**Εργασία στο μάθημα “ΥΠΗΡΕΣΙΕΣ ΚΑΙ ΣΥΣΤΗΜΑΤΑ ΔΙΑΔΙΚΤΥΟΥ”**

ΤΣΙΓΚΡΗΛΑΣ ΧΑΡΑΛΑΜΠΟΣ 2021101

ΣΤΡΑΤΗΣ ΣΠΥΡΙΔΩΝ 2021137

*ΣΥΣΤΗΜΑ ΕΝΗΜΕΡΩΣΗΣ ΛΙΜΑΝΙΩΝ ΓΙΑ ΤΟΝ ΚΑΙΡΟ ΚΑΙ ΤΗΝ ΚΑΤΑΣΤΑΣΗ ΤΗΣ ΘΑΛΑΣΣΑΣ*

Επιλέξαμε την δημιουργία του συστήματός μας για την έγκαιρη ενημέρωση των πλοίων και των επιβατών στην Ελλάδα, προκειμένου να αποφευχθούν ατυχήματα και να πραγματοποιούνται τα ταξίδια με μεγαλύτερη ασφάλεια. Για την επίτευξη του στόχου μας επιλέξαμε να χρησιμοποιήσουμε 2 API από τα οποία λαμβάνουμε τα δεδομένα μας. Το πρώτο μας παρέχει πληροφορίες για την κατασταση του καιρού (όπως θερμοκρασία, βροχές, ταχύτητα του ανέμου κ.α.), ενώ το δεύτερο για την κατάσταση της θάλασσας (όπως το ύψος των κυμάτων, την κατεύθυνση τους και τη διάρκεια τους).

Δεχόμαστε τρέχοντα, ωραία και ιστορικά δεδομένα για 5 λιμάνια επάνω σε ολόκληρη την Ελλάδα (Πειραιάς, Θεσσαλονίκη, Κρήτη, Κέρκυρα και Λέσβος).

**Πηγές δεδομένων (API) και requests**

Τα 2 API μας από όπου παίρνουμε τα δεδομένα μας με http requests είναι τα εξείς:

για τα **current** και **hourly** δεδομένα

* **Open-Meteo Weather Forecast** για το καιρό

<https://open-meteo.com/en/docs>

* και **Open-Meteo Marine Forecast** για τα κύματα της θάλασσας

<https://open-meteo.com/en/docs/marine-weather-api>

για τα **historical** δεδομένα

* **Open-Meteo Historical Weather** για το καιρό

<https://open-meteo.com/en/docs/historical-weather-api>

* και **Open-Meteo Marine Forecast** για τα κύματα της θάλασσας

<https://open-meteo.com/en/docs/marine-weather-api>

Αναλυτικότερα τα http requests που κάνουμε (με GET request):

για τα **current** και **hourly** δεδομένα

* **Open-Meteo Weather Forecast** για το καιρό

για τον Πειραιά

<https://api.open-meteo.com/v1/forecast?latitude=37.944709&longitude=23.638304&current=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&timezone=auto>

<https://api.open-meteo.com/v1/forecast?latitude=37.944709&longitude=23.638304&hourly=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&forecast_days=2&timezone=auto>

για την Κρήτη

<https://api.open-meteo.com/v1/forecast?latitude=35.489126&longitude=24.0755468&current=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&timezone=auto>

<https://api.open-meteo.com/v1/forecast?latitude=35.489126&longitude=24.075546&hourly=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&forecast_days=2&timezone=auto>

για την Θεσσαλονίκη

<https://api.open-meteo.com/v1/forecast?latitude=40.635485&longitude=22.932838&current=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&timezone=auto>

<https://api.open-meteo.com/v1/forecast?latitude=40.635485&longitude=22.932838&hourly=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&forecast_days=2&timezone=auto>

για την Κέρκυρα

<https://api.open-meteo.com/v1/forecast?latitude=39.63028&longitude=19.90177&current=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&timezone=auto>

<https://api.open-meteo.com/v1/forecast?latitude=39.63028&longitude=19.90177&hourly=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&forecast_days=2&timezone=auto>

για την Λέσβο

<https://api.open-meteo.com/v1/forecast?latitude=39.10412&longitude=26.56064&current=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&timezone=auto>

<https://api.open-meteo.com/v1/forecast?latitude=39.10412&longitude=26.56064&hourly=temperature_2m,relative_humidity_2m,rain,showers,snowfall,cloud_cover,wind_speed_10m&forecast_days=2&timezone=auto>

* και **Open-Meteo Marine Forecast** για τα κύματα της θάλασσας

για τον Πειραιά

<https://marine-api.open-meteo.com/v1/marine?latitude=37.944709&longitude=23.638304&current=wave_height,wave_direction,wave_period&timezone=auto>

<https://marine-api.open-meteo.com/v1/marine?latitude=37.944709&longitude=23.638304&hourly=wave_height,wave_direction,wave_period&timezone=auto&forecast_days=2>

για την Κρήτη

<https://marine-api.open-meteo.com/v1/marine?latitude=35.489126&longitude=24.075546&current=wave_height,wave_direction,wave_period&timezone=auto>

<https://marine-api.open-meteo.com/v1/marine?latitude=35.489126&longitude=24.075546&hourly=wave_height,wave_direction,wave_period&timezone=auto&forecast_days=2>

για την Θεσσαλονίκη

<https://marine-api.open-meteo.com/v1/marine?latitude=40.635485&longitude=22.932838&current=wave_height,wave_direction,wave_period&timezone=auto>

<https://marine-api.open-meteo.com/v1/marine?latitude=40.635485&longitude=22.932838&hourly=wave_height,wave_direction,wave_period&timezone=auto&forecast_days=2>

για την Κέρκυρα

<https://marine-api.open-meteo.com/v1/marine?latitude=39.63028&longitude=19.90177&current=wave_height,wave_direction,wave_period&timezone=auto>

<https://marine-api.open-meteo.com/v1/marine?latitude=39.63028&longitude=19.90177&hourly=wave_height,wave_direction,wave_period&timezone=auto&forecast_days=2>

για την Λέσβο

<https://marine-api.open-meteo.com/v1/marine?latitude=39.10412&longitude=26.56064&current=wave_height,wave_direction,wave_period&timezone=auto>

<https://marine-api.open-meteo.com/v1/marine?latitude=39.10412&longitude=26.56064&hourly=wave_height,wave_direction,wave_period&timezone=auto&forecast_days=2>

για τα **historical** δεδομένα

* **Open-Meteo Historical Weather** για το καιρό

για τον Πειραιά <https://archive-api.open-meteo.com/v1/archive?latitude=37.9838&longitude=23.7278&start_date=2023-01-01&end_date=2023-12-31&hourly=temperature_2m,relative_humidity_2m,rain,snowfall,cloud_cover,wind_speed_10m>

για την Κρήτη

<https://archive-api.open-meteo.com/v1/archive?latitude=35.489126&longitude=24.0755468&start_date=2023-01-01&end_date=2023-12-31&hourly=temperature_2m,relative_humidity_2m,rain,snowfall,cloud_cover,wind_speed_10m>

για την Θεσσαλονίκη

<https://archive-api.open-meteo.com/v1/archive?latitude=40.635485&longitude=22.932838&start_date=2023-01-01&end_date=2023-12-31&hourly=temperature_2m,relative_humidity_2m,rain,snowfall,cloud_cover,wind_speed_10m>

για την Κέρκυρα

<https://archive-api.open-meteo.com/v1/archive?latitude=39.63028&longitude=19.90177&start_date=2023-01-01&end_date=2023-12-31&hourly=temperature_2m,relative_humidity_2m,rain,snowfall,cloud_cover,wind_speed_10m>

για την Λέσβο

<https://archive-api.open-meteo.com/v1/archive?latitude=39.10412&longitude=26.56064&start_date=2023-01-01&end_date=2023-12-31&hourly=temperature_2m,relative_humidity_2m,rain,snowfall,cloud_cover,wind_speed_10m>

