

# Κατασκευή REST API και Web Εφαρμογής

Πασχαλίδης Πάυλος  
ΤΕΙ ΘΕΣΣΑΛΙΑΣ  
msf7415012@teilar.gr

## Περίληψη

Το αντικείμενο της παρούσας αναφοράς είναι η επεξήγηση της διαδικασίας για την δημιουργία μιας αλληλεπιδραστικής web εφαρμογής που τροφοδοτείται με δεδομένα μέσω ενός web API τύπου REST στα πλαίσια του μαθήματος Προηγμένες Web Εφαρμογές του μεταπτυχιακού προγράμματος σπουδών με τίτλο "Μηχανική Λογισμικού για Διαδικτυακές και Φορητές Εφαρμογές" του τμήματος Μηχανικών Πληροφορικής του ΤΕΙ Θεσσαλίας.

## Περιεχόμενα

<b>1</b>	<b>Εργαλεία Και Τεχνικές</b>	<b>2</b>
1.1	Πλαίσιο Λογισμικού - PHP	2
1.2	Περιβάλλον Ανάπτυξης	2
1.3	Σύστημα Ελέγχου Εκδόσεων	3
1.4	Βάση Δεδομένων	3
1.5	Καλές Πρακτικές API	4
1.5.1	API Resources	4
1.5.2	Fields	4
1.5.3	Where	5
1.5.4	Pagination	5
1.5.5	Aggregate	5
1.5.6	Group By	6
1.5.7	Between	6
1.5.8	Formats	6
1.5.9	HTTP error	7
1.6	Πλαίσιο Λογισμικού - CSS,JS	8

# 1 Εργαλεία Και Τεχνικές

## 1.1 Πλαίσιο Λογισμικού - PHP

Για την υλοποίηση του REST API χρησιμοποιήθηκε το framework<sup>1</sup> **Lumen-Laravel** [6] που ενδείκνυται για το σκοπό αυτό. Ωστόσο τα προαπαιτούμενα για την χρήση του framework είναι:

- PHP >= 5.6.4
- OpenSSL PHP Extension
- PDO PHP Extension
- Mbstring PHP Extension

τα οποία ήταν διαφορετικά στο τοπικό υπολογιστή μου (PHP 5.61) και για το λόγο αυτό χρησιμοποιήθηκε το εικονικό περιβάλλον **Laravel Homestead** [2] με τη χρήση VirtualBox<sup>2</sup> και Vagrant<sup>3</sup>.

## 1.2 Περιβάλλον Ανάπτυξης

Το **Laravel Homestead**[2] είναι ένα επίσημο πακέτο Vagrant box που παρέχει ένα περιβάλλον ανάπτυξης χωρίς να απαιτείται η εγκατάσταση PHP, Web Server και άλλα προγράμματα λογισμικού στο τοπικό υπολογιστή.

Περιλαμβάνει τα εξής λογισμικά:

- Ubuntu 16.04
- Git
- PHP 7.1
- Nginx
- MySQL
- MariaDB
- Sqlite3
- Postgres
- Composer
- Node (With Yarn, PM2, Bower, Grunt, and Gulp)
- Redis
- Memcached
- Beanstalkd

---

<sup>1</sup>Software framework, a reusable set of libraries or classes for a software system

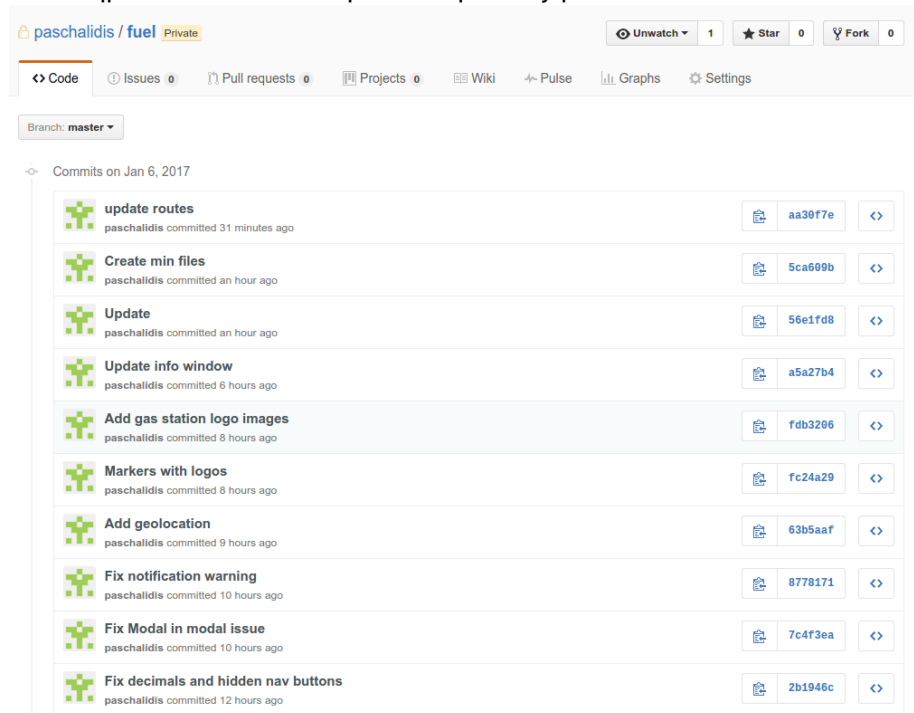
<sup>2</sup>Oracle VM VirtualBox, supports the creation and management of virtual machines

