



Θέματα Προγραμματισμού Διαδικτύου ~ APIs, JSON, AJAX ~

Στελιος Σφακιανάκης
Εαρινό 2020



Αυτή η εργασία χορηγείται με άδεια Creative Commons
Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή

JSON



Data Formats

- Όταν ανταλλάσσονται δεδομένα μεταξύ 2 “προγραμμάτων” θα πρέπει να έχουμε προσδιορίσει εκτός από το πρωτόκολλο (π.χ. HTTP) και το data format
- Το format ορίζει πώς τα δεδομένα θα αναπαρασταθούν σε bytes για να σταλούν από τον “αποστολέα” στον “παραλήπτη”
- Δύο μεγάλες κατηγορίες formats: **Binary** και **Text**-based
 - Η διαφορά τους π.χ. στο πώς αναπαριστούν αριθμούς. Για παράδειγμα σε ένα binary format ο ακέραιος 1073741824 μπορεί να σταλεί ως 32bit (*4 bytes*) ενώ σε ένα text format ως σειρά χαρακτήρων ‘1’, ‘0’, ‘7’, ‘3’, ... οπότε θα χρειαστούμε *10 bytes* (όσα και τα ψηφία του αριθμού)
- Τα πιο γνωστά text data exchange formats είναι η **XML** και το **JSON**

JavaScript Object Notation



- Το πιο δημοφιλές format ανταλλαγής δεδομένων στον Ιστό
- Εμπνευσμένο από τον τρόπο γράφουμε τα “objects” στην Javascript
- Εύκολο, εύχρηστο (text-based), και σχετικά γρήγορο και “ελαφρύ”
- Υποστήριξη από όλες τις γλώσσες προγραμματισμού (όχι μόνο Javascript)
- ISO/IEC 21778:2017 Standard: <https://www.iso.org/standard/71616.html>
- Content type: **application/json** (ενώ π.χ. της HTML είναι **text/html**)



JSON rules

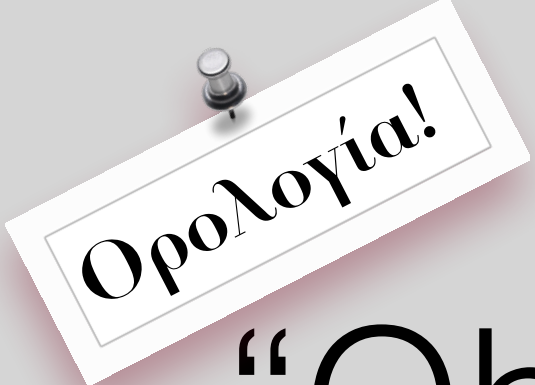
- Βασικοί τύποι : Αριθμοί, Booleans (**true/false**), Strings (με διπλά εισαγωγικά)
- Σύνθετοι τύποι: Arrays (με “τετράγωνες” αγκύλες **[..]**) και “objects” (με αγκύλες **{..}**)
 - Τα Objects έχουν “κλειδιά” (keys) που είναι Strings και values που μπορεί να οτιδήποτε (βασικοί ή σύνθετοι τύποι), χωρισμένα με “:” και κόμματα
- Αν λείπει μια τιμή μπορούμε να βάλουμε το **null**

```
{  
  "firstName": "John",  
  "lastName": "Smith",  
  "isAlive": true,  
  "age": 27,  
  "address": {  
    "streetAddress": "21 2nd Street",  
    "city": "New York",  
    "state": "NY",  
    "postalCode": "10021-3100"  
  },  
  "children": [{"firstname": "Sam", "age": 7}, {"firstname": "Mary", "age": 3}]  
}
```



