



εργασία στο μάθημα

Υπηρεσίες και συστήματα Διαδικτύου

6ο ΕΞΑΜΗΝΟ, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2023-2024

Γιάννης Αντωνιάδης
(ΑΜ: it 22008)

ΜΑΪΟΣ – ΙΟΥΝΙΟΣ 2024

Περιεχόμενα

| | |
|---|---------|
| <u>1. Για το athens-water</u> | σελ. 3 |
| <u>2. Συγκέντρωση δεδομένων από το eydap.gr</u> | σελ. 4 |
| <u>3. Συγκέντρωση δεδομένων από το open-meteo.com</u> | σελ. 7 |
| <u>4. Παρατηρήσεις πάνω στα δεδομένα</u> | σελ. 9 |
| <u>5. Επιλογή δεικτών και κατηγοριοποίηση</u> | σελ. 14 |
| <u>6. Λειτουργία σε πραγματικό χρόνο και σύστημα ειδοποιήσεων</u> | σελ. 16 |
| <u>7. Εγκατάσταση του athens-water</u> | σελ. 20 |
| <u>8. Συνοπτική παρουσίαση των ροών</u> | σελ. 22 |

1. Για το athens-water

Πρόκειται για μια εφαρμογή παρακολούθησης των αποθεμάτων νερού από τα οποία υδροδοτείται η πόλη της Αθήνας, που έχει αναπτυχθεί στο πλαίσιο της εργασίας στο μάθημα 'Υπηρεσίες και Συστήματα Διαδικτύου'.

Έχει δημιουργηθεί στο περιβάλλον του Node-Red με τη λογική των διασυνδεόμενων ροών. Χρησιμοποιούνται ανοιχτά δεδομένα που διατίθενται από την ΕΥΔΑΠ μέσα από τον ιστότοπό της eydap.gr, καθώς και ανοιχτά μετεωρολογικά δεδομένα μέσα από το API του open-meteo.com. Τα δεδομένα αποθηκεύονται σε μια βάση δεδομένων SQLite.

Η λειτουργία της περιλαμβάνει την καθημερινή καταγραφή των αποθεμάτων νερού, της παραγόμενης ποσότητας πόσιμου νερού και της ποσότητας υετού που κατακρημνίζεται. Τη συγκριτική αξιολόγηση της τρέχουσας κατάστασης με τα συσσωρευμένα στοιχεία των προηγούμενων ετών. Και τέλος την αποστολή σχετικών ειδοποιήσεων μέσω ενός RabbitMQ server.

2. Συγκέντρωση δεδομένων από το eydap.gr

Κατά τη διαδικασία αναζήτησης ανοικτών πηγών δεδομένων και διαμόρφωσης ενός σχετικού θέματος/ project, είχα σαν προτεραιότητα να βρω δεδομένα που να αφορούν στην πόλη της Αθήνας. Θεωρώντας ότι αν το θέμα/ project σχετίζεται πιο άμεσα με την καθημερινότητα που ζούμε, ίσως γίνεται πιο παραστατική/ ξεκάθαρη η λειτουργία και η χρησιμότητα ενός τέτοιου project.

Τα ανοικτά δεδομένα από ελληνικές πηγές, δεν είναι στην έκταση αυτών που παρέχονται από ευρωπαϊκές ή αμερικάνικες. Αρχικά υπήρχε δυσκολία στην αναζήτηση και επιλογή μιας κατάλληλης πηγής δεδομένων. Περιστευόμενος όμως σε συνήθεις κατηγορίες με ανοικτά δεδομένα (πχ περιβάλλον, μετακινήσεις, υγεία, ενέργεια κ.α.), ίσως και λίγο τυχαία, βρέθηκα στις πληροφορίες που παρέχει το site της ΕΥΔΑΠ και μου κίνησαν το ενδιαφέρον. Πιο συγκεκριμένα, στις πληροφορίες για τα αποθέματα νερού και την παραγωγή πόσιμου νερού σε ημερήσια βάση.

Η ΕΥΔΑΠ διαθέτει τέσσερις ταμιευτήρες νερού (λίμνες Μαραθώνα, Υλίκης, Μόρνου και Εύηνου) από όπου κυρίως προέρχεται το νερό που επεξεργάζεται και διαθέτει (ένα δευτερεύον και εφεδρικό ρόλο έχουν και οι γεωτρήσεις σε υπόγειους υδατικούς πόρους).

Τα καθημερινά αποθέματα που υπάρχουν στους 4 ταμιευτήρες είναι προσβάσιμα σε σχετική φόρμα στο site της ΕΥΔΑΠ: <https://www.eydap.gr/TheCompany/Water/Savings/>

Ο χρήστης επιλέγει την ημ/νία που τον ενδιαφέρει και λαμβάνει το αποτέλεσμα σε μορφή html:

| ΤΑΜΙΕΥΤΗΡΕΣ: ΑΠΟΛΗΨΙΜΟ ΑΠΟΘΕΜΑ (m ³) Από 22/5/2024 Έως 23/5/2024 | | | | | |
|---|------------|------------|-------------|-------------|-------------|
| Ημερομηνία | ΕΥΗΝΟΣ | ΜΑΡΑΘΩΝΑΣ | ΜΟΡΝΟΣ | ΥΛΙΚΗ | ΣΥΝΟΛΟ |
| 22/05/2024 | 67.414.000 | 26.047.000 | 442.235.000 | 338.832.000 | 874.528.000 |
| 23/05/2024 | 67.414.000 | 26.132.000 | 441.128.000 | 338.644.000 | 873.318.000 |

Επίσης, η ΕΥΔΑΠ διαθέτει τέσσερις μονάδες επεξεργασίας νερού (ΜΕΝ Γαλατσίου, Αχαρνών, Πολυδενδρίου, Ασπρόπυργου), όπου παράγονται οι ποσότητες του πόσιμου νερού που διαθέτει για την ύδρευση της Αθήνας.

Κατά παρόμοιο τρόπο, οι καθημερινές ποσότητες παραγωγής πόσιμου νερού από τις 4 ΜΕΝ είναι προσβάσιμες σε σχετική φόρμα στο site της ΕΥΔΑΠ:

<https://www.eydap.gr/TheCompany/Water/DrinkingWaterProduction>

Ο χρήστης επιλέγει την ημ/νία που τον ενδιαφέρει και λαμβάνει το αποτέλεσμα σε μορφή html:

| Παραγωγή νερού από ΜΕΝ (m ³) Από 21/5/2024 Έως 22/5/2024 | | | | | |
|---|-------------|---------|---------|---------|-----------|
| Ημερομηνία | ΑΣΠΡΟΠΥΡΓΟΣ | ΓΑΛΑΤΣΙ | ΚΙΟΥΡΚΑ | ΜΕΝΙΔΙ | ΣΥΝΟΛΟ |
| 21/05/2024 | 131.020 | 334.564 | 139.389 | 555.391 | 1.160.364 |
| 22/05/2024 | 132.730 | 342.627 | 124.210 | 582.403 | 1.181.970 |

Βέβαια οι html φόρμες δεν είναι ο πιο πρόσφορος τρόπος για επικοινωνία μεταξύ συστημάτων, έτσι διαβάζοντας τον πηγαίο κώδικα πίσω από τις φόρμες βρίσκουμε τα urls στα οποία μπορούμε να κάνουμε αυτόματα requests μέσα από κώδικα:

Για τα αποθέματα νερού →

<https://www.eydap.gr/el/Controls/GeneralControls/SavingsDetails.aspx?DaysSpan=Day&Date=23-05-2024>

Για την παραγωγή πόσιμου νερού →

<https://www.eydap.gr/el/Controls/GeneralControls/DrinkingWaterProductionDetails.aspx?DaysSpan=Day&Date=22-05-2024>

Αυτά τα urls επιστρέφουν επίσης html:

Απόσπασμα από την html απάντηση με τα στοιχεία για τα αποθέματα νερού →

```
<table cellpadding="0" rules="all" border="1" id="grdSavings" style="width:910px;border-collapse:collapse;">
  <tr>
    <th scope="col">Ημερομηνία</th><th scope="col">ΕΥΗΘΟΣ</th><th scope="col">ΜΑΡΑΘΩΝΑΣ</th><th scope="col">ΜΟΡΙΝΟΣ</th><th scope="col">ΥΛΙΚΗ</th><th scope="col">ΣΥΝΟΛΟ</th>
  </tr><tr>
    <td>22/05/2024</td><td>67.414.000 </td><td>26.047.000 </td><td>442.235.000 </td><td>338.832.000 </td><td>874.528.000 </td>
  </tr><tr>
    <td>23/05/2024</td><td>67.414.000 </td><td>26.132.000 </td><td>441.128.000 </td><td>338.644.000 </td><td>873.318.000 </td>
  </tr>
</table>
```

Απόσπασμα από την html απάντηση με τα στοιχεία για την παραγωγή πόσιμου νερού →

```
<table cellpadding="0" rules="all" border="1" id="grdSavings" style="width:910px;border-collapse:collapse;">
  <tr>
    <th scope="col">Ημερομηνία</th><th scope="col">ΑΣΠΡΟΠΥΡΓΟΣ</th><th scope="col">ΓΑΛΑΤΣΙ</th><th scope="col">ΚΙΟΥΡΚΑ</th><th scope="col">ΜΕΝΙΔΙ</th><th scope="col">ΣΥΝΟΛΟ</th>
  </tr><tr>
    <td>21/05/2024</td><td>131.020 </td><td>334.564 </td><td>139.389 </td><td>555.391 </td><td>1.160.364 </td>
  </tr><tr>
    <td>22/05/2024</td><td>132.730 </td><td>342.627 </td><td>124.210 </td><td>582.403 </td><td>1.181.970 </td>
  </tr>
</table>
```

Με τεχνικές web scraping μπορούμε να εντοπίσουμε και να εξάγουμε τα στοιχεία που χρειαζόμαστε, δηλαδή τις τέσσερις ποσότητες αποθέματος των ταμιευτήρων (ή τις τέσσερις ποσότητες παραγόμενου νερού των μονάδων επεξεργασίας).

Στη συνέχεια αναλύουμε τα στοιχεία σε τέσσερις ξεχωριστές εγγραφές που δηλώνουν το απόθεμα για κάθε ταμιευτήρα τη συγκεκριμένη ημερομηνία (ή την παραγόμενη ποσότητα

πόσιμου νερού για κάθε ΜΕΝ τη συγκεκριμένη ημερομηνία). Αυτή είναι και η μορφή που αποθηκεύονται σε μια βάση δεδομένων SQLite. Συγκεκριμένα:

Τα αποθέματα νερού, αποθηκεύονται στο table savings με τη δομή:

```
CREATE TABLE savings (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    reservoir_id INTEGER NOT NULL,  
    date TEXT NOT NULL,  
    quantity INTEGER,  
    FOREIGN KEY( reservoir_id ) REFERENCES reservoirs( id ),  
    UNIQUE( reservoir_id, date )  
);
```

με τους ταμιευτήρες να είναι κωδικοποιημένοι στο table reservoirs:

```
CREATE TABLE reservoirs (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name_el VARCHAR(30) UNIQUE NOT NULL,  
    name_en VARCHAR(30) UNIQUE NOT NULL,  
    lat REAL,  
    lon REAL  
);
```

Με παρόμοιο τρόπο, οι ποσότητες παραγόμενου νερού αποθηκεύονται στο table production:

```
CREATE TABLE production (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    factory_id INTEGER NOT NULL,  
    date TEXT NOT NULL,  
    quantity INTEGER,  
    FOREIGN KEY( factory_id ) REFERENCES factories( id ),  
    UNIQUE( factory_id, date )  
);
```

με τις μονάδες επεξεργασίας να είναι κωδικοποιημένες στο table factories:

```
CREATE TABLE factories (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name_el VARCHAR(30) UNIQUE NOT NULL,  
    name_en VARCHAR(30) UNIQUE NOT NULL,  
    lat REAL,  
    lon REAL  
);
```

3. Συγκέντρωση δεδομένων από το open-meteo.com

Ως επιμέρους διάσταση του θέματος/ project, ήταν και η ιδέα να συσχετιστούν οι ποσότητες υετού (βροχοπτώσεων και χιονοπτώσεων) που καταγράφονται σε συγκεκριμένες περιοχές και χρονικά διαστήματα, επιδρώντας στα αποθέματα των ταμιευτήρων.

