
rfc2696

Extension de contrôle pour la manipulation des résultats paginés

Ce contrôle permet à un client de contrôler la quantité de résultat retourné par le serveur lors d'une opération de recherche. Utile lorsqu'un client a des ressources limitées et peut ne pas être en mesure de traiter le résultat entièrement, ou lorsqu'un client est connecté sur une connexion à faible bande passante.

Ce contrôle est inclus dans les messages **searchRequest** et **searchResultDone**.

La structure de ce contrôle est la suivante :

```
pagedResultsControl ::= SEQUENCE {
    controlType 1.2.840.113556.1.4.319,
    criticality BOOLEAN DEFAULT FALSE,
    controlValue searchControlValue
}
```

searchControlValue est une chaîne encodée BER de la séquence suivante :

```
realSearchControlValue ::= SEQUENCE {
    size INTEGER (0..maxInt),
    - requested page size from client
    - result set size estimate from server
    cookie OCTET STRING
}
```

Intéraction client-serveur

Un client peut spécifier un **pagedResultsControl** avec la taille de page et un cookie vide. La taille de page peut être supérieur à 0 et inférieur à **sizeLimit**.

Si la taille de page est supérieur ou égale à **sizeLimit**, le serveur devrait ignorer ce contrôle. Si le serveur ne supporte pas ce contrôle le serveur doit retourner une erreur **unsupportedCriticalExtension** si le client a positionné **criticality**, sinon le serveur devrait ignorer le contrôle.

Chaque fois que le serveur retourne un jeu de résultat au client, le serveur inclus le contrôle **unsupportedCriticalExtension** dans le message **searchResultDone**. Dans le contrôle retourné, la taille peut être définie à l'estimation du serveur du nombre total d'entrées à retourner. Les serveurs qui ne peuvent pas l'estimer devraient retourner 0. Le cookie doit être mis à une valeur vide s'il n'y a plus d'entrée à retourner.

Le client doit considérer le cookie comme opaque et ne s'occupe pas de son contenu. Quand le client veut récupérer plus d'entrées, il doit envoyer une **searchRequest** avec toutes les valeurs identiques à la requête initiale à l'exception du cookie, et optionnellement une taille de page modifiée. Le cookie à fournir dans la requête est le dernier cookie reçu.

Une séquence est abandonnée par une requête qui contient **pagedResultsControl** de 0 et le dernier cookie retourné par le client. Un client peut utiliser l'opération **abandon** pour abandonner la recherche, mais ce n'est pas encouragé et peut invalider le cookie du client.

Si le serveur ne peut pas continuer une opération paginée, il devrait retourner l'erreur appropriée dans une entrée **searchResultDone**, le client et le serveur devraient considérer le jeu de résultat terminé et non-récupérable.

Exemples

le client envoie une requête avec une résultat paginé à 3 :

C : SearchRequest + pagedResultsControl(3, "")

Le serveur répond avec 3 entrées et une indication du nombre total d'entrées (5) dans le résultat et un cookie à utiliser pour la suite

S : SearchResultEntry

S : SearchResultEntry

S : SearchResultEntry

S : SearchResultDone + pagedResultsControl(5, "opaque")

Le client envoie la même requête en changeant simplement le cookie, demande la page suivante :

C : SearchRequest + PagedResultsControl(3, "opaque")

Le serveur répond avec le 2 dernière entrées en indiquant qu'il n'y a plus d'entrées :

S : SearchResultEntry

S : SearchResultEntry

S : SearchResultDone + pagedResultsControl(5, "")