JSON to Data and back

- Ένα JSON “μήνυμα” είναι String οπότε χρειαζόμαστε 2 συναρτήσεις να υπάρχουν σε μια γλώσσα προγ/μου:
 - Μετατροπή από JSON σε δομές της γλώσσας (“**decoding**”)
 - και το ανάποδο (“**encoding**”)
- Για την Javascript:
 - `JSON.parse(<json-str>)`: μετατρέπει το `<json-str>` σε τύπο της Javascript (JS Object, array, κλπ)
 - `JSON.stringify(<data>)`: μετατρέπει τα δεδομένα `<data>` της Javascript σε JSON
- Για την PHP:
 - `json_decode(<json-str>)`: μετατρέπει το `<json-str>` σε τύπο της PHP
 - `json_encode(<data>)`: μετατρέπει τα δεδομένα `<data>` της PHP σε JSON
- Το “mapping” μεταξύ JSON και δομές της γλώσσας είναι λίγο πολύ το ίδιο (για “δυναμικές” γλώσσες προγ/μου): JSON Arrays σε απλά arrays, JSON Objects σε “associative arrays”



“Objects” in Javascript ..

- Εναλλακτικά ονόματα σε άλλες γλώσσες προγ/μου:
- (Associative) Arrays: PHP
 - Π.χ. `$arr = array("name"=>"Eleni", "age"=>27);`
- Maps : C++, Java, Go, Scala, Clojure, ...
- Dictionaries : Python, .Net, Swift, ...
- Hashes : Perl, Ruby, ...

Application Programming Interfaces (APIs)



APIs

- “Διεπαφή Προγραμματισμού Εφαρμογών”
- Μια δυνατότητα που προσφέρει το προγραμματιστικό “περιβάλλον” ή μια *οντότητα* (server, βιβλιοθήκη, website, κλπ) και η οποία μας επιτρέπει να ζητήσουμε μια συγκεκριμένη “λειτουργία” ή να ανταλλάξουμε δεδομένα
- “διεπαφή” == “διασύνδεση”
- “εφαρμογών”: για την υλοποίηση εφαρμογών για χρήστες



Παραδείγματα APIs

- Javascript (Browser) APIs (HTML5):
 - Εχουμε δει ήδη το DOM, δηλ. `document.getElementById`, `document.createElement`, ...
 - Storage APIs (localStorage, sessionStorage)
 - Geolocation API
 - XMLHttpRequest
 - Και πολλά άλλα: <https://developer.mozilla.org/en-US/docs/Web/API>
- Web (Third-party) APIs, όπου μπορούμε με χρήση του HTTP να ανταλλάξουμε δεδομένα με ένα “ξένο” website, π.χ.:
 - Facebook APIs: <https://developers.facebook.com/docs/>
 - Twitter APIs: <https://developer.twitter.com/en/docs>
 - Google Maps APIs: <https://cloud.google.com/maps-platform/>
- Javascript libraries, π.χ.
 - Google Charts: <https://developers.google.com/chart/>
 - moment.js: <https://momentjs.com/>

I. Browser APIs



1. Storage APIs

- Οι νεότερες εκδόσεις όλων των browsers υποστηρίζουν το Storage API για να κρατάμε δεδομένα στον Browser (δηλ. Client side), σε 2 μορφές:
 - Το **sessionStorage** προσφέρει αποθήκευση των δεδομένων προσωρινά, μέχρι να κλείσει ο ο χρήστης τον browser ή το συγκεκριμένο tab της σελίδας
 - Το **localStorage** προσφέρει (σχεδόν) μόνιμη αποθήκευση των δεδομένων. Επίσης αν 2 ή περισσότερα το ίδιο website φορτωθεί σε 2 ή περισσότερα tabs, αυτά θα “βλέπουν” το ίδιο **localStorage**!
- Τα δεδομένα που αποθηκευονται και με τους 2 τρόπους δεν στέλνονται στο server (όπως τα cookies) αλλά μένουν τοπικά.
- Υπάρχει ένα μέγιστο μέγεθος Storage ανα σελίδα που εξαρτάται από τον Browser, αλλά συνήθως είναι 5MB (Safari) με 10MB (Firefox/Chrome)



