



# Projektbeskrivning: analys av jobannonser med Arbetsförmedlingens öppna API:er

## Syfte och avgränsning

Det här delprojektet syftar till att bygga en datamängd som gör det möjligt att analysera **antal annonser över tid per yrkesklassificering (SSYK)** och **geografi** samt att beskriva vilka arbetsgivare och vilka kompetenser som förekommer i annonserna. Du kommer att integrera det här i ett större projekt och därför behöver datateknikern som tar över att få tydliga instruktioner kring vilka öppna API:er som ska användas, hur de anropas och vilka fält som behövs.

Flera API:er från Arbetsförmedlingens JobTech-plattform kombineras:

- **Historiska platsannonser** – innehåller annonser som varit publicerade i Platsbanken och som har tagits ned. Detta API bygger på samma sökmodell som JobSearch och innehåller annonser från 2016 och framåt. Varje annons är berikad med kompetenser och distansarbeteinformation <sup>1</sup>. API:et kräver ingen API-nyckel och används för att hämta tidsserier av tidigare annonser.
- **JobStream** – realtidsström över alla annonser som är **öppna för ansökan**. Endpoints `/snapshot` ger en fullständig ögonblicksbild av alla öppna annonser och `/stream` ger löpande uppdateringar med nya/uppdaterade/borttagna annonser. Tidsstämplar och filter gör det möjligt att hämta alla annonser efter ett givet klockslag eller mellan två tidpunkter <sup>2</sup>. Här behövs en API-nyckel.
- **JobSearch** – sök-API för annonser som är öppna för ansökan och lämpar sig för interaktiva sökningar. Endpoints `/search` för fritext och filtersökning, `/complete` för autocomplete, `/ad` och `/ad/{id}/logo` <sup>3</sup>. Det stöder filter för yrken, arbetsgivare, geografi och tidsstämplar. API:et är inte tänkt för att ladda ned alla annonser; för bulk-nedladdning används JobStream <sup>4</sup>. API-nyckel krävs.
- **Taxonomy-API** – referensdata över yrkesbenämningar, kompetenser, geografiska koder m.m. Det finns endpoints för att hämta koncept (t.ex. SSYK-nivåer) och relationer mellan dem; exempelvis kan du hämta alla SSYK-nivåer 1–3 med en begäran som anger flera typer <sup>5</sup>. GraphQL-gränssnittet används för att hämta hierarkier som kopplar yrkesbenämningar till SSYK-koder och för att hämta regioner/kommuner <sup>6</sup>. API:et kräver inte nyckel.
- **JobAd Enrichments** – AI-baserat API som extraherar kompetenser, mjuka färdigheter, yrkestitlar och geografiska platser ur en jobbannons. Enrichments-API:et returnerar termer och sannolikheter för om termen verkligen efterfrågas. Huvudendpoints är `/enrichtextdocuments` (alla termer med sannolikhetsvärden), `/enrichtextdocumentsbinary` (endast termer över ett tröskelvärde) och `/synonymdictionary` <sup>7</sup>. API:et ger ett synonymlexikon och används för att automatiskt extrahera strukturerad kompetensdata ur annonsens text. Kontakta JobTech innan produktionsanvändning <sup>8</sup>.

## Registrering av API-nyckel

JobStream och JobSearch kräver en API-nyckel. En nyckel registreras via <https://apirequest.jobtechdev.se> (informationen finns i JobSearch-dokumentationen <sup>9</sup>). Taxonomy och Historiska annonser kräver **ingen** nyckel <sup>10</sup>.

## Identifiera yrken och geografi (Taxonomy-API)

För att räkna annonser per SSYK-nivå och geografi måste du kunna översätta filtreringsparametrar (occupation-name, occupation-group osv.) till concept-ID:n. Den nya taxonomin använder unika **concept-id** snarare än SSYK-koder, men du kan hämta SSYK-koder och relationer:

- **Hämta alla SSYK-nivåer:**

```
GET https://taxonomy.api.jobtechdev.se/v1/taxonomy/specific/concepts/ssyk?  
type=ssyk-level-1%20ssyk-level-2%20ssyk-level-3%20ssyk-level-4
```

Detta returnerar alla SSYK-koder med metadata <sup>11</sup>.

