



API Web

exemple de XediX Tera Solution

Pierre Brochard

pierre.brochard@cea.fr

21/2/2013

15/10/2013



Plan



- Pierre Brochard -

- **Introduction**
- **APIs Web**
- **Architecture d'une base avec le Web**
- **URLs de XediX**
- **Racine X2Admin**
- **Racine X2Import**
- **Racine X2Documents**
- **Racine X2Imgelems**
- **Racine X2Accueil**
- **Racine X2Xsearch**
- **Exemple de requêtes HTTP**
- **Exemples en Javascript de génération d'URLs**
- **Exemples avec une base XediX**
- **TD**

Introduction

15/10/2013



Type d'APIs pour bases de données



API pour Application Programming Interface ou Interface de Programmation qui peut-être multi-langages ou indépendant d'un langage en particulier. On cite quelques exemples ci-dessous.

- Monde des bases relationnelles : API *ODBC* qui implémente la norme ISO/IEC 9075 qui fusionne CLI et SQL.
- *JDBC* : sorte d'*ODBC* pour Java qui permet à Java de s'interfacer avec des bases relationnelles.
- *Web SQL Database* : API de type script (Javascript) implémentée par *Sqlite* et spécifié par le *W3C* (<http://www.w3.org/TR/webdatabase/>).
- *Web Storage* : stockage de données persistantes de type paires (nom,valeur) pour les clients Web spécifié par le *W3C* (<http://www.w3.org/TR/webstorage/>)
- Bases *NoSQL* avec interfaces spécifiques qui typiquement stockent des paires (nom,valeur) indexées par des fonctions type fonctions de hashage :
 - *CouchDB* (orientée documents),
 - *MongoDB* (orientée documents et dynamique),
 - *RavenDB* (écrit en .net et orientée documents),
 - *Redis* (de type paires (nom, valeur) sous licence BSD),
 - *Cassandra* (cassandra.apache.org) qui vient de *Facebook*,
 - *SimpleDB* (vient d'*Amazon*).

1/53

15/10/2013



Autres types de bases



- Bases de données orientées colonnes qui stockent les données du même *type* les unes à la suite des autres :
 - *BigTable* (base de données de Google),
 - *HBase* (implémente *BigTable* - sous-projet de *Hadoop* - sera utilisé par *Facebook*),
 - *Cassandra* (est aussi NoSQL),
 - *HyperTable* (concurrent de *BigTable* qui utilise *Hadoop*).
- Bases XML natives :
 - *eXist*,
 - *SEDNA*,
 - *Oracle Berkeley DB XML*,
 - *Tamino* (de la société *Software AG*),
 - *XIndice* (<http://xml.apache.org/xindice/>).

2/53

15/10/2013



Autres types de bases (II)



- Bases de données orientées objets :
 - *db4o* (permet la persistance d'objets Java et .net),
 - *ZODB* (persistance d'objets en Python),
 - *NEO* (stockage transactionnel distribué et objet),
 - *PERST* (base de données embarquée orientée-objet pour Java et .net).
- Bases de données orientées graphe :
 - *HypergraphDB* (pour Java),
 - *AllegroGraph* (stocke du RDF),
 - *InfiniteGraph* (<http://www.infinitegraph.com/> - base distribuée qui traite des graphes),
 - *InfoGrid* (base de graphes d'objets pour Java avec rendu Web).

3/53

15/10/2013



Exemple de l'API de Cassandra



```
/etc/init.d/cassandra start
```

```
cassandra-cli  
Welcome to Cassandra CLI.
```

```
Type 'help' or '?' for help. Type 'quit' or 'exit' to quit.  
cassandra> connect localhost/9160  
Connected to: "Test Cluster" on localhost/9160
```

```
cassandra> show cluster name  
Test Cluster
```

```
cassandra> show keyspaces  
Keyspace1  
system
```

```
cassandra> show config file  
...(on ajoute People comme Keyspaces sous forme XML dans storage-conf.xml....
```

```
cassandra> show keyspaces  
Keyspace1  
system  
People
```

