

Användarhandledning för avnsSearch

Datum skapad: 2013-10-11

Datum uppdaterad: 2016-09-19

Version: 1.1

Dokumentkontroll

Bilagor

Bilaga	Namn	Sökväg
1	RESTletGuide.pdf	
2	Search.pdf	
3		
4		
5		
6		
7		
8		

Ändringskontroll

Datum	Namn	Version	Kommentar
2012-10-11		1.0	Dokument skapat
2016-09-19		1.1	Reviderat

Innehåll

Dokumentkontroll.....	2
Bilagor.....	2
Ändringskontroll.....	2
Granskare	Error! Bookmark not defined.
Distribution.....	Error! Bookmark not defined.
Introduktion.....	4
Sammanfattning.....	4
REST funktionen.....	5
Header (icke token baserad)	5
Body.....	6
Uthämtning av stort antal poster.....	8
Parametrar i body.....	10
Operatorer för filter	11
Resultat.....	12

Introduktion

Sammanfattning

Detta dokument beskriver hur funktionen AVNS search är implementerat i affärssystemet samt hur funktionen skall användas. Funktionens uppgift är att läsa ut affärsdata ur Netsuite via en så kallad REST funktion. Huvudkomponenterna i systemet är sparade sökningar och ett skript som definierar hur REST funktionen fungerar. En sparad sökning definierar vilken posttyp som skall läsas ut och kan, men måste inte, innehålla filter för att begränsa urvalet. En sökning beskriver också vilka fält som skall utläsas ur en post.

REST anropet kan innehålla filter som skall läggas till sökningen.

REST funktionen

REST funktioner är ett mycket flexibelt REST-baserat API för att kommunicera med Netsuite. Funktionerna kan skräddarsys för att anpassas exakt efter kundens behov. Http anropet består av två delar, dels en header och dels en body. Innehållets format i bodyn är i JSON format, vilket måste avspeglas i headern där man anger att Content-Type är application/json.

För att testa REST-anrop används med fördel gratisprogrammet Postman eller Fiddler. Fiddler finns endast för Windows men enligt uppgift går det att använda i OS X om man använder Wine, annars kan man använda en App i webbläsaren Chrome som heter "Advanced REST Client Application" (ARC). ARC är dock långsamt om många poster ska utläsas.

Header (icke token baserad)

Anropet av REST funktionen sker med ett http POST anrop där autentiseringen sker via autentiseringsuppgifter i http headern.

För att finna den korrekta webbadressen till funktionen, gå till Anpassning/Skriptkörning/Produktionssätt skript. Högst upp väljer du filtret "RESTlet". Leta rätt på avnsSearch välj visa. Adressen finns i fältet EXTERN WEBBADRESS.

Header information som skall finnas med är: Host: rest.netsuite.com

Authorization: NLAAuth nlauth_account=aaa, nlauth_email= zzz,nlauth_signature= xxxx, nlauth_role=yyy

Content-Type: application/json; charset=UTF-8

Content-Length: nnn

Observera att Authorization måste skrivas exakt som ovan för att inte få fel i autentiseringen (se fiddler exemplet i Figur 1). I Headern skall account, e-postadressen, xxxx och rollen bytas ut mot korrekta värden. Content-Length är antalet tecken i body-texten.