- **Bygga hierarki yrke → SSYK → yrkesområde:** Använd </v1/taxonomy/main/graph> med [edge-relation-type=broaden](#) för att koppla samman concept-typer. Exempelvis kan du hämta relationer från yrkesbenämning till SSYK-nivå 4 och sedan till yrkesområde <sup>12</sup>.
- **Hämta regioner och kommuner:** GraphQL-endpoint </v1/taxonomy/graphql> gör det möjligt att hämta alla regioner (län) och deras kommuner i en enda förfrågan. Dokumentationen visar ett exempel där man utgår från ett id för Sverige och hämtar regioner och deras kommuner <sup>13</sup>. Geografiska filtreringsparametrar i JobSearch/JobStream använder concept-id för [country](#), [region](#) och [municipality](#) <sup>14</sup>.

## Historiska platsannonser

Historiska annonser används för att analysera **tidstrender** och för att bygga statistiska tidsserier. API-dokumentationen finns inte i GitLab, men ett forskningsarbete (Appelberg 2025) visar hur gränsnittet används. Förfrågan görs mot <https://historical.api.jobtechdev.se/search> med följande parametrar <sup>15</sup>:

Parameter	Beskrivning
<a href="#">published-before</a>	Slutdatum (ISO-tidstämpel) – annonser publicerade <b>före</b> detta datum returneras <sup>15</sup> .
<a href="#">published-after</a>	Startdatum (ISO-tidstämpel) – annonser publicerade <b>efter</b> detta datum returneras <sup>15</sup> .

Parameter	Beskrivning
occupation-name	Concept-ID för yrkesbenämningen du vill studera <sup>16</sup> . Concept-ID hittas via Taxonomy-API eller Atlas (webbgränssnitt).
offset	Paginering; startposition för resultaten (0 ger första sidan) <sup>17</sup> .
limit	Antal annonser som returneras per sida <sup>17</sup> .
request-timeout	Frivillig parameter (sekunder) för hur länge servern får arbeta.

Exempel från avhandlingen:

```
https://historical.api.jobtechdev.se/search?published-before=2024-12-31T00:00:00&published-after=2024-01-01T00:00:00&occupation-name=n9RX_nnz_ZYF&offset=0&limit=10&request-timeout=300
```

Den här anropet hämtar annonser för concept-id `n9RX_nnz_ZYF` (skolbibliotekarie) mellan 1 jan 2024 och 31 dec 2024. För att hämta nästa sida ökar du `offset` med samma värde som `limit` <sup>18</sup>.

Parametrarna för historiska annonser liknar JobSearch (fri text och filter). Exempelvis finns även `municipality`, `region`, `occupation-field`, `occupation-group` etc., men de dokumenteras i Swagger (behöver API-nyckel). Varje historisk annons är berikad med **kompetenser och distansarbete**, vilket gör dem lämpliga för kompetensanalys <sup>1</sup>.

## JobStream – hämta alla öppna annonser

JobStream används för att bygga en **lokal databas** med alla annonser som är öppna just nu och för att uppdatera denna kontinuerligt. Viktiga endpoints <sup>19</sup>:

Endpoint	Syfte	Parametrar
/snapshot	Returnerar alla öppna annonser som en stor JSON-fil (~300 MB). Den används för att bygga startdatabasen <sup>20</sup> .	Inga parametrar (men API-nyckel behövs).
/stream	Returnerar förändringar (nya annonser, uppdateringar, borttagningar) efter en given tidpunkt. Kräver <code>date</code> som anger starttid (ISO-format). Man kan ange <code>updated-before-date</code> för att avgränsa ett tidsintervall <sup>21</sup> .	<code>date</code> (obligatorisk starttid), <code>updated-before-date</code> (slut), <code>occupation-concept-id</code> (filtrera på yrke), <code>location-concept-id</code> (filtrera på geografi) <sup>22</sup> .

Vid realtidsuppdatering anropar du `/stream` en gång per minut och använder tidsstämpeln från förra anropet som `date`. Du kan filtrera på flera koncept-ID för geografi eller yrke; flera filter kombineras med **AND** (både yrke och geografi måste uppfyllas) <sup>23</sup>.

