

data-pipeline



- Processing pipelines are defined in a configuration file and not code
- Input and output works through streaming data
- Natively based on data-packages

[https://github.com/
frictionlessdata/
datapackage-pipelines](https://github.com/frictionlessdata/datapackage-pipelines)

Why declarative?

Contribution Oriented

- Declarative → Structured
- Structured →
 - Maintainable
 - Readable
 - Understandable
- Limited flexibility
(but in a good way)
- Less code →
More people can contribute

```
ottoman-association-registry:
  title: רשימת אגודות עותומניות
  description: עותומניות, מתוך תשובה לבקשת חופש מידע
  pipeline:
    - run: add_metadata
      parameters:
        name: ottoman-association-registry
        title: רשימת אגודות עותומניות
        sources:
          - title: אתר היחידה לחופש המידע
            path: https://foi.gov.il/he/node/1908
    - run: add_resource
      parameters:
        name: foi-response
        url: "https://foi.gov.il/sites/default/fi"
        headers:
          - id
          - name
          - address
        skip_rows: 1
        constants:
          kind: "ottoman-association"
    - run: stream_remote_resources
    - run: sample
    - run: concatenate
      parameters:
        target:
          name: ottoman-association-registry
```


datapackage-pipelines

- Processing pipelines are defined in a configuration file and not code
- Input and output works through streaming data
- Natively based on data-packages

<https://github.com/frictionlessdata/datapackage-pipelines>