Storage API

- Το API είναι το ίδιο για το sessionStorage και το localStorage
- Επιτρέπει την αποθήκευση ζεύγων “κλειδιών” και “τιμών” (key-value pairs).
- Κάθε key και κάθε value πρέπει να είναι string
- Μέθοδοι:
 - `setItem(<key>, <value>)` : αποθηκεύει το `<value>` με το `<key>`
 - `getItem(<key>)` : επιστρέφει την τιμή που αντιστοιχεί στο `<key>`
 - `removeItem(<key>)` : σβηνει το `<key>` και την αντίστοιχη τιμή
 - `clear()` : σβήνει όλα τα περιεχόμενα (δηλ. όλα τα key-value pairs)
 - `key(<index>)` : επιστρέφει το key στη θέση `<index>` (όπου το `<index>` είναι ακέραιος)
 - `length` : επιστρέφει το πλήθος των αποθηκευμένων “ζευγών”

Storage στα “Inspector tools”



The screenshot shows the Hellenic Mediterranean University (HMU) website in a web browser. The browser's address bar shows the URL `https://hmu.gr/`. The website's header includes the HMU logo and navigation links: **Το Πανεπιστήμιο**, **Σπουδές**, **Σχολές και Τμήματα**, and **Διεθνείς Σχέσεις**. The main content area features a large image of the university building.

The Chrome DevTools Application tab is open, showing the **Storage** section. The **Local Storage** for `https://hmu.gr` is expanded, displaying a table of stored data:

Key	Value
date	June 10
user	ssfak

The **Console** tab is also open, showing the following JavaScript commands and their output:

```
> localStorage.setItem('user', 'ssfak')
< undefined
> localStorage.setItem('date', 'June 10')
< undefined
> |
```




2. Geolocation API

- Με αυτό το API μπορούμε να βρούμε την ακριβή γεωγραφική τοποθεσία του χρήστη (“γεω-εντοπισμός”)
- Είναι προσβάσιμο μέσω του **navigator.geolocation** και μπορούμε να ανακτήσουμε τις γεωγραφικές συντεταγμένες (γεωγραφικό πλάτος και μήκος, latitude - longitude)
- Όμως: Υπάρχουν ζητήματα ιδιωτικότητας στη χρήση του geolocation API, οπότε οι Browsers συνήθως ζητάνε την άδεια του χρήστη πριν δώσουν πρόσβαση σε αυτή την πληροφορία





2. Geolocation API

- Το API αποτελείται από μια και μόνη μέθοδο: `navigator.geolocation.getCurrentPosition` στην οποία περνάμε ένα `callback` (συνάρτηση).
- Αν πετύχει ο εντοπισμός και ο χρήστης το επιτρέψει, καλείται η συνάρτησή μας με μια παράμετρο τύπου `GeolocationPosition` που περιέχει τις συντεταγμένες (coordinates):

```
// Παράδειγμα:  
navigator.geolocation.getCurrentPosition(function(pos) {  
    console.log(pos.coords.latitude);  
    console.log(pos.coords.longitude);  
});
```




3. XMLHttpRequest

- Επιτρέπει να κάνουμε κλήσεις HTTP στο server με Javascript — χωρίς να γίνει reload της σελίδας!
 - Και επομένως μπορούμε να κάνουμε δυναμικές ενημερώσεις στη σελίδα μας βάσει των server-side δεδομένων σχεδόν “μαγικά”!
- Αρχικά χρησιμοποιήθηκε για την μεταφορά δεδομένων με XML format. Εναλλακτικό όνομα: “Asynchronous Javascript And XML” -> **AJAX**
- Η καλύτερη ιδέα της Microsoft!

```
let xhr = new XMLHttpRequest();
xhr.responseType = 'json';
xhr.onload = function () {
    let data = xhr.response;
    console.log(data)
}
xhr.open('GET', '/json.php');
xhr.send();
```



