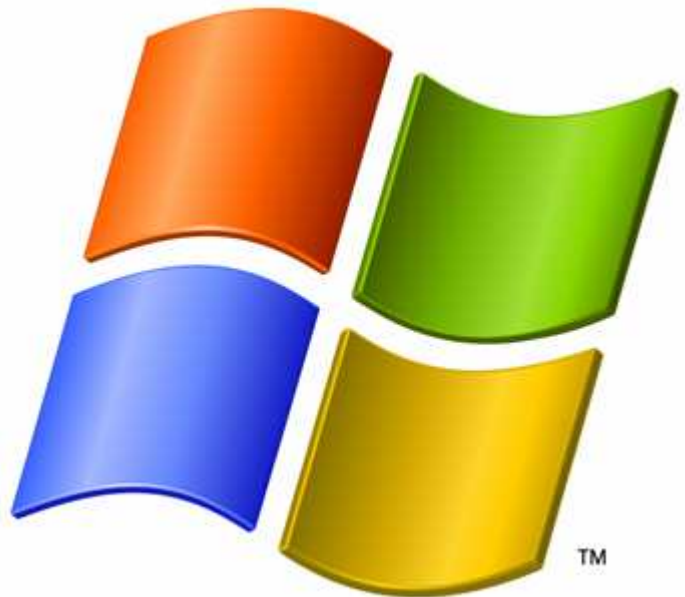


TP RE41 : MOTEUR DE RECHERCHE IIS

Microsoft®
IIS®



Sommaire

Introduction.....	- 3 -
I/ Installation du serveur IIS	- 4 -
II/ Conception de l'interface de recherche.....	- 5 -
1) Le formulaire de recherche.....	- 5 -
2) La requête	- 7 -
3) Affichage des résultats	- 8 -
Conclusion	- 10 -
Annexes	- 11 -

Introduction

IIS, le serveur Internet de Microsoft, offre de nombreux services aux développeurs Web, et notamment une série de fonctions permettant de réaliser des recherches dans les fichiers disponibles sur le serveur. Elle laisse libre choix au développeur de définir différents paramètres de recherches tels que les informations affichées ou le nombre de résultats à fournir.

Le client aura donc la possibilité de formuler des requêtes en utilisant un navigateur Internet (via un formulaire en HTML). Une fois la requête envoyée et interprétée par le serveur, ce dernier renverra les résultats les plus pertinents et le client pourra alors les consulter grâce à une page HTML.

Nous allons présenter notre travail selon deux points principaux ; nous verrons tout d'abord l'installation du serveur IIS puis nous étudierons ensuite plus précisément la conception et la mise en place de l'interface de recherche. Enfin, vous pourrez consulter le code source dans son intégralité en annexes.

I/ Installation du serveur IIS

L'installation et le paramétrage d'IIS se fait très simplement et en quelques étapes.

Tout d'abord, il faut installer ce nouveau service dans Windows 2003 Server. Pour cela, il suffit d'ajouter un composant de Windows nommé *Service Internet (IIS)*.

Puis, il est nécessaire d'installer le *Service d'Indexation* qui fait partie des composants de Windows. Il faut ensuite donner à ce service les autorisations nécessaires : *Gestionnaire des Services/Service d'Indexation/Autoriser*.

Enfin, comme nous avons pu le voir dans précédents TP, nous devons créer un *Répertoire Virtuel*, ce qui permettra notamment de rendre accessible notre nouveau moteur de recherche depuis le réseau, grâce à un simple *alias*.