Δουλεύοντας αρχικά πάνω σε αυτό, αποδείχτηκε αρκετά πολύπλοκο και υπερέβαινε τις παρούσες δυνατότητες ανάπτυξης του θέματος. Απαιτούνταν γνώσεις από τον τομέα της υδρολογίας, εξειδικευμένες γεωγραφικές καταγραφές για λεκάνες συγκέντρωσης υδάτων από τις βροχοπτώσεις/ χιονοπτώσεις και αναλύσεις για το πως μπορούν να συσχετιστούν με τα αποθέματα των ταμιευτήρων. Επίσης λεπτομερή μετεωρολογικά δεδομένα συγκεκριμένων οριοθετημένων περιοχών (πολύγωνα) που δεν βρίσκονται εύκολα διαθέσιμα χωρίς χρεώσεις.

Ωστόσο επιλέχτηκε δοκιμαστικά, μια ενδεικτική συγκέντρωση στοιχείων υετού, από την πολύ ευρύτερη περιοχή που βρίσκονται οι ταμιευτήρες. Από οκτώ γνωστές πόλεις της Στερεάς Ελλάδας και της Αιτωλοακαρνανίας (Αθήνα, Άμφισσα, Χαλκίδα, Καρπενήσι, Λαμία, Λιβαδειά, Θήβα, Αργίριο).

Ως πηγή μετεωρολογικών δεδομένων επιλέχτηκε το open-meteo, λόγω του ότι παρέχει χωρίς χρέωση ιστορικά δεδομένα σε μεγάλο χρονικό βάθος (από το 1940 όπως ισχυρίζεται).

Στα καθημερινά requests προς το API δηλώνουμε τις γεωγραφικές συντεταγμένες των οκτώ πόλεων, τους μετεωρολογικούς δείκτες που μας ενδιαφέρουν (ελάχιστη, μέση και μέγιστη θερμοκρασία, καθώς και ποσότητα βροχόπτωσης, χιονόπτωσης και συνολικού υετού), καθώς και ένα εύρος προηγούμενων ημερών για τις οποίες θα επιστραφούν στοιχεία (ανάμεσα σε αυτές τις ημέρες είναι και η ημ/νία που κάθε φορά μας ενδιαφέρει):

```
https://archive-api.open-meteo.com/v1/archive?daily=weather_code,
temperature_2m_min,temperature_2m_mean,temperature_2m_max,
precipitation_sum,rain_sum,snowfall_sum
&timezone=Europe/Athens&past_days=7
&latitude=37.9842,38.5253,38.4625,38.9121,38.8972,38.4361,38.321,38.6263
&longitude=23.7281,22.3753,23.595,21.795,22.4311,22.875,23.3178,21.409
```

Η απάντηση από το API είναι σε μορφή JSON και περιλαμβάνει έναν πίνακα με οκτώ θέσεις, όσες και οι γεωγραφικές συντεταγμένες των πόλεων που ζητήσαμε. Η επόμενη εικόνα είναι ενδεικτική για δύο μόνο γεωγραφικές συντεταγμένες:

```
[{
  "latitude": 37.996483, "longitude": 23.709677, "generationtime_ms": 0.31006336212158203, "utc_offset_seconds": 10800,
  "timezone": "Europe/Athens", "timezone_abbreviation": "EEST", "elevation": 92.0,
  "daily_units": { "time": "iso8601", "weather_code": "wmo code", "temperature_2m_min": "°C", "temperature_2m_mean": "°C",
  "temperature_2m_max": "°C", "precipitation_sum": "mm", "rain_sum": "mm", "snowfall_sum": "cm" },
  "daily": { "time": ["2024-05-20", "2024-05-21", "2024-05-22", "2024-05-23"], "weather_code": [1, 1, 1, null],
  "temperature_2m_min": [19.3, 19.5, 20.9, null], "temperature_2m_mean": [24.2, 24.1, null, null], "temperature_2m_max":
  [29.1, 29.2, 22.0, null], "precipitation_sum": [0.00, 0.00, null, null], "rain_sum": [0.00, 0.00, null, null],
  "snowfall_sum": [0.00, 0.00, null, null] }
}, {
  "latitude": 38.558872, "longitude": 22.377718, "generationtime_ms": 0.2340078353881836, "utc_offset_seconds": 10800,
  "timezone": "Europe/Athens", "timezone_abbreviation": "EEST", "elevation": 206.0, "location_id": 1,
  "daily_units": { "time": "iso8601", "weather_code": "wmo code", "temperature_2m_min": "°C", "temperature_2m_mean": "°C",
  "temperature_2m_max": "°C", "precipitation_sum": "mm", "rain_sum": "mm", "snowfall_sum": "cm" },
  "daily": { "time": ["2024-05-20", "2024-05-21", "2024-05-22", "2024-05-23"], "weather_code": [51, 2, 0, null],
  "temperature_2m_min": [18.6, 18.9, 18.3, null], "temperature_2m_mean": [24.0, 24.4, null, null], "temperature_2m_max":
  [29.3, 30.9, 19.3, null], "precipitation_sum": [0.10, 0.00, null, null], "rain_sum": [0.10, 0.00, null, null],
  "snowfall_sum": [0.00, 0.00, null, null] }
}]
```

Από κάθε θέση φιλτράρουμε μόνο τις τιμές των μετεωρολογικών δεικτών για την ημ/νία που μας ενδιαφέρει. Έτσι έχουμε μια εγγραφή για κάθε μία από τις οκτώ πόλεις με τιμές για μια σειρά από μετεωρολογικούς δείκτες. Αυτή είναι και η μορφή που αποθηκεύονται στη βάση δεδομένων.

Συγκεκριμένα:

```
CREATE TABLE weather (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  location_id INTEGER NOT NULL,
  date TEXT NOT NULL,
  weather_code INTEGER,
  temperature_2m_mean REAL,
  temperature_2m_min REAL,
  temperature_2m_max REAL,
  precipitation_sum REAL,
  rain_sum REAL,
  snowfall_sum REAL,
  FOREIGN KEY( location_id ) REFERENCES locations( id ),
  UNIQUE( location_id, date )
);
```

με τις πόλεις/ τοποθεσίες να είναι κωδικοποιημένες στο table locations:

```
CREATE TABLE locations (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name_el VARCHAR(30) UNIQUE NOT NULL,
  name_en VARCHAR(30) UNIQUE NOT NULL,
  lat REAL NOT NULL,
  lon REAL NOT NULL
);
```

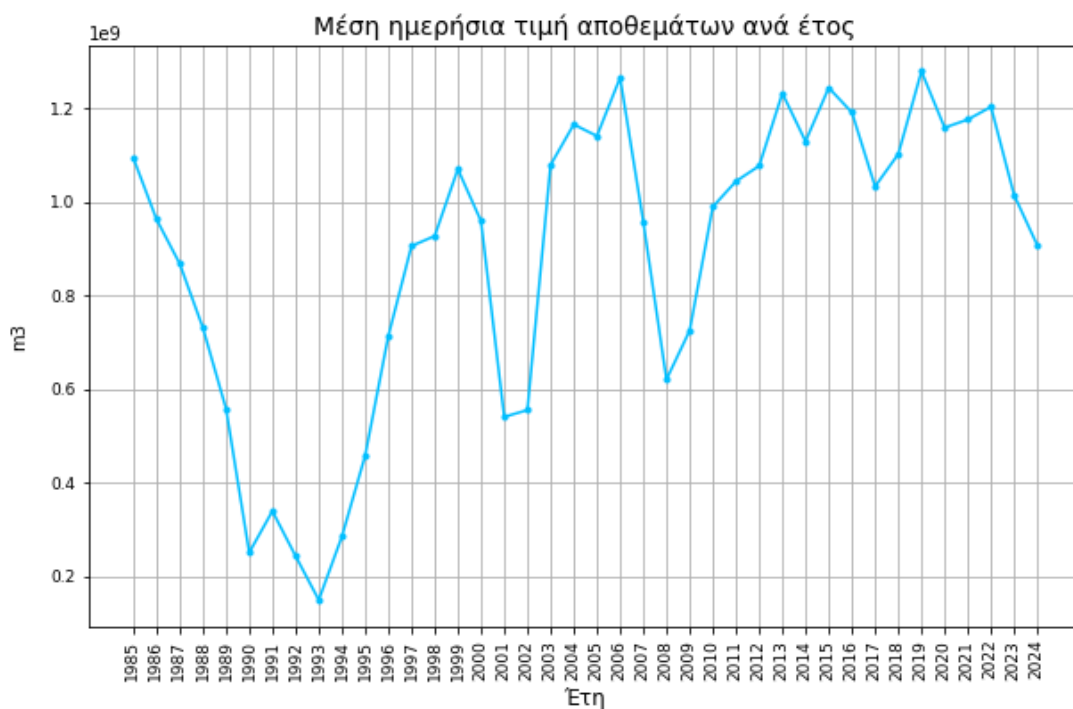
Εδώ να σημειώσω ότι όπως θα δούμε στη συνέχεια, μόνο η ποσότητα του συνολικού υετού θα χρησιμοποιηθεί για τις ανάγκες του συγκεκριμένου project. Ωστόσο κατά τη διάρκεια της εξέλιξης του project ήταν ακόμα ασαφές αν θα μπορούσε να υπάρξει επεξεργασία και να αξιοποιηθούν επιπλέον δείκτες (κυρίως οι θερμοκρασίες). Τελικά δεν χρειάστηκαν, αλλά επιλέχτηκε η διατήρησή τους για την πιθανότητα να υπάρξει κάποια περαιτέρω μελλοντική επεξεργασία.

4. Παρατηρήσεις πάνω στα δεδομένα

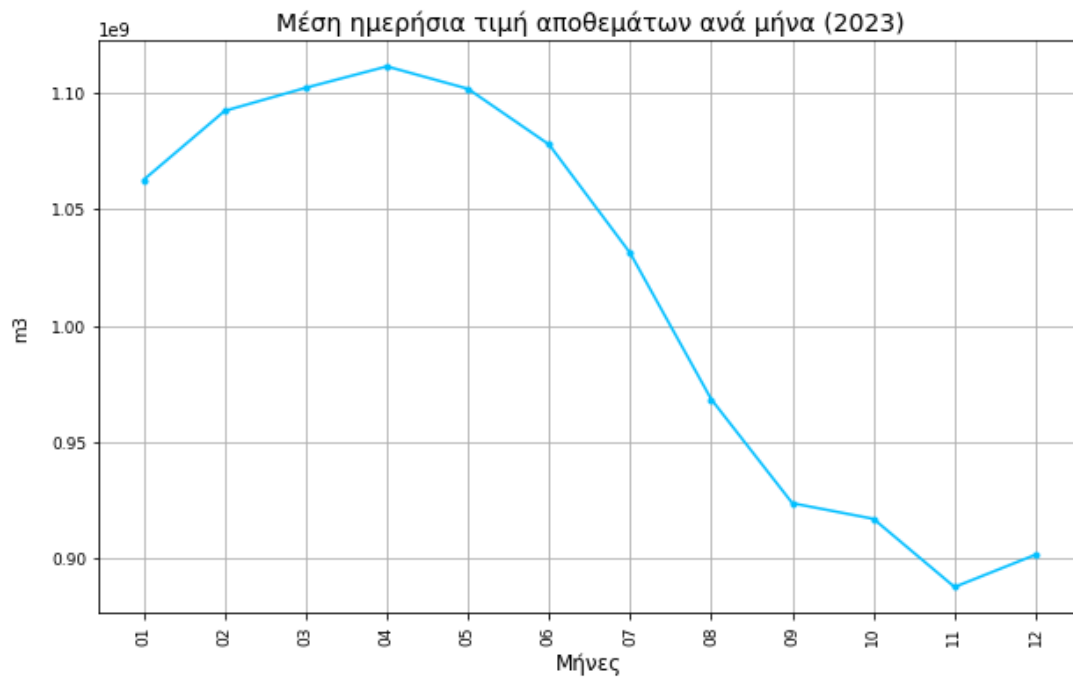
Η τρέχουσα λειτουργία του συστήματος στηρίζεται στα στοιχεία των προηγούμενων χρόνων που έχουν ήδη συγκεντρωθεί (με scripts σε nodejs). Στη βάση δεδομένων έχουν αποθηκευτεί όλα τα ημερήσια αποθέματα από το 1985 έως σήμερα (το χρονικό διάστημα που παρέχει η ΕΥΔΑΠ), όλες οι ημερήσιες ποσότητες παραγωγής νερού από το 1996 έως σήμερα (το χρονικό διάστημα που παρέχει η ΕΥΔΑΠ), καθώς και οι μετεωρολογικοί δείκτες για τις οκτώ πόλεις που εξετάζουμε, για κάθε ημέρα από το 1996 μέχρι σήμερα.

