

# API



## Extract

Retrieves and verifies data  
from various sources



## Transform

Processes and organizes  
extracted data so it is usable



## Load

Moves transformed data  
to a data repository

# Read data from a file

```
import pandas as pd  
  
df = pd.read_csv("data/grades.csv")  
  
print(df)
```

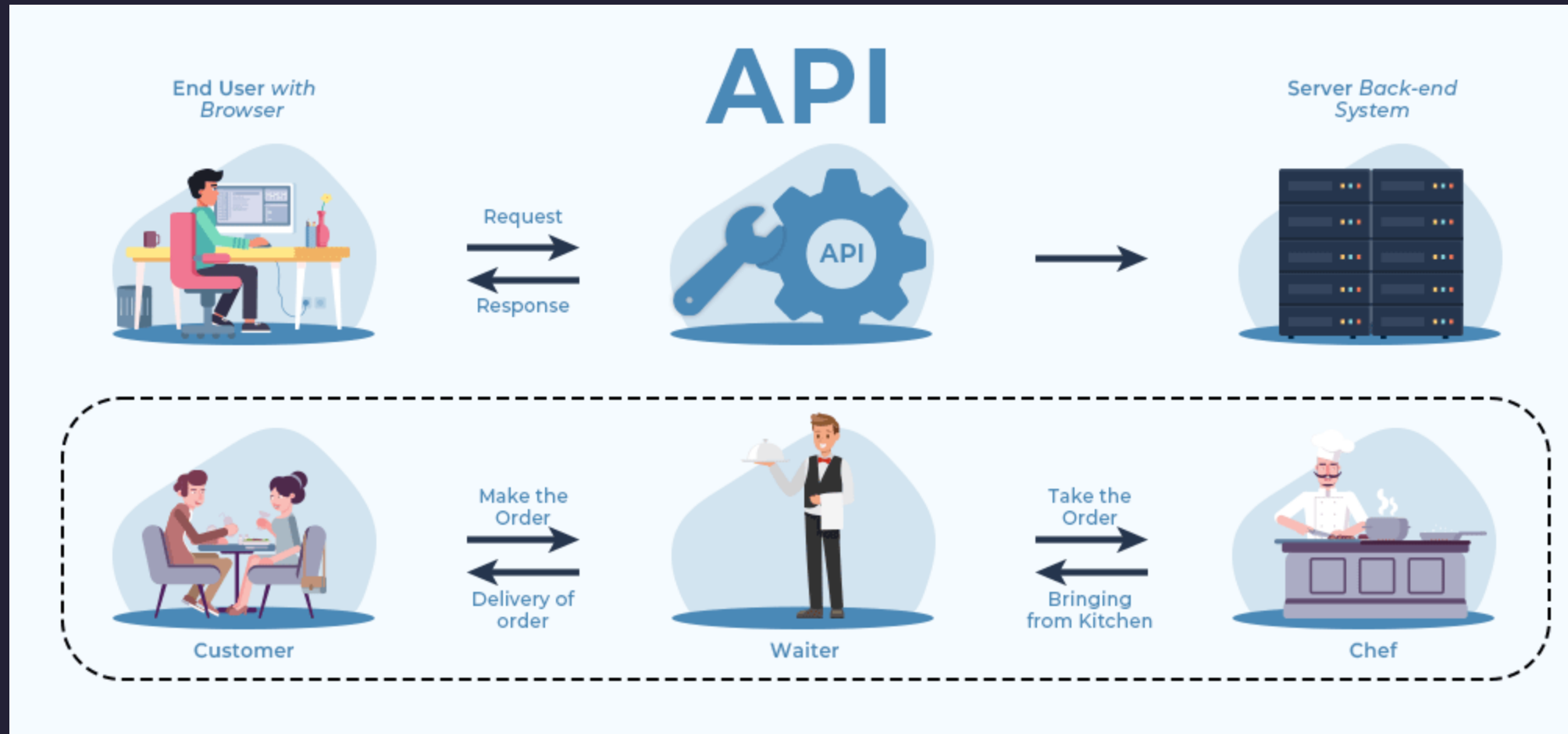
# Extract data from a source

- Manually download a file
- Extract data using SQL
- **Extract data using an API**

# What is an API?

<https://youtu.be/s7wmiS2mSXY?si=Qb3zgVmPj2jB9DPP>

# Application Programming Interface



# iTunes API

Web API that allows you to search for movies, music, apps, etc on iTunes.

---

[https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/index.html#//apple\\_ref/doc/uid/TP40017632-CH3-SW1](https://developer.apple.com/library/archive/documentation/AudioVideo/Conceptual/iTuneSearchAPI/index.html#//apple_ref/doc/uid/TP40017632-CH3-SW1)

<https://itunes.apple.com/search?entity=movie&term=avengers&limit=1>



# API response in JSON (JavaScript Object Notation)

```
{
  "resultCount": 1,
  "results": [
    {
      "wrapperType": "track",
      "kind": "feature-movie",
      "collectionId": 1470195095,
      "trackId": 533654020,
      "artistName": "Joss Whedon",
      "collectionName": "Avengers 4-Movie Collection",
      "trackName": "The Avengers",
      ...
    }
  ]
}
```

# REST API syntax

```
https://itunes.apple.com/search?entity=movie&term=avengers&limit=1
```

- endpoint: `itunes.apple.com/`
- path (or database): `search`
- query parameters: `entity=movie&term=avengers&limit=1`

---

<https://www.ibm.com/docs/en/informix-servers/12.10?topic=api-rest-syntax>

# More on query parameters

```
https://itunes.apple.com/search?entity=movie&term=avengers&limit=1
```

- Start after the question mark `?`
- Key-value pairs separated by `&`
  - `entity=movie`
  - `term=avengers`
  - `limit=1`

# Understanding API documentation

- Endpoint
- Path
- Query parameters

# COVID-19 Canada Open Data Working Group

<https://opencovid.ca/api/>

- Endpoint: ?
- Path: ?
- Query parameters: ?