4/53

15/10/2013



Exemple de l'API de Cassandra (II)



```
cassandra> describe keyspace People
People.Users
Column Family Type: Standard
Columns Sorted By: org.apache.cassandra.db.marshall.BytesType@1b22920
...

cassandra> get People.Users['1']
Returned 0 results

cassandra> set People.Users['1']['email'] = 'toto.tata@tata.fr'
cassandra> set People.Users['1']['first_name'] = 'Toto'
cassandra> set People.Users['1']['last_name'] = 'Tata'

cassandra> get People.Users['1']
=>(column=....., value=Tata, timestamp=....)
=>...
Returned 3 results

cassandra> count People.Users['1']
3 columns
```

5/53

15/10/2013

APIs Web

15/10/2013



APIs Web



- Surcouches Web plus ou moins complètes ou dédiées peuvent exister au-dessus d'autres protocoles publics ou privés comme par exemple :
 - *PhpMyAdmin* comme surcouche Web de *MySQL*.
- modules du serveur Web *Apache* :
 - *mod_dbd* (http://httpd.apache.org/docs/2.2/mod/mod_dbd.html),
 - *mod_sql* ou autres (<http://apache.webthing.com/database/>),
 - ...
- Langages standardisés par le W3C comme *XQUERY* (<http://www.w3.org/TR/xquery/>) pour extraire des données en XML.

6/53

15/10/2013



Autres APIs ou protocoles Web



■ SOAP

- Originellement projet de Microsoft,
- Devenu recommandation du W3C en 2003,
- Peut utiliser comme protocole de transport HTTP, HTTPS, SMTP ou autres,
- Originellement veut dire Simple Object Access Protocol,
- Permet d'échanger des informations structurées en implémentant des Services Web,
- Repose sur XML pour le format de ses messages en entrée et en sortie,
- Protocoles de type applicatif comme RPC et HTTP pour la négociation et la transmission de ses messages,
- Trois parts pour ses messages : une enveloppe, un jeu de règles d'encodage et une convention pour représenter les appels de procédures et leur réponses.

7/53

15/10/2013



Serveur Web



- Sert des requêtes au standard *HTTP* en provenance de clients.
- Conception modulaire comme pour *Apache* (<http://httpd.apache.org/>).
- La chaîne de traitement des requêtes et de leur réponse est une suite de modules :
 - Module filtres de la requête,
 - Modules pour lire les données utiles pour composer la réponse qui viennent :
 - de fichiers en provenance d'un système de fichiers stockés,
 - de données en provenance d'une base de données avec connections persistantes, semi-persistantes ou atomiques,
 - qui appellent des processus externes via un module *mod_cgi* qui implémente le standard *CGI* (Common Gateway Interface),
 - ou *FastCGI* qui maintient des connections persistantes avec un processus externe,
 - ou autres modules dédiés.
 - Modules filtres de la réponse :
 - Ajouts d'en-têtes standards ou spécifiques,
 - Encodages de la réponse.

8/53

15/10/2013



HTTP - rappels



- Standard défini par l'IETF (*RFC 2616* de l'Internet Society - <http://www.w3.org/Protocols/rfc2616/rfc2616.html>).
- Existe en version 1.0 ou 1.1.
- *HTTP 1.1* permet de maintenir des connections sockets persistantes très utiles pour les temps de réponses des sites Web :
 - Les sockets persitantes nécessitent l'en-tête **Content-Length:** pour la réponse qui donne la taille de la réponse.
 - Cette taille peut être difficile ou pénalisante à calculer (XML généré comme un flux par exemple).
 - Dans ce cas, le serveur Web génère automatiquement une suite de blocs de taille connue et formate la réponse avec un encoding particulier en ajoutant l'en-tête **Transfer-Encoding: chunked**. Le client sait ainsi quand il doit arrêter de lire la réponse.
- Méthodes *GET, HEAD, POST,*

9/53

15/10/2013



HTTP - exemple



- Exemple de requête HTTP 1.1 sans demande de fermeture de connexion :