Από την συγκέντρωση αυτών των στοιχείων και την απεικόνισή τους σε διαγράμματα (με scripts σε python), μπορούμε γενικά να παρατηρήσουμε ότι:

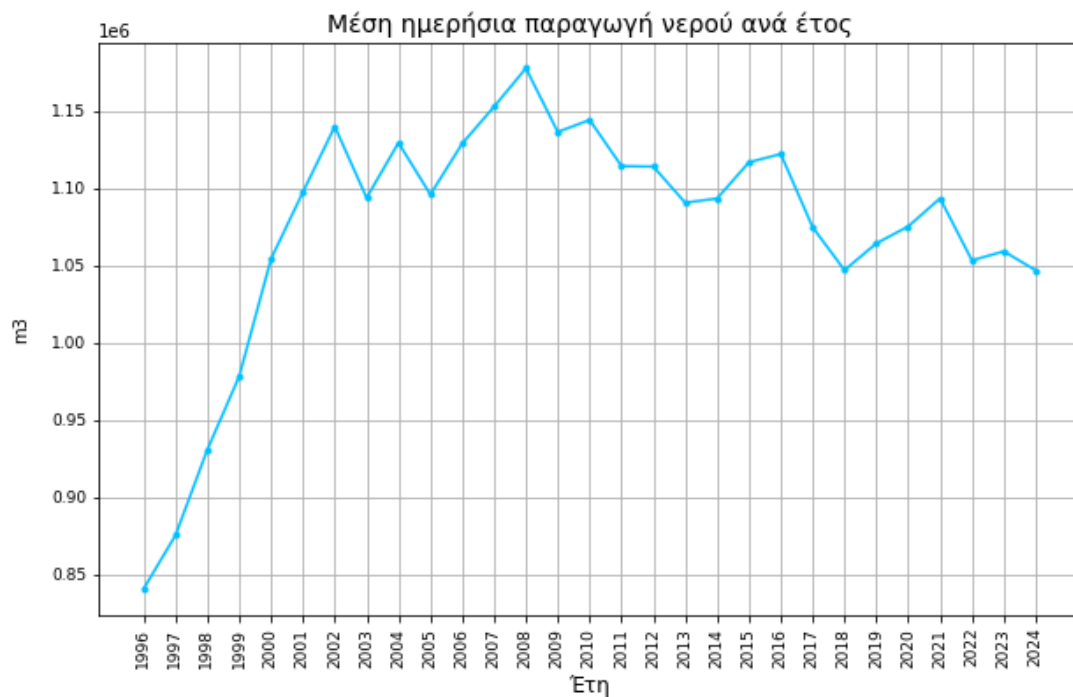
- Η μέση τιμή των ημερήσιων αποθεμάτων ανά έτος, μπορεί να εμφανίζει σημαντικές διακυμάνσεις ανάμεσα στα χρόνια (περίοδοι ξηρασίας).



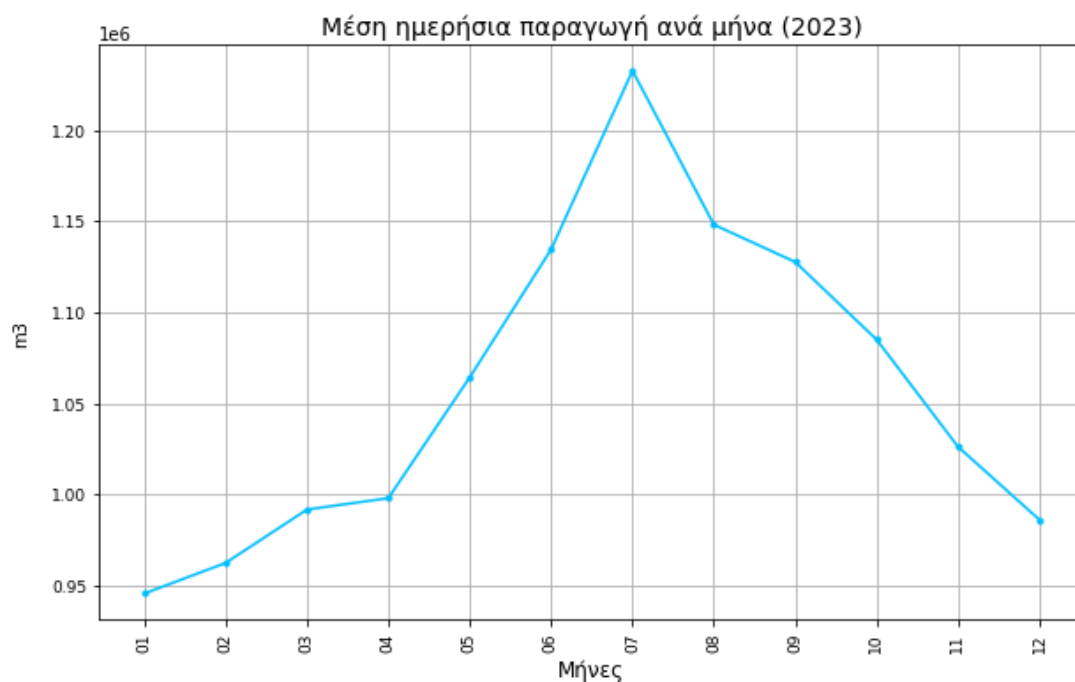
- Επίσης η μέση ημερήσια τιμή των αποθεμάτων ανά μήνα, ακολουθεί ένα κοινό μοτίβο με άνοδο της ποσότητας την χειμερινή περίοδο και μείωσή τους το καλοκαίρι.



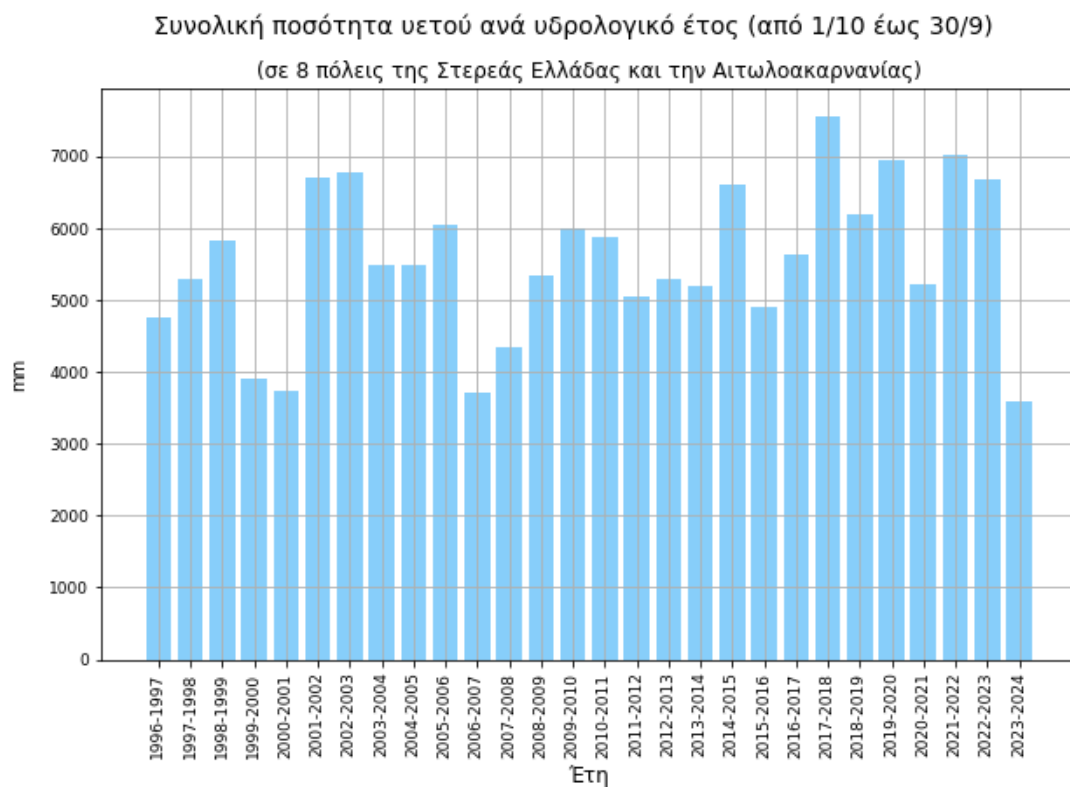
- Η μέση ημερήσια παραγωγή πόσιμου νερού (άρα και της κατανάλωσης) ανά έτος, παρουσιάζει κάποιες μακροπρόθεσμες μάλλον τάσεις, μια έντονα αυξητική τάση μέχρι το 2008 και έκτοτε μια μικρή πτωτική τάση (με ενδιάμεσες αυξομειώσεις).



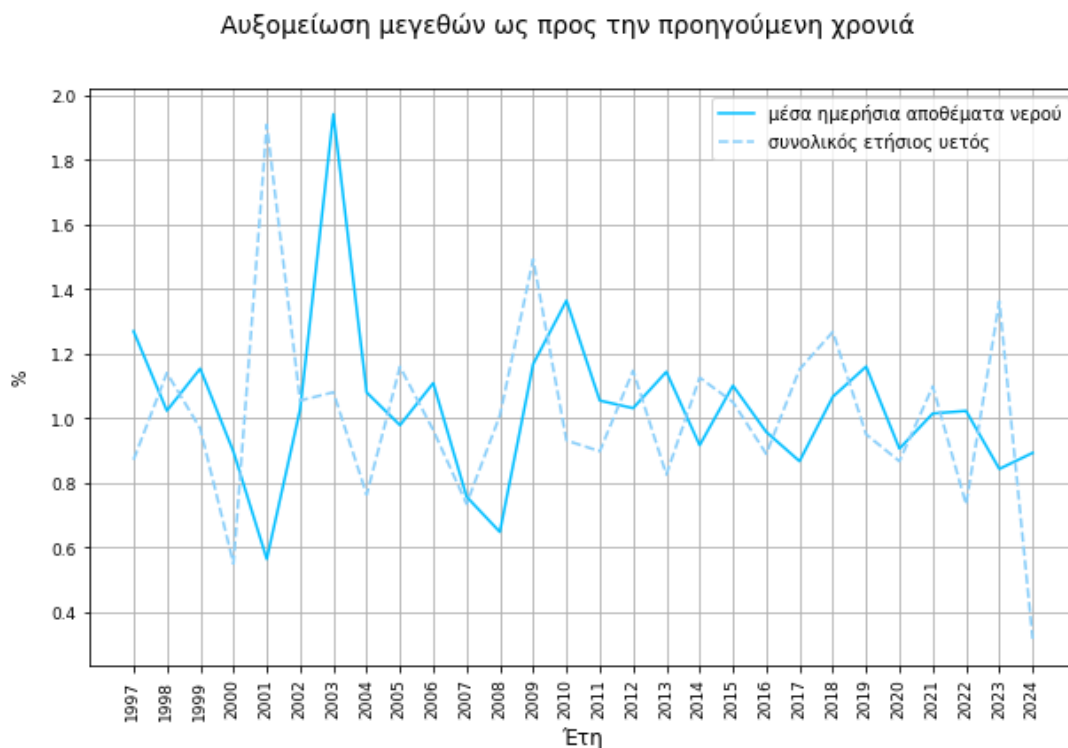
- Επίσης η μέση ημερήσια παραγωγή πόσιμου νερού ανά μήνα, ακολουθεί ένα κοινό μοτίβο με αυξημένες τιμές τους καλοκαιρινούς μήνες. Η μείωση της παραγωγής τον Αύγουστο ίσως οφείλεται στην απουσία σημαντικού μέρους του πληθυσμού λόγω των καλοκαιριών διακοπών.