XMLHttpRequest API

- `xhr = new XMLHttpRequest()` : Δημιουργεί ένα XMLHttpRequest αντικείμενο για να κάνουμε μια κλήση στο server
- `xhr.open(<method>, <url>)` : Προσδιορίζει ποια HTTP μέθοδος θα χρησιμοποιηθεί (π.χ. "GET") και σε ποιο URL θα σταλεί η αίτηση
- `xhr.onload = function() {...}` : Δίνουμε ένα callback που θα κληθεί όταν ο browser πάρει την απάντηση (δεδομένα) της HTTP κλήσης από τον server. Μέσα στη συνάρτηση μπορούμε με το `this.response` (ή `this.responseText`) να πάρουμε τα δεδομένα που έστειλε ο server
 - Αν ο server στέλνει JSON, μπορούμε να θέσουμε `xhr.responseType = "json"` και μετά το `xhr.response` θα έχει τα decoded data (οπότε δεν χρειάζεται να καλέσουμε την `JSON.parse`)
- `xhr.send()` : Εκτελεί την HTTP κλήση / στέλνει το HTTP request

II. Web APIs



Τι είναι ένα Web API

- Χρήση του HTTP και άλλων τεχνολογιών του Ιστού (π.χ. JSON) για να “τραβήξουμε” (ή να στείλουμε) πληροφορία από ένα τρίτο website
 - Π.χ. από το <https://exchangeratesapi.io/> μπορούμε να παίρνουμε την τρέχουσα συναλλαγματική ισοτιμία των νομισμάτων της Ευρωπαϊκής Κεντρικής Τράπεζας
- Αν χρησιμοποιήσουμε περισσότερα του ενός web APIs φτιάχνουμε ένα “mash-up”
 - Για παράδειγμα, παίρνουμε τα τοπικά νέα (ειδήσεις) από ένα ειδησεογραφικό API και τη λίστα με τις πιο πρόσφατες ταινίες και τα αντίστοιχα reviews από ένα movies API



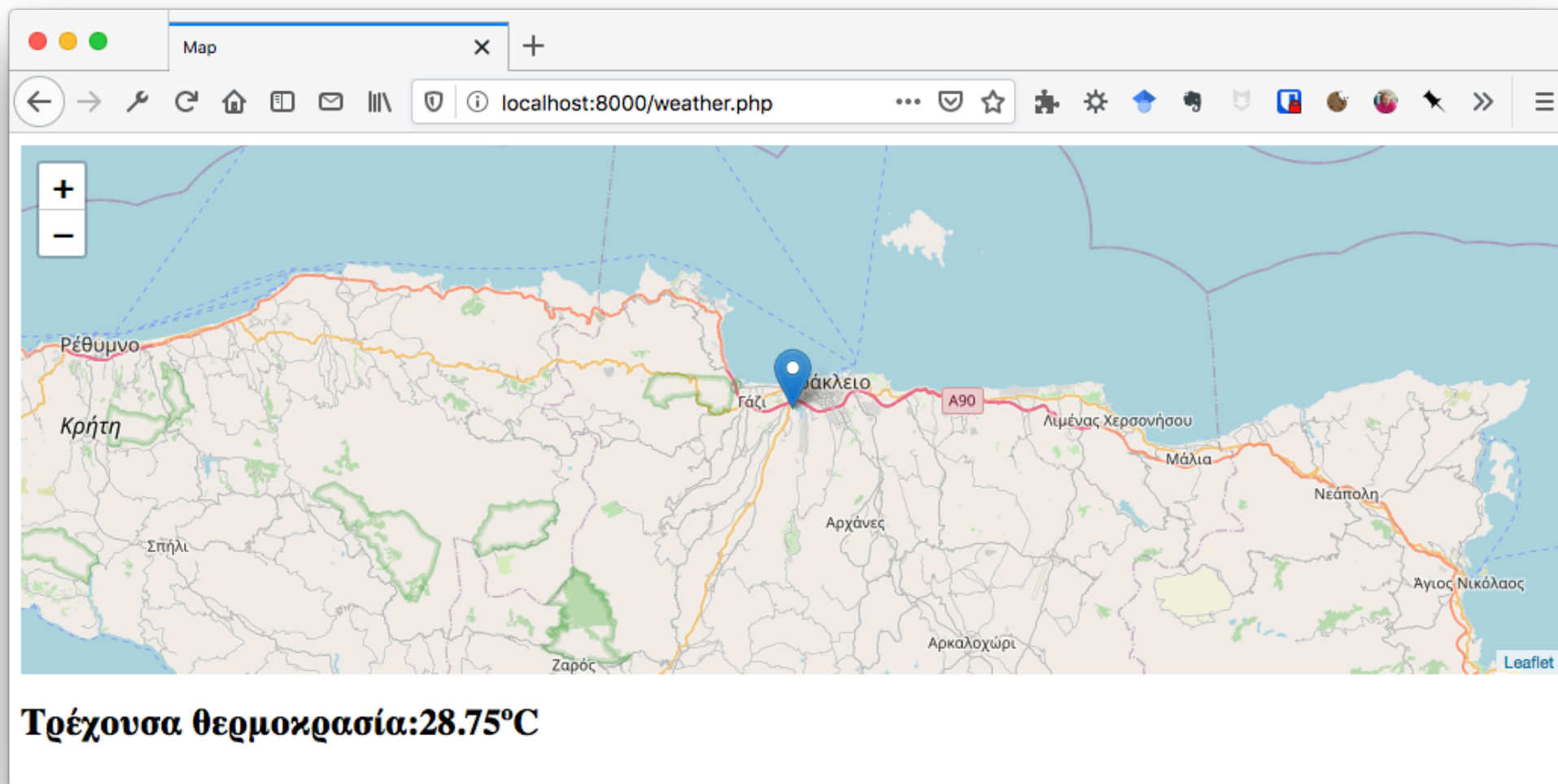
- GET <https://api.exchangeratesapi.io/latest>

```
{
  "rates": {
    "CAD": 1.5187,
    "HKD": 8.753,
    ...
    "CHF": 1.077,
    "SGD": 1.5708,
    "PLN": 4.4464,
    "BGN": 1.9558,
    "TRY": 7.6859,
    "CNY": 8.0025,
    "NOK": 10.5418,
    "NZD": 1.7375,
    "ZAR": 18.8935,
    "USD": 1.1294,
    ...
  },
  "base": "EUR",
  "date": "2020-06-09"
}
```