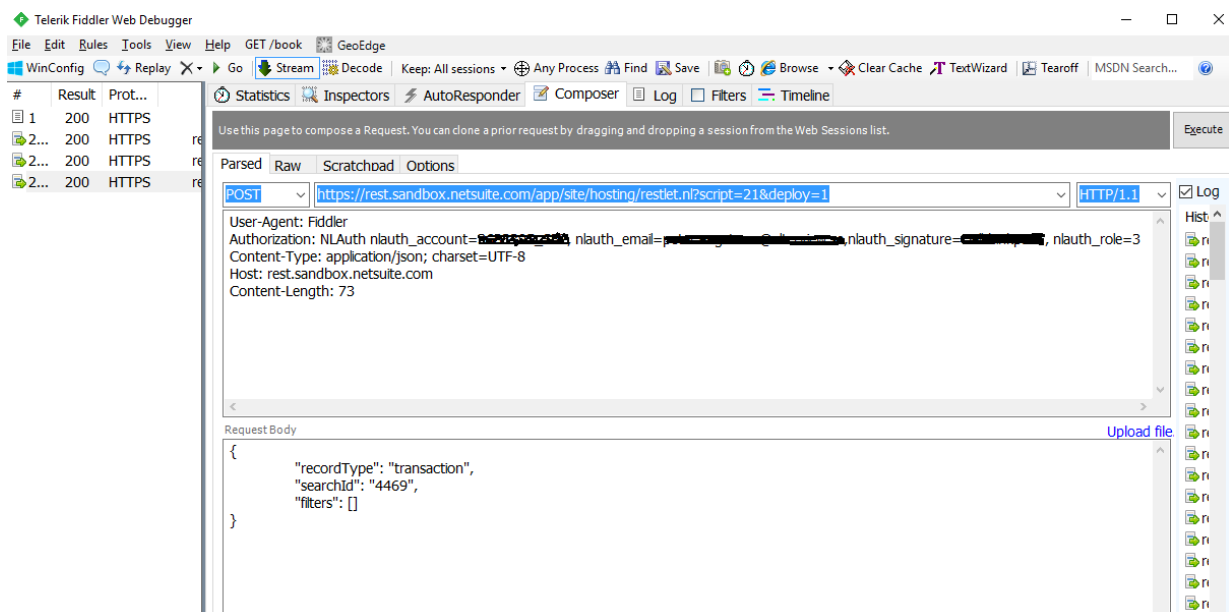
Body

Exempel 1: Body för en sökning i sin enklaste form (inga extra filter) kan se ut så här

```
{
  "recordType": "transaction",
  "searchId": "customsearch_my_first_search",
  "filters": []
}
```

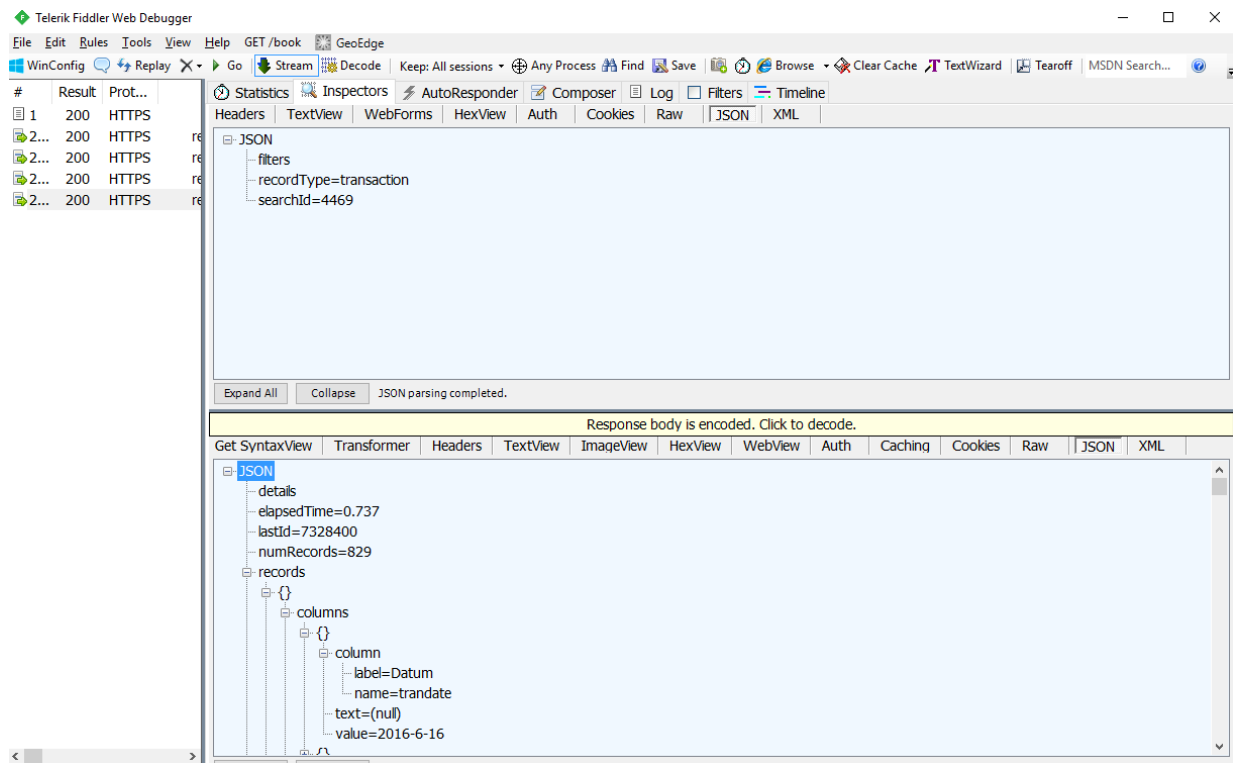
I exempel 1 är den sparade sökningen "customsearch_my_first_search" en enkel sökning på transaktioner och innehåller inga filter. I body:n för anropet så är "recordType" satt till transaction. Då detta är samma post typ, så kommer alla typer av transaktioner att returneras. Om vi bara ville ha fakturor så kunde vi satt "recordType" i body till "invoice" utan att ändra den sparade sökningen. Eftersom body inte heller innehåller några filter så kommer vi att få maximalt 10000 poster. Hur man får ut valbart antal poster kommer vi till senare i detta dokument.