- Το ετήσιο άθροισμα της ποσότητας υετού μπορεί να εμφανίσει σημαντικές διακυμάνσεις ανάμεσα στα χρόνια.



Όπως έχει ήδη αναφερθεί, η συγκέντρωση των ποσοστών υετού είναι ενδεικτική από οκτώ πόλεις της ευρύτερης γεωγραφικής περιοχής που βρίσκονται οι ταμιευτήρες. Έτσι δεν υπήρχε συγκεκριμένη γνώση για το αν και πως μπορούν να επιδρούν οι μετρήσεις αυτές με τα αποθέματα νερού. Από τις παρατηρήσεις όμως των δεδομένων σε γραφήματα, φαινόταν ίσως μια αναλογία στις μετρήσεις του υετού κάθε χρονιάς με τα αποθέματα της επόμενης χρονιάς.



Έπειτα από συζήτηση με τον διδάσκοντα του μαθήματος και σχετική υπόδειξή του, δοκίμασα να ελέγξω τα δεδομένα με τους δείκτες συσχέτισης Pearson και Spearman. Και οι δύο δείκτες είναι κατάλληλοι για συνεχείς τυχαίες μεταβλητές (εδώ οι ποσότητες των αποθεμάτων και του υετού). Ο Pearson έχει ως προϋπόθεση οι μεταβλητές να ακολουθούν την κανονική κατανομή, ενώ ο Spearman δεν έχει τέτοιο περιορισμό.

Και στις δύο μεθόδους, ο δείκτης συσχέτισης r μπορεί να πάρει τιμή από -1 έως 1. Ακολουθεί μια ενδεικτική ερμηνεία του $|r|$

| | |
|-------------|----------------------|
| $0.0 < 0.1$ | καμία συσχέτιση |
| $0.1 < 0.3$ | μικρή συσχέτιση |
| $0.3 < 0.5$ | μεσαία συσχέτιση |
| $0.5 < 0.7$ | υψηλή συσχέτιση |
| $0.7 < 1$ | πολύ υψηλή συσχέτιση |

Αν ο δείκτης είναι θετικός, δηλώνεται θετική συσχέτιση, δηλαδή η αύξηση της μιας μεταβλητής συσχετίζεται με την αύξηση της άλλης. Αν είναι αρνητικός, δηλώνεται αρνητική συσχέτιση, δηλαδή η αύξηση της μιας μεταβλητής συσχετίζεται με την μείωση της άλλης.

Ο υπολογισμός των δεικτών (με python) έγινε ανάμεσα στα μέσα ημερήσια αποθέματα νερού και την ετήσια συνολική ποσότητα υετού για κάθε χρόνο από το 1996 μέχρι σήμερα. Αρχικά τα δύο μεγέθη ελέγχθηκαν ως προς το ίδιο έτος και το αποτέλεσμα ήταν απογοητευτικό:

Pearson $r = 0.19889884657386991$, $pvalue = 0.300963829475726$

Spearman $r = 0.17980295566502458$, $pvalue = 0.3506577360844545$

Στη συνέχεια ελέγχθηκε η συνολική ποσότητα υετού κάθε χρονιάς με τα μέσα ημερήσια αποθέματα της επόμενης. Το αποτέλεσμα τώρα ήταν αρκετά ενθαρρυντικό:

Pearson $r = 0.6280443952269333$, $pvalue = 0.0003459287807183462$

Spearman $r = statistic=0.5342090859332238$, $pvalue = 0.003409173869857708$

Και με τους δύο δείκτες εμφανίζεται μια σχετικά υψηλή συσχέτιση, ενώ και οι πολύ χαμηλές τιμές των p -values καθιστούν ισχυρή την υπόθεση της συσχέτισης των δύο μεταβλητών.

5. Επιλογή δεικτών και κατηγοριοποίηση

Εφόσον είναι βάσιμες οι προηγούμενες παρατηρήσεις -η σημαντική διακύμανση των μεγεθών που καταγράφουμε (αποθέματα, παραγωγή νερού, υετός) μέσα στα χρόνια, καθώς και η υπόθεση της συσχέτισης του υετού με τα αποθέματα νερού- επέλεξα να χρησιμοποιήσω τους παρακάτω δείκτες για την αποτύπωση και αξιολόγηση της καθημερινής εικόνας των αποθεμάτων:

- Μέση τιμή των αποθεμάτων νερού (συνολικά, αλλά και για κάθε ταμιευτήρα χωριστά).
- Μέση τιμή της παραγωγής πόσιμου νερού.
- Λόγος της μέσης τιμής των αποθεμάτων νερού προς τη μέση τιμή της παραγωγής νερού (ώστε να αξιολογείται η διαθέσιμη ποσότητα και σε συνάρτηση με τις ανάγκες κατανάλωσης όπως αλλάζουν μέσα στα χρόνια).
- Άθροισμα ποσότητας υετού.

Τα μεγέθη συγκεντρώνονται σε ημερήσια βάση και η αρχική σκέψη ήταν να γίνεται σύγκριση με τις τιμές που είχαν την αντίστοιχη ημέρα τα προηγούμενα χρόνια. Αυτή όμως η αντιστοιχία φαινόταν προβληματική για την παραγόμενη ποσότητα νερού και πολύ περισσότερο για την ποσότητα υετού. Σε επίπεδο μιας μόνο ημέρας εύκολα μπορούν να εμφανίζονται συγκυριακές τιμές που να μην αντιστοιχούν στην πραγματική τάση.

Έτσι υιοθετήθηκε η σύγκριση διαστημάτων αρκετών ημερών. Κάθε μέρα που ανανεώνεται η τιμή ενός μεγέθους (αποθέματα, παραγωγή νερού ή υετός), συγκεντρώνονται τα στοιχεία των τελευταίων 30 ημερών για το μέγεθος αυτό και συγκρίνονται με εκείνα των αντίστοιχων διαστημάτων των προηγούμενων χρόνων. Για παράδειγμα, όταν καταγραφεί ένα μέγεθος στις 18/6 θα υπολογιστεί και θα συγκριθεί η τιμή για την περίοδο 18/5-18/6.

Σε όλες τις περιπτώσεις των συγκρίσεων που αναφέρθηκαν, τα στοιχεία ομαδοποιούνται με τον αλγόριθμο K-Means Clustering και κατατάσσονται σε 5 κατηγορίες: lower, low, mid, high, higher. Η επιλογή μιας κλίμακας 5 επιπέδων, έγινε διαισθητικά. Η εφαρμογή απευθύνεται στο ευρύ κοινό. Μια κλίμακα με περιορισμένα επίπεδα μπορεί να αποδώσει με απλό και κατανοητό τρόπο καταστάσεις αποθεμάτων, παραγωγής νερού ή ποσότητας υετού ως προς τα ιστορικά τους στοιχεία. Λιγότερα επίπεδα διακινδύνευαν την εξάλειψη των διαβαθμίσεων και την επικράτηση

ομοιομορφίας, ενώ περισσότερα ίσως καθιστούσαν την πληροφορία πιο εξειδικευμένη και δυσνόητη.

| | | |
|------------|--|--------|
| savings | : μέση τιμή των αποθεμάτων νερού (συνολικά) | higher |
| reservoirs | : μέση τιμή των αποθεμάτων νερού (για κάθε ταμιευτήρα χωριστά) | high |
| production | : μέση τιμή της παραγωγής πόσιμου νερού | mid |
| water | : λόγος αποθεμάτων προς παραγωγή νερού | low |
| weather | : άθροισμα ποσότητας υετού | lower |

Ωστόσο στη συνέχεια θέλησα να ελέγξω ποιο πλήθος ομάδων (πχ. από 2 έως 9) θα μπορούσε να είναι πιο κατάλληλο για τα συγκεκριμένα δεδομένα. Διαβάζοντας στο Διαδίκτυο βρήκα ότι υπάρχουν σχετικές μέθοδοι και επέλεξα να χρησιμοποιήσω (με python) την Silhouette analysis. Η μέθοδος αυτή υπολογίζει έναν συντελεστή που δηλώνει πόσο συνεκτικά είναι τα στοιχεία μέσα στις ομάδες ενός clustering και ταυτόχρονα πόσο σαφείς είναι οι αποστάσεις ανάμεσα στις ομάδες. Ο συντελεστής παίρνει τιμές από -1 έως 1, ένα clustering θεωρείται ισχυρό όταν η τιμή είναι πάνω από 0.7, λογικό όταν η τιμή είναι πάνω από 0.5 και αδύναμο όταν είναι απλώς πάνω από 0.25. Επαναλαμβάνοντας το K-Means Clustering σε ένα σύνολο δεδομένων με άλλο κάθε φορά πλήθος ομάδων, βρίσκουμε τον συντελεστή Silhouette για κάθε περίπτωση και ελέγχουμε για ποιο πλήθος ομάδων εμφανίζει υψηλότερες τιμές.

Για όλα τα σύνολα δεδομένων ο συντελεστής για 5 κατηγορίες ήταν πάνω από 0.5. Γενικότερα ήταν πολύ μικρές οι διαφορές στις τιμές του συντελεστή για όλα τα πλήθη ομάδων (ένα προβάδισμα έχουν μάλλον οι 2 ομάδες στα δεδομένα παραγωγής νερού). Ως συμπέρασμα θεώρησα ότι δικαιολογείται και μπορεί να διατηρηθεί η ομαδοποίηση σε 5 κατηγορίες.

| μέσα ημερήσια αποθέματα ανά έτος | | μέσης ημερήσια παραγωγή νερού ανά έτος | | συνολικής ποσότητας υετού ανά έτος | |
|-------------------------------------|--------------------|---|--------------------|---------------------------------------|---------------------|
| clusters | score | clusters | score | clusters | score |
| 2 | 0.608740519357838 | 2 | 0.7496309553309379 | 2 | 0.5244150314190505 |
| 3 | 0.6057366050220415 | 3 | 0.5635778070943428 | 3 | 0.46525582883369804 |
| 4 | 0.5655270880340955 | 4 | 0.595064046814738 | 4 | 0.6130748614065478 |
| 5 | 0.5938099737223771 | 5 | 0.5683630316657202 | 5 | 0.6526768167789606 |
| 6 | 0.5596965225415288 | 6 | 0.5740311176967804 | 6 | 0.6250568893811029 |
| 7 | 0.5617033573585946 | 7 | 0.5515363509309978 | 7 | 0.5779708203403137 |
| 8 | 0.5798210498804273 | 8 | 0.5733603709034976 | 8 | 0.5668217170740983 |
| 9 | 0.5885225158746209 | 9 | 0.5257484281950453 | 9 | 0.5248408299617268 |

6. Λειτουργία σε πραγματικό χρόνο και σύστημα ειδοποιήσεων

Τα δεδομένα στις πηγές ενημερώνονται σε ημερήσια βάση, αλλά αυτό δεν συμβαίνει πάντα με έναν ίδιο σταθερό τρόπο. Η ΕΥΔΑΠ φαίνεται πως ενημερώνει τα στοιχεία τις εργάσιμες ημέρες. Οι τιμές για τα ΣΚ και τις αργίες ενημερώνονται την επόμενη εργάσιμη. Επίσης οι ενημερώσεις στις τιμές των αποθεμάτων και της παραγωγής νερού δεν είναι ταυτόχρονες, μπορούν να διαφέρουν αρκετές ώρες μεταξύ τους. Στα δεδομένα του open-meteo υπάρχει επίσης κάποια καθυστέρηση. Κάποιοι μετεωρολογικοί δείκτες (ανάμεσά τους και ο συνολικός υετός) δεν εμφανίζονται ενημερωμένοι για την αμέσως προηγούμενη, καταλήγοντας η πιο πρόσφατη ενημέρωση να αφορά συνήθως την προχθεσινή μέρα.