* και **Open-Meteo Marine Forecast** για τα κύματα της θάλασσας

για τον Πειραιά

<https://marine-api.open-meteo.com/v1/marine?latitude=37.9838&longitude=23.7278&hourly=wave_height,wave_direction,wave_period&start_date=2023-01-01&end_date=2023-12-31>

για την Κρήτη

<https://marine-api.open-meteo.com/v1/marine?latitude=35.489126&longitude=24.0755468&hourly=wave_height,wave_direction,wave_period&start_date=2023-01-01&end_date=2023-12-31>

για την Θεσσαλονίκη

<https://marine-api.open-meteo.com/v1/marine?latitude=40.635485&longitude=22.932838&hourly=wave_height,wave_direction,wave_period&start_date=2023-01-01&end_date=2023-12-31>

για την Κέρκυρα

<https://marine-api.open-meteo.com/v1/marine?latitude=39.63028&longitude=19.90177&hourly=wave_height,wave_direction,wave_period&start_date=2023-01-01&end_date=2023-12-31>

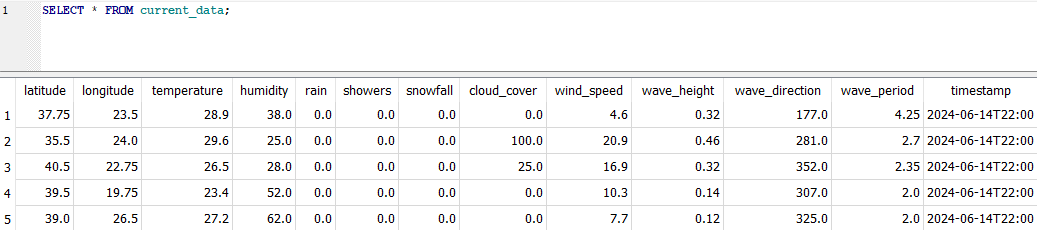
για την Λέσβο

<https://marine-api.open-meteo.com/v1/marine?latitude=39.10412&longitude=26.56064&hourly=wave_height,wave_direction,wave_period&start_date=2023-01-01&end_date=2023-12-31>

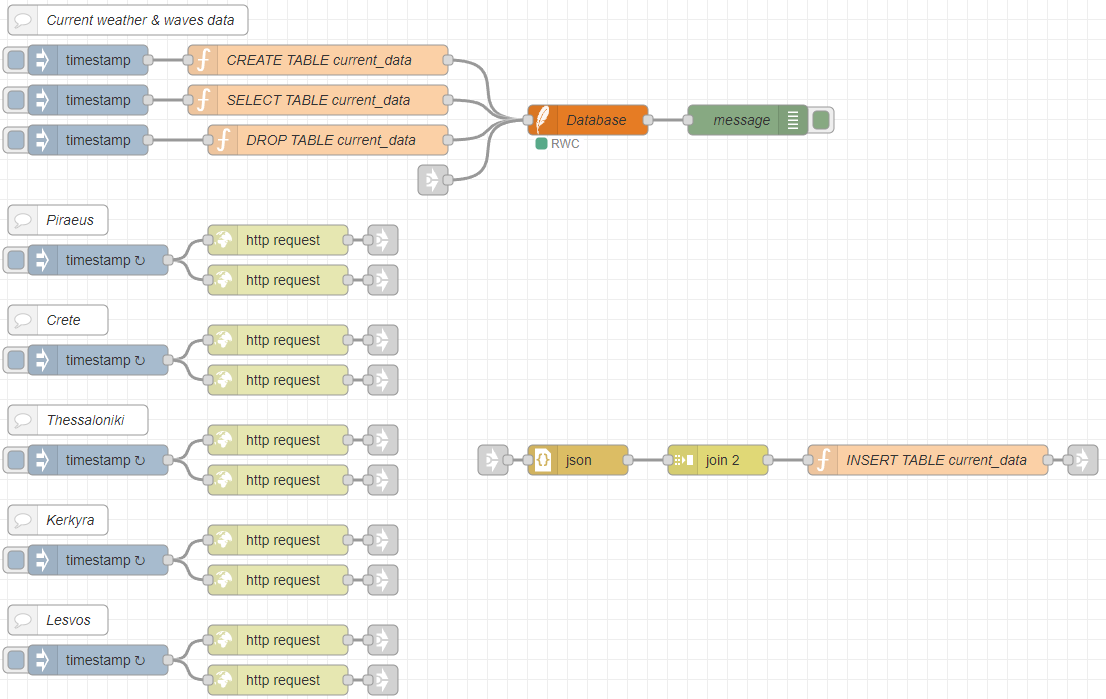
**Αποθήκευση και επεξεργασία δεδομένων**

Η αποθήκευση και ανάκτηση των δεδομένων γίνεται μέσω queries σε μια sqlite database (weather.db) με κάποιους πίνακες. Οι πίνακες που χρησιμοποιούμε είναι οι εξείς:

* **current\_data** για ζωντανά δεδομένα καιρού και κυμάτων της τωρινής ώρας. Ο πίνακας **current\_data** είναι κοινός και για τα 5 λιμάνια και έχει τα εξείς γνωρίσματα:

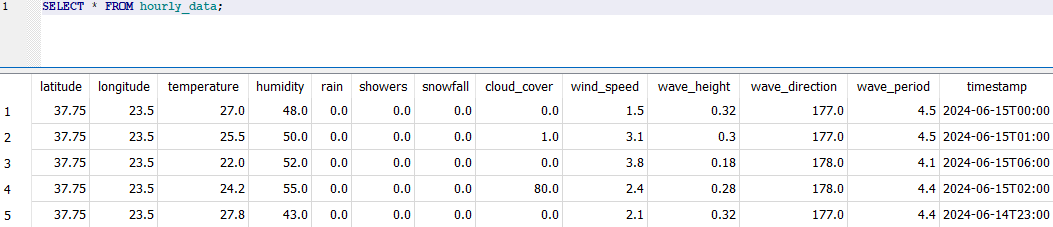


Η δομή στο node red είναι η ακόλουθη:

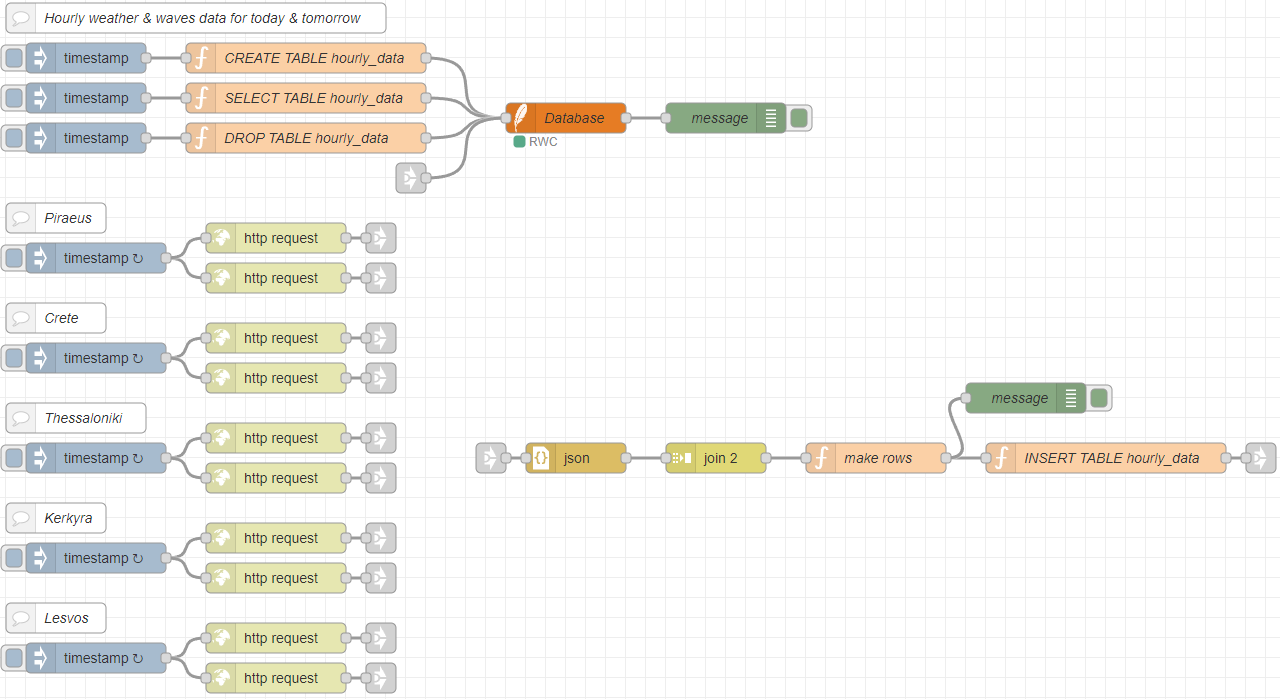


όπου το κάθε function εκτελεί το αντίστοιχο query μέσω μηνυμάτων msg.topic. Τα requests, φέρνουν 1 json το καθένα και μέσω του join τα περνάμε στο ίδιο μήνυμα και ύστερα κάνουμε τα στοιχεία που μας αφορούν insert στο table της βάσης. Τα requests καλούνται αυτόματα κάθε 1 ώρα.

* **hourly\_data** για ωριαία δεδομένα πρόγνωσης του καιρού και των κυμάτων για τη σημερινή και την αυριανή μέρα. Ο πίνακας **hourly\_data** είναι κοινός και για τα 5 λιμάνια και έχει τα εξείς γνωρίσματα:

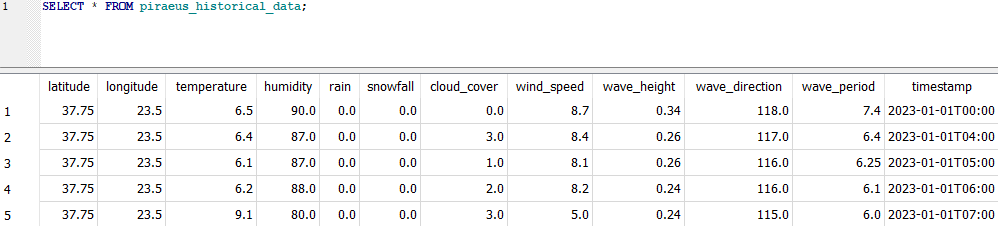


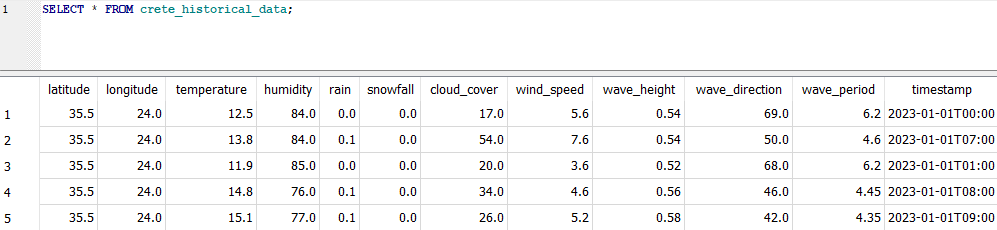
Η δομή στο node red είναι η ακόλουθη:

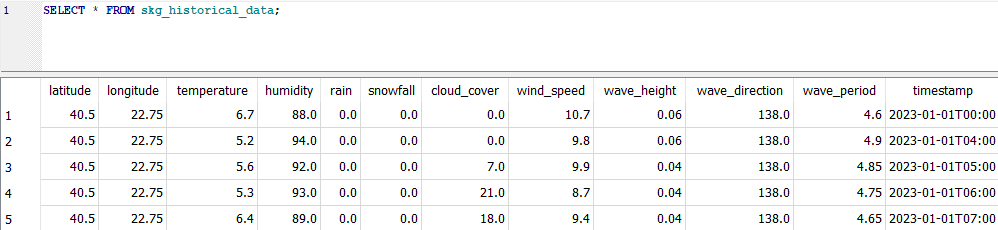


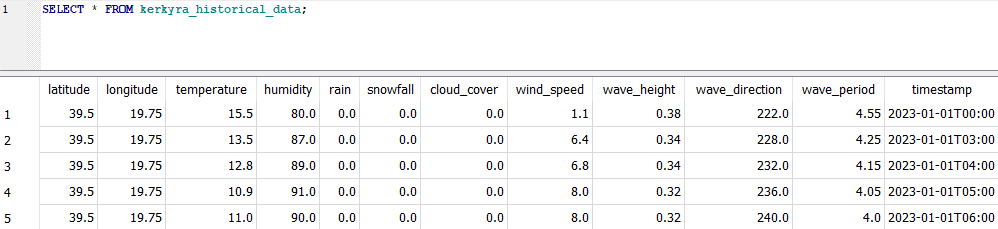
όπου το κάθε function εκτελεί το αντίστοιχο query μέσω μηνυμάτων msg.topic. Τα requests, φέρνουν 1 json το καθένα και μέσω του join τα περνάμε στο ίδιο μήνυμα. Για να φτιάξουμε rows στέλνουμε επαναληπτικά από το “make rows” function ένα msg στο “INSERT TABLE hourly\_data” function και κάνουμε στη συνέχεια insert τα στοιχεία που μας αφορούν στο table της βάσης. Τα requests καλούνται αυτόματα κάθε μέρα στις 23.55 το βράδυ ώστε να παίρνουμε δεδομένα για τη πρόγνωση της επόμενης μέρας από τις 00:00 μέχρι και τις 23:00 σύμφωνα με το format των APIs μας.

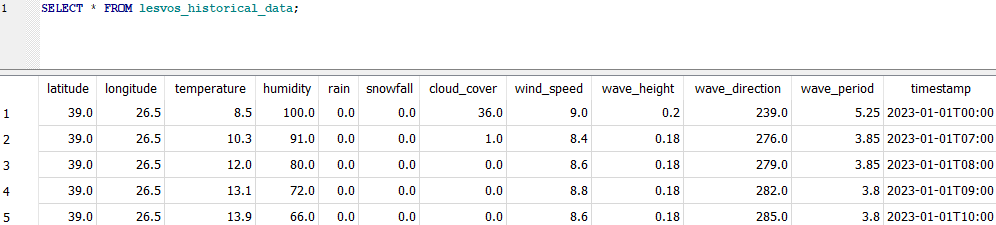
* **piraeus\_historical\_data, crete\_historical\_data, skg\_historical\_data, kerkyra\_historical\_data, lesvos\_historical\_data,** για ιστορικά δεδομένα ανά ημέρα και ανά ώρα για κάθε ένα από τα 5 λιμάνια που μελετάμε για το έτος 2023. Οι πίνακες αυτοί περιέχουν τα παρακάτω στοιχεία:



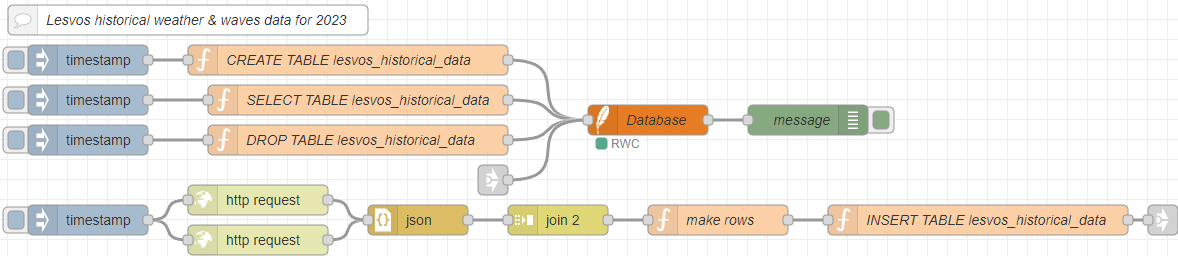
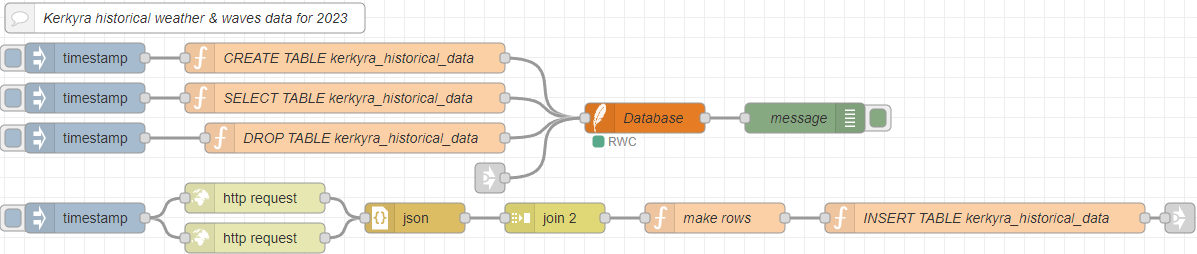
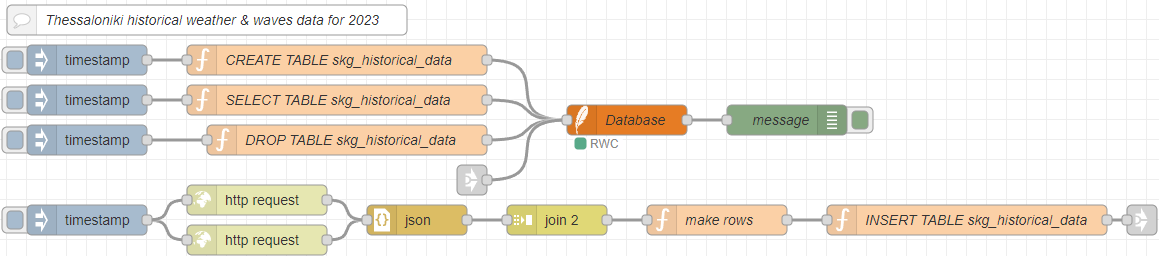
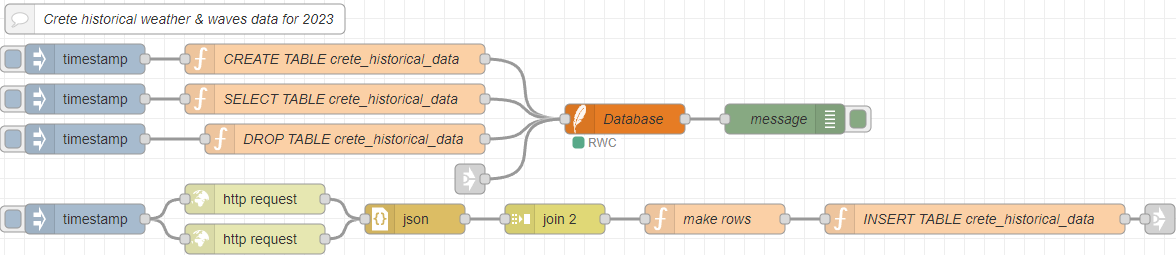
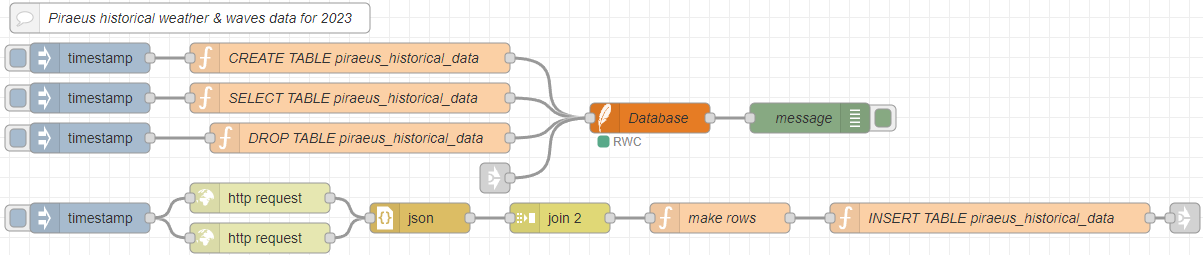








Στο node red αυτοί οι πίνακες υλοποιήθηκαν με αυτό το τρόπο:

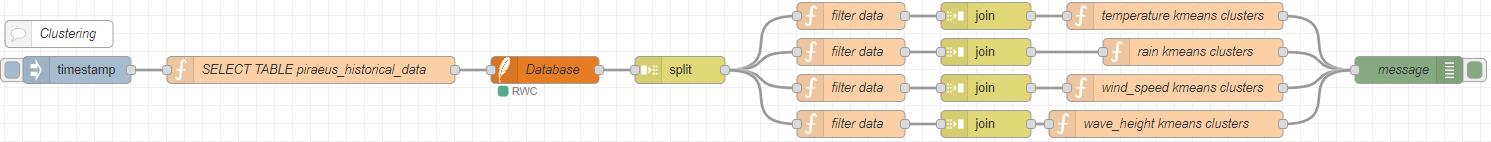


όπως και στα current\_data και hourly\_data, έτσι και εδώ στα functions εκτελούμε τα αντίστοιχα queries για τα tables στη βάση. Η διαδικασία των inserts εδώ είναι ίδια ακριβώς με τα inserts των hourly data παραπάνω. Μόνο που αυτή τη φορά έχουμε πολλά περισσότερα δεδομένα λόγω του χρονικού διαστήματος που συλλέγουμε δεδομένα (1-1-2023 έως 31-12-2023). Τα στοιχεία ωστόσο παραμένουν όλα τα ίδια, πέρα του showers που δεν υπάρχει στα historical data, ενώ στα current και hourly υπάρχει.

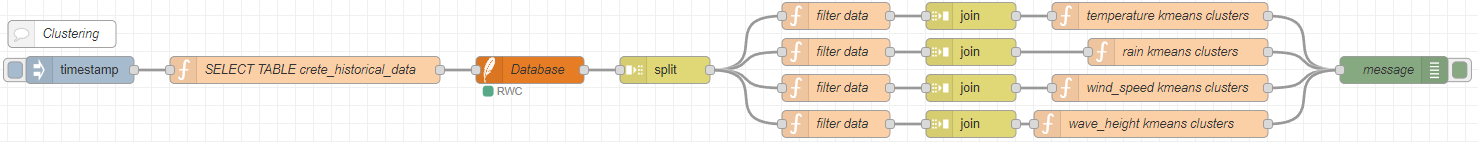
**Καθορισμός ορίων για τις τιμές των δεδομένων (clustering)**

Τα όρια των τιμών καθορίζονται με τη διαδικασία του clustering για τα δεδομένα temperature, rain, wind\_speed και wave\_height, για κάθε ένα από τα 5 λιμάνια, καθώς αυτές οι 4 τιμές καθορίζουν το αν οι συνθήκες καιρού και κυμάτων είναι κατάλληλες για τον απόπλου πλοίων. Η διαδικασία αυτή πραγματοποιείται με τα ιστορικά δεδομένα όπως βλέπουμε παρακάτω:

