

# Робота з даними в форматі JSON

Керівник наукової роботи  
Проценко В.С.  
Виконала студентка ІПЗ-3  
Олійник Дарина

# Постановка задачі

Дослідження та вивчення особливостей роботи з даними в форматі JSON, зокрема у функціональній мові програмування Haskell.

Вимоги:

- ✓ Надати детальний опис формату JSON
- ✓ Виконати аналіз проблем та особливостей використання JSON у функціональній мові програмування Haskell.
- ✓ Надати можливі шляхи для вирішення наведених проблем.
- ✓ Описати бібліотеки для роботи з JSON в Haskell.
- ✓ Продемонструвати роботу з JSON.

# Короткий опис JSON

JSON (JavaScript Object Notation) — простий формат для обміну даними, зрозумілий людині. Має досить невеликий набір правил для форматування. Може містити чотири базових типи (стрічки, числа, логічні значення (false та true) та null), а також два структурні типи (об'єкти та масиви).

```
{
  "faculty": "FI",
  "students": [
    {
      "name": "Daryna",
      "surname": "Oliinyk",
      "course": 3
    },
    {
      "name": "Petro",
      "surname": "Petrov",
      "course": 2
    }
  ]
}
```

# Підтримка мовами програмування



# Схожі формати

## XML

```
<?xml version="1.0"
encoding="UTF-8" ?>
<root>
  <worker>
    <name>Ivan</name>
    <id>1</id>
  </worker>
  <worker>
    <name>Petro</name>
    <id>2</id>
  </worker>
</root>
```

## CSV

```
worker/name,worker/id
Ivan,1
Petro,2
```

## YAML

```
---
worker:
- name: Ivan
  id: '1'
- name: Petro
  id: '2'
```

## JSON

```
{ "worker" :
[ {
  "name": "Ivan",
  "id": "1"
}, {
  "name": "Petro",
  "id": "2"
}] }
```

# Робота програми

Завантаження JSON  
даних з API та їх запис  
у файл



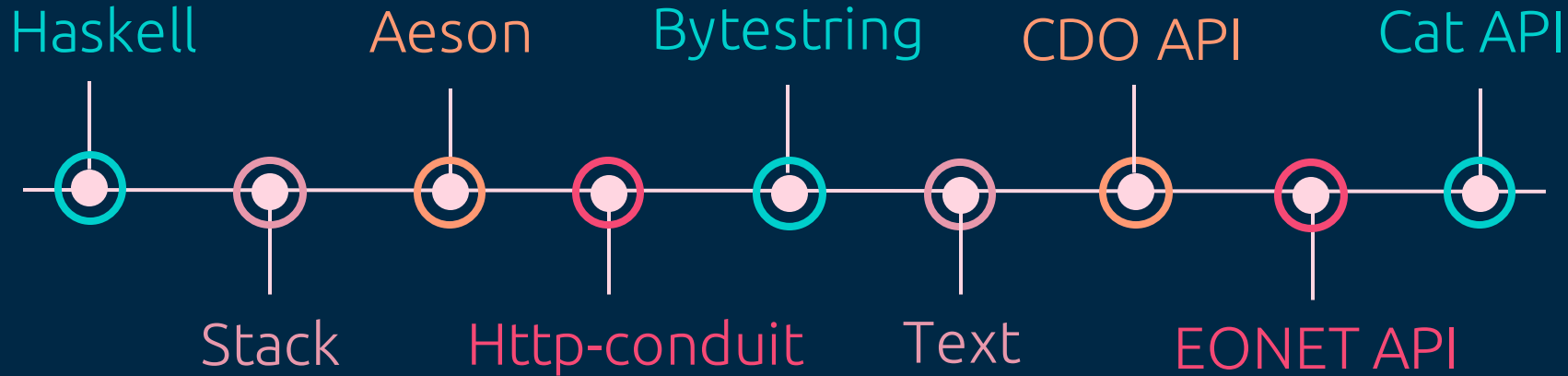
Обробка отриманих  
даних та виведення  
потрібної інформації на  
екран

Створення нових  
типів даних з  
існуючих



Збереження створеного  
типу даних у JSON  
форматі

# Використані інструменти



# Представлення даних

```
{ "uid": "gov.noaa.ncdc:C00861",  
  "mindate": "1763-01-01",  
  "maxdate": "2020-05-06",  
  "name": "Daily Summaries",  
  "datacoverage": 1,  
  "id": "GHCND" }
```