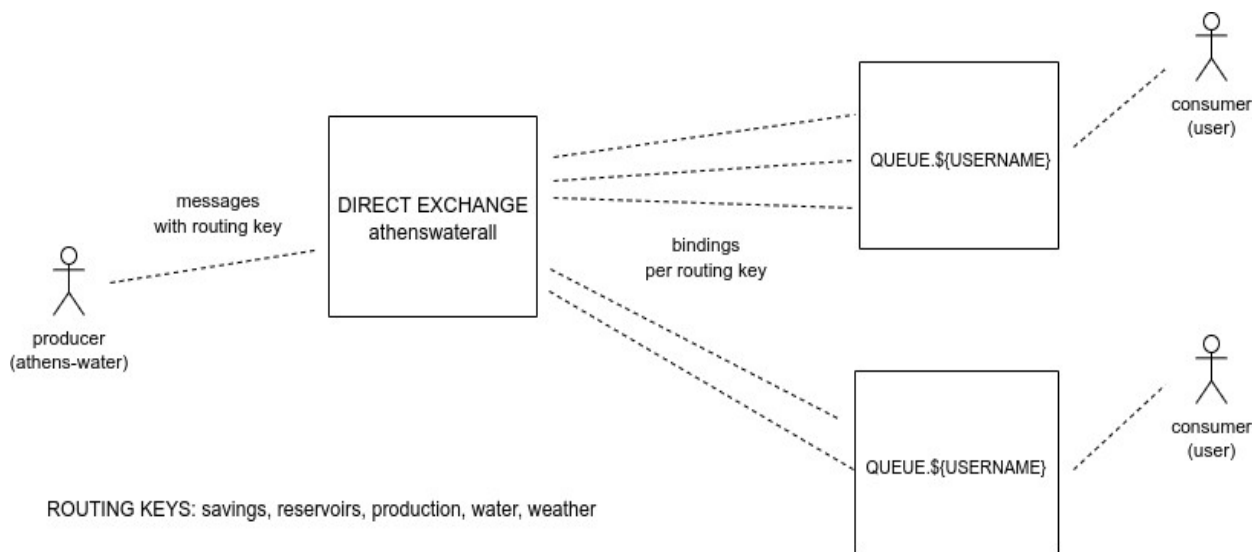
Το athenes-water θεωρείται πλήρως ενημερωμένο, όταν έχουν ενημερωθεί και τα τρία μεγέθη (αποθέματα, παραγωγή νερού και υετός) για τη χθεσινή μέρα. Όσο εκκρεμεί κάποιο ή κάποια από τα μεγέθη, η εφαρμογή επαναλαμβάνει όποια request απαιτούνται με ενδιάμεσες καθυστερήσεις δέκα λεπτών. Αν έχουν ολοκληρωθεί όλες οι ενημερώσεις για τη χθεσινή μέρα, τότε κάνει περιοδικούς ελέγχους ανά μία ώρα για αλλαγή της ημερομηνίας.

Κάθε φορά που ένα μέγεθος ενημερώνεται, γίνεται ο υπολογισμός του για την περίοδο των τελευταίων 30 ημερών και ξαναδημιουργούνται όποια clustering σχετίζονται. Με την ενημέρωση των αποθεμάτων, σχετίζονται τα clustering για το συνολικό απόθεμα καθώς και για κάθε ταμιευτήρα ξεχωριστά. Με την παραγωγή νερού, το clustering για την παραγωγή νερού, πιθανώς και το clustering του λόγου των αποθεμάτων προς την παραγωγή (εφόσον προηγουμένως έχουν ανανεωθεί και τα αποθέματα). Με την ποσότητα υετού, το clustering για την ποσότητα υετού.

Τα queries στη βάση δεδομένων για τους υπολογισμούς των τιμών σε διαστήματα 30 ημερών, καθώς και τα K-Means clusterings πραγματοποιούνται πολύ γρήγορα χωρίς καθυστερήσεις. Δεδομένου επίσης ότι οι ανανεώσεις των δεδομένων γίνονται με αργό ρυθμό (σε επίπεδο ημέρας), η άμεση ανανέωση του clustering ταυτόχρονα με την ανανέωση των μεγεθών διεκπεραιώνεται χωρίς ουσιαστικό κόστος και χωρίς την πολυπλοκότητα της διαχείρισης πιο σύνθετων περιοδικών διαδικασιών. Όπως για παράδειγμα προκαταβολικά clusterings για κάθε μέρα του μήνα ή κάθε μέρα του χρόνου, με τα διάφορα αποτελέσματά τους να διατηρούνται στη μνήμη ή στη βάση δεδομένων και να ανασύρονται για την κατηγοριοποίηση των νέων ενημερώσεων.

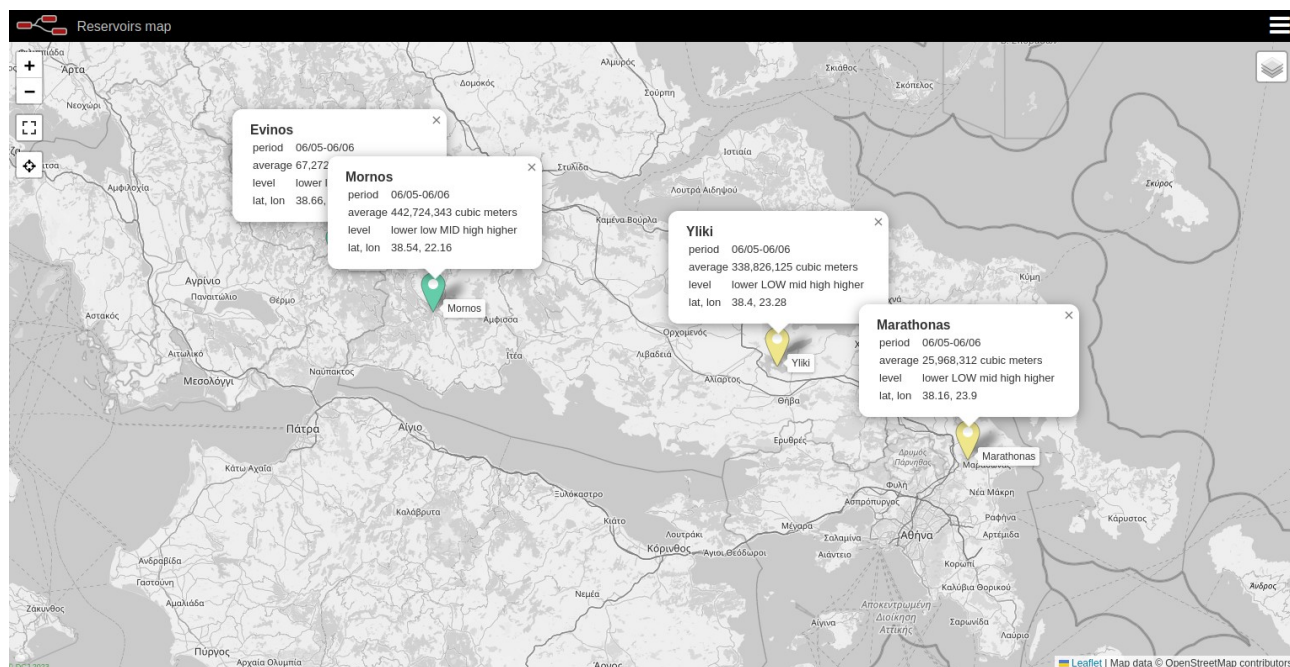
Όταν κάποιο μέγεθος και τα αντίστοιχα clusterings ανανεωθούν, στέλνονται και ειδοποιήσεις μέσω RabbitMQ με τους αντίστοιχους δείκτες και την κατηγοριοποίησή τους. Το athens-water στέλνει ως producer ειδοποιήσεις σε μία exchange τύπου direct με όνομα athenswaterall. Οι ειδοποιήσεις χαρακτηρίζονται με διαφορετικά routing keys αναλόγως ποιους δείκτες αφορούν:

- savings (για τα αποθέματα)
- reservoirs (για τα αποθέματα ανά ταμιευτήρα)
- production (για την παραγωγή νερού),
- water (για το λόγο των αποθεμάτων προς την παραγωγή)
- weather (για τις ποσότητες υετού)



Οποιοσδήποτε χρήστης εγγεγραμμένος στην exchange athenswaterall και για όποια routing keys ενδιαφέρεται, μπορεί να λαμβάνει αυτές τις ειδοποιήσεις και να τις αξιοποιεί με τον δικό του τρόπο.

Το athens-water χρησιμοποιεί και το ίδιο ως consumer αυτές τις ειδοποιήσεις για την παρουσίαση των αποτελεσμάτων του μέσω του dashboard και του worldmap που παρέχει το node-red. Αυτό δεν είναι αναγκαίο (πιθανώς να μην είναι και ενδεδειγμένο), οι ενημερώσεις μπορούν να γίνουν και απευθείας μέσω των ροών, αλλά το προτίμησα για λόγους παρουσίασης, για να φανεί η αξιοποίηση των ειδοποιήσεων με έναν ρεαλιστικό τρόπο.



Επιπλέον, μέσω του dashboard παρέχεται μια φόρμα εγγραφής χρήστη στο rabbitmq και στην exchange athenswaterall. Ο χρήστης δηλώνει username, password καθώς και για ποια routing keys ενδιαφέρεται να λαμβάνει ειδοποιήσεις. Το σύστημα μέσω του RabbitMQ api:

- δημιουργεί το νέο χρήστη
- δημιουργεί ουρά με την εξής μορφή ονόματος: `queue.${username}`,
- δημιουργεί bindings ανάμεσα στην ουρά και την exchange athenswaterall για κάθε ένα από τα routing keys που έχει δηλώσει ενδιαφέρον ο χρήστης
- δημιουργεί τέλος permissions για τον χρήστη με τη μορφή:
configure: `"^$ "`, write: `"^$ "`, read: ``queue.${username}``

The screenshot shows a web interface for registering a user in RabbitMQ. The page has a blue header with a hamburger menu icon and the text "Register". The main content area is light gray. In the center, there is a white box titled "Registration to RabbitMQ". Inside this box, there are two input fields: "Username *" and "Password *". Below these fields are five checkboxes, each with a label: "key - savings", "key - reservoirs", "key - production", "key - water", and "key - weather". At the bottom of the form are two buttons: "SUBMIT" and "CANCEL". Below the form, there is an error message in small text: "Error from Athens-water. Invalid registration. Username admin not available."

7. Εγκατάσταση του athens-water

- Χρησιμοποιούνται οι **βιβλιοθήκες** που μπορούν να εγκατασταθούν από το Manage Palette:

```
node-red-node-sqlite
@meowwolf/node-red-contrib-amqp
node-red-contrib-web-worldmap
node-red-dashboard
```

- Χρησιμοποιείται επίσης το **πακέτο clusters** που πρέπει να εγκατασταθεί με npm:

```
sudo npm install clusters --save
```

- Χρησιμοποιείται ένας **RabbitMQ server** και πρέπει να έχει δημιουργηθεί χρήστης με πλήρη δικαιώματα, που χρησιμοποιεί η εφαρμογή για πρόσβαση στο RabbitMQ api, με credentials:

```
username: admin, password: pass
```

- Στο **settings.js** χρειάζονται να οριστούν τα ακόλουθα:

```
...
/* Using environmental variables (system or user defined) ... */
process.env["NODE_TLS_REJECT_UNAUTHORIZED"] = 0;
process.env.AW_VHOST='athenswater';
process.env.AW_EXCHANGE='athenswaterall';
process.env.AW_ROUTING_KEYS='reservoirs,savings,production,water,weather';
...
module.exports = {
  MW_CONTRIB_AMQP_USERNAME: 'admin',
  MW_CONTRIB_AMQP_PASSWORD: 'pass',
  ...
  functionGlobalContext: {
    // os:require('os'),
    clusters:require( 'clusters' ),
    clusterHelpers:require( './projects/athens-water/helpers/cluster.js' ),
    dateHelpers:require( './projects/athens-water/helpers/date.js' ),
  },
  ...
}
```

- Τα **αρχεία της εφαρμογής** πρέπει να βρίσκονται μέσα στον κατάλογο:

`./node-red/projects/athens-water/`

`./node-red/projects/athens-water/flows.json`

`./node-red/projects/athens-water/flows-cred.json`

`./node-red/projects/athens-water/athens-water.sqlite` (βάση δεδομένων)

