Data Model - brainstorming

This is a brainstorming document; I'll simply jot down what I think could be valuable for our data model.

Ressources

- Guide to data modelling with MongoDB
- UML Data Modelling
  Our example will be simpler I guess.

My thoughts

A t first I thought we should get the data from different sources, transform them and only after the transformation store them in the database.
Maybe I was wrong here and we could benefit if we store the data from each source first in the database, transform and save again afterwards. I think we are mainly interested in the uniform data model (which includes all your different sources) but maybe in the future we change our minds about this uniform structure and in this case we would have
no raw data from the sources and would be forced to get all the data again (in some caes this may be not possible at all, who knows).

So we could make a seperate collection for all data sources.

So far we have 3 data sources, I try to add new ones before our meeting on Wedenesday, I promis at least one new source :)

Data Sources

This is a description of the data we have collected so far. Can be found in data/processed/< name of source >

Muse.com

- **job_title**: name of the job, should not be NULL
- **min_salary**: value can be :"NOT_FOUND" which equals NULL
- **max_salary**: value can be :"NOT_FOUND" which equals NULL
- **currency**: value can be :"NOT_FOUND" which equals NULL
- **skills**: value can be [] => empty list which equals NULL
- **publication_date**: when the offer was published, should not be NULL
- **id**: every data I save contains an id, this id is distinct for the data source
- **location**: list that contains somethimes multiple entries, one of them should be city, country, field has to be checked, I have discovered so far something like this:

```
"location": [
{
"name": "Flexible / Remote"
},
{
"name": "London, United Kingdom"
}
]
```

- **experience**: level of exp required, have not seen yet but I guess could be NULL
- **html_link** : link to full job offer description , could valuable if we get data from multiple sources to identify doubles (our sources are often services that query data sources, for example okjobs queries LinkIn), should not be NULL
- **type**: describes tpye as "external" (sample data has no other values), could be NULL imho
- **company_name**: should not be NULL
- **company_id**: muse company id => not useful for us !?
- **source**: redundant (see html_link above) !?
- **created** : I would say publication_date is enough !?

reed.com

- **employerId**: internal id, should not be NULL but is of no use for us !?
- **employerName**: should not be NULL
- **employerProfileId**: for internal us, do we need this (I think not)
- **employerProfileName**: see above
- **jobTitle**: job name, should be not NULL
- **locationName**: cntains name of city or areas code ?, looks like
  `"LS110BT", "SG138PQ", "ST150AL", "B248HH", "CM12TS"`
  I have to check but I guess the api was done by country !?
- **minimumSalary**: float, can be NULL
- **maximumSalary**: float, can be NULL
- **currency**: name, can be NULL
- **expirationDate**: a date, do we need this ?
- **date**: date when offer was published, should not be NULL
- **jobDescription**: first lines from job offer, too few and unspecified for us to use ?!
- **applications**: integer, seems too unique fro me (info not found in any other source)
- **jobUrl**: url of job offer, can be important to identify duplicates
- **id**: every data I save contains an id, this id is distinct for the data source

okjob.com

- **Company-ID**: internal id we don´t need ?!
- **LinkedIn-Job-Link**: "url of job offer, can be important to identify duplicates
- **Company-Name**: name, should not be null
- **Job-Title**: job name, should be not NULL
- **Location**: different formats possible:
  `"Linz, Upper Austria, Austria", "Framwellgate Moor, England, United Kingdom", # city, region, country "Los Angeles Metropolitan Area" # only region ? "New Zealand" # only country "Sacramento, CA" # city and state`
  not very uniform format, needs further checks
- **Job-Description**: url of job offer, can be important to identify duplicates
- **Apply-Link**: if don´t think we need this
- **Region**: I get confused, see location but something like "North America", we don´t need
- **Job-Type**: sample:

```
"Hybrid, 100% Salary, Four Days",
"Remote, Four Days, 100% Salary",
"On-site, Four Days, 100% Salary",
"Flexible, 100% Salary, Four Days",
  "4 Day Week, Remote, 100% Salary",
```

- **Job-Tags**: job skills, comma separated string
- **Job-Category**: can contain multiple entries, comma separated string
- **Hours**: integer
- **Salary-Min**: can be NULL
- **Salary-Max**: can be NULL
- **id**: every data I save contains an id, this id is distinct for the data source

For this source I have also saved the full html job description in a subfolder of `data/processed/okjob`
The reason behind this is that we may find information about

- location
- jobskills
- experience
- education !!! <- we have no information about this
- maybe currency
- maybe work hours

and this data mining requires a bit more effort than a simple web scrapping effort can produce (see Boris´s stuff
on your data transformation). In the moment it is still too much data, I try to identify parts of the job offer that could contain
the information above, so for example I will try to identitfy the parts of the job offer that could contain info about required skills. This part can then be saved per source and can then be used for "skill mining".

So far our data needs only a collection per source but for this part we could make sperate collections and link them together,
see folder for oksamples and the subfolder with the job descriptions. " collections but they are not independent but have to be linked.

Final DataSet

So what information do we need to solve our buisiness case ?

- **source-id** : to avoid adding the same data twice, see id field of sources
- **job-title** : must not be NULL
- **job-category** : would be nice to have
- **job-skills** : would be nice to have
- **experience** : would be nice to have
- **work-hours** : would be nice to have

EITHER:

- **salary** we also have to know the dimension, is it calculated by hour, month or year ?

OR

- **salary-min** we also have to know the dimension, is it calculated by hour, month or year ?
- **salary-max**

- **currency** : must not be NULL , in some cases we have to add this
- **job-source** : url of original job offer, to detect duplicates between sources, would be nice to have
- **date-published** : when was the job offer published, would be nice to have but should be available for all

Transformation needed

We have to transform our raw data for:

- **job-title** => find a basis for the names (i think okjobs offers a nice list) and try to link the findings from each source to it
- **job-skills** =>find a basis for the names (i think okjobs offers a nice list) and try to link the findings from each source to it
- **job-category** => find a basis for the names (i think okjobs offers a nice list) and try to link the findings from each source to it
- **experience** => ???
- **location** => this has to be a uniform format for all sources, I saw that Boris tried some Geolocator, I think
  this is a good approach
- **salray** => identitfy source dimension and transform to common basis (per month?)
- **currency** => when not delivered by source, identify by country ?

TODO for 2nd milestone

We have to produce a uml diagramm for our data model (see ressources at the start of the document).
These are my thoughts on the topic. Of course it still has to be written in the right format.
Forgive me, I´m still tired today ;)