I Fiddler kan anropet se ut så här:



Figur 1:Fiddler anrop

Svaret från Netsuite för ovanstående anrop:



Exempel 2: En body med ett filter:

```
{  "recordType": "transaction",  "searchId": "customsearch_my_first_search",  "filters": [    [ {      "name": "trandate",      "operator": "after",      "value1": "2016-01-01"    } ]  ]}
```

Som synes så kan man använda antingen id-numret eller id för att ange searchId.

Varken den sparade sökningen eller body behöver innehålla några filter. De filter som man anger i body adderas till de eventuella filter som finns i den sparade sökningen, d v s de filter man anger i body byter inte ut befintliga filter.

Observera att i sökningen ovan som innehåller datum, så måste datumformatet vara samma som det datumformat som angivits i inställningarna för användaren som gör anropet. Användaren är den e-postadress som man angav i header:ns Authorization.

Exempel 3: En body med två filter som tar ut poster med ett debitvärde större än 2090 och bokförda efter ett visst datum. maxRecords talar om att vi vill ha max 20 poster.

```
{
  "recordType": "transaction",
  "searchId": "customsearch_my_first_search",
  "maxRecords": "20",
  "filters": [
    [
      {
        "name": "trandate",
        "operator": "after",
        "value1": "2016-01-01"
      }
    ], "AND", [
      {
        "name": "debitamount",
        "operator": "greaterthan",
        "value1": "2090"
      }
    ]
  ]
}
```

Uthämtning av stort antal poster

För att göra uthämtning av många poster eller poster som i sig innehåller stora mängder data så finns parametrar för att styra storleken på svaret från en sökning. Parametern "maxRecords" begränsar antalet returnerade poster men passar inte alltid då det kan finnas fler poster än vad som angetts i parametern. I dessa fall så använder man istället "page" och "pageSize".

"pageSize" anger hur många poster man vill utläsa (i det exempel som visas nedan så är storleken satt till 100, om man utelämnar denna parameter så är storleken förvald till 1000) för varje sida. "page" anger vilken sida vi vill hämta ut (observera att sidnumreringen startar på noll).