II/ Conception de l'interface de recherche

1) Le formulaire de recherche

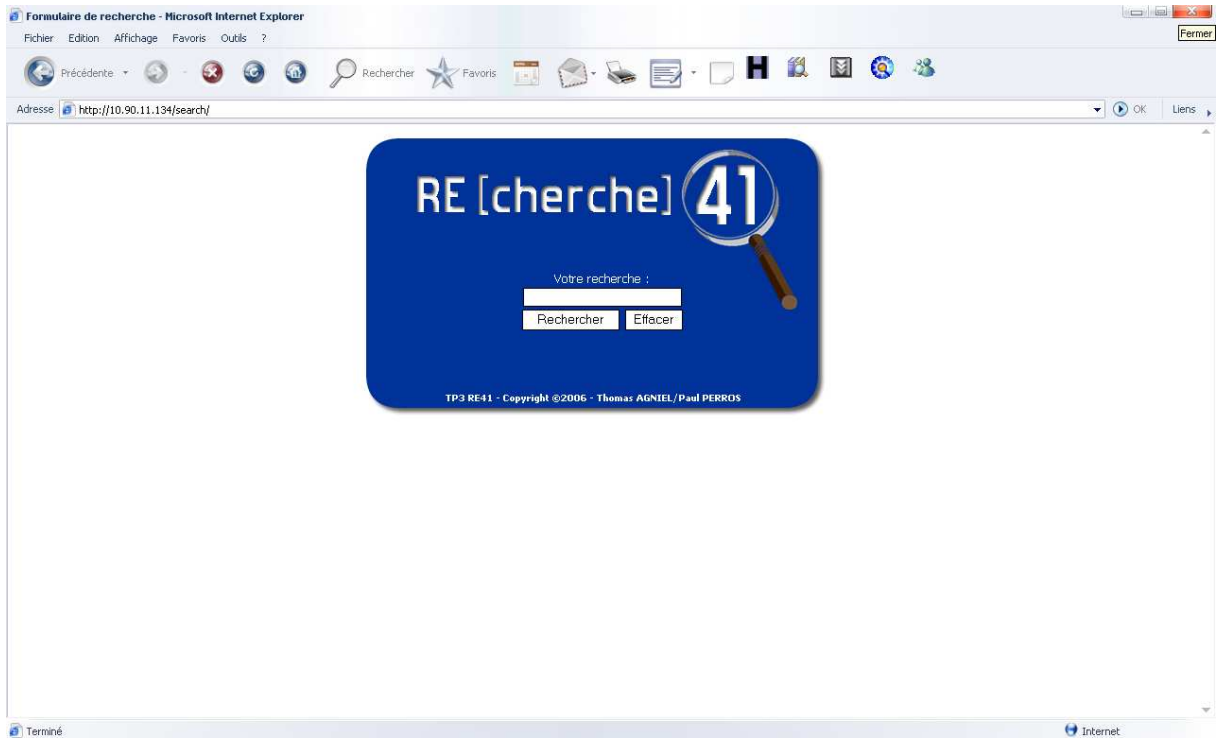


Figure 1 : Page initiale

A l'aide d'une feuille de style CSS, et d'un simple fichier HML *query.htm* nous avons réalisé la première étape du moteur de recherche.

Cette interface a pour simple but de permettre à n'importe quel utilisateur d'indiquer quelle recherche il souhaite effectuer.

Une fois la recherche lancée, ce formulaire permet d'envoyer la requête au fichier *.idq* (nous reviendrons plus tard sur son rôle et fonctionnement).

Ce formulaire donne donc la possibilité à l'utilisateur de préciser différents critères pour sa recherche, et de les transmettre sous forme de paramètres au fichier *.idq* qui se chargera de la requête de recherche.

```
<FORM ACTION="query.idq?" methode="POST" name="formulaire" onSubmit="return  
champsok()">  
  <font color="white">Votre recherche :</font><br>  
  <INPUT TYPE="TEXT" class="bouton" NAME="CiRestriction" SIZE="24" MAXLENGTH="100"  
VALUE=""><BR>  
  <INPUT TYPE="SUBMIT" VALUE="Rechercher">  
  <INPUT TYPE="RESET" VALUE="Effacer"><BR>  
</FORM>
```

On remarque donc que ce formulaire de recherche est en effet un formulaire très simple et très classique. Cependant, on peut repérer une légère amélioration par rapport à un formulaire classique : *onSubmit="return champsok()"*

En effet, lorsque l'utilisateur clique sur *Rechercher*, avant de lancer la requête, une fonction (en JavaScript) est préalablement utilisée pour vérifier que l'utilisateur a bel et bien renseigné le champ de recherche, et qu'il l'a fait de manière correcte :

```
function champsok() {
    if ((document.formulaire.CiRestriction.value.length == 0) ||
        (document.formulaire.CiRestriction.value == " ") ||
        (document.formulaire.CiRestriction.value.indexOf('*') == 0))
    {
        alert("Veuillez entrer une recherche valide");
        return false;
    }
    return true;
}
```

Nous vérifions donc dans cette fonction que l'utilisateur a rempli le champ de recherche, qu'il n'a pas mis qu'un seul espace, et que sa recherche ne commence pas par une étoile *. Si l'un des cas cités est détecté, un message d'erreur s'affiche donc, et la recherche ne s'effectue pas.

