

Node.js API

Εισαγωγή

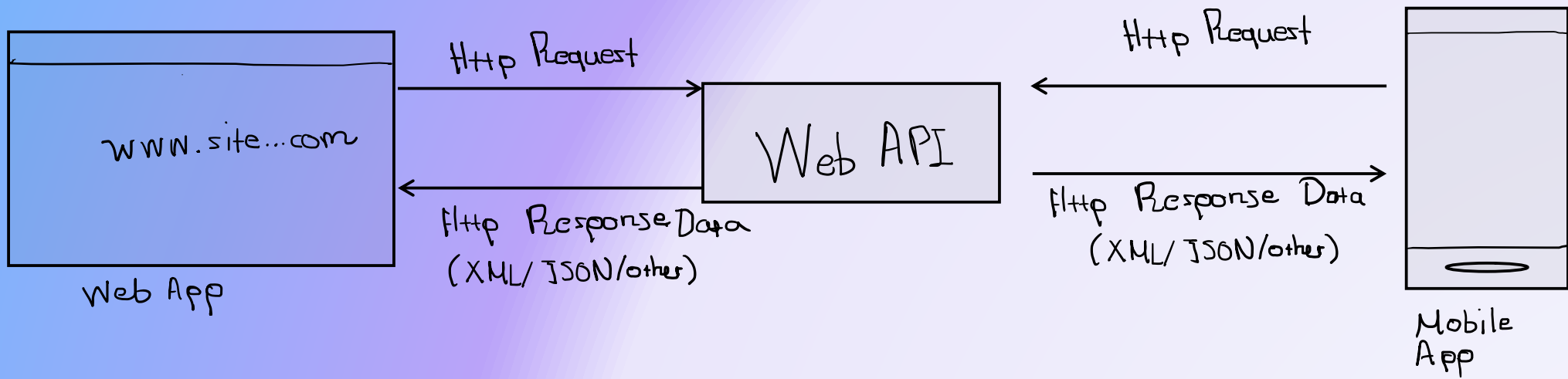
- API (Application Programming Interface): διεπαφή προγραμματισμού εφαρμογών
- Το API αποτελεί μία διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λογισμικό-λειτουργικό σύστημα-βιβλιοθήκη ώστε να γίνεται δυνατή η **αποστολή** αιτημάτων ή/και η **ανταλλαγή** δεδομένων με άλλα λογισμικά.
- Με άλλα λόγια, ένα API μπορεί να θεωρηθεί ένα είδος διεπαφής που έχει ένα σύνολο λειτουργιών που επιτρέπουν στους προγραμματιστές να έχουν πρόσβαση σε συγκεκριμένες δυνατότητες ή δεδομένα μιας εφαρμογής, ενός λειτουργικού συστήματος ή μιας άλλης υπηρεσίας.

Εισαγωγή

- web API -> χρησιμοποιείται για την ανάπτυξη διαδικτυακών εφαρμογών
- περιέχει ένα σύνολο από μεθόδους που εκτελούν HTTP requests σε έναν server
- Αποτελεί μια «έννοια» και όχι μια συγκεκριμένη τεχνολογία
- Μπορούμε να δημιουργήσουμε Web APIs χρησιμοποιώντας διαφορετικές τεχνολογίες όπως Javascript, Java, .NET κ.λπ.

Εισαγωγή

- Άρα το Api είναι ένα service από το οποίο μπορούμε να «ζητήσουμε» κάποια δεδομένα
- Ένα API επιτρέπει σε μια εφαρμογή να εξάγει πληροφορίες από ένα κομμάτι λογισμικού και να τις χρησιμοποιεί



Παράδειγμα

- Θα δούμε ένα παράδειγμα όπου η εφαρμογή μας θα διαβάζει ένα json και θα μας επιστρέφει τα δεδομένα που περιέχει.

1^{ος} τρόπος

```
    res.end('Here are all Countries');
} else if (path_name === '/api') {
    // read JSON file
    filesystem.readFile(`${__dirname}/countries.json`, "utf8", (err, jsonString) => {
        if (err) {
            console.log("File read failed:", err);
            return;
        }
        console.log("File data:", jsonString);
        //lets send back the data we have parsed
        //we need to let the browser know that we send back json
        res.writeHead('200', {'Content-Type': 'application/json'});
        res.end(jsonString); //end the response
    });
} else {
```

2^{ος} τρόπος

- Υπάρχει βέβαια καλύτερος τρόπος
- Στο παραπάνω παράδειγμα, κάθε φορά που κάποιος μεταβαίνει στο url /api γίνεται ανάγνωση του json αρχείου και στη συνέχεια επιστρέφεται
- Το καλύτερο είναι να κάνουμε ανάγνωση του json αρχείου μια φορά στην αρχή και στη συνέχεια κάθε φορά που μεταβαίνει στο url /api απλά να επιστρέφει τα δεδομένα χωρίς να τα διαβάζει ξανά και ξανά

1ο βήμα, διαβάζουμε το αρχείο json στον top level κώδικα

```
5
6 //TOP LEVEL CODE
7 //use sync version, because it is easier to handle-> we put data in a var and we use it right away
8 // we do not care if data is blocking code execution because this top level code
9 //is going to be executed only once, at the beggining
10
11 const data=filesystem.readFileSync(`${__dirname}/countries.json`, "utf8");
12 //use Js inbuilt function to convert text into a JavaScript object
13 const countriesObject= JSON.parse(data);
14
15
```

- Μια κοινή χρήση του JSON είναι η ανταλλαγή δεδομένων προς/από έναν web server.
- Κατά τη λήψη δεδομένων από έναν web server, τα δεδομένα είναι πάντα μια συμβολοσειρά.
- JSON.parse() ->δεδομένα γίνονται αντικείμενο JavaScript.

Στον server απλά εμφανίζουμε τα δεδομένα που διαβάσαμε

```
//create a server
//creatServer method will accept a callback function that will fire
//every time a new request hits our server
const server= http.createServer((req, res) => {
  //we store the requested url in a variable
  const path_name= req.url;
  console.log(path_name);

  if(path_name=== '/' || path_name=== '/all'){
    res.end('Here are all Countries');
  }else if(path_name=== '/api'){
    //lets send back the data we have parsed in the top level code
    //we need to let the browser know that we send back json

    res.writeHead('200',{ 'Content-Type': 'application/json'});
    res.end(data); //end the response
  }
});
```

To be continued...