<sup>3</sup>Open-source software for building and maintaining portable virtual development environments

### 1.3 Σύστημα Ελέγχου Εκδόσεων

Για την διαχείριση των εκδόσεων του Rest API και του Web Application χρησιμοποιήθηκε το git<sup>4</sup> και ένα private repository απο το git-hub<sup>5</sup>

Σχήμα 1: Εικόνα από το private repository με τα τελευταία commits.



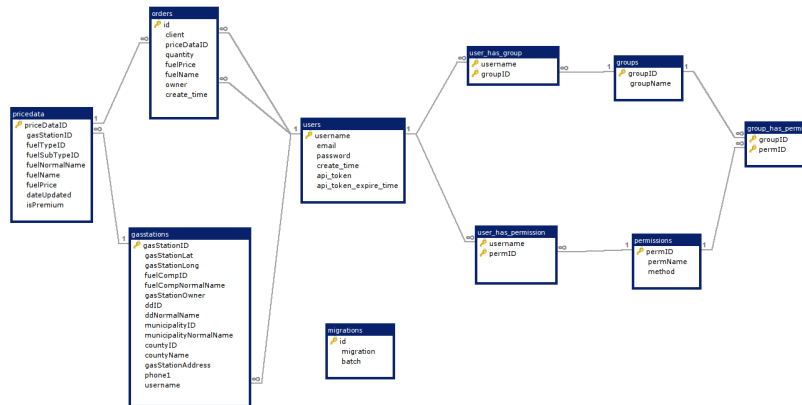
### 1.4 Βάση Δεδομένων

Για λόγους ευχρηστίας έγινε πρόσθεση του πεδίου **id** στους πίνακες **pricedata** και **gasstations**, ενώ για τα δικαιώματα των χρηστών (πελάτης, πρατηριούχος) υλοποιήθηκε με το σχεσιακό μοντέλο **users-and-rights-schema** περισσότερες λεπτομέρειες στο σχεσιακό σχήμα.2. Επιπλέον δημιουργήθηκαν τρία views: user-permission, fuel-analytics, pricedata-view.

<sup>4</sup>Free software, used for software development and other version control tasks

<sup>5</sup>Web-based git repository hosting service

Σχήμα 2: Εικόνα από το Σχεσιακό Μοντέλο.



## 1.5 Καλές Πρακτικές API

Χρησιμοποιήθηκαν καλές πρακτικές σχεδίασης όπως αυτές παρουσιάστηκαν στις διαλέξεις και για τον έλεγχο και την υλοποίηση των resources η εφαρμογή chrome-postman [1]. API URL: **domain/api/v1/**

### 1.5.1 API Resources

- GET - gasstations
- GET - gasstations/count
- GET - gasstations/id/pricedata
- GET - pricedata
- GET - gasstations/analytics
- POST - login
- POST - register
- PUT - pricedata/id
- POST - orders
- GET - orders

### 1.5.2 Fields

Επιλογή συγκεκριμένων πεδίων με την παράμετρο fields, πχ fields=gasStationID,gasStationLat.

