
rfc2891

Extension de contrôle pour le trie des résultats de recherche coté serveur

Ces contrôles permettent à un client de spécifier les attributs et règles de correspondance qu'un serveur devrait utiliser pour retourner les résultats. Ces contrôles peuvent être utiles quand un client a des fonctionnalités limitées ou ne peuvent pas trier les résultats, mais nécessite que le résultat soit trié. Les contrôles de trie permettent de retourner un code de résultat pour le trie qui est indépendant du code de résultat retourné.

Requête

controleType vaut "1.2.840.113556.1.4.473"

controlValue vaut :

```
SortKeyList ::= SEQUENCE OF SEQUENCE
{ attributeType AttributeDescription,
  orderingRule [0] MatchingRuleId OPTIONAL,
  reverseOrder [1] BOOLEAN DEFAULT FALSE }
```

SortListKey est une séquence ordonnée pour le trie.

MatchingRuleId devrait être un valide pour le type d'attribut auquel il s'applique.

Réponse

Ce contrôle est inclus dans le message `searchResultDone`.

```
SortResult ::= SEQUENCE {
  sortResult ENUMERATED {
    success (0), -- results are sorted
    operationsError (1), -- server internal failure
    timeLimitExceeded (3), -- timelimit reached before sorting was completed
    strongAuthRequired (8), -- refused to return sorted results via insecure protocol
    adminLimitExceeded (11), -- too many matching entries for the server to sort
    noSuchAttribute (16), -- unrecognized attribute type in sort key
    inappropriateMatching (18), -- unrecognized or inappropriate matching rule in sort key
    insufficientAccessRights (50), -- refused to return sorted results to this client
    busy (51), -- too busy to process
    unwillingToPerform (53), -- unable to sort
    other (80)
  },
  attributeType [0] AttributeDescription OPTIONAL }
```

Interaction Client-Serveur

sortKeyRequestControl spécifie un ou plusieurs types d'attributs et de règles de correspondance pour les résultats retournés par une recherche.

Le serveur devrait retourner tous les résultats dans l'ordre spécifié. Si **reverseOrder** est à TRUE le résultat sera inversé.

Il y'a 6 scénarios possibles qui peuvent se produire en résultat au contrôle de trie incluse dans la requête :

- 1- Si le serveur ne supporte pas ce contrôle et que le client met le champs **criticality**, le serveur doit retourner **unavailableCriticalExtension**.
- 2- Si le serveur ne supporte pas ce contrôle et que **criticality** est FALSE, le serveur doit ignorer le contrôle et traiter la requête.
- 3- Si le serveur supporte ce contrôle mais ne peut pas trier le résultat avec les clés spécifiées et que le champs **criticality** est mis, le serveur devrait retourner **unavailableCriticalExtension**
- 4- Si le serveur supporte ce contrôle mais ne peut pas trier le résultat avec les clés spécifiées et que le champs **criticality** est FALSE, le serveur devrait retourner tous les résultats non triés. et inclure **sortKeyResponseControl** dans **searchResultDone**.
- 5- Si le serveur supporte ce contrôle et peut trier le résultat en utilisant les clés spécifiées, il devrait inclure le **sortKeyResponseControl** dans **searchResultDone** avec **sortResult** à **success**.
- 6- Si la recherche échoue ou qu'il n'y a pas de résultat le serveur devrait omettre le **sortKeyResponseControl** du **searchResultDone**.

Le client est assuré que le résultat est trié uniquement si le code de résultat dans **sortKeyResponseControl** est à **success**.

Si un client demande un trie à un serveur qui chaîne la requête et récupère des entrées depuis plusieurs serveurs, et que **criticality** est positionné, le serveur doit fusionner le résultat avant de les retourner au client ou doit retourner **unwillingToPerform**

Quand **sortKeyRequestControl** est utilisé avec **pagedResultsControl** , le serveur devrait envoyer les messages **searchResultEntry** triés avec les clés appliqués au résultat total, et non sur les pages. **sortKeyList** doit être présent sur chaque message **searchRequest** pour le résultat paginé. Il ne doit pas changer entre les **searchRequests** pour le même résultat.