## Fält i JobStream/JobSearch

Både JobStream och JobSearch returnerar annonsobjekt som JSON. Modellbeskrivningen finns i Swagger, men följande fält är särskilt viktiga:

- `id` – unikt annons-ID (kan användas med `/ad/{id}`)
- `headline` och `description.text` – titel och beskrivning
- `published` och `last_application_date` – publicerings- och sista ansökningsdatum
- `employer.name` och `employer.organisation_number` – arbetsgivarens namn och organisationsnummer <sup>24</sup>
- `occupation_group` / `occupation_name` – concept-ID för yrkesgrupp respektive benämning
- `workplace_address.coordinates` – latitud/längdut för platsen <sup>25</sup>
- `remote` – information om distansarbete (finns i berikade historiska annonser)

## JobSearch – sök annonser

JobSearch är ett sök-API för annonser som är öppna för ansökan. Det används för att hämta annonser efter specifika filter snarare än för att ladda ned allt. Du behöver en API-nyckel. Några viktiga filter:

Parameter	Beskrivning
<code>q</code>	Fritextsökning i rubrik och beskrivning <sup>26</sup> . Jokertecken ( <code>*</code> ) stöds för prefixsökningar <sup>27</sup> ; frassökning görs med citattecken och URL-kodade tecken <sup>28</sup> .
<code>occupation-name</code> / <code>occupation-field</code> / <code>occupation-group</code>	Filtrering på yrke via concept-ID från Taxonomy <sup>29</sup> ; <code>occupation-field</code> motsvarar yrkesområde.
<code>employer</code>	Organisationsnummer eller prefix för att hämta alla annonser från en viss arbetsgivare <sup>24</sup> .
<code>country</code> , <code>region</code> , <code>municipality</code>	Concept-ID för geografi. Om flera geografiska filter anges kommer den mest lokala platsen att prioriteras <sup>30</sup> .
<code>offset</code> , <code>limit</code>	Paginering; <code>offset</code> + <code>limit</code> definierar vilken sida som returneras.
<code>date-published-from</code> och <code>date-published-to</code>	Begränsar tidsintervall (finns i Swagger).
<code>x-fields</code> header	Du kan i HTTP-headern <code>X-Fields</code> ange vilka fält som ska returneras för att minimera datamängden <sup>31</sup> .

För enkla autocomplete-funktioner finns `/complete?q=` som returnerar vanliga termer i annonser <sup>32</sup>. Vid massnedladdning ska du **inte** använda JobSearch utan JobStream <sup>4</sup>.

## Exempel – filtrera annonser på en arbetsgivare i en region

```
GET https://jobsearch.api.jobtechdev.se/search?  
employer=2021002114&region=9hXe_F4g_eTG&q=java
```

Anropet hämtar Java-jobb för arbetsgivare med organisationsnummer som börjar på `2021002114` i Norrbottens län (region-ID `9hXe_F4g_eTG`) <sup>33</sup>.

## JobAd Enrichments – extrahera kompetenser

Det här API:et används för att analysera själva annonstexten. Du skickar in råtexten och får tillbaka strukturerad information om yrkestitlar, kompetenser och mjuka färdigheter samt sannolikheten för att termen efterfrågas. Exempel på endpoints <sup>34</sup>:

Endpoint	Syfte	Viktiga parametrar
<code>/enrichtextdocuments</code>	Returnerar alla identifierade termer med ett sannolikhetsvärde (0 – 1). Högre värde betyder att termen sannolikt är ett krav <sup>35</sup> .	Body: JSON-lista med dokument där varje dokument innehåller <code>id</code> och <code>text</code> .
<code>/enrichtextdocumentsbinary</code>	Returnerar endast termer som överstiger ett tröskelvärde.	Body: <code>classification-threshold</code> (t.ex. 0.7), <code>id</code> , <code>text</code> <sup>36</sup> .
<code>/synonymdictionary</code>	Returnerar synonymer som används i API:et; används för att mappa synonymer till ett gemensamt koncept <sup>37</sup> .	

API:et kan hantera singular/plural, missstavningar och sammansatta ord <sup>38</sup>. Utdata innehåller en lista över termer, deras typ (yrkestitel, kompetens, mjuk färdighet, plats) och en sannolikhet för om termen är efterfrågad <sup>39</sup>. Detta gör det möjligt att skapa frekvenslistor över efterfrågade kompetenser eller mjuka färdigheter.

## Projektets datamodell och bearbetning

### 1. Hämta och lagra data

- 1. Bygg baskartotek för SSYK och geografi:**
- Anropa Taxonomy-API:t för att hämta alla SSYK-nivåer och deras relationer. Spara även region- och kommunlistor via GraphQL. Dessa tabeller gör att du kan mappa concept-ID från annonserna till SSYK-kod och geografinamn.
- 3. Ladda ned historiska annonser:**
- Bestäm ett datumintervall (t.ex. från 2016-01-01 till idag) och för varje SSYK-nivå eller yrkesområde hämta annonser via Historiska-API:t. Använd `published-after` / `published-before` och concept-ID för `occupation-name` eller `occupation-field` för att filtrera <sup>15</sup>. Loop:a över `offset` tills alla annonser är hämtade (100 annonser per sida). Spara relevanta fält (se ovan).