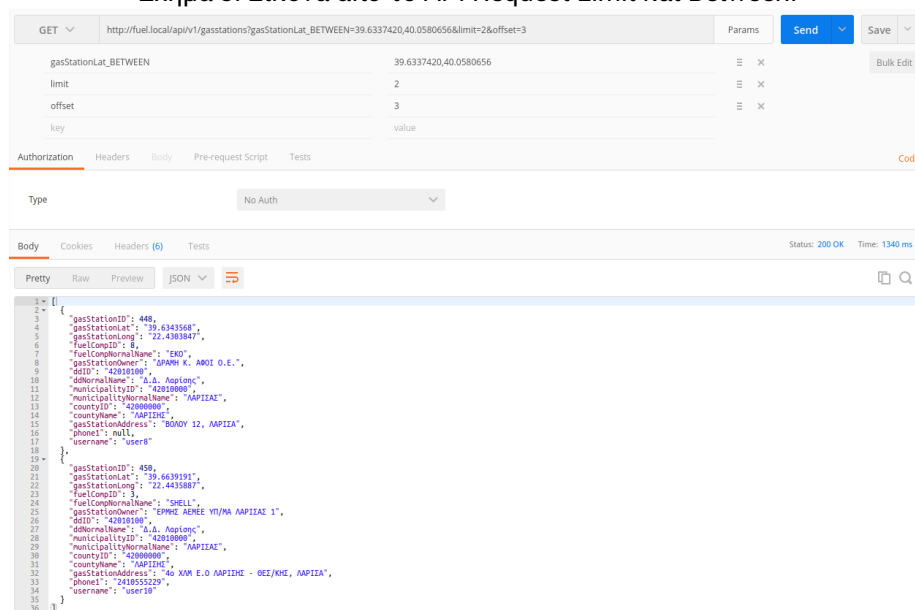
### 1.5.3 Where

Υλοποίηση where statement με το όνομα του πεδίου και την τιμή ή τις τιμές για IN, πχ gasStationID=1 ή gasStationID=1,2,3. 5

### 1.5.4 Pagination

Χρήση pagination με τις εντολές limit και offset και την τιμή του καθενός, πχ limit=10,offset=100. 3

Σχήμα 3: Εικόνα από το API Request Limit και Between.



### 1.5.5 Aggregate

Υλοποίηση συναρτήσεων max, min, avg, count πχ max=fuelPrice. 4

Σχήμα 4: Εικόνα από το API Request με συναρτήσεις και group by.

The screenshot displays an API client interface. The top section shows a GET request to the URL `http://fuel.local/api/v1/pricedata?max=fuelPrice&min=fuelPrice&avg=fuelPrice&groupby=fuelTypeID&count=priceDataID`. Below the URL bar, there are tabs for Authorization, Headers, Body, Pre-request Script, and Tests. The Body tab is selected, showing a JSON response in Pretty format. The response is a list of objects, each representing a fuel type with its maximum, minimum, and average fuel price, and the count of price data points.

max	min	avg	groupby	count
fuelPrice	fuelPrice	fuelPrice	fuelTypeID	priceDataID
				value

```

1  {
2    "max_fuelPrice": "1.535",
3    "min_fuelPrice": "1.355",
4    "avg_fuelPrice": "1.4282649",
5    "count_priceDataID": 151
6  },
7  {
8    "max_fuelPrice": "1.875",
9    "min_fuelPrice": "1.459",
10   "avg_fuelPrice": "1.624692",
11   "count_priceDataID": 65
12  },
13  {
14    "max_fuelPrice": "1.478",
15    "min_fuelPrice": "1.445",
16    "avg_fuelPrice": "1.4616099",
17    "count_priceDataID": 3
18  },
19  {
20    "max_fuelPrice": "1.219",
21    "min_fuelPrice": "1.025",
22    "avg_fuelPrice": "1.122398",
23    "count_priceDataID": 213
24  },
25  {
26    "max_fuelPrice": "1.029",
27    "min_fuelPrice": "0.848",
28    "avg_fuelPrice": "0.9159483",
29    "count_priceDataID": 67
30  },
31  {
32    "max_fuelPrice": "0.759",
33    "min_fuelPrice": "0.618",
34    "avg_fuelPrice": "0.675688",
35    "count_priceDataID": 30
36  },
37  {
38    "max_fuelPrice": "0.968",
39    "min_fuelPrice": "0.868",
40    "avg_fuelPrice": "0.9136876",
41    "count_priceDataID": 79
42  },
43  }
44  ]
  
```

### 1.5.6 Group By

Επιλογή για groupby με την παράμετρο groupby, πχ groupby=fuelTypeID.

