# Kom godt igang med ElasticSearch

Når du har fået adgang til CVR data (dvs. du har fået udleveret dit brugerId og password) kan du begynde at fremsøge CVR data fra indekset i form af JSON dokumenter. Men inden du går igang, vil vi ikke undlade, at gøre opmærksom på, at ElasticSearch produktet både har en glimrende [guide](https://www.elastic.co/guide/en/elasticsearch/guide/1.x/index.html) og ligeledes en omfattende [reference dokumentation](https://www.elastic.co/guide/en/elasticsearch/reference/1.7/index.html).

Du vil med fordel kunne læse guiden, inden du evt. forsøger dig frem, idet du hermed får en bedre forståelse for, hvad der er muligt, hvilket igen kan være med til at inspirere dig til lave en bedre løsning, her forstået som en løsning, der arbejder hurtigt, og som udnytter ElasticSearch på en optimal måde. Hvis du laver en løsning der ikke er optimal, i forhold til vores drift af ElasticSearch, kan du komme ud for, at vi midlertidigt bliver nødt til at spærre for din adgang.

Du har kun fået adgang til at fremsøge data via RESTful API’et, så du kan koncentrere dig om at forstå ElasticSearch forespørgsler, dvs. de forskellige typer af forespørgsler og det tilhørende forespørgsels-sprog (Query DSL).

For tiden anvender vi ElasticSearch v.1.7.4, men vil senere opgradere til en nyere version af Elastic-Search. Idet der er ”breaking changes” mellem 1.x serien og 2.x serien og også den kommende 5.x serie, planlægger vi en periode med parallel drift, således at alle vores brugere vil få tid til, at omlægge deres løsninger til den nyere version. Det er klart, at det vil lette overgangen, hvis man på forhånd sætter sig ind i forskellene, således at man så vidt muligt kun anvender forespørgsler, der vil fungere korrekt på tværs af ElasticSearch versionerne.

Som ekstern bruger har man som sagt kun adgang til at fremsøge data, hvilket via RESTful API’et gøres ved at sende enten en http GET eller en POST request til /indeksnavn/type/\_search endepunktet på frontend serveren (<http://distribution.virk.dk>) der sørger for at sende requestet videre til ElasticSearch, vente på svaret og returnere det til brugeren.

Man kan lave simple forespørgsler via GET request parametre, men det anbefales, at man bruger ElasticSearch query DSL sproget, som sendes i body af request’et (både GET og POST er understøttet). Query DSL giver mange flere muligheder, ligesom man enkelt kan begrænse størrelsen af svaret man modtager. Der jo ikke nogen grund til at modtage alle registrerede data om en virksomhed (også historiske), hvis man kun har behov for virksomhedens aktuelle navn.

Eksempler på forespørgsler vil i det efterfølgende blive vist på følgende formel:

POST /cvr-permanent/virksomhed/\_search

{

"\_source":["Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn"],

"query":{"term":{"Vrvirksomhed.cvrNummer":"12345678"}}

}

Betydningen af ovenstående er: søg efter en virksomhed med cvr nr. 12345678 og returner virksomhedens aktuelle navn. Hvis der ikke findes et virksomhed type dokument, der opfylder søgebetingelserne, får man følgende svar:

{"took":1,"timed\_out":false,"\_shards":{"total":3,"successful":3,"failed":0},"hits":{"total":0,"max\_score":null,"hits":[]}}

Hvor det interessante her er sektionen: "hits":{"total":0, …}, der viser hvor mange dokumenter i alt, der opfylder søgekriteriet - her naturligvis ingen.

Hvis man selv vil afprøve nogen af de viste eksempler, kan det f.eks. gøres med kommando linje værktøjet curl eller man kan bruge et browser plugin som f.eks. postman, det vigtige er blot at værktøjet understøtter, at man kan afsende et http request med en Authorization header og et request body der indeholder forespørgslen. Ovenstående eksempel vil med curl f.eks. se ud som følger (vi bruger cvr nr. 30714024, så vi får et svar):

curl -XPOST -u 'BrugerId:Password' http://distribution.virk.dk/cvr-permanent/virksomhed/\_search -d '

{

"\_source":["Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn"],

"query":{"term":{"Vrvirksomhed.cvrNummer":"30714024"}}

}' | python -mjson.tool

I ovenstående sender vi svaret gennem python JSON fortolkeren via en pipe, så vi får et pænt formateret resultat:

{

"\_shards": {

"failed": 0,

"successful": 3,

"total": 3

},

"hits": {

"hits": [

{

"\_id": "4001646987",

"\_index": "cvr-permanent-prod-20161007",

"\_score": 17.506895,

"\_source": {

"Vrvirksomhed": {

"virksomhedMetadata": {

"nyesteNavn": {

"navn": "NINECONSULT A/S"

}

}

}

},

"\_type": "virksomhed"

}

],

"max\_score": 17.506895,

"total": 1

},

"timed\_out": false,

"took": 3

}

Som viser os at vi – som forventet - har fundet et resultat ("total": 1) og vi har kun fået returneret virksomhedens navn: "NINECONSULT A/S" præcis som vi bad om. Prøv evt. at gentage forespørgslen, men denne gang uden linien: "\_source":["Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn"], så du kan se alle data, der er registreret om virksomheden.

Når vi ser på hvordan de udstillede data anvendes af vores brugere, er der to hovedscenarier der går igen.

Det ene er, hvor man typisk ønsker at lave simple valideringer af dataregistreringer i eget system, hvor man derfor f.eks. ønsker at validere et indtastet cvr nr. og vise det tilhørende virksomhedsnavn. Det kan også være hjælp til at finde et cvr nr. på basis af et indtastet virksomhedsnavn, eller en del af et indtastet virksomhedsnavn.

For begge eksempler har man oftest kun behov for at validere i forhold til de aktuelle virksomhedsdata, og man vil derfor med fordel kunne søge i "Vrvirksomhed.virksomhedMetadata" sektionen i virksomhed typen i ElasticSearch cvr-permanent indekset (som i eksemplet ovenfor).

Til den omvendte situation – find et cvr nr. på basis af et virksomhedsnavn eller en del af et virksomhedsnavn kan man f.eks. anvende denne søgning:

POST /cvr-permanent/virksomhed/\_search

{

"\_source":[

"Vrvirksomhed.cvrNummer",

"Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn"

],

"query":{

"query\_string":{

"default\_field":"Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn",

"query":"parken AND sport"

}

},

"size":20

}

Hvilket vil returnere følgende svar (det var i hvert fald tilfældet, da dette blev skrevet):

{

"\_shards": {

"failed": 0,

"successful": 3,

"total": 3

},

"hits": {

"hits": [

{

"\_id": "2834786",

"\_index": "cvr-permanent-prod-20161007",

"\_score": 8.383249,

"\_source": {

"Vrvirksomhed": {

"cvrNummer": 15107707,

"virksomhedMetadata": {

"nyesteNavn": {

"navn": "PARKEN SPORT & ENTERTAINMENT A/S"

}

}

}

},

"\_type": "virksomhed"

}

],

"max\_score": 8.383249,

"total": 1

},

"timed\_out": false,

"took": 4

}

Her skal man forestille sig at brugeren har indtastet ”parken sport”, hvilket i ”query” delen udtrykkes som begge ord skal findes (AND) i attributten ”Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn”. Det er også muligt at anvende wild-card søgninger med query\_string operatoren. Ovenstående viser også at det var tilstrækkeligt for brugeren at indtaste ”parken sport”, idet der kun blev fundet et resultat, der opfyldte søgekriteriet. En anden ting man skal bemærke er ”size” attributten i forespørgslen, der beskriver hvor mange dokumenter, man maksimalt vil have i resultatet (standardværdien er 10), og nej, man kan ikke få alle dokumenter tilbage i en søgning, ved at specificere et tilstækkeligt stort tal.

Dette bringer os naturligt til det andet typiske hovedscenarie: man ønsker af forskellige grunde, at have f.eks. et udsnit af alle virksomhedsdata liggende i lokal kopi. Det betyder så også, at man skal lave en søgning, der giver en mulighed for at hente alle de ønskede data, og at man med mellemrum skal fremsøge ændringer, i hvert fald hvis man vil holde sin egen lokale kopi opdateret.

Til dette formål skal man anvende en scan / scroll søgning, hvor ElasticSearch returnerer data i passende klumper, svarende til ”size” attributten gange med antallet af primære shards i indekset (i skrivende stund 3), så hvis man anfører en størrelse på 200, vil man modtage 200 \* 3 = 600 dokumenter ad gangen, hvilket er en passende størrelse. ElasticSearch læser data transaktionelt, således at man vil kunne gemme starttidspunktet for scan / scroll søgningen, som efterfølgende vil kunne anvendes som starttidspunkt, når man senere skal fremsøge ændringer, så man kan vedligeholde den lokale kopi.