POST /web/search.php HTTP/1.1

Host: www.toto.fr

Keep-Alive: 300

Connection: keep-alive

Content-Type: application/x-www-form-urlencoded

Content-Length: 48

keyword=ddd&submit=%EF%EE%E8%F1%EA+%EF%EE+Web

10/53

15/10/2013



HTTP - exemple (II)



Exemple de réponse qu'*Apache* encode automatiquement avec **Transfer-Encoding: chunked** :

- Si utilisé avec d'autres mécanismes, doit être en dernier et appliqué une seule fois,
- Composé de suite de blocs comprenant :
 - la taille du bloc codée en hexadécimal suivi de paramètres optionnels et du retour à la ligne '\r\n',
 - les données du bloc de la taille codée ci-dessus et de '\r\n',
- Le dernier bloc est de taille 0.

■

```
HTTP/1.1 200 OK
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/plain
```

```
23
This is the data in the first chunk
1C
and this is the second one
```

```
3
con
8
sequence
0
```

11/53

15/10/2013



Protocole OAI-PMH



- Protocole fondé par l'*Open Archives Initiative* pour échanger des métadonnées.
- Il permet de constituer et de mettre à jour automatiquement des entrepôts centralisés où les métadonnées de sources diverses peuvent être interrogées.
- Utilisé notamment par les Archives Ouvertes et les entrepôts institutionnels ou les bibliothèques.
- Version 2 en juin 2002.

12/53

15/10/2013



Protocole OAI-PMH : exemples d'entrepôts



- CiteseerX - 1,6 Millions de références informatique et sciences de l'information - gestion des citations.
- Pubmed Central - 2,6 millions de références médecine et biologie.
- XTCAT - 1,5 millions de références de thèses.
- ARXIV - plus de 1 million de références scientifiques - maths, physique, ...
- BNF - plus de 10 millions d'oeuvres.

13/53

15/10/2013



Protocole OAI-PMH : syntaxe



- `oai?verb=val` où `val` vaut :
 - GetRecord
 - Identify
 - ListIdentifiers
 - ListMetadataFormats
 - ListRecords
 - ListSets
- `from=date&until=date&set=filtre&metadataPrefix=oai_dc&resumptionToken=code` où `code` permet d'avoir le bloc suivant filtré par la date du premier paquet pour ListIdentifiers et ListRecords.
- Gestion des erreurs

14/53

15/10/2013

Architecture d'une base avec le Web

15/10/2013



Mécanisme entre serveur de base et serveur Web



- Un client communique avec XediX en HTTP 1.0 ou 1.1 via un serveur Web (comme Apache)
 - Par les méthodes GET, HEAD, POST ou autres
 - Supporte le HTTP Range pour le non XML
 - Permet de demander un extrait de document en spécifiant une liste de plages
 - Range: bytes=10-20,30-
 - Le type au sens du standard MIME est spécifié par l'en-tête Content-Type: standard du HTTP

15/53

15/10/2013



Mécanisme entre serveur de base et serveur Web (II)



- Entre XediX et serveur Web, au choix :
 - Binaire au standard *CGI*
 - Module xedix pour Apache (**mod_xedix**) fourni avec XediX plus performant car
 - supporte des connexions persistantes vers le serveur XediX (transparent pour les utilisateurs)
 - Elimine le lancement d'un processus à chaque connexion
 - Permet à *XediX* de suivre les règles de duplication des processus Apache
 - Une instance (processus) Apache pour une instance de XediX.
 - Paramètres Apache **MAXCLIENTS** et autres importants (les défauts sont pas mal pour une machine type 4 coeurs, 4 Go MEM)

16/53

15/10/2013



Mécanisme entre serveur de base et serveur Web (III)



- Plusieurs serveurs XediX possibles par base
 - En pratique peu utilisé
 - On en lance un qui se duplique ou non sur option
- Les serveurs XediX se dupliquent fonction des besoins
 - Chaque serveur se duplique (fork)
 - Suit donc les forks d'Apache avec le module XediX pour Apache (mod_xedix)
 - Connection socket persitantes entre XediX (ou autres bases) et Apache
 - Entre client et Apache pas forcément persistante
 - Car l'en-tête **Content-Length** n'est pas forcément envoyé. Faire une requête HTTP 1.1 au lieu de 1.0 pour avoir cette notion de socket persistante.
 - Voir standard HTTP et la notion de **Transfer-Encoding: chunked** vu plus haut.

