ».

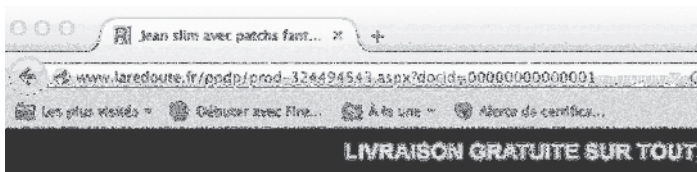


Figure 1.8 – Transmission des champs de saisie.

Définition 1.6 : Serveur dynamique

Dans les deux exemples présentés, la page cible contient un **programme** écrit dans un **langage de programmation** exécutable sur le serveur. Les données saisies sont transmises à des variables définies dans la page cible. Le programme cible manipule ces variables et produit « **dynamiquement** » un résultat inséré dans la page cible.

La page ainsi modifiée est transmise au client comme dans le premier exemple. Le serveur est qualifié de **dynamique** dans le sens où les pages sur le serveur ont été produites et/ou modifiées par un programme informatique.

Cette situation met en évidence la nécessité d'un langage de programmation sur le serveur afin de récupérer des données, les manipuler et produire une page dont le contenu dépend des données fournies et des calculs faits avec elles. Ces notions sont abordées en partie 4.

Synthèse : la dynamique de fonctionnement

- Une **application web** est composée d'un ensemble de pages décrites par un langage de programmation. Elle est organisée autour de trois composants :
 - une partie client, le **navigateur**. Il émet des requêtes (identification de la page à afficher), reçoit la page demandée, affiche la page ;
 - une partie **serveur** qui héberge les pages ;
 - un **réseau** qui assure le transport des requêtes et des réponses (les pages demandées). Les échanges entre le client et le serveur se font au travers d'un modèle client/serveur piloté par le protocole HTTP.
- La dynamique de fonctionnement d'une application web peut se résumer par le schéma de la figure 1.9 :
 - 1 - le navigateur émet une requête vers le serveur et précise l'adresse de la page recherchée ;
 - 2 - les couches réseau permettent l'accès à l'ordinateur cible à l'aide de son adresse internet ;
 - 3 - le suffixe « http » de l'URL indique au système d'exploitation de s'adresser au démon « httpd ». Celui-ci s'exécute en tâche de fond sur le serveur. « httpd » est le logiciel serveur (par exemple Apache) ;
 - 4 - le démon « httpd » adresse le fichier cible et fabrique un fichier préfixé par un en-tête spécifiant notamment le type du fichier à interpréter par le navigateur et l'adresse du client ;
 - 5 - à réception par le navigateur, l'entête permet de sélectionner l'interpréteur correspondant au type du fichier ;
 - 6 - l'interprétation déclenche l'affichage.

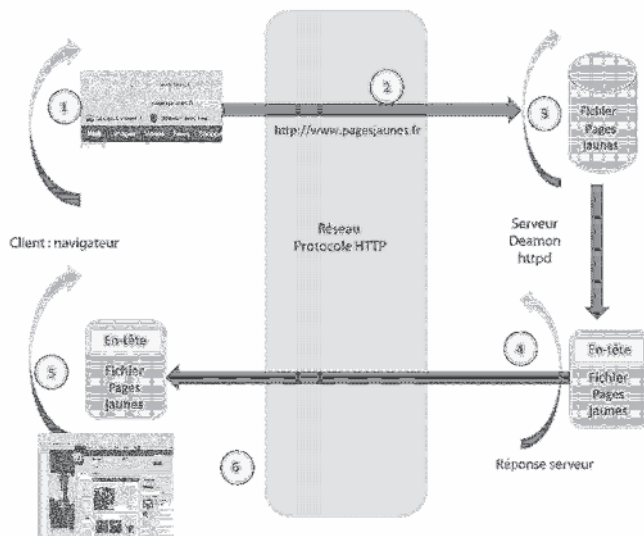


Figure 1.9 - Dynamique de fonctionnement.

STRUCTURE D'UNE APPLICATION WEB

2

PLAN

- 2.1 Le réseau
- 2.2 Le serveur
- 2.3 Le client
- 2.4 Fonctionnement général
- 2.5 Serveur statique et dynamique
- 2.6 Dynamique de fonctionnement
- 2.7 Un peu de pratique : les outils de développement

OBJECTIFS

- Définir les principaux composants
- Comprendre le fonctionnement et l'architecture générale d'une application web

Une application web est organisée autour de trois composants : un client, un serveur et un réseau. Généralement, ces applications manipulent des données stockées dans des bases de données.

Le **serveur web** désigne tout à la fois le serveur matériel (*hardware*) et le serveur logiciel composé du système d'exploitation (compilateurs, interpréteurs de code), des applications (Apache, Java, NodeJS) et des données (les ressources).

Le **réseau** est composé par l'interconnexion mondiale Internet et l'utilisation pour les applications web du protocole HTTP.

Le **client web** désigne tout à la fois le client matériel (*hardware*) et le client logiciel à savoir le navigateur.