Вигляд JSON даних

Відповідний тип для для  
роботи з ними



```
data DatasetResult = DatasetResult  
  { uid :: T.Text  
    , minimalDate :: T.Text  
    , maximalDate :: T.Text  
    , name :: T.Text  
    , datacoverage :: Double  
    , id :: T.Text  
  } deriving Show
```

```
instance FromJSON DatasetResult where  
  parseJSON (Object v) =  
    DatasetResult <$> v .: "uid"  
    <*> v .: "mindate"  
    <*> v .: "maxdate"  
    <*> v .: "name"  
    <*> v .: "datacoverage"  
    <*> v .: "id"
```



# Функціонал програми

```
printCatFact :: IO ()
printCatFact = do
  response <- httpLBS catRequest
  let status = getResponseStatusCode response
  if status == 200
  then do
    let responseBody = getResponseBody response
    let cat = eitherDecode responseBody :: Either String CatData
    case cat of
      Left l -> print l
      Right r -> print (fact r)
  else print "cat request failed with error"
```

Отримання  
даних, їх аналіз  
та виведення  
поля fact

Результат  
виконання

["Many people fear catching a protozoan disease, Toxoplasmosis, from cats. This disease can cause illness in the human, but more seriously, can cause birth defects in the unborn. Toxoplasmosis is a common disease, sometimes spread through the feces of cats. It is caused most often from eating raw or rare beef. Pregnant women and people with a depressed immune system should not touch the cat litter box. Other than that, there is no reason that these people have to avoid cats."]

# Функціонал програми

```
readStations :: IO()
readStations = do
    jsonData <- B.readFile stationFile
    let response = decodeStrict jsonData :: Maybe
    StantionResponse
    case resultsS <$> response of
        Nothing -> print "error loading data"
        (Just r) -> printNames (Stantion r)
```



Обробка  
отриманих  
даних, та  
виведення  
значень name

```
saveData :: IO()
saveData = do
    response <- decodeFileStrict jsonData :: IO (Maybe
    DatasetResponse)
    let resultsData = results <$> response
    let d = getElems resultsData
    encodeFile "datasetsName.json" d
```



Створення  
нових даних та  
збереження їх у  
JSON форматі



# Результати виконання програми

```
"saving response about datasets to file"  
"saving response about stations to file"  
"Dataset names"  
"Daily Summaries"  
"Global Summary of the Month"  
"Global Summary of the Year"  
"Weather Radar (Level II)"  
"Weather Radar (Level III)"  
"Normals Annual/Seasonal"  
"Normals Daily"  
"Normals Hourly"  
"Normals Monthly"  
"Precipitation 15 Minute"  
"Precipitation Hourly"  
"Station names"  
"ABBEVILLE, AL US"  
"ADDISON, AL US"  
"ADDISON CENTRAL TOWER, AL US"  
"ALABASTER SHELBY CO AIRPORT, AL US"  
"BELLE MINA 2 N, AL US"  
"ALAGA, AL US"  
"ALBERTA, AL US"  
"ALBERTVILLE, AL US"  
"ALEXANDER CITY, AL US"
```

