



Node.js API

Εισαγωγή

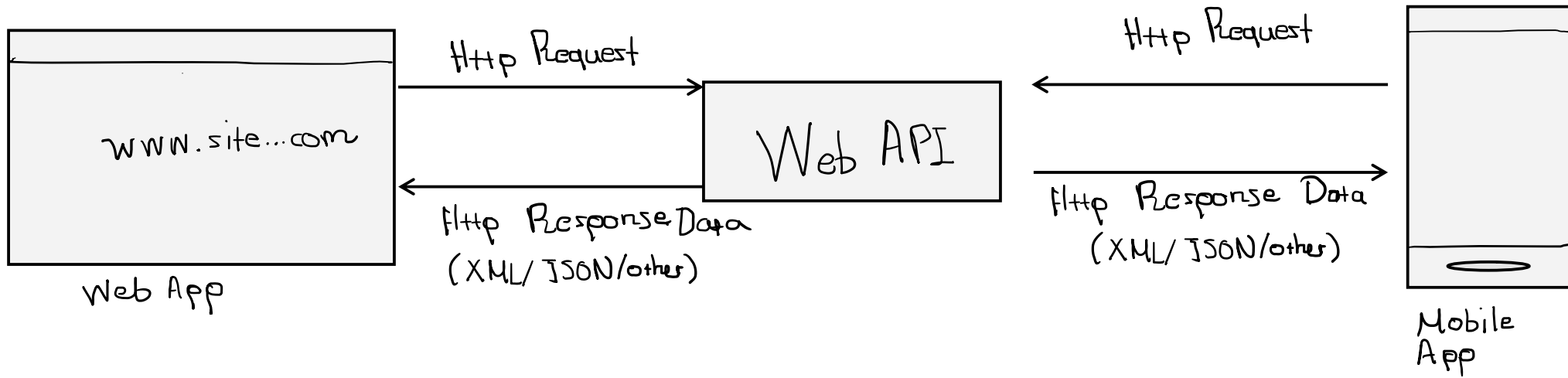
- API (Application Programming Interface): διεπαφή προγραμματισμού εφαρμογών
- Το API αποτελεί μία διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λογισμικό-λειτουργικό σύστημα-βιβλιοθήκη ώστε να γίνεται δυνατή η **αποστολή** αιτημάτων ή/και η **ανταλλαγή** δεδομένων με άλλα λογισμικά.
- Με 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 Αρι 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 διαβάσει ξανά και ξανά

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

top-level code in a Node.js module: it's code that is executed when the module is first loaded by Node.js. In a Node.js file, code that is not inside a function or a block is considered top-level code.

```
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

```
API > JS 2nd_api.js > server > http.createServer() callback
14
15
16 //create a server
17 //creatServer method will accept a callback function that will fire
18 //every time a new request hits our server
19 const server= http.createServer((req, res) => {
20   //we store the requested url in a variable
21   const path_name= req.url;
22   console.log(path_name);
23
24   if(path_name==='/' || path_name==='all'){
25     res.end('Here are all Countries');
26   }else if(path_name==='api'){
27     //lets send back the data we have parsed in the top level code
28     //we need to let the browser know that we send back json
29
30     res.writeHead('200',{'Content-Type': 'application/json'});
31     res.end(data); //end the response
32   }else{
33     //inspect web page and check the console
34     //we can also send headers, a piece of info regarding content
35     res.writeHead('404',{'Content-Type': 'text/html'});
36     res.end('<h2> 404 <br> Page not Found </h2>');
37   }
38   //res.end(); //end the response
39 });
```

To be continued...