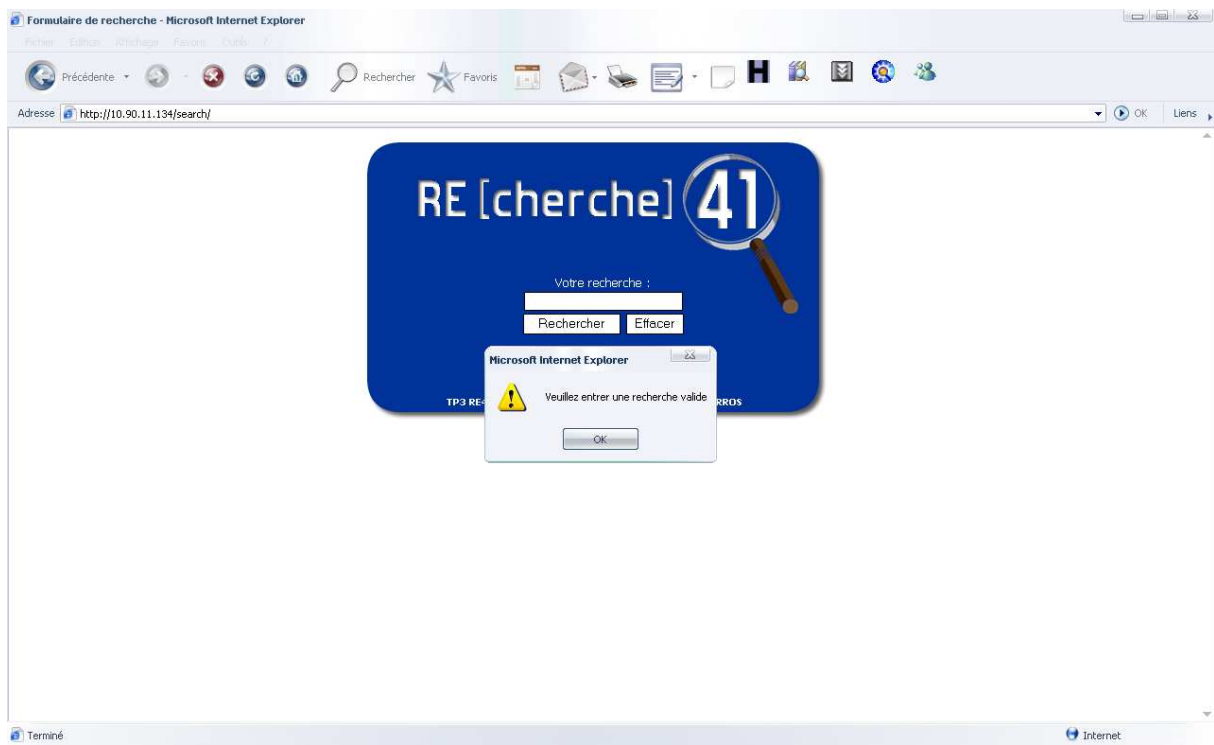


Figure 2 : Message d'erreur suite à la validation d'une recherche incorrecte

Améliorations possibles : Dans notre formulaire, nous permettons à l'utilisateur de renseigner seulement le mot clé recherché. Cependant, il est possible d'ajouter certains paramètres tel que le nombre de réponses affichées par page, ou encore le tri des réponses (par ordre alphabétique, par date, par taille...)

2) La requête

Pour que la requête soit effectuée sur le serveur IIS, un certains nombres de paramètres doivent être spécifiés.

Ces informations sont stockées dans un fichier IDQ (Internet Data Query), qui sera interprété par IIS pour effectuer la recherche.

```
[Query]
CiColumns=filename,size,rank,characterization,vpath,DocTitle,write
CiFlags=DEEP
CiRestriction=%CiRestriction%
CiMaxRecordsInResultSet=150
CiMaxRecordsPerPage=5
CiScope=C:\
CiTemplate=search/query.htx
CiSort=rank[d]
```

Détails :

[Query]	Précise qu'il va s'agir d'une recherche
CiColumns	Donne les informations qui devront être renvoyées par le serveur. Ici le nom du fichier, sa taille, son rang, son titre, son adresse sur le disque, son type et sa date de création
CiFlags	Différents drapeaux utiles pour la recherche. Ici DEEP précise que la recherche sera récursive
CiRestriction	Mot à rechercher (mot indiqué dans le formulaire)
CiMaxRecordsInResultSet	Nombre de résultats maximum
CiMaxRecordsPerPage	Nombre de résultats affichés par page
CiScope	Adresse où l'on doit rechercher
CiTemplate	Fichier utilisé pour afficher le résultat de la recherche
CiSort	Classement des résultats. Ici par rang décroissant.

3) Affichage des résultats

Une fois le formulaire de recherche renseigné, et la requête effectuée, il faut maintenant afficher les résultats de la recherche.