Hvis ens lokale register f.eks. indeholder cvr nr., aktuelt navn og adresse, kan man udføre følgende scan / scroll søgning:

POST /cvr-permanent/virksomhed/\_search?search\_type=scan&scroll=1m

{

"\_source":[

"Vrvirksomhed.cvrNummer",

"Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn",

"Vrvirksomhed.virksomhedMetadata.nyesteBeliggenhedsadresse"

],

"query":{"match\_all":{}},

"size":200

}

Resultatet af denne første søgning er et scroll\_id, der sendes med til den næste søgning (vist herunder), som i sin tur returnerer et nyt scrool\_id, samt første klump af de ønskede data. Inden udløbet af scroll perioden (i eksemplet ovenfor 1m = 1 minut), skal man have lavet det næste kald igen, med det senest modtagne scroll\_id. Dvs. sålænge man bliver ved med at kalde indenfor scroll perioden holder ElasticSearch læse transaktionen aktiv. Man bliver ved med at kalde indtil man ikke længere modtager nogen hits, dvs. indtil hits array er tomt.

Eksempel på resultat af første søgning (der indeholder første version af scroll\_id):

{

"\_scroll\_id": "c2NhbjszOzEyMTk2NDcyOkd6ZVlKRlVOVHF5UkkwNVZLM2Rhb0E7MTIxMTEzNTU6ei1JaXRWWXhTenF5SElUTTFHRk9OdzsxMjE5NjQ3MzpHemVZSkZVTlRxeVJJMDVWSzNkYW9BOzE7dG90YWxfaGl0czoxNTY2Mjk3Ow==",

"\_shards": {

"failed": 0,

"successful": 3,

"total": 3

},

"hits": {

"hits": [],

"max\_score": 0.0,

"total": 1566297

},

"timed\_out": false,

"took": 19

}

De efterfølgende søgninger, der skal udføres har alle formen (scroll\_id fremsendes i body (en lang linie)):

GET /\_search/scroll?scroll=1m

c2NhbjszOzEyMTk2NDcyOkd6ZVlKRlVOVHF5UkkwNVZLM2Rhb0E7MTIxMTEzNTU6ei1JaXRWWXhTenF5SElUTTFHRk9OdzsxMjE5NjQ3MzpHemVZSkZVTlRxeVJJMDVWSzNkYW9BOzE7dG90YWxfaGl0czoxNTY2Mjk3Ow==

Sålænge ens scan / scroll transaktion er aktiv vil kaldet returnerer det næste scroll\_id og den næste klump data (eksempel):

{

"\_scroll\_id": "c2NhbjszOzEyMTk2NDcyOkd6ZVlKRlVOVHF5UkkwNVZLM2Rhb0E7MTIxMTEzNTU6ei1JaXRWWXhTenF5SElUTTFHRk9OdzsxMjE5NjQ3MzpHemVZSkZVTlRxeVJJMDVWSzNkYW9BOzE7dG90YWxfaGl0czoxNTY2Mjk3Ow==",

"\_shards": {

"failed": 0,

"successful": 3,

"total": 3

},

"hits": {

"hits": [

{

"\_id": "5282642",

"\_index": "cvr-permanent-prod-20161007",

"\_score": 0.0,

"\_source": {

"Vrvirksomhed": {

"cvrNummer": 32329675,

"virksomhedMetadata": {

"nyesteBeliggenhedsadresse": {

"bogstavFra": "B",

"bogstavTil": null,

"bynavn": null,

"conavn": null,

"etage": "st",

"fritekst": null,

"husnummerFra": 51,

"husnummerTil": null,

"kommune": {

"kommuneKode": 101,

"kommuneNavn": "K\u00d8BENHAVN",

"periode": {

"gyldigFra": "2007-01-01",

"gyldigTil": null

},

"sidstOpdateret": "2006-11-13T00:00:00.000+01:00"

},

"landekode": "DK",

"periode": {

"gyldigFra": "2015-04-24",

"gyldigTil": "2015-06-26"

},

"postboks": null,

"postdistrikt": "K\u00f8benhavn K",

"postnummer": 1253,

"sidedoer": null,

"sidstOpdateret": "2015-06-28T17:00:22.000+02:00",

"vejkode": 7540,

"vejnavn": "Toldbodgade"

},

"nyesteNavn": {

"navn": "CLAUS MEYER KORN ApS"

}

}

}

},

"\_type": "virksomhed"

},

.