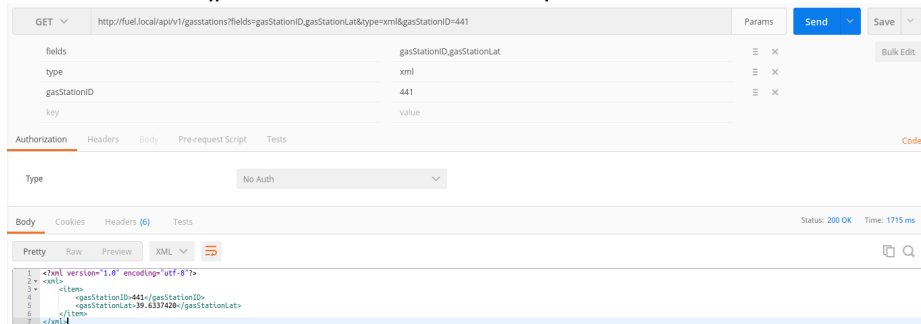
### 1.5.7 Between

Υλοποίηση του between, πχ gasStationLat\_BETWEEN=39.6337420,40.0580656.

### 1.5.8 Formats

Επιλογή μορφή δεδομένων με την παράμετρο type=json ή xml.

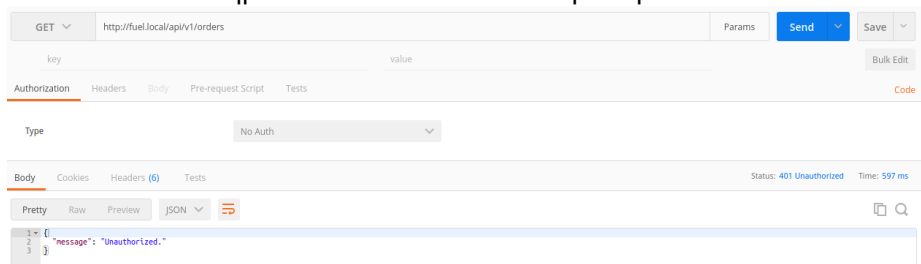
Σχήμα 5: Εικόνα από το API Request XML Format.



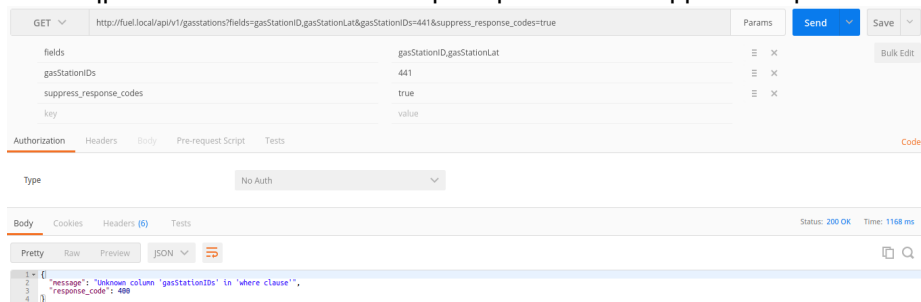
### 1.5.9 HTTP error

Με την παράμετρο `suppress_response_code` το API επιστρέφει πάντα `http status 200` με μήνυμα λάθους και το `response status` που θα έπαιρνε. Χωρίς 6 Με 7

Σχήμα 6: Εικόνα από το API Request με error.



Σχήμα 7: Εικόνα από το API Request με error και suppress response.





## 1.6 Πλαίσιο Λογισμικού - CSS,JS

Για την υλοποίηση της εφαρμογής Single Page RIA χρησιμοποιήθηκε το framework **Bootstrap** [3] το οποίο παρέχει πολλές λειτουργίες σε HTML, CSS και JS. Για τις ajax κλήσης χρησιμοποιήθηκε η βιβλιοθήκη **jQuery** [4] που είναι εύχρηστη και για τα μηνύματα προς τον χρήστη το plugin **Notify.js** [5].

## Αναφορές

- [1] Development apis tool. <https://chrome.google.com/webstore/detail/postman/fhbjgbflijnjbdbggehcdcbncdddomop>. [last access 7-Jan-2017].
- [2] Development environment. <https://laravel.com/docs/5.3/homestead>. [last access 13-Dec-2016].
- [3] Html, css, js framework. <http://getbootstrap.com/>. [last access 7-Jan-2017].
- [4] Javascript library. <https://jquery.com/>. [last access 7-Jan-2017].
- [5] jquery plugin. <https://notifyjs.com/>. [last access 7-Jan-2017].
- [6] Taylor Otwell. micro-framework by laravel. <https://lumen.laravel.com>. [last access 13-Dec-2016].