Nous allons utiliser pour cela un fichier possédant l'extension *.htx*. Un fichier HTX se présente de la même manière qu'un fichier HTML classique. Il est écrit dans le même langage et utilise les mêmes balises. La différence se situe dans le fait que ce fichier est interprété par IIS.

Pour cette raison, un fichier HTX peut utiliser un certain nombre de variables fournies par le serveur, afin de créer le fichier résultat de différentes manières selon ces dernières. Les variables que nous utiliserons pour afficher les résultats dans notre cas sont les suivantes :

%CiMatchedRecordCount%	: Nombre de résultats trouvés
%CiFirstRecordNumber%	: Numéro du premier résultat
%CiLastRecordNumber%	: Numéro du dernier résultat
%CiCurrentRecordNumber%	: Numéro du résultat courant
%CiMaxRecordsPerPage%	: Nombre de résultats à afficher par page
%CiRecordsNextPage%	: Nombre de résultats restant à afficher
%CiCurrentPageNumber%	: Numéro de la page courante
%CiTotalPageNumber%	: Nombre total de pages de résultats
%CiQueryIncomplete%	: Indicateur de requête complète
%CiContainsLastRecord%	: Indicateur pour le dernier résultat
%CiRestriction%	: Requête envoyée par le client
%begindetail%	: Balise de début de description des détails
%DocTitle%	: Titre du document correspondant au résultat
%filename%	: Nom du lien vers l'URL du résultat

```
<H5>
  <%if CiMatchedRecordCount eq 0%>
    Aucun document correspondant &agrave; la requ&ecirc;te "<%CiRestriction%>".
  <%else%>
    Documents <%CiFirstRecordNumber%> &agrave; <%CiLastRecordNumber%> des
    <%if CiMatchedRecordCount eq CiMaxRecordsInResultSet%>
      meilleurs
    <%endif%>
    <%CiMatchedRecordCount%> documents correspondants &agrave; la requ&ecirc;te
    "<%CiRestriction%>".

  <%endif%>
</H5>
```

Ces lignes permettent l'affichage des résultats si ils existent, sinon cela affiche un message indiquant à l'utilisateur qu'aucun document n'a été trouvé.

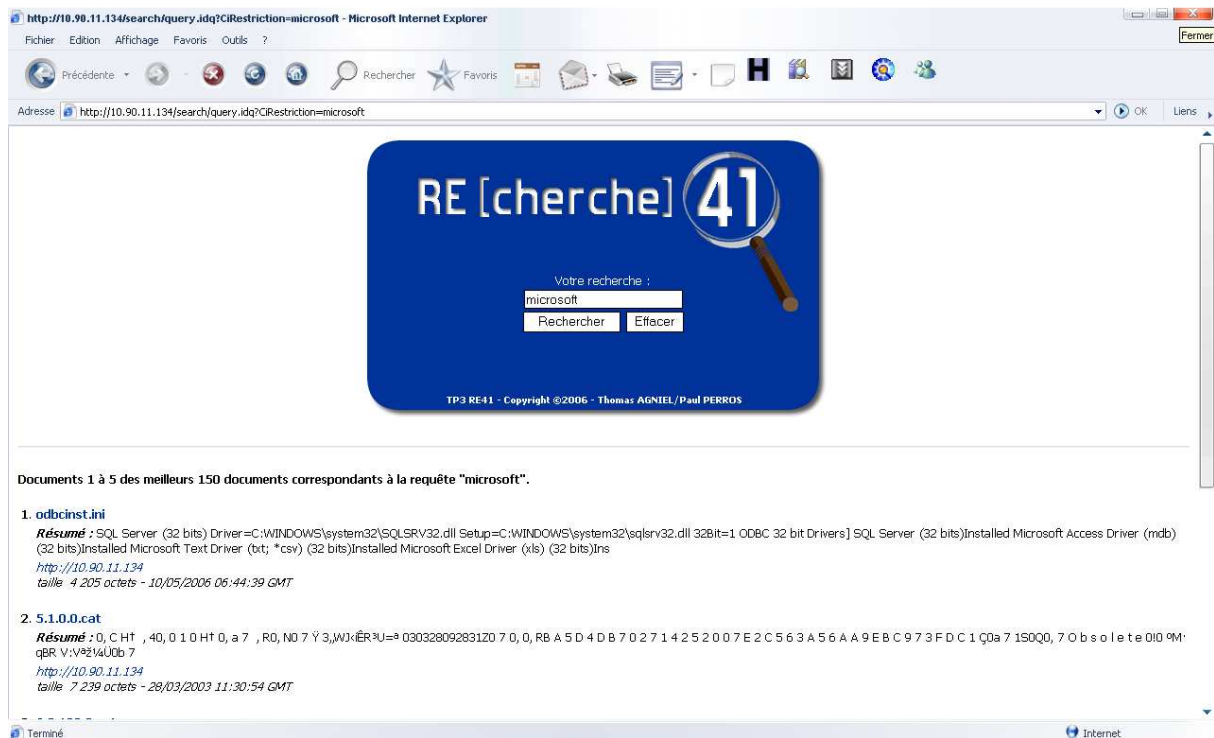


Figure 3 : Affichage des résultats suite à une requête valide de l'utilisateur

Nous avons ensuite intégré des boutons *Suivant* et *Précédent* pour visualiser les résultats sur la page suivante, ou précédente (cf code en annexe).