# COVID-19 Canada Open Data Working Group

- Endpoint: `https://api.opencovid.ca`

- Path & Query parameters

- `timeseries`

- `loc`

- `after`

- ...

- `summary`

- `loc`

- `date`

- ...

# COVID-19 Canada Open Data Working Group

- <https://api.opencovid.ca/timeseries?loc=qc>
- <https://api.opencovid.ca/timeseries?loc=qc&after=2023-09-01>
- <https://api.opencovid.ca/summary?loc=qc>

# Open Brewery DB

<https://www.openbrewerydb.org>

- Endpoint: ?
- Path: ?
- Query parameters: ?



# Open Brewery DB

- Endpoint: `https://api.openbrewerydb.org/v1`
- Path & Query parameters
  - `breweries`
    - `by_city`
    - `by_state`
    - `by_type`
    - ...



## Open Brewery DB

- `https://api.openbrewerydb.org/v1/breweries?by\_city=detroit`
- `https://api.openbrewerydb.org/v1/breweries?by\_state=michigan`
- `https://api.openbrewerydb.org/v1/breweries?by\_state=michigan&by\_type=micro`

# Make `requests` programmatically

- `requests.get()` : **get data from a URL**
- `requests.post()` : post data to a URL
- `requests.put()` : update data on a URL
- `requests.delete()` : delete data from a URL

## requests.get()

```
import requests  
  
url = "itunes.apple.com/search?entity=movie&term=avengers&limit=1"  
  
response = requests.get(url)
```

# response

```
print(response)          # <Response [200]>
print(response.text)      # raw text
print(response.json())    # convert to python dictionary
```

## `requests.get()` dynamically

```
import requests

entity = "movie"
term = "avengers"
limit = 1

url = f"itunes.apple.com/search?entity={entity}&term={term}&limit={limit}"

response = requests.get(url)
```

## `requests.get()` dynamically with `params`

```
import requests

url = "https://itunes.apple.com/search"

params = {
    "entity": "movie",
    "term": "avengers",
    "limit": 10
}

response = requests.get(url, params=params)

print(response.json())
```

# Accessing items in a list/tuple/dict

- list/tuple: `[index]`
- dict: `[key]`



# How to access the following items?

```
cities = [  
    {"name": "Montreal", "state": "QC", "country": "CA"},  
    {"name": "Toronto", "state": "ON", "country": "CA"},  
    {"name": "Vancouver", "state": "BC", "country": "CA"},  
    {"name": "Detroit", "state": "MI", "country": "US"}  
]
```

- {"name": "Vancouver", "state": "BC", "country": "CA"}
- "Vancouver"
- "Montreal"

```
cities = {  
    "name": ["Montreal", "Toronto", "Vancouver", "Detroit"],  
    "state": ["QC", "ON", "BC", "MI"],  
    "country": ["CA", "CA", "CA", "US"]  
}
```

- ["Montreal", "Toronto", "Vancouver", "Detroit"]
- "Montreal"

```
cities = {  
    "Montreal": {"state": "QC", "country": "CA"},  
    "Toronto": {"state": "ON", "country": "CA"},  
    "Vancouver": {"state": "BC", "country": "CA"},  
    "Detroit": {"state": "MI", "country": "US"}  
}
```

- {"state": "QC", "country": "CA"}
- "QC"

```
cities = {
    "location": {
        "Montreal": {"state": "QC", "country": "CA"},
        "Toronto": {"state": "ON", "country": "CA"},
        ...
    },
    "stats": {
        "Montreal": [
            {"year": 2013, "population": 2000000, "area": 431.5},
            {"year": 2014, "population": 1980000, "area": 431.5}
        ],
        "Toronto": [
            {"year": 2013, "population": 2800000, "area": 630.2},
            ...
        ],
        ...
    }
}
```

- {"year": 2013, "population": 2000000, "area": 431.5}
- "QC"
- 2000000

```
{
  "resultCount": 1,
  "results": [
    {
      "wrapperType": "track",
      "kind": "feature-movie",
      "collectionId": 1470195095,
      "trackId": 533654020,
      "artistName": "Joss Whedon",
      "collectionName": "Avengers 4-Movie Collection",
      "trackName": "The Avengers",
      ...
    }
  ]
}
```

- "Joss Wheldon"
- "The Avengers"

```
[
  {
    "id": "8aadd633-ee8b-4550-9b3a-ffdf6fee2b44",
    "name": "Brew Detroit",
    "brewery_type": "micro",
    "address_1": "1401 Abbott St",
    ...
  },
  {
    "id": "9ef70c89-a7e9-4759-a71e-4adca64c543a",
    "name": "Eastern Market Brewing Company",
    "brewery_type": "micro",
    "address_1": "2515 Riopelle St",
    ...
  },
]
```

- "Brew Detroit"

## `requests.get()` dynamically with `params` - Open Brewery DB

<https://www.openbrewerydb.org>

1. Make a request to the Open Brewery DB API to get a list of breweries in California that are micro breweries.
2. Print the name and address (`address_1`) of each brewery.