17/53

15/10/2013



Mémoire virtuelle



- Les systèmes d'exploitation actuels comme Linux réservent en mémoire virtuelle.
- Les pages de mémoires sont allouées en mémoire physique lorsqu'on y accède (pages partagées en lecture et dupliquées en lecture-écriture).
- La duplication de processus avec **fork** commence par réserver en mémoire virtuelle autant de mémoire que le processus à dupliquer même si cette dernière n'est pas utilisée dans le processus fils créé.
- D'où la notion d'*overcommit* qui permet de réserver plus de mémoire virtuelle qu'il n'y a de mémoire physique (RAM + swap).
- Mais ça ne suffit pas forcément, donc :
 - Qui doit au moins être réservée en mémoire sur disque dur (swap).
 - Donc pensez à déclarer de la mémoire sur disque pour des configs de machines supportant des serveurs Web associés à des serveurs de bases de données.
 - Même si cette mémoire n'est pas vraiment utilisée (on ne "racle" pas forcément le disque) à cause des mécanismes de sur-réservation de la mémoire virtuelle sur les systèmes actuels.
- Voir infos sur Linux et la notion d'*overcommit* et de mémoire virtuelle sur Internet.

18/53

15/10/2013



Mémoire et fragmentation



- La mémoire RAM de l'ordinateur peut-être fragmentée.
- Certains *malloc/free* sont meilleurs de ce point de vue et diminuent aussi le besoin en *overcommit*.
- Exemple de *jemalloc* (utilisé dans *FreeBSD* et *Firefox*) à comparer à *glibc*.
- Sur disque, pour un système de fichier ou une base de données, la notion de fragmentation existe aussi, en particulier sur les index.
- Les *index inverses* de XediX se défragmentent partiellement à la volée même si on indexe document par document.
- Une base de donnée a un *taux d'expansion* par rapport aux données "brutes" (en CSV ou XML).
- Ce taux dépend du nombre et de la nature des index, ainsi que de l'âge de la base (lié au nombre d'ajouts/modifications dans la base).

19/53

15/10/2013



Compression et bases de données



- On peut avoir un système de fichier qui comprime à la volée.
- Utilisé dans certaines bases :
 - Machine Oracle - deux niveaux de compression pour données proches et lointaine.
 - Option sur Oracle 11g.
 - Compression des données et sauvegardes sur SQL Serveur (Microsoft).
 - Autres...
- XediX comprime sur options définies dans le fichier de config par base.
 - La décompression a un débit plus rapide que le débit du disque dur.
 - On peut tirer partie du format XML pour améliorer les performances de la compression (notion de patron).
 - Comprime environ de 4 à 6 fois les index de champs pour **CLIOCOMPACT=5** et **ZLIBCBEST=14**.
 - Technique type *LZ77* et *Huffman*.

20/53

15/10/2013

URLs de XediX

15/10/2013



URLs de XediX



- Les URLs de XediX peuvent se générer dans le langage de son choix pourvu que l'on sache ouvrir une connexion de type socket et communiquer avec le protocole standard HTTP à travers cette socket.
- Ceci peut-être fait côté client (navigateur Web et applications du poste de travail) ou serveur avec, par exemple, les langages :
 - Javascript, HTML, CSS, XSLT côté client
 - PHP, Java, C, C++, modules Apache en C côté serveur
- La forme générale des URLs de XediX est :
 - de la forme **http://serveur:port/cgi-bin**
/client?racine+nufonc+user,session+params avec :
 - **Racine** est une chaîne de caractères caractérisant la nature de l'action demandée (liste de documents, document, recherche,...) ce qui permet un classement des types d'URLs
 - **Nufonc** est le numéro de fonction précisant la nature de l'action demandée
 - **Params** est de la forme **motcle1=valeur1&motcle2=valeur2&...** sont les paramètres des fonctions - ils varient suivant **Racine** et **Nufonc**
 - **User** est le nom de l'utilisateur qui s'est connecté à la base et a donc ouvert une session
 - **Session** est la clé de session générée après connexion réussie de l'utilisateur dans la base

