



# REST APIs

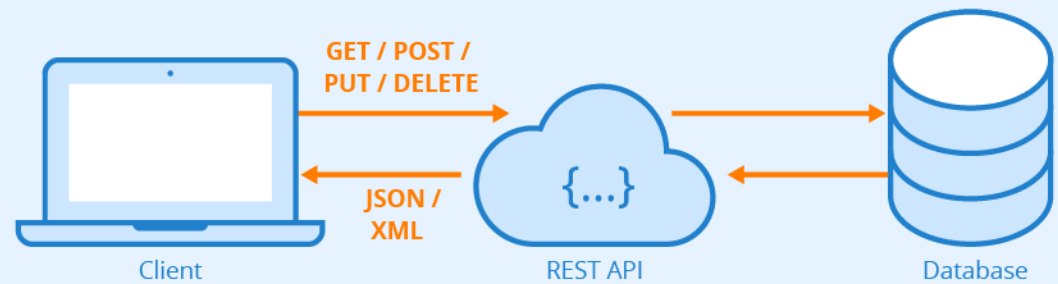
---

**Data Engineering Workshops pt 2**

Ionian IEEE Student Branch

# Τι είναι ένα API

- API: Application Programming Interface
- Διεπαφή επικοινωνίας μεταξύ εφαρμογών (και όχι μόνο)



# Glossary

---

- Web: Σύνολο HTTP Endpoints
- JSON (JavaScript Object Notation): Τύπος αρχείου που περιέχει δεδομένα (μοιάζει με Excel Spreadsheet)
- REST: Μοντέλο – Πρωτόκολλο για τη δημιουργία και χρήση API

# JSON

- File Format (όπως .txt, .py, κλπ)
- Χρησιμοποιείται στα REST APIs
- Εύκολη αποθήκευση και μεταφορά δεδομένων
- Τα αρχεία που θα χρησιμοποιούμε θα είναι σχεδόν πάντα .json
- Κάθε γλώσσα έχει βιβλιοθήκες για την ανάγνωση και επεξεργασία τους

```
1 {  
2     "name": "Michael",  
3     "age": 16,  
4     "school": "JSON High School",  
5     "athlete": true,  
6     "Classes": [  
7         "math",  
8         "english",  
9         "french",  
10        "physics"  
11    ],  
12    "GPA": null  
13 }
```

# REST API

---

## Client - Server

Ξεχωριστές διεπαφές για Client (χρήστη) και Server (διαχειριστή)

---

## Stateless

Κάθε call και request είναι αυτόνομα και δεν αποθηκεύονται πληροφορίες

---

## Cacheable

Κάθε απάντηση σε κλήση είναι cacheable ή όχι

---

## Uniform Interface

Η πληροφορία (αρχεία κλπ) έχει standard μορφή

---

## Layered System

Ιεραρχική αρχιτεκτονική, κάθε χρήστης έχει πρόσβαση μόνο στο δικό του layer



# HTTP Methods

---

Όλα τα REST APIs λειτουργούν κάνοντας request σε μεθόδους HTTP:

- GET – Διάβασε / Κατέβασε κάποιο resource (αρχείο, σελίδα κλπ)
- POST – Δημιούργησε / Ανέβασε κάποιο resource
- PUT – Ενημέρωσε ήδη υπάρχον resource
- DELETE – Διέγραψε ήδη υπάρχον resource

# Μέθοδος GET

---

- Χρησιμοποιείται για να λάβουμε resources από τον server
- Κάθε σελίδα στον browser έρχεται με GET request
- Ζητάει το resource σε συγκεκριμένο url
  - *GET http://www.example.com/url\_to\_folder/* - Παίρνει ολόκληρο φάκελο
  - *GET http://www.example.com/url\_to\_folder/file\_id* - Παίρνει συγκεκριμένο αρχείο

# Μέθοδος POST

---

- Χρησιμοποιείται για να ανεβάζουμε resources στον server
- Πχ Upload φωτογραφίας στο drive
- Ανεβάζει το resource σε συγκεκριμένο url
  - *POST http://www.example.com/folder/ - Δημιουργεί νέο directory*
  - *POST http://www.example.com/folder/file\_id body=/path/to/data - Δημιουργεί νέο αρχείο που \*ΔΕΝ\* υπήρχε ήδη με τα περιεχόμενα του data*



# Μέθοδος PUT

---

- Χρησιμοποιείται για να ενημερώσουμε resources στον server
- Πχ Edit post σε social media
- Ενημερώνει το resource σε συγκεκριμένο url
  - *PUT http://www.example.com/folder/file\_id body=/path/to/data – Ενημερώνει συγκεκριμένο αρχείο με τα περιεχόμενα του data*

# Μέθοδος DELETE

---

- Χρησιμοποιείται για να διαγράψουμε resources στον server
- Πχ Διαγραφή post σε social media
- Διαγράφει το resource σε συγκεκριμένο url
  - *DELETE http://www.example.com/folder* – Διαγράφει ολόκληρο φάκελο
  - *DELETE http://www.example.com/folder/file\_id* – Διαγράφει συγκεκριμένο αρχείο

# Παράμετροι σε Μεθόδους

---

- Μπορούμε να εισάγουμε παραμέτρους στις HTTP μεθόδους
- Στο τέλος του url προσθέτουμε “?” και στη συνέχεια τις παραμέτρους
- Πχ:
  - *GET http://www.example.com/customers?name=john*
- Μπορούμε να εισάγουμε και πολλές παραμέτρους, με τη χρήση του “&”:
  - *GET http://www.example.com/customers?name=john&surname=smith*

# Χρήση API

---

- Terminal & Curl
- Γλώσσα προγραμματισμού & requests
- Συγκεκριμένη βιβλιοθήκη για το API