

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)
```

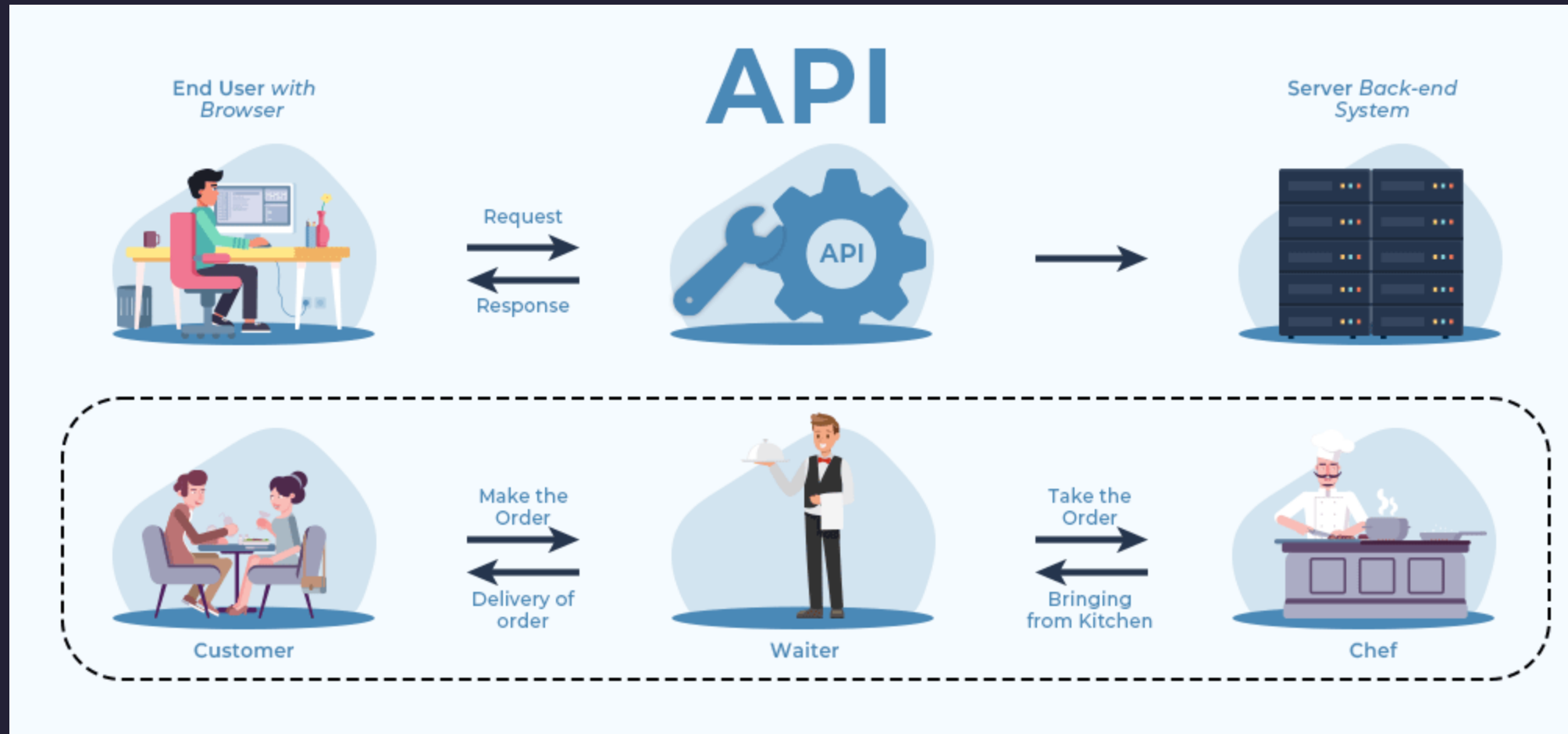
Extract data from a source

- 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://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())
```

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

<https://www.openbrewerydb.org>

1. Make a request to the Open Brewery DB API to get all breweries in the db.
2. Make a request to the Open Brewery DB API to get a list of breweries in California that are nano breweries.


```
import requests

url = "https://api.openbrewerydb.org/v1/breweries"

params = {
    "by_state": "california",
    "by_type": "nano"
}

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

print(response.json())
```