21/55

15/10/2013



URLs de XediX (II)



- Si l'URL est transmise par la méthode *HTTP GET* et si params contient \$0 à la fin
 - Les paramètres suivants sont lus avec la méthode *HTTP POST* et concaténés à l'URL avant traitement et envoi à XediX (par le module où le CGI)
 - Exemple :
 - X2Accueil+0+clef+beginid=10
 - X2Accueil+0+clef+\$0
 - Beginid=10 en *POST* (HTML FORM)
 - Voir avec IHM par défaut de XediX

22/53

15/10/2013



URLs de XediX (III)



- Les *racines* de XediX classent les fonctions de XediX en familles de fonctions. Les *racines* existantes pour XediX sont :
 - **X2Admin** : administration des utilisateurs et de leurs droits,
 - **X2Import** : alimentation (importation et destruction de données) de la base,
 - **X2Documents** : accès à un document ou une portion de document en XML ou HTML,
 - **X2Imgelems** : Accès aux documents non XML (comme des vidéos par exemple),
 - **X2Accueil** : liste globale de documents ou concaténation d'extraits de documents,
 - **X2Xsearch** : recherche de documents ou d'extraits de document éventuellement concaténés à partir des indexs de mots et de champs de XediX.

23/53

15/10/2013

Racine X2Admin

15/10/2013



Racine X2Admin



■ X2Admin

- Nufonc=0
 - Création d'une nouvelle session utilisateur (connexion) avec réponse en HTML
- Nufonc=1
 - Panneau principal en HTML des opérations d'administration de la base
- Nufonc=2
 - Déconnecte l'utilisateur et renvoie le panneau HTML de connexion à la base
- Nufonc=3
 - Panneau HTML pour un utilisateur

24/53

15/10/2013



Racine X2Admin (II)



- Nufonc=4
 - Panneau HTML pour un groupe d'utilisateur
- Nufonc=5
 - Panneau HTML pour un groupe de documents
- Nufonc=6
 - Supprime un utilisateur
- Nufonc=7
 - Supprime un groupe d'utilisateur
- Nufonc=8
 - Supprime un groupe de documents

25/53

15/10/2013



Racine X2Admin (III)



- Nufonc=9
 - Ajoute ou modifie un utilisateur
- Nufonc=10
 - Ajoute ou modifie un groupe d'utilisateurs
- Nufonc=11
 - Ajoute ou modifie un groupe de documents
- Nufonc=13
 - Création d'une nouvelle session utilisateur (connexion) avec réponse en XML
- Nufonc=14
 - Déconnecte l'utilisateur (renvoie OK en XML)

26/53

15/10/2013



Racine X2Admin (IV)



- Nufonc=15
 - Ajoute ou modifie un utilisateur et renvoie OK en XML (identique à nufonc=9 et paramètre output=xml)
- Nufonc=16
 - Etat des connexions en mode métaserveur

27/53

15/10/2013

Racine X2Import

15/10/2013



Racine X2Import



- X2Import
 - Nufonc=0
 - Suppression de documents
 - Nufonc=1
 - Relecture de références de documents (synchronisation des serveurs)
 - Nufonc=2
 - Envoi d'un ordre de relecture de documents (appelé automatiquement)
 - Nufonc=3
 - Panneau d'ajouts asynchrone de documents (de type spooler ou gestionnaire d'import)

28/53

15/10/2013



Racine X2Import (II)



- Nufonc=4
 - Upload de documents et soumission de travaux d'import sur ceux-ci
- Nufonc=5
 - Destruction de répertoires (cas) contenant des documents uploadés
- Nufonc=6
 - Création d'un répertoire (cas) dans lequel on pourra uploader des documents
- Nufonc=7
 - Panneau de gestion des logs des travaux d'imports de documents

