

Backend Engineer | Programming Task

Overview:

This programming challenge consists of two tasks. First task requires you to crawl the data from a credit web source and parse the crawled data. In the second part, you will have to develop a backend engine that will expose a web API. These APIs can be used by client to search the data based on a specific criteria.

Please note that you are required to use Python version 3.x.x throughout this task. Choice of other supporting frameworks and libraries is up to you.

Submission:

You are required to submit this task using github. Please create a profile on github if you don't already have one. Upload all the code, graphs and results to a github repository and share the link to repository in the email.

You will have **one week** to submit this assignment.

Scoring:

Primary scoring will be done on the correctness of your code and output. However, We will be considering several other factors for evaluation e.g.

- Code readability
- Performance
- Documentation
- Error handling
- Test cases
- Evaluation/accuracy scores.

You are encouraged to improvise on the above list, as this is only for the hint.

Task-01:

The website you need to crawl is the Indonesia Stock Market Research. The website link is: <http://www.idx.co.id/en-us/listed-companies/company-profiles/>

This site contains company information and mainly ticker symbols.

This task has two parts:

1) You need to collect the basic raw data available in the link. Bonus points will be given if you collect more. The desired output format should be

```
[
  {
    "ticker symbol": "AALI",
    "company name": "Astra Agro Lestari Tbk",
```

```

        "url":
        "http://www.idx.co.id/en-us/listed-companies/company-profiles/company-profile-detail/?kodeEmiten=AALI",
        "listing_date": "1997-12-09",
        "crawled_at": "2018-06-04",
    },
    ...
]

```

Save this output to file named **company_index.json**.

2) In the second part you need to fetch detailed company profile based on the links you crawled in part 1. The detailed company profile should include:

- company name	- Security Code
- Office address	- Email Address
- Phone	- Fax
- NPWP	- Site
- IPO Date	- Board
- Sector	- Sub Sector
- Registrar	- Corporate Secretary
- Director	- Subsidiary

The output format should be like

```

[
  {
    "Company Name": "Astra Agro Lestari Tbk",
    "Security Code": "AALI",
    "Office address": "Jl Pulo Ayang Raya Blok OR No. 1 Kawasan Industri Pulogadung Jakarta",
    "Email Address": "Investor@astra-agro.co.id.",
    "Country": "Indonesia", (default)
    "Phone": "+62 461-65-55",
    "Fax": "[+62 461-6655, +62 461-6677, +62 461-6688]",
    "NPWP": "01.334.427.0-054.000",
    "Company Website": "http://www.astra-agro.co.id",
    "IPO Date": "1997-12-09",
  }
]

```

```

    "Board": "UTAMA",
    "Sector": "AGRICULTURE",
    "Sub Sector": "PLANTATION",
    "Registrar": "PT. Raya Saham Registrar (dulu bernama PT. Risjad Salim Registrar)",
    "Corporate Secretary": [{"name": "Mario Casimirus Surung Gultom", "email": "investor@astra-agro.co.id", "phone": "+62 021-4616555"}],
    "Director": [{"name": "Santosa", "position": "PRESIDEN DIREKTUR"}, {"name": "Joko Supriyono", "position": "WAKIL PRESIDEN DIREKTUR"}],
    "Subsidiary": [{"name": "PT Agro Menara Rachmat", "type": "Perkebunan Sawit", "total asset": 10,316, "percentage": "100%"}, ...]
  },
  ...
]

```

Save the output in **company_profiles.json**.

Task 2:

This task involves building a backend web API. You are free to choose any web Framework e.g. *Flask*, *Django*, *Bottle*, *Pyramid* etc. Please use the crawled datafile **company_profiles.json**. For the first step of this task you should load this Data in the database of your choice *MySQL*, *PostgreSQL*, *MongoDb*, *SQLite* etc. Then using the web framework, provide an API for the consumer to search company information.

The consumer should also be able to apply filters. The API endpoint should support following operations:

1) Fetch list of companies:

The endpoint should be able to fetch list of all companies without applying any filters.

Endpoint: /companies

Output:

```

{
  "status_code": 200,
  "message": "successful",
  "data": [
    {
      "id": 1
      "Company Name": "Astra Agro Lestari Tbk",
      "Security Code": "AALI",
      "Office address": "Jl Pulo Ayang Raya Blok OR No. 1 Kawasan Industri Pulogadung Jakarta"
      "Email Address": "Investor@astra-agro.co.id.",

```

```

    "Country": "Indonesia", (default)
    "Phone": "+62 461-65-55",
    "Fax": "[+62 461-6655, +62 461-6677,+62 461-6688]",
    "NPWP": "01.334.427.0-054.000",
    "Company Website": "http://www.astra-agro.co.id",
    "IPO Date": "1997-12-09",
    "Board": "UTAMA",
    "Sector": "AGRICULTURE",
    "Sub Sector": "PLANTATION",
    "Registrar": "PT. Raya Saham Registrar (dulu bernama PT. Risjad Salim
    Registra"
    "Corporate Secretary": [{"name": "Mario Casimirus Surung
    Gultom", "email": "investor@astra-agro.co.id", "phone": "+62 021-4616555"}],
    "Director": [{"name": "Santosa", "position": "PRESIDEN DIREKTUR"}, {"name": "Joko
    Supriyono", "position": "WAKIL PRESIDEN DIREKTUR"}],
  },...
]
}

```

2) Fetch specific company:

This endpoint can fetch a specific company given its *name*

Endpoint: companies?company_name=Astra%20Astra (Fetch company where name=Astra Agro.')

Output:

```

{
  "status_code": 200,
  "message": "successful",
  "data": [
    {
      "id": 1
      "Company Name": "Astra Agro Lestari Tbk",
      "Security Code": "AALI",
      "Office address": "Jl Pulo Ayang Raya Blok OR No. 1 Kawasan Industri
      Pulogadung Jakarta"
      "Email Address": "Investor@astra-agro.co.id.",
      "Country": "Indonesia", (default)
      "Phone": "+62 461-65-55",
      "Fax": "[+62 461-6655, +62 461-6677,+62 461-6688]",
      "NPWP": "01.334.427.0-054.000",
      "Company Website": "http://www.astra-agro.co.id",
      "IPO Date": "1997-12-09",
      "Board": "UTAMA",

```

```
"Sector": "AGRICULTURE",
"Sub Sector": "PLANTATION",
"Registrar": "PT. Raya Saham Registrar (dulu bernama PT. Risjad Salim
Registra"
"Corporate Secretary": [{"name": "Mario Casimirus Surung
Gultom", "email": "investor@astra-agro.co.id", "phone": "+62 021-4616555"}],
"Director": [{"name": "Santosa", "position": "PRESIDEN DIREKTUR"}, {"name": "Joko
Supriyono", "position": "WAKIL PRESIDEN DIREKTUR"}],
```

```
}
    ]
}
```

Please provide answer to following questions:

1. Which database engine you choose and why?
2. Which web framework you choose and why?
3. Briefly describe the architecture of your application?