

- 1) What is an API?
 - a) a set of rules that allows software programs to communicate with each other.
 - b) makes it easier to use external data, services, or functionality without knowing the internal workings
- 2) Common Concepts
 - a) Endpoint: URL where the API can be accessed.
Ex: <https://api.github.com/users/octocat>
 - b) Request: The message you send to the API asking for data or to perform an action.
 - c) Response: The data or confirmation returned by the API, usually in JSON or XML.
 - d) HTTP Methods:
 - i) **GET** → Retrieve data
 - ii) **POST** → Send data / create new resource
 - iii) **PUT** → Update data
 - iv) **DELETE** → Remove data
- 3) Authentication
 - a) API Key: A unique code identifying the user/application.
 - b) OAuth: A protocol allowing apps to access data on behalf of a user securely.
 - c) Token: A string sent in the authorization header for secure access.
- 4) Data Formats
 - a) JSON (JavaScript Object Notation): Most common, lightweight, human-readable.
Ex:

```
{  
  "name": "Sherlock Holmes",  
  "books": 4  
}
```
 - b) XML: Older, more verbose alternative to JSON.
 - c) CSV / Plain Text: Sometimes APIs provide raw data files.
- 5) Using APIs in R
 - a) Packages: **httr**, **jsonlite**, **rvest** (for web scraping if needed).

```
library(httr)  
library(jsonlite)  
  
url <- "https://api.example.com/data"  
response <- GET(url)  
data <- fromJSON(content(response, as = "text"))
```
 - b) Handling authentication:

```
GET(url, add_headers(Authorization = "Bearer YOUR_TOKEN"))
```
- 6) Common Challenges
 - a) Rate limits: APIs often limit the number of requests per minute/hour.
 - b) Errors: 400s → bad request, 401 → unauthorized, 404 → not found, 500 → server error.
 - c) Data structure changes: APIs can update formats or fields over time.