29/53

15/10/2013



Racine X2Import (III)



- Nufonc=8
 - Purge d'un travail soumis au gestionnaire d'import
- Nufonc=9
 - Affichage des logs liées à un travail donné
- Nufonc=10
 - Purge des logs liées à un travail donné
- Nufonc=11
 - Affichage d'une log
- Nufonc=12
 - Ajout ou modification synchrone d'un document XML
(sans passer par le gestionnaire d'import)

30/53

15/10/2013



Racine X2Import (IV)



- Nufonc=13
 - Panneau de saisie ou de modification d'un document XML qui appelle X2Import+12 comme traitement

31/53

15/10/2013

Racine X2Documents

15/10/2013



Racine X2Documents



- Paramètre **name=iddoc** pour le nom du document
- Autre paramètres dans la documentation technique
- **X2Documents**
 - **Nufonc=0**
 - Affichage page de garde d'un document en HTML
 - **Nufonc=1**
 - Affichage du document entier en HTML
 - **Nufonc=2**
 - Affichage d'un document entier en HTML avec mise en évidence des mots trouvés par le moteur de recherche (REX ou un autre si existant)

32/53

15/10/2013



Racine X2Documents (II)



- Nufonc=3
 - Affichage du résumé d'un document en HTML
 - Nufonc=4
 - Affichage de deux frames HTML pour naviguer dans un document
 - Nufonc=5
 - Affichage en HTML des annexes d'un document
 - Nufonc=6
 - Affichage en HTML de la liste de diffusion d'un document
- 33/53 15/10/2013



Racine X2Documents (III)



- Nufonc=7
 - Affichage en HTML de la table des matières cliquable d'un document
- Nufonc=8
 - Affichage en HTML d'une division d'un document correspondant à un titre de la table des matières
- Nufonc=9
 - Affichage du document en XML
- Nufonc=10
 - Affichage du document en XML avec mise en évidence des mots trouvés par le moteur de recherche (REX ou autre)

34/53

15/10/2013



Racine X2Documents (IV)



- Nufonc=11
 - Affichage en HTML du contenu correspondant à une référence interne du document
- Nufonc=12
 - Affichage en HTML de la section des métadonnées du document
- Nufonc=13
 - Affichage de la DTD du document
- Nufonc=14
 - Affichage en HTML du document sous forme simplifiée
- Nufonc=15
 - Affichage en HTML de la liste des références internes du document

35/53

15/10/2013

Racine X2Imgelems

15/10/2013



Racine X2Imgelems



- Supporte l'en-tête HTTP Range pour spécifier des sous-ensembles de ressources non XML.
 - Utile aux lecteurs audio-vidéo ou autres logiciels de lecture par exemple qui supportent nativement cet en-tête.
- X2Imgelems
 - Nufonc=0
 - Affichage en HTML d'une image seule avec sa légende
 - Nufonc=1
 - Affichage de l'image elle-même avec son type au standard MIME fourni dans l'en-tête HTTP Content-Type

36/53

15/10/2013



Racine X2Imgelems (II)



- Nufonc=2
 - Affichage de l'image elle-même avec son type au standard MIME enacspulée en HTML/Java pour zoom éventuel
- Nufonc=3
 - Affichage d'une page numérisée correspondant au document XML (si existe associée au document) avec navigation
- Nufonc=4
 - Panneau impression de pages numérisées d'un document

37/53

15/10/2013



Racine X2Imgelems (III)



- Nufonc=5
 - Ordre d'impression de pages numérisées d'un document
- Nufonc=6
 - Envoie sous formes de playliste paramétrable des URLs du lien
- Nufonc=7
 - Sauvegarde sur le serveur des pages numérisées d'un document (pour l'administrateur)
- Nufonc=8
 - Affichage de frames HTML pour les pages numérisées

38/53

15/10/2013



Racine X2Imgelems (IV)