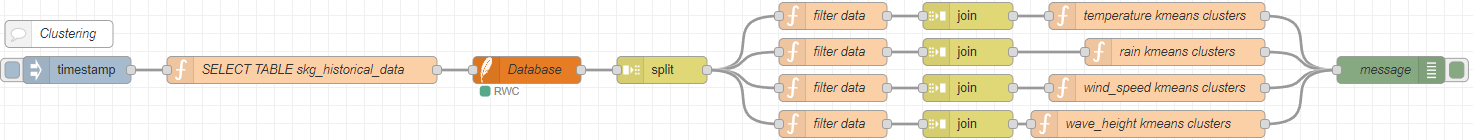
για τον Πειραιά



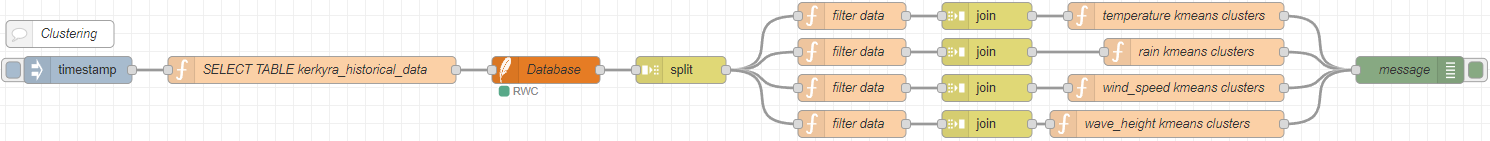
για την Κρήτη



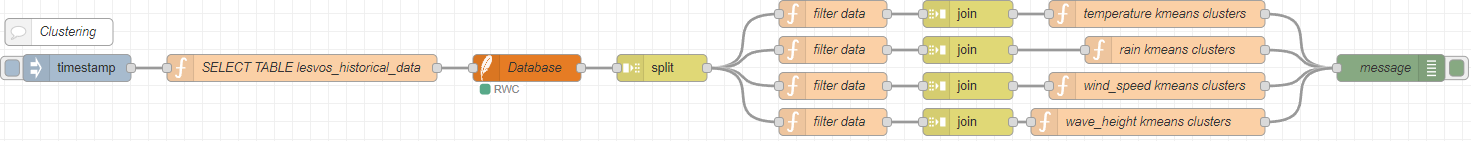
για την Θεσσαλονίκη



για την Κέρκυρα



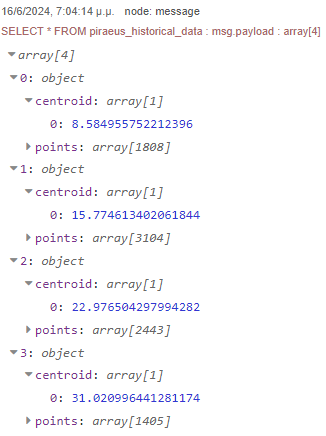
για την Λέσβο



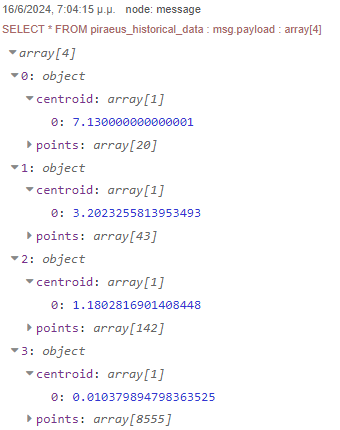
Κάνουμε ένα select table, από τα ιστορικά δεδομένα του κάθε λιμανιού, χωρίζουμε με split τη κάθε γραμμή του πίνακα που μας φέρνει το query, ύστερα φιλτράρουμε τα δεδομένα που θέλουμε κάθε φορά, ανάλογα τη παράμετρο που μας ενδιαφέρει κάθε φορά (temperature, rain, wind\_speed ή wave\_height), κάνουμε join στη συνέχεια όλες τις τιμές της τρέχουσας παραμέτρου και τέλος πραγματοποιούμε το clustering για τη κάθε παράμετρο με τη δημιουργία δικών της k-means global clusters. Το πλήθος των k-means clusters σε όλες τις παραμέτρους είναι 4 επίκεντρα clusters δηλαδή 4 states. Το clustering ουσιαστικά γίνεται με τη ταξινόμηση όλων των τιμών μιας παραμέτρου και τη δημιουργία (4) centroids. Αυτή η διαδικασία γίνεται και για τα 5 λιμάνια, ώστε κάθε λιμάνι να έχει το δικό του εύρος τιμών και τα δικά του centroids. Επομένως δημιουργούμε διαφορετικούς clusters για κάθε λιμάνι και για κάθε παράμετρο.

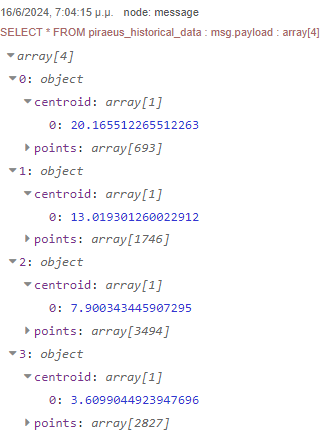
Εδώ έχουμε τα αποτελέσματα με τα centroids στους clusters του Πειραιά

temperature wind speed

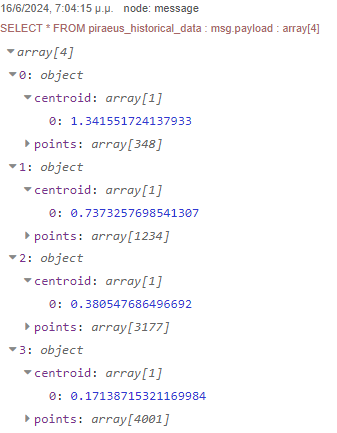


rain





wave height

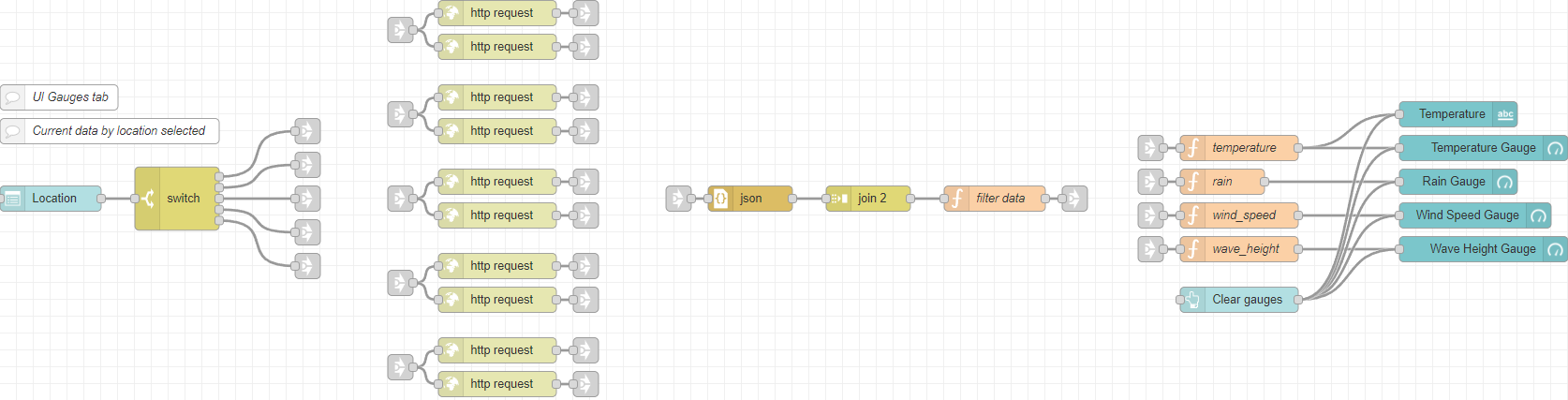


Αντίστοιχα παράγονται οι clusters και για τα υπόλοιπα λιμάνια.