. (her er der klippet 598 virksomheder ud af resultatet)

.,

{

"\_id": "4001550321",

"\_index": "cvr-permanent-prod-20161007",

"\_score": 0.0,

"\_source": {

"Vrvirksomhed": {

"cvrNummer": 29499772,

"virksomhedMetadata": {

"nyesteBeliggenhedsadresse": {

"bogstavFra": "D",

"bogstavTil": null,

"bynavn": null,

"conavn": null,

"etage": null,

"fritekst": null,

"husnummerFra": 13,

"husnummerTil": null,

"kommune": {

"kommuneKode": 461,

"kommuneNavn": "ODENSE",

"periode": {

"gyldigFra": "2007-01-01",

"gyldigTil": null

},

"sidstOpdateret": "2006-11-12T21:00:00.000+01:00"

},

"landekode": "DK",

"periode": {

"gyldigFra": "2009-07-10",

"gyldigTil": "2010-02-24"

},

"postboks": null,

"postdistrikt": "Odense C",

"postnummer": 5000,

"sidedoer": null,

"sidstOpdateret": "2013-11-22T19:44:48.000+01:00",

"vejkode": 9378,

"vejnavn": "Vilhelminevej"

},

"nyesteNavn": {

"navn": "Christoffersen VVS v/Jesper Christoffersen"

}

}

}

},

"\_type": "virksomhed"

}

],

"max\_score": 0.0,

"total": 1566297

},

"timed\_out": false,

"took": 50

}

Scan / scroll eksempel forespørgslen henter alle virksomheder (der bruges en match\_all query), så man får data om alle virksomheder også de historiske. Hvis man kun ønsker data om aktive virksomheder, kan man i stedet bruge følgende query udtryk:

POST /cvr-permanent/virksomhed/\_search?search\_type=scan&scroll=1m

{

"\_source":[

"Vrvirksomhed.cvrNummer",

"Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn",

"Vrvirksomhed.virksomhedMetadata.nyesteBeliggenhedsadresse"

],

"query":{

"constant\_score":{

"filter":{

"missing":{

"field":"Vrvirksomhed.livsforloeb.periode.gyldigTil"

}

}

}

},"size":200

}

Af ovenstående kan man slutte, at så længe en virksomhed har et livsforløb med en åben periode (gyldigTil mangler), er den at betragte som aktiv.

Hermed er der ikke sagt noget om virksomhedens status, om den er f.eks. er normal eller underkonkurs (se Vrvirksomhed.virksomhedMetadata.sammensatStatus der indeholder den aktuelle værdi).

Når man skal hente opdateringer og nyoprettelser til sin lokale kopi, skal man igen lave en scan / scroll søgning, hvor man skal fremsøge de virksomheder, der har et Vrvirksomhed.sidstOpdateret tidspunkt, som er større end eller lig med det tidspunkt, man gemte fra sidst man opdaterede. Den kan f.eks. se således ud:

POST /cvr-permanent/virksomhed/\_search?search\_type=scan&scroll=1m

{

"\_source":[

"Vrvirksomhed.cvrNummer",

"Vrvirksomhed.virksomhedMetadata.nyesteNavn.navn",

"Vrvirksomhed.virksomhedMetadata.nyesteBeliggenhedsadresse"

],

"query":{

"range":{"Vrvirksomhed.sidstOpdateret":{"gte":"2016-11-20T13:23:31.000"}}

},

"size":200

}

Her skal man blot være opmærksom på, at selv om virksomheden er blevet opdateret, er det ikke sikkert, at det vedrører det udsnit af data, man har i sin lokale kopi, så ovenstående fremsøger sikkert for meget data, men til gengæld er det et simpelt søgeudtryk.

Det er naturligvis muligt kun at fremsøge f.eks. ændrede navne og/eller adresser, men det vil føre vidt i forhold til dette dokuments formål. Skulle man være interesseret skal man læse op på nested query, således at man får udformet sin forespørgsel korrekt.

De viste eksempler har alle taget udgangspunkt i virksomhed typen, der også indeholder referencer til produktionsenheder, som man kan finde alle detaljer om ved at søge i produktionsenhed typen på VrproduktionsEnhed.pNummer. Tilsvarende indeholder virksomhed typen også informationer om deltagerne, men også her gælder det, at den fulde information om en deltager findes i deltager typen ved at søge på Vrdeltager.enhedsNummer.