`./node-red/projects/athens-water/athens-water.log` (log αρχείο)

`./node-red/projects/athens-water/eydap.gr.crt` (TLS πιστοποιητικό ΕΥΔΑΠ)

`./node-red/projects/athens-water/helpers/date.js` (βοηθητικές συναρτήσεις)

`./node-red/projects/athens-water/helpers/cluster.js` (βοηθητικές συναρτήσεις)

- Έχει σημασία για τη σωστή λειτουργία των relative paths της βάσης δεδομένων και του log αρχείου, να **ξεκινάμε το ned-red** από το home directory (μέσα εκεί θεωρούμε ότι βρίσκεται και το subdirectory `.node-red`):

`~$ node-red`

- Για το **άνοιγμα των ροών** μέσα στο περιβάλλον του node-red, εφόσον έχουμε ενεργοποιημένη τη διαχείριση projects:

`projects > open project > athens-water`

- **Εναλλακτικά**, μπορούμε να κάνουμε απλώς import τις ροές:

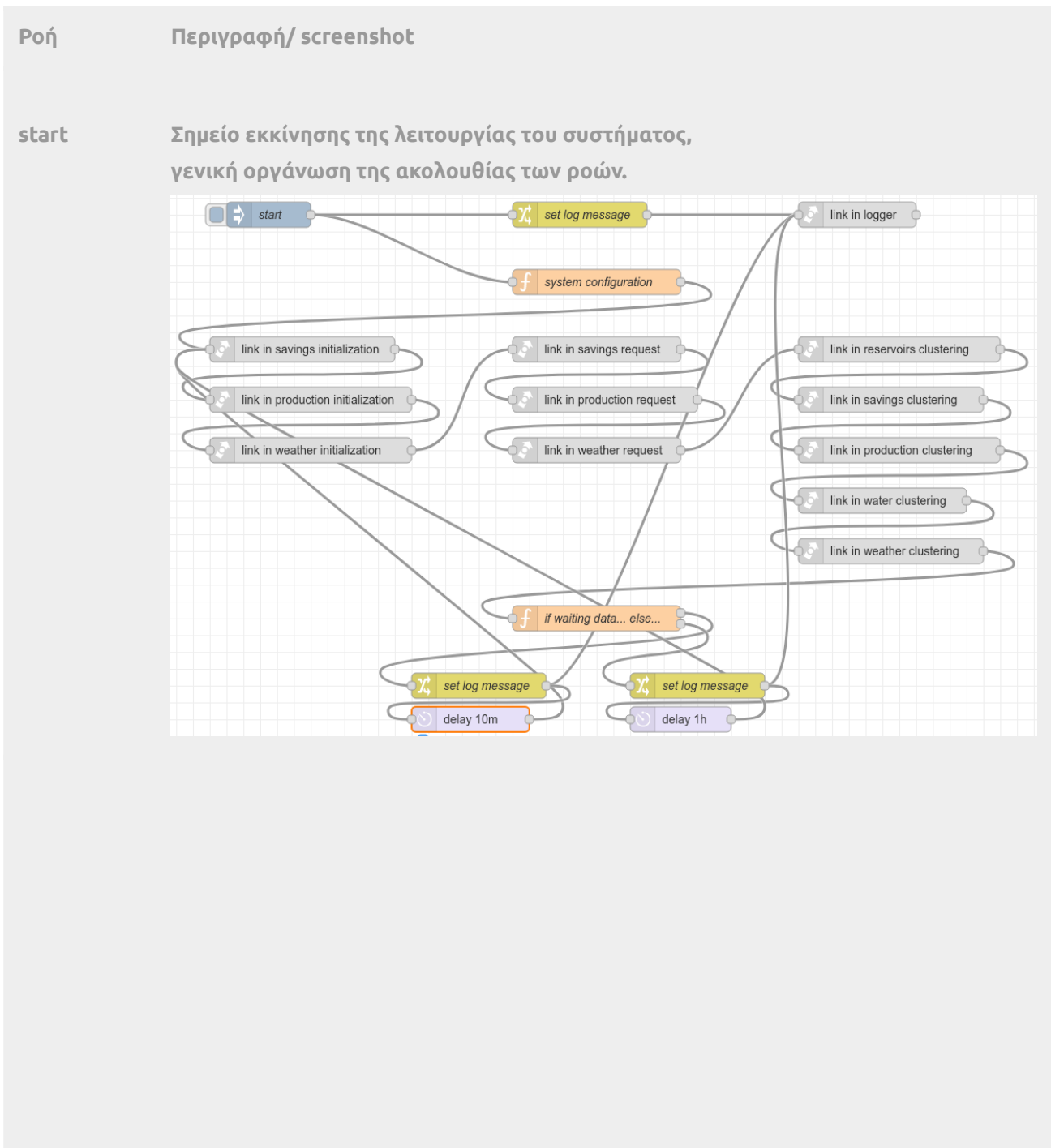
`./node-red/projects/athens-water/flows.json`

- Την πρώτη φορά πριν την εκτέλεση της εφαρμογής, δημιουργούμε **το vhost και το exchange** για το RabbitMQ μέσα από τη ροή 'rabbitmq admin' (inject node 'go create vhost').

- **Η εκκίνηση του athens-water** γίνεται μέσα από τη ροή 'start' (inject node 'start').

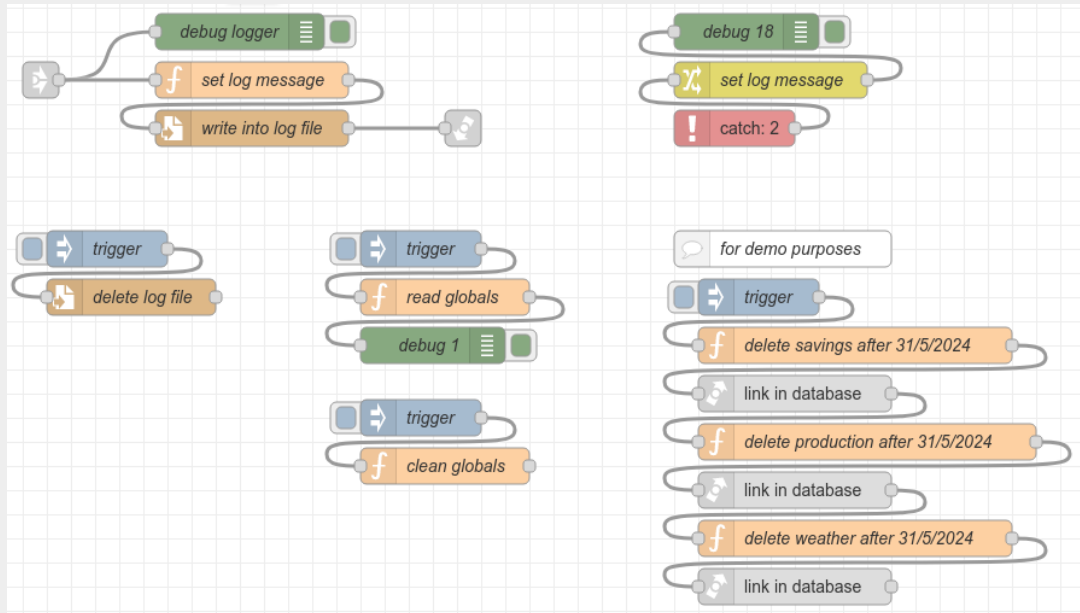
8. Συνοπτική παρουσίαση των ροών

Για την λειτουργία του athens-water χρησιμοποιείται ένας συνδυασμός αρκετών διαφορετικών ροών. Ο επιμερισμός σε διαφορετικές ροές, διευκολύνει την επαναχρησιμοποίηση τμημάτων κώδικα, τη λογική οργάνωσή του και το debugging. Ως τρόπος επικοινωνίας και περάσματος από τη μία ροή στην άλλη, έχουν χρησιμοποιηθεί εκτενώς οι κόμβοι link in, link out, link call. Ακολουθεί μια συνοπτική περιγραφή και screenshots των ροών.



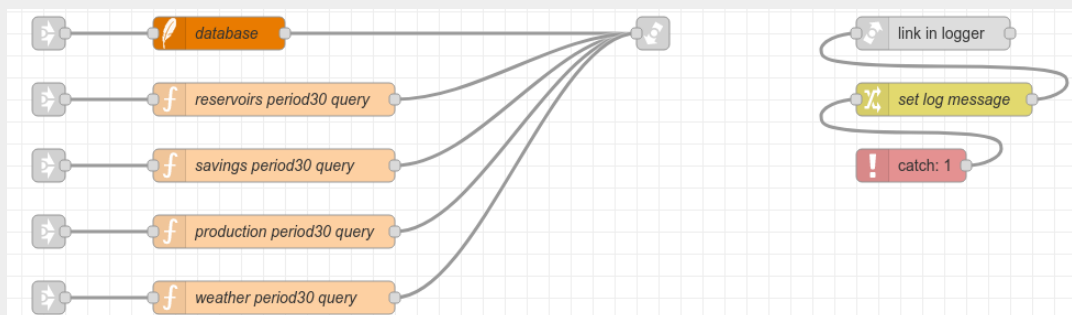
admin

Βοηθητικές διαχειριστικές λειτουργίες, διατήρηση ενός logger για εγγραφή μηνυμάτων σε αρχείο, παρακολούθηση/ καθαρισμός global μεταβλητών.



database

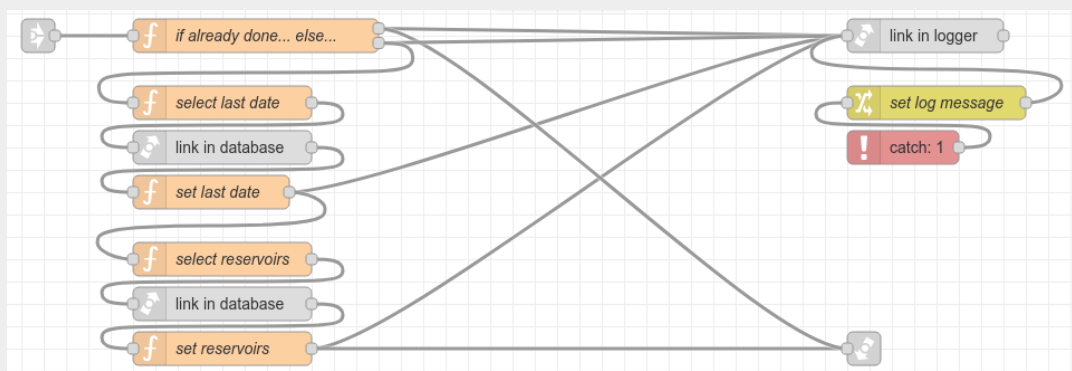
Πρόσβαση στη βάση δεδομένων και κάποια σύνθετα επαναχρησιμοποιούμενα queries.



savings

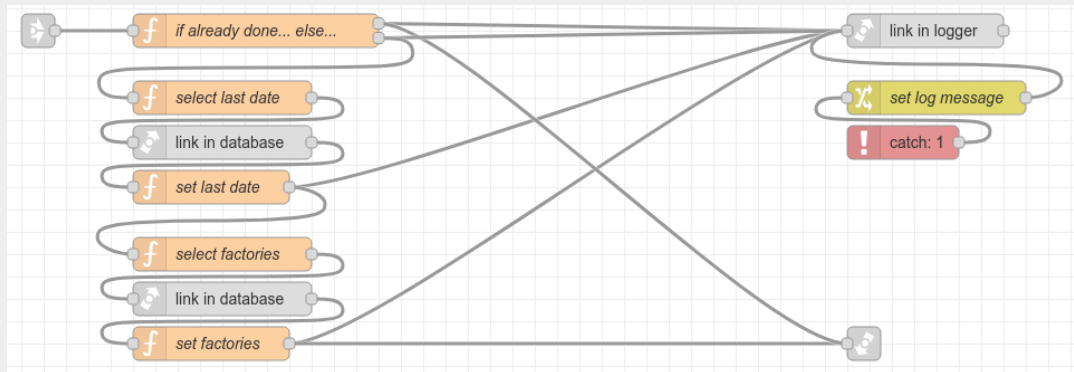
initialization

Αρχικοποίηση global μεταβλητών που αφορούν στα δεδομένα αποθεμάτων.