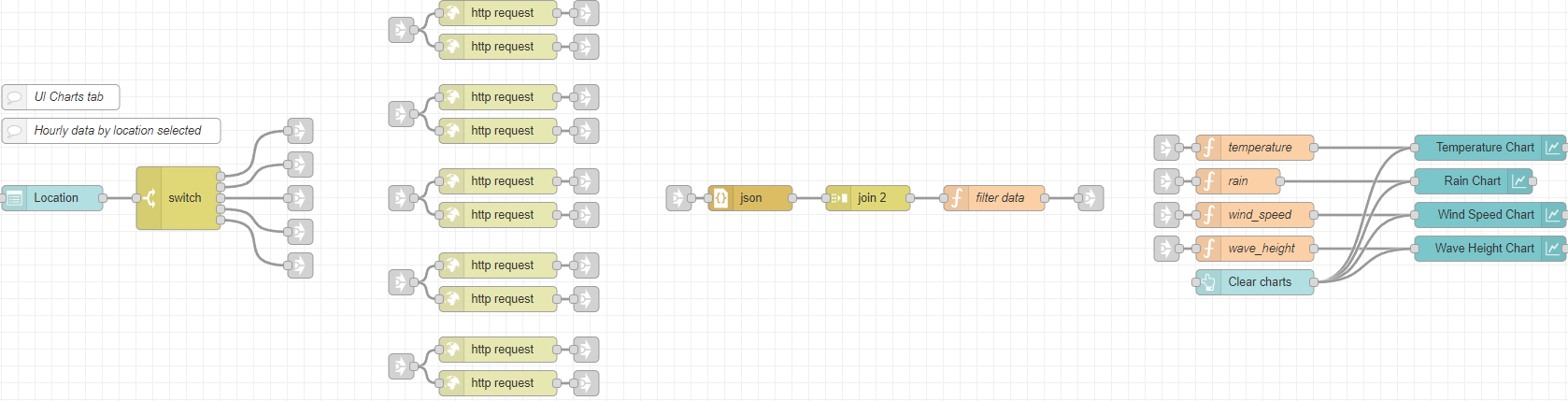
**Dashboards και περιβάλλον χρήστη (UI)**

* **Gauges:** Εμφανίζουν σε πραγματικό χρόνο τιμές για την θερμοκρασία, βροχή, ταχύτητα του ανέμου και ύψος των κυμάτων.
* **Charts:** Δείχνουν τιμές για θερμοκρασία, βροχή, ταχύτητα του ανέμου και ύψος των κυμάτων σε γραμμική αναπαράσταση με την πάροδο του χρόνου.
* **World Map:** Αποτελεί οπτική αναπαράσταση των δεδομένων για κάθε λιμάνι καθώς παρέχει πληροφόρηση σχετικά με την κατάσταση των συνθηκών που επικρατούν σε κάθε περιοχή τη τρέχουσα χρονική στιγμή (εμφανίζεται στο ίδιο tab με τα gauges).

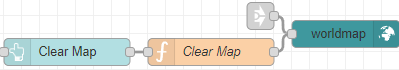
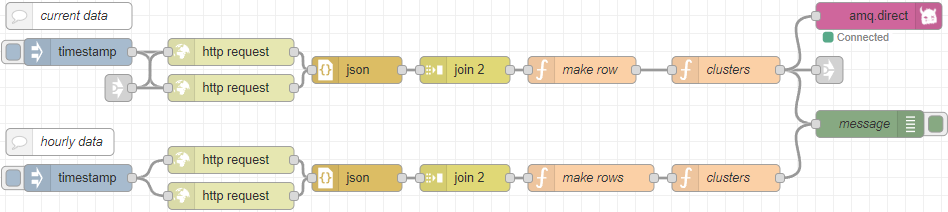
Η ροή για τα gauges



Η ροή για τα charts



Η ροή για το worldmap

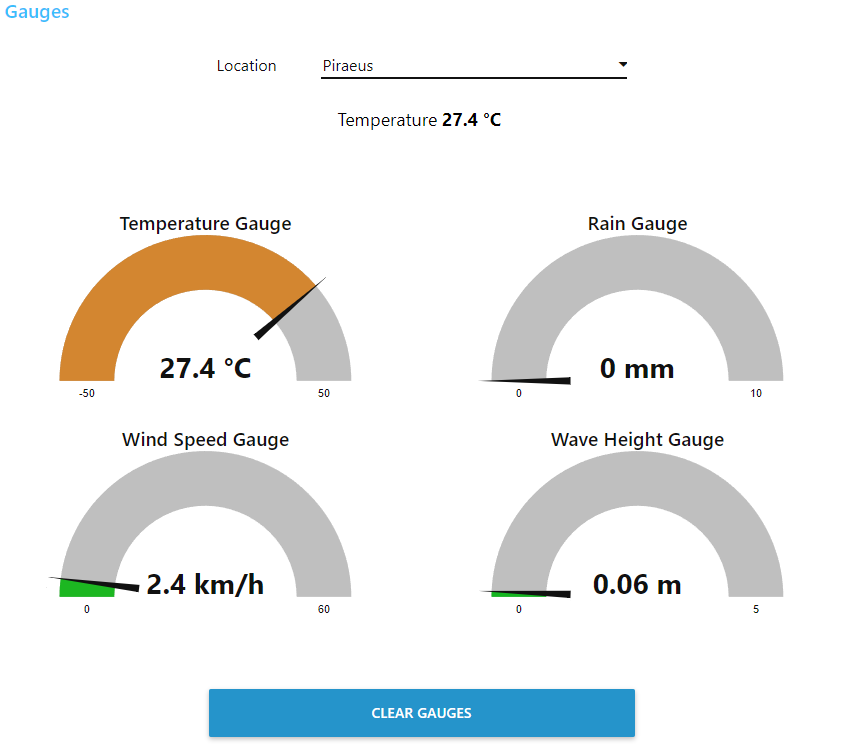


Με ένα dropdown list δίνουμε την δυνατότητα στον χρήστη να επιλέξει ένα από τα λιμάνια (Πειραιάς, Θεσσαλονίκη, Κρήτη, Κέρκυρα ή Λέσβος). Στην συνέχεια με βάση την επιλογή με ένα switch κάνουμε ένα request σε κάθε API (όπως κάνουμε και για την αποθήκευση των current και hourly data) και επιστρέφουμε σε ενα json τα δεδομένα για το λιμάνι που έχει επιλεχθεί. Φιλτράρουμε τα δεδομένα με την συνάρτηση filter data, επιλέγοντας θερμοκρασία, βροχόπτωση, ταχυτητα του ανεμου και ύψος των κυμάτων, τα οποία στην συνέχεια εμφανίζονται είτε με μορφή gauge (με τα αντίστοιχα δεδομενα να εμφανίζονται και επάνω στον χάρτη με το αντίστοιχο χρώμα που έχει καθοριστεί από το clustering), είτε ως chart αναλόγως το tab που έχει επιλέξει ο χρήστης.

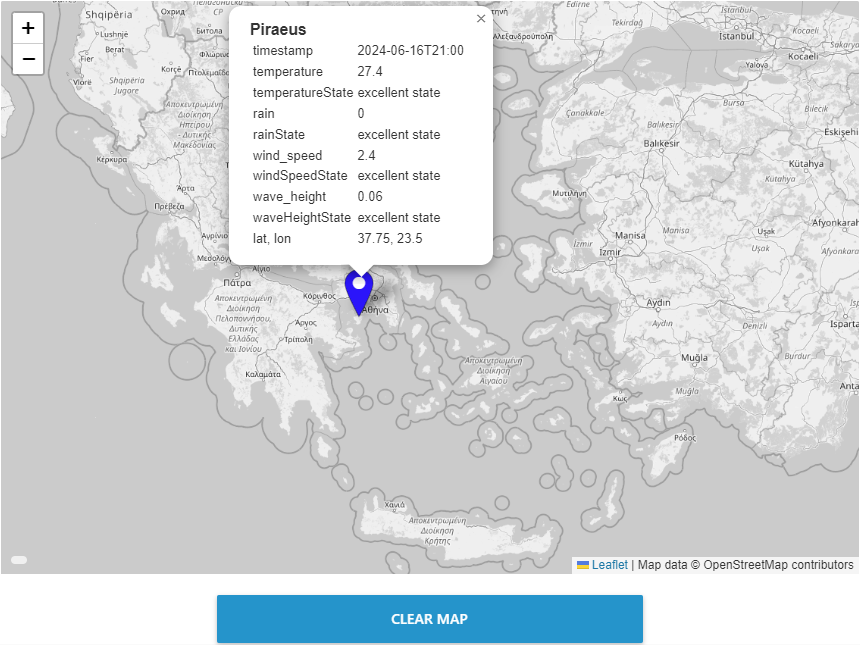
Όταν κάνουμε requests για την σύγκριση των live δεδομένων και την εμφάνιση των states στο debug και στον χάρτη, στέλνουμε και μια ενημέρωση με amq.direct όπως θα δούμε και παρακάτω.