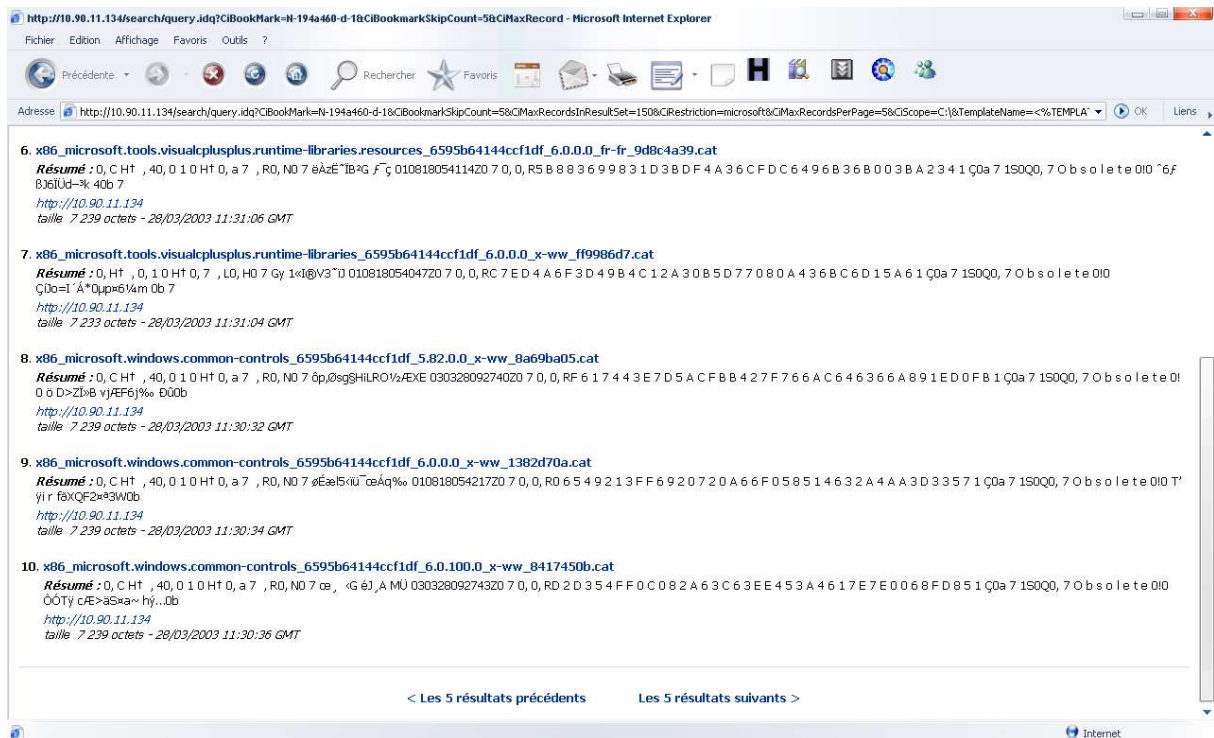


Figure 4 : Pied de page de l'affichage des résultats (présence des boutons précédents et suivants)

Conclusion

Grâce à la mise en place d'un tel serveur et des 3 fichiers principaux (query.htm, query.idq et query.htx), nous avons pu réellement comprendre le mécanisme d'IIS. Nous nous sommes aussi aperçu que le développeur web pouvait spécifier et personnaliser son serveur de manière assez précise sans trop de difficultés, ce qui est un avantage certain.

Le client, quant à lui, peut facilement utiliser le moteur de recherche mis en place : en effet, l'interface de recherche étant une interface Web est très simple, très rapide et très efficace. On peut donc aussi apprécier cette simplicité du point de vue de l'utilisateur.

En conclusion, je dirais que ce TP nous a aussi apporté pas mal de connaissances sur les moteurs de recherche en général, puisque les autres moteurs de recherche utilisent à priori le même principe (indexation des fichiers puis affichage des résultats correspondants aux différentes requêtes).