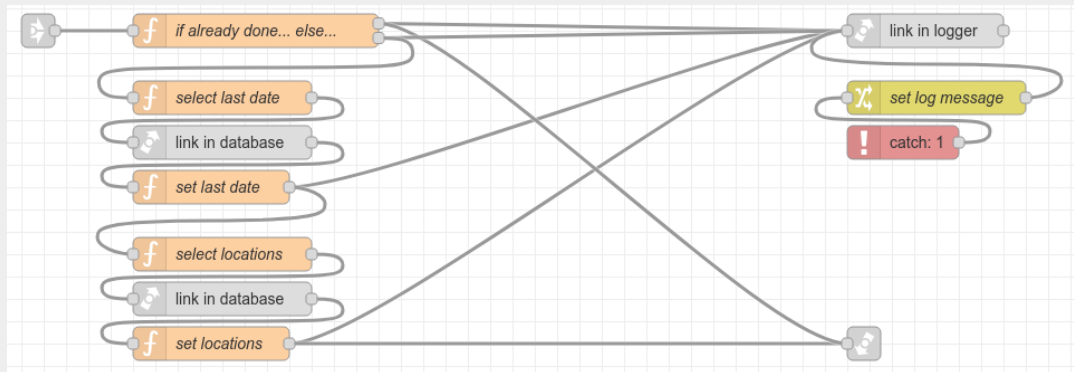
production
initialization

Αρχικοποίηση global μεταβλητών που αφορούν στα δεδομένα παραγωγής νερού.



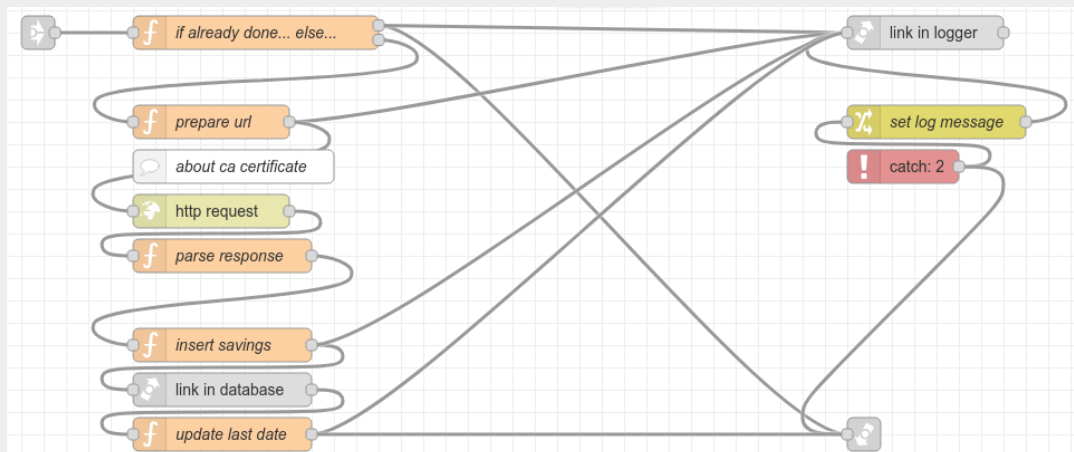
weather
initialization

Αρχικοποίηση global μεταβλητών που αφορούν στα μετεωρολογικά δεδομένα.



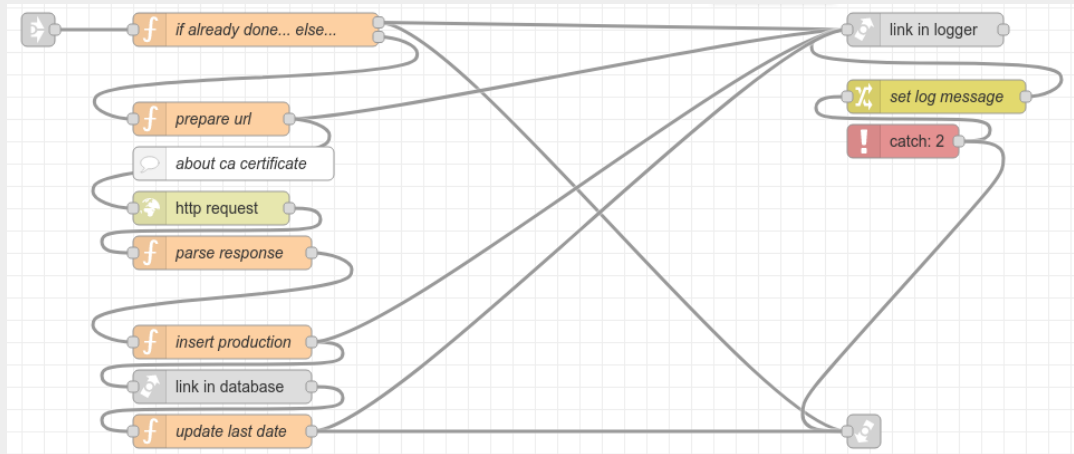
savings
request

Request για δεδομένα αποθεμάτων, φιλτράρισμα
και αποθήκευση στη βάση δεδομένων.



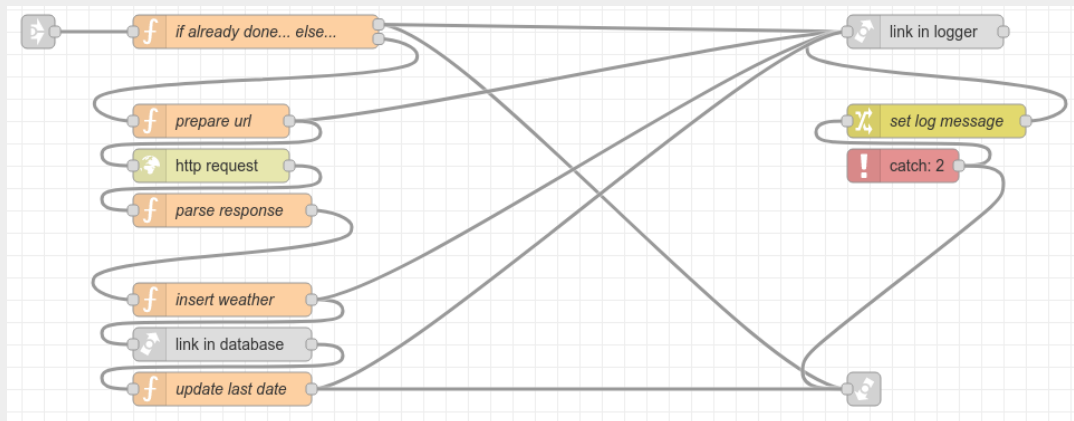
production
request

Request για δεδομένα παραγωγής νερού, φιλτράρισμα
και αποθήκευση στη βάση δεδομένων.



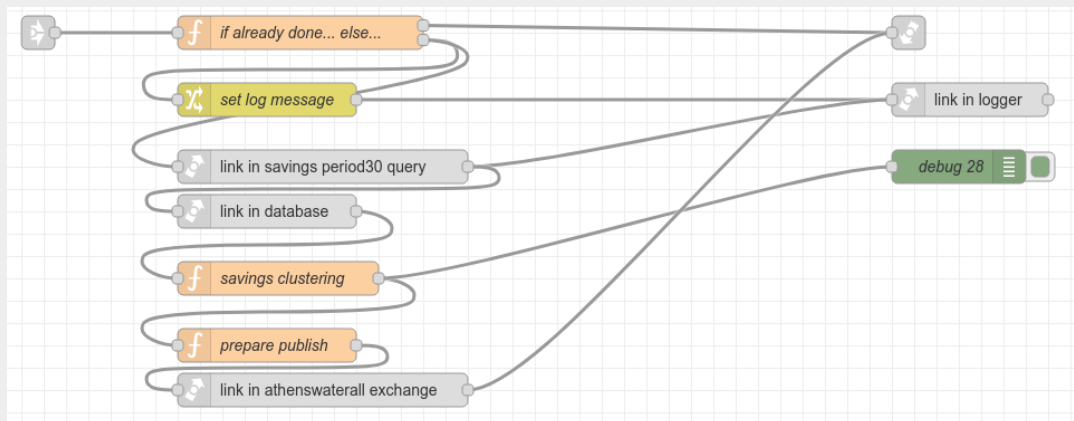
weather
request

Request για μετεωρολογικά δεδομένα, φιλτράρισμα
και αποθήκευση στη βάση δεδομένων.



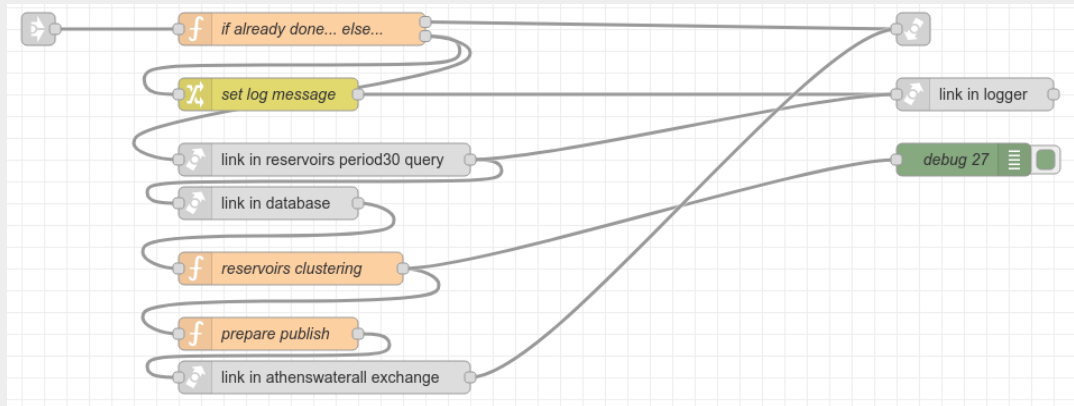
savings
clustering

Ανάκτηση δεδομένων αποθεμάτων (συνολικά) όλων των χρόνων
για ίδιο διάστημα 30 ημερών και κατηγοριοποίηση με K-Means clustering.



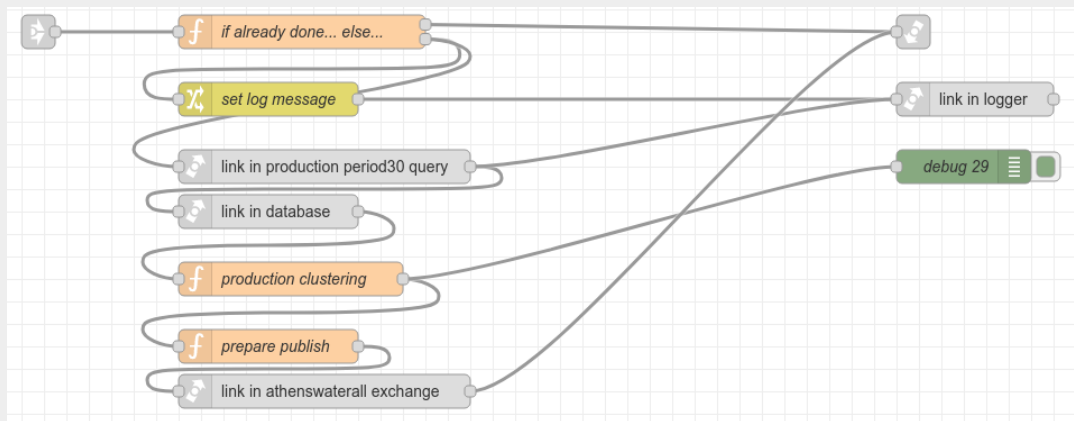
reservoirs clustering

Ανάκτηση δεδομένων αποθεμάτων (ανά ταμιευτήρα) όλων των χρόνων για ίδιο διάστημα 30 ημερών και κατηγοριοποίηση με K-Means clustering.