**Παραδείγματα χρήσης του UI**

Gauge Tab (Gauges Group):



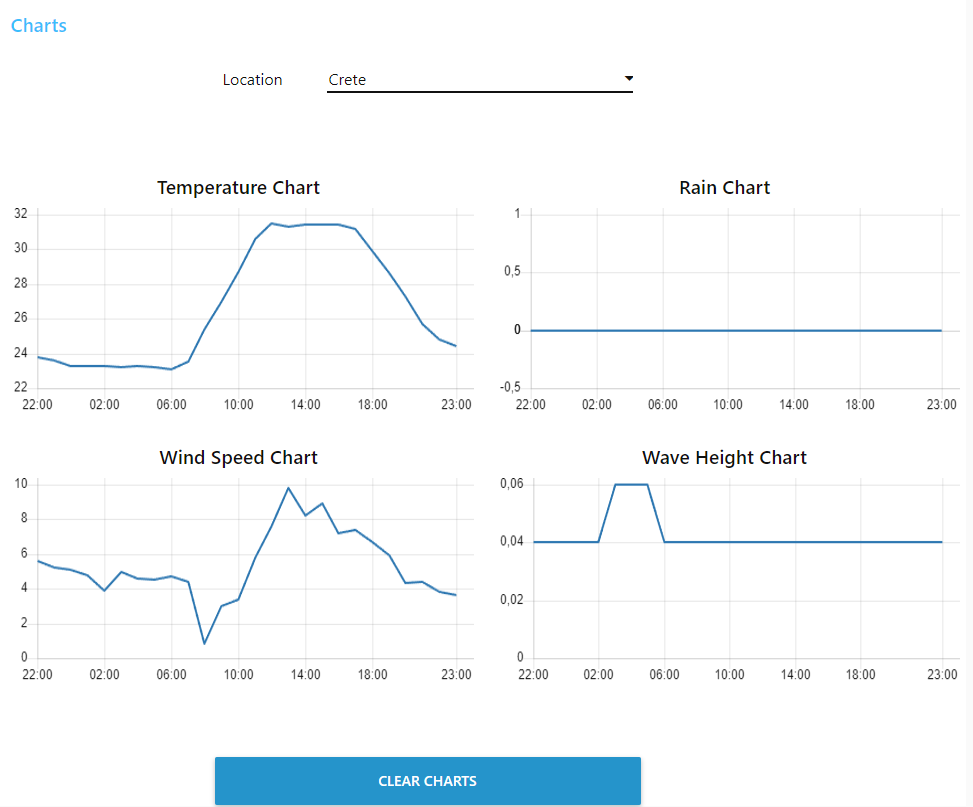
Με το κουμπί “CLEAR GAUGES” καθαρίζουμε τα gauges αν επιθυμούμε.



Στο worldmap αν έστω και μία από τις 4 παραμέτρους (temperature, rain, wind\_speed, wave\_height) έχει χειρότερη τιμή, δηλαδή χειρότερο state από τις υπόλοιπες, τότε η τελική συνθήκη (state) και τελικό χρώμα του marker επάνω στο χάρτη, καθορίζεται από αυτή τη μία παράμετρο.

Για να εμφανίζονται markers με states στο χάρτη, απαραίτητη προϋπόθεση είναι να έχει προηγηθεί η διαδικασία του clustering για το κάθε λιμάνι.

Με το κουμπί “CLEAR MAP” καθαρίζουμε όλα τα markers από τον χάρτη αν επιθυμούμε.

****

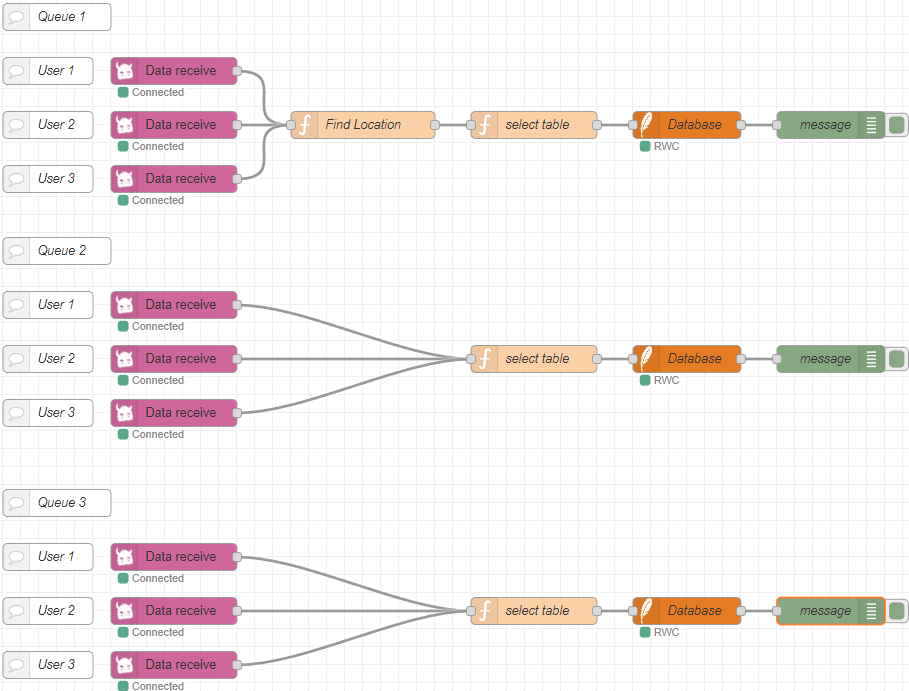
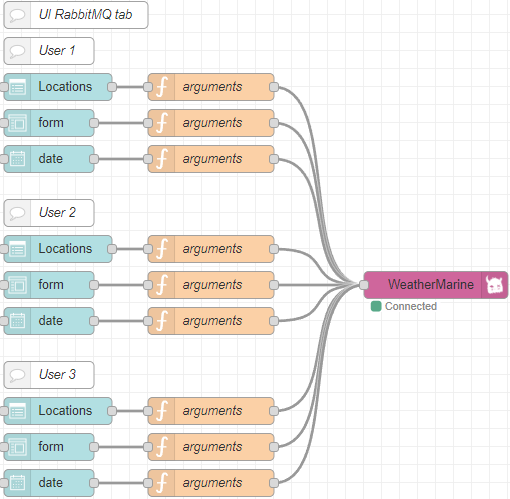
Tα charts απεικονίζουν δεδομένα για τις υπολειπόμενες ώρες της ημέρας καθώς και για όλο το 24ωρο της επόμενης μέρας.

Για να πάρουμε τα δεδομένα του κάθε λιμανιού, πρέπει πρώτα να κάνουμε εκκαθάριση του προηγούμενου πατώντας το κουμπί “CLEAR CHARTS”.

**RabbitMQ**

Με την χρήση του RabbitMQ διαχειριζόμαστε τα δεδομένα για πολλούς χρήστες οργανώνοντας και διανέμοντας μηνύματα μέσω exchanges και queues. Κύριος στόχος είναι τα δεδομένα που λαμβάνονται, να επεξεργάζονται και αποθηκεύονται αποτελεσματικά, επιτρέποντας την παρακολούθηση σε πραγματικό χρόνο αλλά και παλαιότερα.