Παραδείγματα Εφαρμογών



Εφαρμογή 1: εμφάνιση χάρτη με την τρέχουσα τοποθεσία του χρήστη και την θερμοκρασία



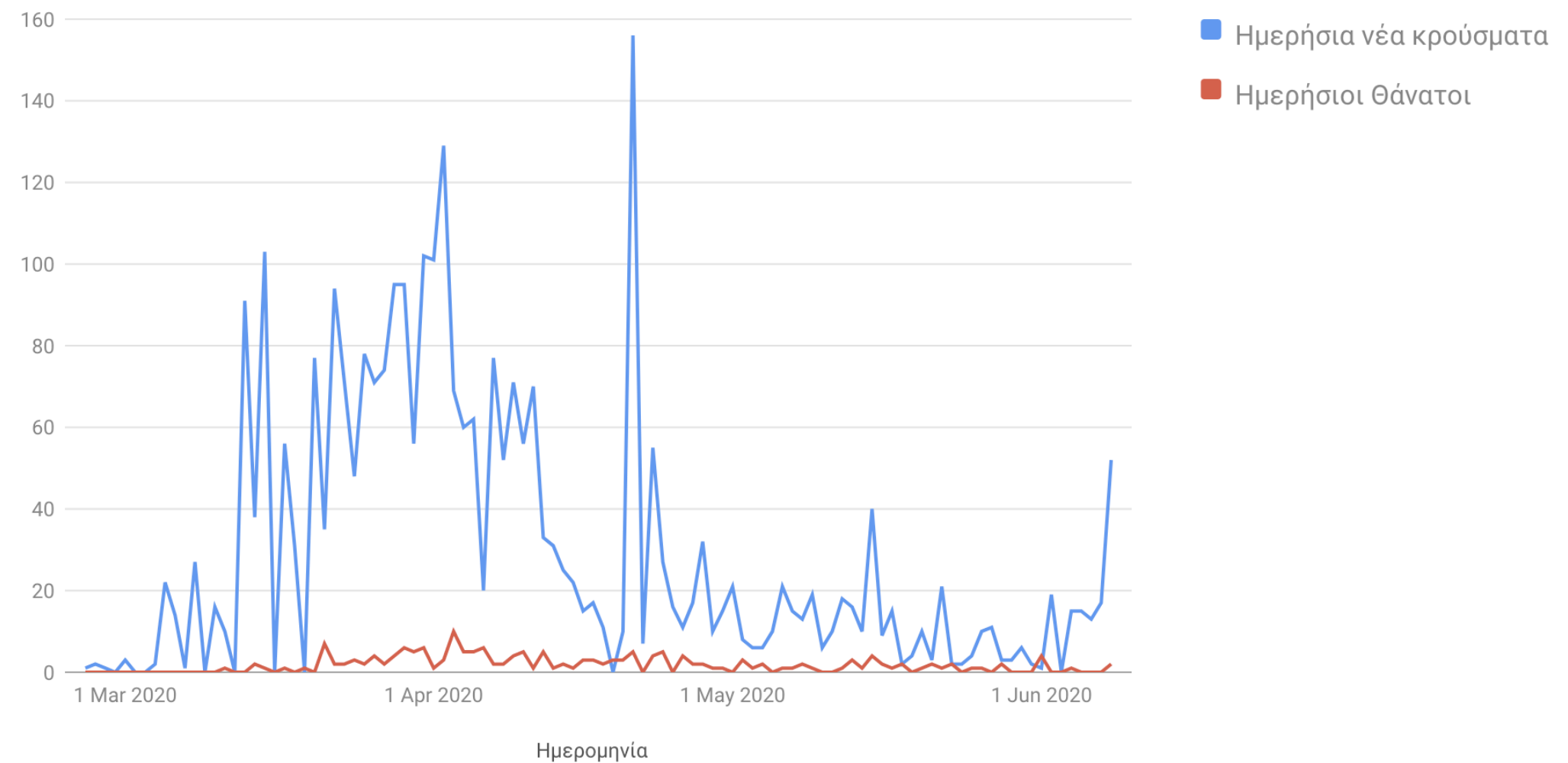


- Πώς θα βρούμε την τρέχουσα τοποθεσία του χρήστη?
 - Geolocation API!
- Πώς θα πάρουμε πληροφορίες για τον καιρό σε αυτή την τοποθεσία?
 - OpenWeatherMap: <https://openweathermap.org/api>
- Πώς θα ενσωματώσουμε έναν χάρτη μέσα στη σελίδα μας?
 - Google Maps, αλλά πρέπει να δώσουμε πιστωτική κάρτα!
 - OpenStreetMap : <https://www.openstreetmap.org>
 - Leaflet.js : <https://leafletjs.com>



Εφαρμογή 2: Γράφημα της εξέλιξης των κρουσμάτων κορωνοϊού στην Ελλάδα

Κρούσματα COVID19 στην Ελλάδα





- Πώς θα βρούμε τα δεδομένα?
 - “Country Timeline” API από το <https://thevirustracker.com/api>
 - Περισσότερα: <https://covid-19-apis.postman.com>
- Πώς θα δημιουργήσουμε το διάγραμμα μέσα στη σελίδα?
 - Google Charts, line chart: <https://developers.google.com/chart/interactive/docs/gallery/linechart?hl=en>
 - Εναλλακτικά: <https://www.chartjs.org> <https://www.highcharts.com/>



- Δείτε τα στο e-class:
- Κορωνοϊός: <https://eclass.hmu.gr/modules/document/file.php/TP327/ΘΕΩΡΙΑ/covid.html>
- Χάρτης: <https://eclass.hmu.gr/modules/document/file.php/TP327/ΘΕΩΡΙΑ/map.html>