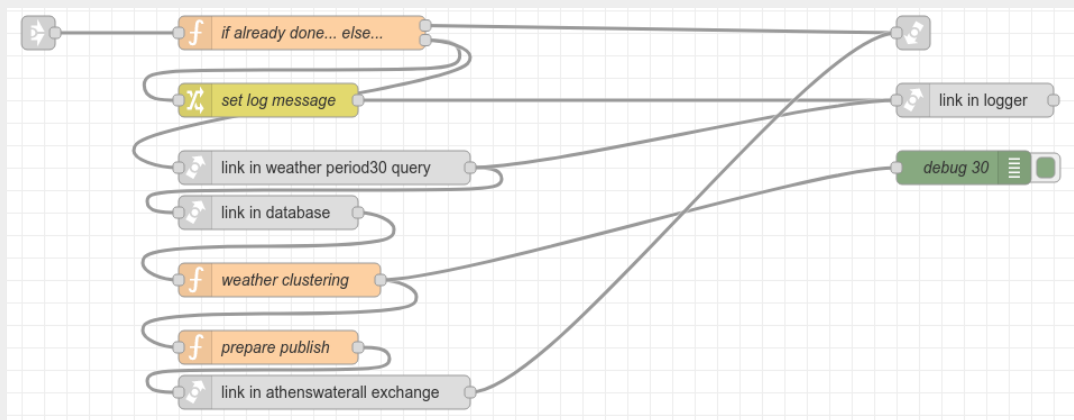
production clustering

Ανάκτηση δεδομένων παραγωγής νερού όλων των χρόνων για ίδιο διάστημα 30 ημερών και κατηγοριοποίηση με K-Means clustering.



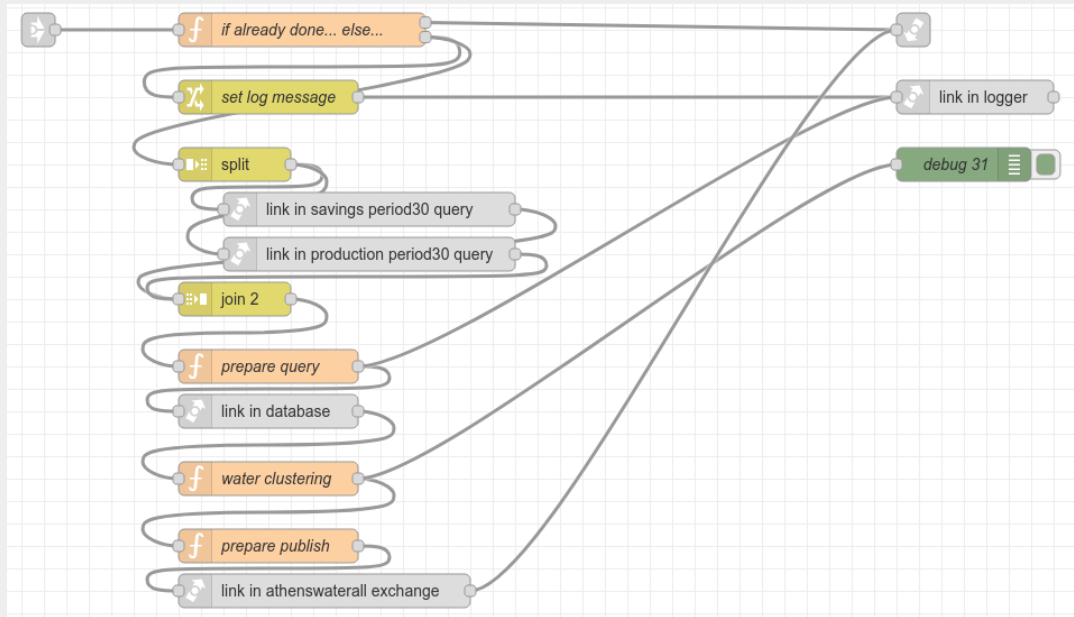
Weather clustering

Ανάκτηση δεδομένων υετού όλων των χρόνων για ίδιο διάστημα 30 ημερών και κατηγοριοποίηση με K-Means clustering.

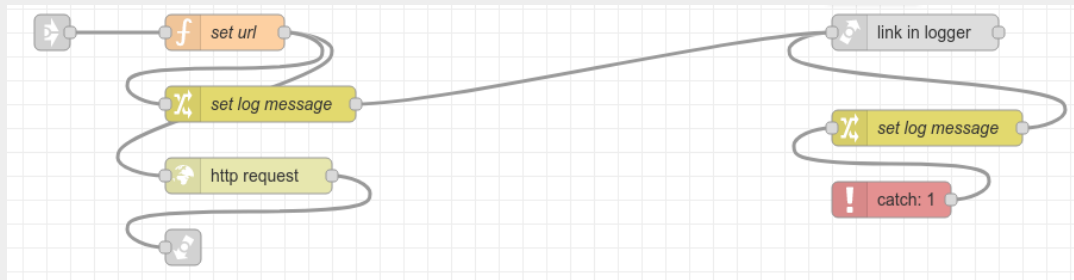


water
clustering

Ανάκτηση δεδομένων αποθεμάτων και παραγωγής νερού όλων των χρόνων
για ίδιο διάστημα 30 ημερών και κατηγοριοποίηση με K-Means clustering.

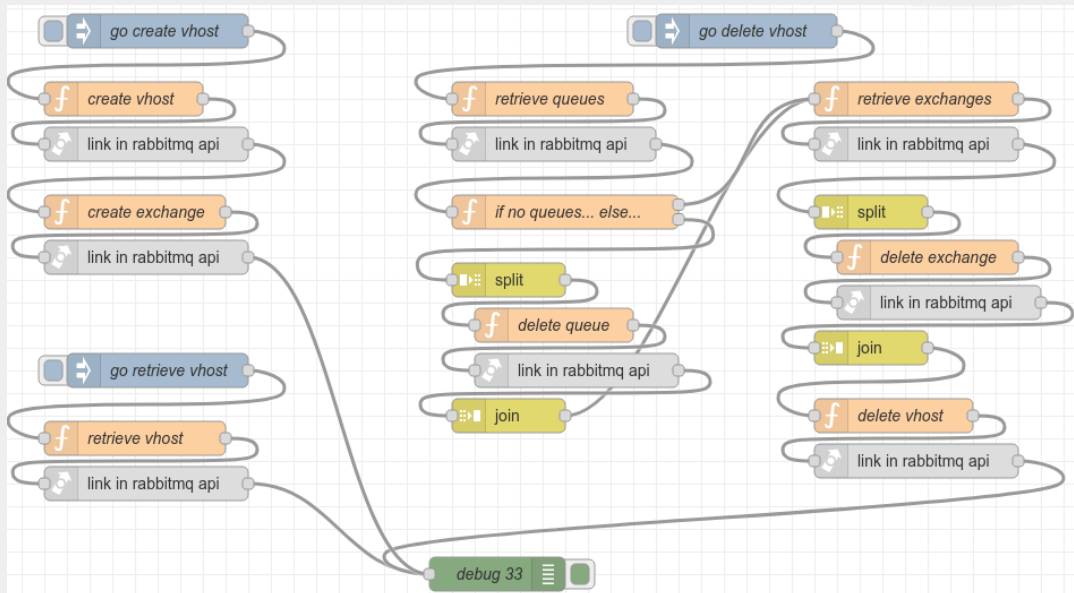


rabbitmq api Πρόσβαση στο api του RabbitMQ.



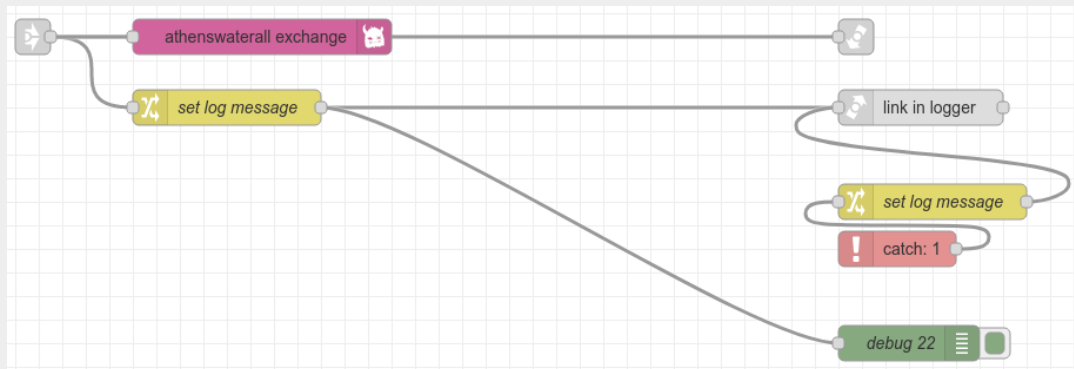
rabbitmq admin

Βοηθητικές διαχειριστικές λειτουργίες, για τη δημιουργία ή τη διαγραφή του vhost και του exchange που χρησιμοποιούνται στο RabbitMQ για τις ανάγκες της εφαρμογής.



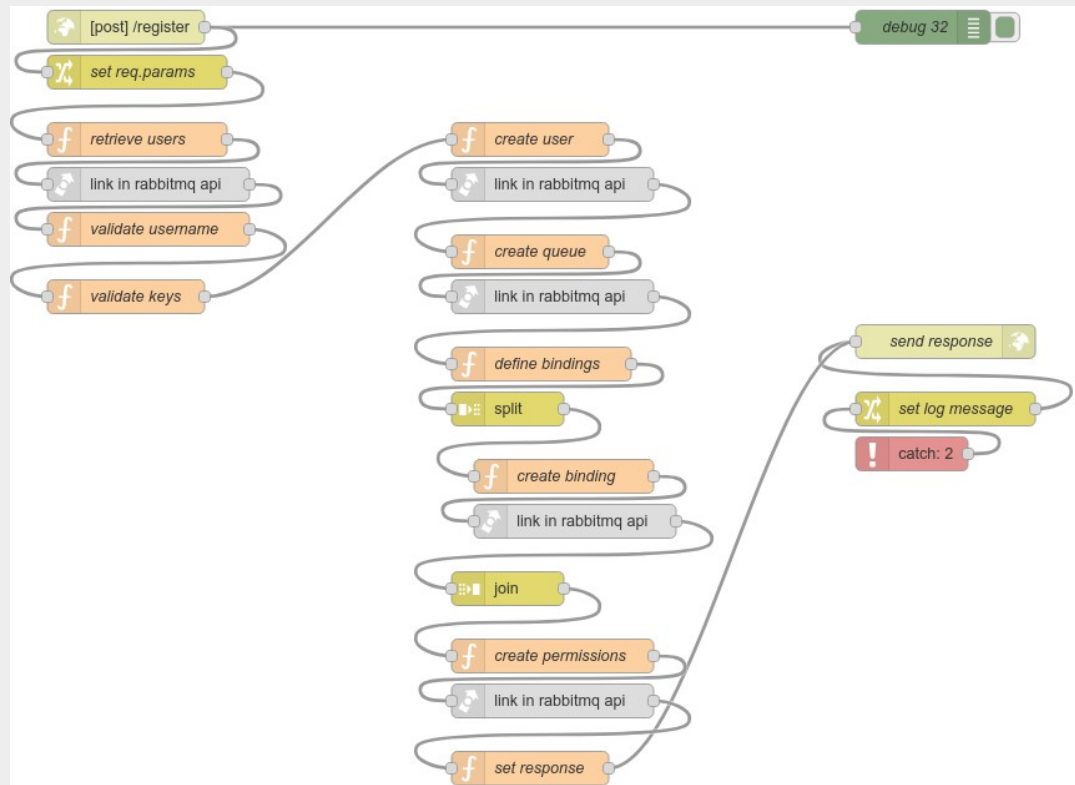
rabbitmq publish

Λειτουργία της εφαρμογής ως producer στο RabbitMQ με αποστολή ειδοποιήσεων.



rabbitmq
register

Http service που διεκπεραιώνει την εγγραφή νέου χρήστη στο RabbitMQ
(συμπεριλαμβάνεται δημιουργία queue, bindings, permissions).



dashboard

Λειτουργία της εφαρμογής ως consumer στο RabbitMQ και παρουσίαση των ειδοποιήσεων μέσα από το dashboard και το worldmap.

Επίσης φόρμα στο dashboard για την εγγραφή χρήστη στο RabbitMQ.