* **Broker:** Ο broker διαχειρίζεται τα exchanges και τα queues, και τις συνδέσεις μεταξύ producers και consumers.
* **Exchanges:** Για τα exchanges χρησιμοποιούμε direct exchange, με αυτόν τον τρόπο το κάθε exchange βασίζεται στο δικό του routing key.
* **Queues:** Η ροή μας περιλαμβάνει 3 queues (Queue1, Queue2, Queue3) οι οποίες διατηρούνται και δεν διαγράφονται.
* **Routing Keys:** Κάθε queue συνδέεται με διαφορετικά routing keys (User1Key1, User1Key2,... ,User3Key3). Το κάθε κλειδί καθορίζει σε ποία ουρά θα παραδοθεί το μήνυμα.

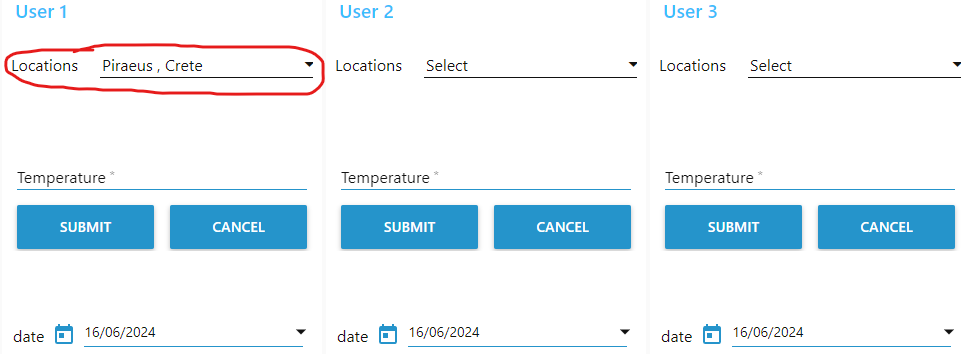


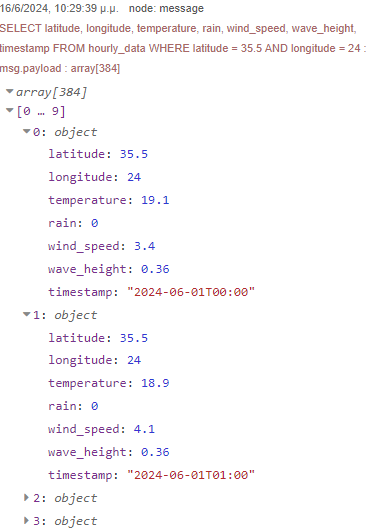
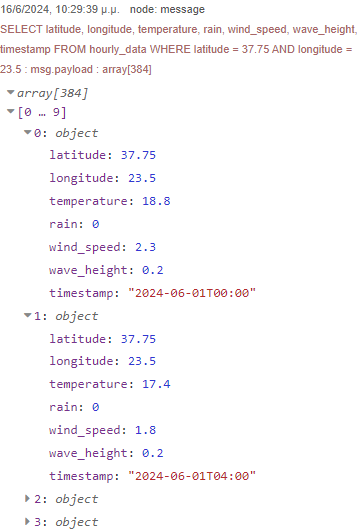
Δημιουργούμε τρία κλειδιά για κάθε έναν από τους 3 Users μας, για την επιλογή τοποθεσίας (μιας ή περισσοτέρων ταυτόχρονα), θερμοκρασίας και ημερομηνίας. Όταν ένας χρήστης επιλέξει ένα από όλα τα παραπάνω (τοποθεσίες, θερμοκρασία ή ημερομηνία) στο “RabbitMQ” tab στο UI, στέλνεται μέσω του “arguments” function το Routing Key του χρήστη στον amqp-out node ο οποίος δημιουργεί ένα direct exchange με όνομα WeatherMarine (για όλους τους χρήστες) και Routing Key το εισερχόμενο. Ύστερα δημιουργούμε 3 queues, τα οποία με ένα select table επιλέγονται και επιστρέφονται τα δεδομένα που ζήτησε ο χρήστης. Η συνάρτηση Find Location καθορίζει την κάθε τοποθεσία σύμφωνα με τις συντεταγμένες (latitude, longitude) και ύστερα τη ξεχωρίζει για να γίνουν ξεχωριστά queries για τη κάθε τοποθεσία που επέλεξε ο χρήστης.

**Παραδείγματα χρήσης του RabbitMQ σε συνδυασμό με το UI**

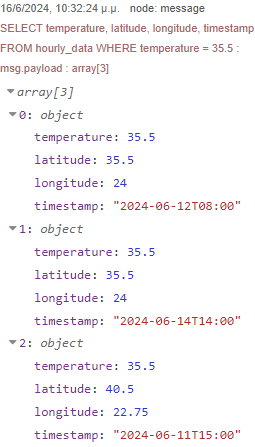
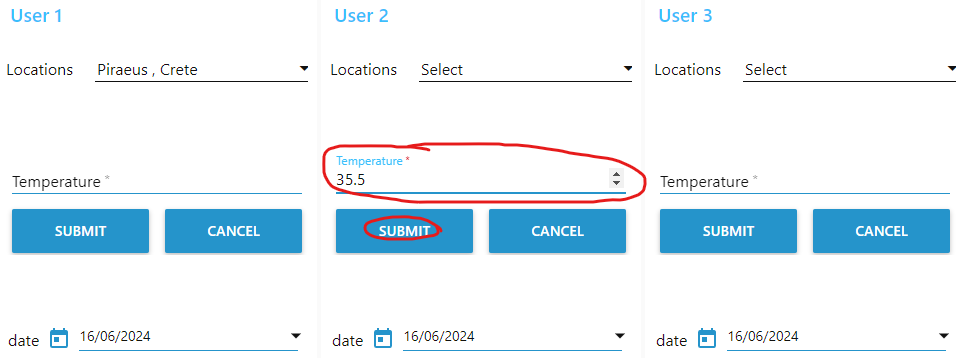
RabbitMQ tab (Groups: User1, User2, User3):

για User1Key1 (Queue1 για τα Locations)

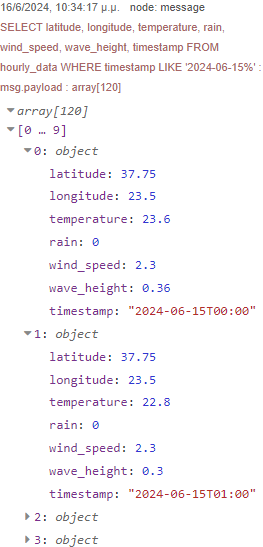
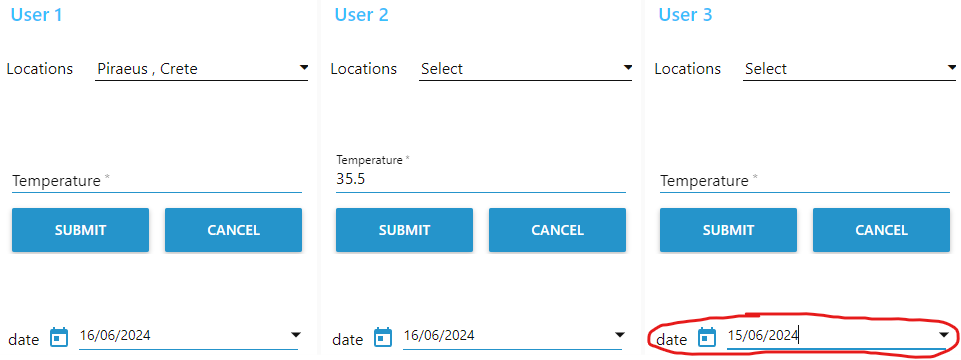




για User2Key2 (Queue2 για τα Temperature)



για User3Key3 (Queue3 για τα date)



Link για το Github που έχουμε το database με όλα τα αποθηκευμένα δεδομένα, καθώς και τα flows και την αναφορά της εργασίας:

<https://github.com/ZenXaris03/Internet-Services-and-Systems-Excercise>