Виведення значень name з файлів  
datasets.json та stations.json

Виведення вмісту файлу categories.json

```
{  
  "ShortCategory": {  
    "cTitle": "Drought",  
    "cDescription": "Long lasting absence of precipitation affecting agriculture and livestock, and the overall availability of food and water."},  
  "ShortCategory": {  
    "cTitle": "Dust and Haze",  
    "cDescription": "Related to dust storms, air pollution and other non-volcanic aerosols. Volcano-related plumes shall be included with the originating eruption event."},  
  "ShortCategory": {  
    "cTitle": "Earthquakes",  
    "cDescription": "Related to all manner of shaking and displacement. Certain aftermath of earthquakes may also be found under landslides and floods."},  
  "ShortCategory": {  
    "cTitle": "Floods",  
    "cDescription": "Related to aspects of actual flooding--e.g., inundation, water extending beyond river and lake extents."},  
  "ShortCategory": {  
    "cTitle": "Landslides",  
    "cDescription": "Related to landslides and variations thereof: mudslides, avalanche."},  
  "ShortCategory": {  
    "cTitle": "Manmade",  
    "cDescription": "Events that have been human-induced and are extreme in their extent."},  
  "ShortCategory": {  
    "cTitle": "Sea and Lake Ice",  
    "cDescription": "Related to all ice that resides on oceans and lakes, including sea and lake ice (permanent and seasonal) and icebergs."},  
  "ShortCategory": {  
    "cTitle": "Severe Storms",  
    "cDescription": "Related to the atmospheric aspect of storms (hurricanes, cyclones, tornadoes, etc.). Results of storms may be included under floods, landslides, etc."},  
  "ShortCategory": {  
    "cTitle": "Snow",  
    "cDescription": "Related to snow events, particularly extreme/anomalous snowfall in either timing or extent/depth."},  
  "ShortCategory": {  
    "cTitle": "Temperature Extremes",  
    "cDescription": "Related to anomalous land temperatures, either heat or cold."},  
  "ShortCategory": {  
    "cTitle": "Volcanoes",  
    "cDescription": "Related to both the physical effects of an eruption (rock, ash, lava) and the atmospheric (ash and gas plumes)."},  
  "ShortCategory": {  
    "cTitle": "Water Co"
```

# Приклад вмісту отриманих файлів

## datasets.json

```
{
  "metadata": {
    "resultset": {
      "offset": 1,
      "count": 11,
      "limit": 25
    },
    "results": [
      {
        "uid": "gov.noaa.ncdc:C00861",
        "mindate": "1763-01-01",
        "maxdate": "2020-05-06",
        "name": "Daily Summaries",
        "datacoverage": 1,
        "id": "GHCND"
      }, ...
    ]
  }
}
```

## stations.json

```
{
  "metadata": {
    "resultset": {
      "offset": 1,
      "count": 139465,
      "limit": 25
    },
    "results": [
      {
        "elevation": 139,
        "mindate": "1948-01-01",
        "maxdate": "2014-01-01",
        "latitude": 31.5702,
        "name": "ABBEVILLE, AL US",
        "datacoverage": 0.8813,
        "id": "COOP:010008",
        "elevationUnit": "METERS",
        "longitude": -85.2482
      }, ... ]
    ]
  }
}
```

## nasa.json

```
{
  "title": "EONET Categories",
  "description": "List of all the available event categories",
  "link": "https://eonet.sci.gsfc.nasa.gov/api/categories",
  "categories": [
    {
      "id": 6,
      "title": "Drought",
      "link": "https://eonet.sci.gsfc.nasa.gov/api/v2.1/categories/6",
      "description": "Long lasting absence of precipitation affecting agriculture and livestock.",
      "layers": "https://eonet.sci.gsfc.nasa.gov/api/v2.1/layers/6"
    }, ... ]
  }
}
```

# Приклад вмісту створених файлів

## datasetsName.json

```
[
  {
    "idDataset": "GHCND",
    "nameDataset": "Daily Summaries"
  },
  {
    "idDataset": "GSOM",
    "nameDataset": "Global Summary of the Month"
  },
  {
    "idDataset": "GSOY",
    "nameDataset": "Global Summary of the Year"
  },
  {
    "idDataset": "NEXRAD2",
    "nameDataset": "Weather Radar (Level II)"
  },
  ...
]
```

## categories.json

```
[
  {
    "cDescription": "Long lasting absence of precipitation affecting agriculture and livestock, and the overall availability of food and water.",
    "cTitle": "Drought"
  },
  {
    "cDescription": "Related to dust storms, air pollution and other non-volcanic aerosols. Volcano-related plumes shall be included with the originating eruption event.",
    "cTitle": "Dust and Haze"
  },
  ...
]
```

# Висновок

Було реалізовано проект, який демонструє роботу з форматом JSON у функціональній мові програмування Haskell. Результати програми чітко дають зрозуміти, що даний формат дійсно дуже зручний для передачі даних по мережі. До того ж нескладна та зрозуміла структура дозволяє легко перетворювати об'єкти JSON формату у відповідні типи, структури або класи обраної мови програмування (та навпаки).

The background is a dark blue field decorated with a pattern of thin white vertical lines and small squares in teal, pink, and orange. The squares are scattered across the frame, some appearing as solid colors and others as white outlines. The text is centered in a white, sans-serif font.

Дякую за увагу