5. Ladda ned aktuella annonser:
6. För att analysera nuläget, skapa en lokal databas med hjälp av `/snapshot` från JobStream. Kör sedan ett script som ropar `/stream?date=<timestamp>` var minut/timme för att uppdatera databasen med nya eller ändrade annonser <sup>21</sup>.

## 2. Bearbeta data för statistik

- **Tidsserier per SSYK och geografi:** Använd publiceringsdatumet (`published`) från varje annons och mappa `occupation_name` eller `occupation_group` till SSYK-kod. Aggregera antalet annonser per vecka eller månad och per SSYK-nivå och region/kommun. För historiska data finns endast annonser som har avpublicerats, men de ger ändå en god bild av efterfrågan.
- **Topparbetsgivare:** Gruppera annonser per `employer.organisation_number` eller `employer.name` och räkna antalet annonser. Kombinera med geografi för att se vilka arbetsgivare som domineras i olika regioner.
- **Distansarbete:** Använd fältet `remote` (historiska annonser) för att räkna hur många annonser som erbjuder distansarbete över tid.
- **Kompetenstreder:** Kör annonsernas texter genom JobAd Enrichments. Fokusera på termtyperna `skill` och `trait`. Räkna hur ofta olika kompetenser förekommer per år och per SSYK-grupp för att upptäcka trender.

## 3. Visualisering och kartor

- **Kartdata:** Annonserna innehåller `workplace_address.coordinates` (latitud/längditud) och concept-ID för region/kommun. Kombinera detta med geodata (t.ex. shapefiler över svenska regioner/kommuner) för att skapa interaktiva kartor där varje polygon färgas efter antal annonser per period.
- **Tidsseriediagram:** Visualisera antal annonser per SSYK-grupp över tid för att se trender. Du kan även visa andelen distansarbete över tid.
- **Stapel/bubbeldiagram för arbetsgivare:** Rangordna arbetsgivare efter antal annonser; visa per region.

## Vidare läsning och resurser

- **Swagger / API-dokumentation:** För JobSearch, JobStream och Taxonomy finns Swagger-gränssnitt; de visar alla parametrar och svarsfält. Använd dem aktivt under utveckling: <https://jobsearch.api.jobtechdev.se>, <https://jobstream.api.jobtechdev.se>, <https://taxonomy.api.jobtechdev.se>.
- **Atlas:** Ett interaktivt verktyg för att bläddra i taxonomin och hitta concept-id (<https://atlas.jobtechdev.se>).
- **GitHub-repos:** Jobtech Swe har exempel på hur API:erna används (sök efter `getting-started-code-examples`).

## Slutsats

Genom att kombinera **historiska annonser**, **JobStream/JobSearch** och **Taxonomy** kan du bygga en rik datamängd för analys av den svenska arbetsmarknaden. Använd Taxonomy-API för att mappa yrken och geografi till koncept och SSYK-koder, JobStream för att hålla en aktuell databas med öppna annonser,

Historiska API för tidsserier bakåt i tiden och JobAd Enrichments för att extrahera kompetensinformation. Med dessa data kan du skapa tidsserier, kartor och arbetsgivarrankningar som ger insikt i var efterfrågan finns och vilka kompetenser som efterfrågas.

---

1 Öppna data från Arbetsförmedlingen

<https://data.arbetsformedlingen.se/>

2 19 20 21 22 23 Jobstream - Prenumerera på platsannonser - JobTech Dev

<https://arbetsformedlingen.gitlab.io/www/docs/docs/api/jobstream/>

3 4 9 14 24 25 26 27 28 29 30 31 32 33 Jobsearch - Sök platsannonser - JobTech Dev

<https://arbetsformedlingen.gitlab.io/www/docs/docs/api/jobsearch/>

5 6 11 12 13 Taxonomy - referensdata som yrken, kompetenser mm - JobTech Dev

<https://arbetsformedlingen.gitlab.io/www/docs/docs/api/taxonomy/>

7 8 34 35 36 37 38 39 raw.githubusercontent.com

<https://raw.githubusercontent.com/Jobtechdev-content/JobAdEnrichments-content/master/>

GettingstartedJobAdEnrichmentsSE.md

10 Driftproblem - JobTech Dev

<https://arbetsformedlingen.gitlab.io/www/docs/docs/driftproblem/log/>

15 16 17 18 FULLTEXT01.pdf

<https://www.diva-portal.org/smash/get/diva2:1993349/FULLTEXT01.pdf>