- Nufonc=9
 - Navigation pour les pages numérisées
- Nufonc=10
 - Affichage d'une page numérisée seule correspondant au document XML
- Nufonc=11
 - Logo de la base

39/53

15/10/2013

Racine X2Accueil

15/10/2013



Racine X2Accueil



- Le paramètre **targetcoll=(liste|documents|...)** contrôle le type de sortie
- Le paramètre **filtreid=expr** permet le filtrage par expression régulière sur les identifiants
- Le paramètre **elems=selecteur** génère des liens vers les documents qui ont ce paramètre ou extrait les portions le vérifiant
- **X2Accueil**
 - **Nufonc=0**
 - Affichage de la liste de documents par paquets ou de frames si targetcoll=liste

40/53

15/10/2013



Racine X2Accueil (II)



- Nufonc=1
 - Liste XML des identifiants des documents
- Nufonc=2
 - Une page de la liste de documents en HTML encapsulé dans du XML (utilisé par le métaserveur)
- Nufonc=3
 - Paramètres de contrôle pour mode targtecoll=liste
- Nufonc=4
 - Concaténation d'extraits de documents

41/53

15/10/2013

Racine X2Xsearch

15/10/2013



Racine X2Xsearch



- Paramètres comme **X2Accueil**
- Paramètre **allrequest=requete** pour la requête REX elle-même
- **X2Xsearch**
 - **Nufonc=0**
 - Affichage du formulaire général de saisie d'une requête vers REX (moteur de recherche de XediX)
 - **Nufonc=1**
 - Résultat d'une requête vers REX par paquets ou frames si targetcoll=liste
 - **Nufonc=2**
 - Liste des balises donnée par l'index de REX

42/53

15/10/2013



Racine X2Xsearch (II)



- Nufonc=3
 - Affichage en XML de la liste des ids des documents et de la position des mots trouvés pour les documents répondants à allrequest=requete
- Nufonc=14
 - Affichage en XML de la liste des ids des documents et de la position des mots trouvés pour les documents répondants à allrequest=requete ainsi que les mots ou les champs correspondants

43/53

15/10/2013



Racine X2Xsearch (III)



- Nufonc=4
 - Affichage par paquets du résultat de la requête en HTML encapsulé dans du XML (appelé par le métaserveur)
- Nufonc=5
 - Affichage en XML de la liste des balises via l'index REX
- Nufonc=6
 - Paramètres de contrôle pour mode targtecoll=liste
- Nufonc=7
 - Concaténation d'extraits de documents en réponse à la requête REX

44/53

15/10/2013



Racine X2Xsearch (IV)



- Nufonc=8
 - Formulaire de saisie d'une requête dans le champ de type mot clés (paramétrable fonction de la DTD) des documents
- Nufonc=9
 - Formulaire de saisie d'une requête dans le champ de type légendes d'images (paramétrable fonction de la DTD) des documents
- Nufonc=10
 - Liste en HTML des mots dans la base via l'index REX

45/53

15/10/2013



Racine X2Xsearch (V)



- Nufonc=11
 - Liste en XML des mots dans la base via l'index REX
- Nufonc=12
 - Liste en HTML des champs dans la base via l'index REX
- Nufonc=13
 - Liste en XML des champs dans la base via l'index REX

46/53

15/10/2013

Exemple de requêtes HTTP

15/10/2013



Exemple de requêtes HTTP



- Fichier HTML sur le serveur Web avec en-tête Range

```
GET /atos/html/accueil_7.html HTTP/1.1
Host: homere:5225
Range: bytes=0-20
```

- Demande d'une URL XediX avec en-tête Range (car URL qui renvoie un binaire)

```
GET /cgi-bin/client?X2Imgelems+11 HTTP/1.1
Host: homere:5225
Range: bytes=0-20
```

- Demande d'une URL XediX avec en-tête Range (car URL qui renvoie un binaire)

```
GET /cgi-bin/client?X2Imgelems+1+admin,16228428651679283010+name=elect30000&img_rang=0
Host: homere:5225
Host: homere:5225
```

47/53

15/10/2013



Exemple de requêtes HTTP (II)