Exempel 4: Uthämtning av data, sida för sida.

```
{
  "recordType": "transaction",
  "searchId": "customsearch_my_first_search",
  "page": 0,
  "pageSize": 100,
  "filters": [
    [
      {
        "name": "trandate",
        "operator": "between",
        "value1": "2010-01-01",
        "value2": "2016-09-09"
      }
    ]
  ]
}
```

För att på detta sätt hämta ut alla poster som finns i sökningen så ökar man "page" med ett för varje anrop, tills man inte får några träffar i svaret d v s "numRecords" är noll.

Exempel 5: I det här exemplet så används en formel för att hämta alla poster vars moms inte är i hela kronor.

```
{
  "recordType": "transaction",
  "searchId": "customsearch_my_first_search",
  "page": 0,
  "pageSize": 20,
  "filters": [
    [
      {
        "name": "formulanumeric:{taxamount}-TRUNC({taxamount},0)",
        "operator": "greaterthan",
        "value1": "0"
      }
    ]
  ]
}
```

Parametrar i body

Parameter	Kommentar
recordType	Posttyp. Giltiga värden finns beskrivet i Netsuite Record Browser, där Interna Id kolumnen anger giltigt värde. I detta fall skall den vara "transaction"
Searchid	Sökningens internalid. I detta fall "customsearch_my_first_search" eller "4469"
maxRecords	Giltiga värden > 1. Förval om denna inte anges är 10000. Om man använder parametern "page" har "maxRecords" ingen betydelse.
pageSize	Används när man vill dela upp resultatet av sökningen i sidor. Giltiga värden är 1 - 4000. Förvald storlek om man inte anger parametern är 1000.
page	Här anger man vilken sida man vill komma åt i sökresultatet. 0 är första sidan. För att hämta ut alla träffar så ökar man page för varje anrop tills man i svaret får numRecords = 0. När denna parameter används så har maxRecords ingen betydelse.
skiplabel	Anger om man vill lämna etiketter tomma i resultatet. Giltiga värden "T"/"F"
skiptext	Anger om man vill lämna textuella värden tomma i resultatet. Giltiga värden "T"/"F"
skipname	Anger om man vill låta namn på fält vara tomma i resultatet. Giltiga värden "T"/"F"
Filters	Extra filter att applicera på sökningen. Denna funktion kan användas för ytterligare filtrera ner sökresultatet. Syntaxen för att skapa filterobjekt följer Netsuites dokumentation på setFilterExpression förutom att varje filter i array:en inte i sig är en array utan är ett javascript objekt med namngivna parametrar: name, operator och value1/value2.

Operatorer för filter

För att veta vilka operatorer som är tillåtna i filter så sök på "Search Operators" i Netuite Help Center. Det finns även relativa datumbeteckningar som kan användas i "value1"/"value2" för sökning på datum, t ex "today" och "fourDaysAgo". Dessa finner man genom att söka på "Search Date Filters" i Netuite Help Center.

Resultat

Parameter	Kommentar
elapsedTime	Anger total processtid på server i sekunder
numRecords	Antal returnerade poster
lastId	Anger internalid på sista utlästa posten.
status	OK eller ERROR. Om ERROR kommer array:en details innehålla ett eller flera meddelanden om vad som orsakat felet
details	Array med felmedelanden
records	<p>Innehåller en array av returnerade poster. Varje post innehåller 3 parametrar (fet stil):</p> <p>Id: Internalid för transaktionen</p> <p>recordType: Typ av transaktion. Giltiga värden finns beskrivet i Netsuite Help under rubriken "SuiteScript Supported Records"</p> <p>columns: Array av kolumner. Varje kolumn har följande parametrar (fet kursiv stil):</p> <p><i>value:</i> Värdet på kolumnen</p> <p><i>text:</i> Textuellt värde på kolumnen(speciellt för fält av List/Record typ). Kan stängas av genom att sätta "skiptext" till "T"</p> <p><i>column:</i> Ett objekt med två parametrar(kursiv understruken stil):</p> <p><u><i>name:</i></u> Namn på kolumnen. Kan stängas av genom att sätta "skipname" till "T"</p> <p><u><i>label:</i></u> Kolumnrubrik. Kan stängas av genom att sätta "skiplabel" till "T"</p>