- Avec la méthode HTTP POST :

POST /cgi-bin/client?X2Imgelems+1+admin,16228428651679283010+\$0 HT

Content-Length: 15

Content-Type: application/x-www-form-urlencoded

name=elect30000

48/53

15/10/2013



Exemple de requêtes HTTP (III)



■ Avec la méthode POST en multipart (pour upload de fichiers)

```
POST /cgi-bin/client?X2import+4+admin,16228428651679283010+$0 HTTP/1.0
Content-Type: multipart/form-data; boundary=16838575810113
Content-Length: xxxx
```

```
--16838575810113
Content-Disposition: form-data; name="filtre"
```

```
rempli_base
--16838575810113
Content-Disposition: form-data; name="traite"
```

```
yes
--16838575810113
Content-Disposition: form-data; name="nomcas"
```

```
XEDIX_admin_20071023_164146_13471
--16838575810113
Content-Disposition: form-data; name="ficimport"; filename="notes605.sgm"
Content-Type: application/octet-stream
```

```
<!DOCTYPE DOCCEA PUBLIC "CEA/DIF//DTD Rapports 1.4//EN">
<!--ArborText, Inc., 1988-1998, v.4002-->
<?Pub Inc>
<doccea fontfam="helvetica" fonttai="11" espace="grand" legende="global" pagegardes="n
<fmt></fmt>
--16838575810113--
```

49/53

15/10/2013

Exemples en Javascript de génération d'URLs

15/10/2013



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Concepts</title>
<!-- Changed by: Pierre Brochard, 20-Oct-2008 -->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link rel="stylesheet" href="../styles/panneau_ecran.css" title="Default" media="screen" />
<script language="JavaScript" type="text/javascript" src="../../scripts/xapi.js"></script>
<noscript>
<p align="center">Veuillez activer Javascript dans votre navigateur.</p>
</noscript>
</head>
<body bgcolor="#FFFFFF" background="../../Images/fond.jpg">
<ul>
<li><b>Liens vers volcanisme</b>
```

15/10/2013



Exemples en Javascript de génération d'URLs (II)



■ Le code javascript :

```
// Fonction xsearch (liste réponse à une requête sur l'index de Verity des mots
// des documents de la base)
function xsearch(nufonc,target,request,targetcoll,elems,filtreid,begin,output,tableau,
    dispdocs,display,tri)
{
    var som2_loc = top.dir_cgi + '/client?X2Search+' + nufonc + '+';

    som2_loc += top.login + ',';
    som2_loc += top.clefsession + '+request=' + request;
    som2_loc += '&allrequest=' + request + '&begin=' + begin;
    som2_loc += '&targetcoll=' + targetcoll;
    som2_loc += '&elems=' + elems + '&filtreid=' + filtreid;
    som2_loc += '&output=' + output + '&tableau=' + tableau;
    som2_loc += '&dispdocs=' + dispdocs + '&display=' + display;
    som2_loc += '&tri=' + tri;
    target.location = som2_loc;
}
```

51/53

15/10/2013

Exemples avec une base XediX

15/10/2013



Exemples avec une base XediX



- 45 Go sur le portable
- IHM par défaut permet de gérer les utilisateurs et illustre toutes les URLs d'accès au serveur XediX
 - Panneau d'accès à la liste globale des documents
 - Panneaux de saisie d'une requête au moteur de recherche
 - Type de réponses
 - Formattage des documents
 - En HTML
 - Gestion table des matières et liens internes à un document (au sens SGML/XML)
 - En XML

52/53

15/10/2013

TD

15/10/2013



TD



- Poursuite des Tds précédent du module
- Ecrivez vos panneaux de requêtes dédiées en HTML + Javascript + CSS
 - Ajouts à web/concepts/xedix.html
 - Liste des noms (sélecteur à faire) formatage par défaut
 - Liste des noms sous forme de tableau (+XSLT+XHTML+CSS)
 - Idem pour fiches qui répondent à une requête qui demande un mot et une expression régulière sur un champ à choisir.

53/53

15/10/2013