

Εργασία στο Μάθημα της Τεχνολογίας Λογισμικού

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2019-2020

Διδάσκοντες: Ν. Παπασπύρου, Β. Βεσκούκης, Κ. Σαΐδης

Θεματικό πεδίο

Η εργασία αφορά τη διαχείριση δεδομένων της αγοράς ηλεκτρικής ενέργειας στην Ευρώπη, τα οποία διατίθενται ως ανοικτά δεδομένα (Open Data) από το δικτυακό τόπο transparency.entsoe.eu. Τα δεδομένα αυτά είναι κρίσιμα για τη διαφάνεια στη λειτουργία της αγοράς ηλεκτρικής ενέργειας και η διάθεση, οπτικοποίηση και ανάλυσή τους είναι απαραίτητες υπηρεσίες προς τους συμμετέχοντες στην αγορά, ανεξάρτητα από το ρόλο τους (παραγωγοί, πωλητές, συνεταιρισμοί κ.ά.).

Η εμβέλεια της εργασίας περιορίζεται σε 3 επιλεγμένα datasets, τα οποία θα σας διατεθούν και επί των οποίων θα αναπτύξετε ένα σύστημα λογισμικού το οποίο περιγράφεται στη συνέχεια.

Διατύπωση ζητουμένου

Το ζητούμενο της εργασίας είναι να αναπτυχθεί ένα σύστημα λογισμικού για την εισαγωγή επιλεγμένων συνόλων δεδομένων αγοράς ηλεκτρικής ενέργειας που διατίθενται από το transparency.entsoe.eu σε μία ΒΔ (data import) και τη διάθεσή τους, με ελεγχόμενη πρόσβαση, μέσω ενός RESTful Web API. Ανάλογα με το πλήθος των μελών της ομάδας, θα αναπτύξετε επίσης διάφορες εφαρμογές-πελάτες (clients) του Rest API, όπως περιγράφεται παρακάτω.

Τα επιλεγμένα σύνολα δεδομένων είναι τα ακόλουθα:

- **ActualTotalLoad.** Περιέχει την πραγματική ενέργεια που καταναλώθηκε (φορτίο) σε μια χωρική εμβέλεια στην οποία αναφέρεται (χώρα ή ζώνη αγοράς). Οι τιμές που περιέχει (MWh) έχουν προκύψει από μετρήσεις του συστήματος διανομής ηλεκτρικής ενέργειας μετά την κατανάλωση, πχ την επόμενη μέρα.
- **DayAheadTotalLoadForecast.** Περιέχει μια πρόβλεψη για τη συνολική ζήτηση σε ενέργεια κατά την επόμενη μέρα, σε μια χωρική εμβέλεια στην οποία αναφέρεται (χώρα ή ζώνη αγοράς). Οι τιμές που περιέχει (MWh) υπολογίζονται από μαθηματικά μοντέλα πρόβλεψης.
- **AggregatedGenerationPerType.** Αφορά την ανάλυση της πραγματικής παραγωγής ενέργειας στη χώρα ή ζώνη στην οποία αναφέρεται, ανάλογα με τον τρόπο παραγωγής (θερμική, αιολική, ηλιακή, πυρηνική κ.ά.). Περιέχει τιμές (MWh) που έχουν έχουν αθροιστεί ανά τρόπο παραγωγής ηλεκτρικής ενέργειας.

Όλα τα παραπάνω σύνολα δεδομένων αναλύονται χρονικά σε ώρες, δηλαδή οι τιμές που δίνονται είναι ωριαίες. Περισσότερες πληροφορίες για τα περιεχόμενα των συνόλων δεδομένων μπορούν να βρεθούν στις πηγές που αναφέρονται στο τέλος του παρόντος, θα παρουσιαστούν αναλυτικά στο μάθημα και θα δίνονται εξηγήσεις όπως απαιτείται. Σημειώνεται ότι η εκτενής μελέτη της παρεχόμενης από το entso-e τεκμηρίωσης δεν είναι αναγκαία για την εκπόνηση της εργασίας, καθώς απαιτείται βασική κατανόηση της εννοιολογίας (semantics) των συγκεκριμένων συνόλων δεδομένων και μόνο.

Το σύστημα που θα κατασκευάσετε θα αποτελείται από:

1. Ένα back-end υποσύστημα, το οποίο θα υποστηρίζει δυνατότητες διαχείρισης χρηστών (εγγραφή, σύνδεση, αποσύνδεση), θα προσφέρει υπηρεσίες διάθεσης των δεδομένων από τα παραπάνω σύνολα μέσω REST API και θα εφαρμόζει περιορισμούς κατανάλωσης του REST API από τους χρήστες (User Quotas)
2. Μία εφαρμογή CLI (Command Line Interface) για την προσπέλαση στα δεδομένα. Η εφαρμογή θα λειτουργεί ως client του REST API που παρέχεται από το back-end υποσύστημα, παρέχοντας στο χρήστη της τη δυνατότητα να εκτελέσει όλες τις υποστηριζόμενες λειτουργίες (εγγραφή, σύνδεση, προσπέλαση στα σύνολα δεδομένων).
3. **(Μόνο για ομάδες 4 ή 5 ατόμων)** Μία δικτυακή εφαρμογή (Web Application), η οποία θα προσφέρει στο χρήστη περαιτέρω δυνατότητες παρουσίασης και οπτικοποίησης των δεδομένων (διαγράμματα, οπτικοποίηση σε χάρτη). Η εφαρμογή αυτή θα λειτουργεί σε περιβάλλον Web browser και θα αποτελεί το Web-based front-end του συστήματος (δεύτερος - εναλλακτικός client του REST API).
4. **(Μόνο για ομάδες των 5 ατόμων, ένα από τα παρακάτω)**
 - a. Μία εφαρμογή για φορητές συσκευές (Mobile Application), η οποία θα προσφέρει την ίδια ακριβώς λειτουργικότητα με το Web Application (διαγράμματα, χάρτες, λοιπές λειτουργίες) και θα λειτουργεί ως τρίτος - εναλλακτικός client του REST API.
 - b. Ενσωμάτωση λειτουργιών χρέωσης (billing) στην κατανάλωση του REST API, όπου και όπως απαιτείται.
 - c. Ανάλυση των επιδόσεων του back-end υποσυστήματος (benchmarking).

Στοιχεία που θα σας διατεθούν

Πρόκειται να σας διατεθούν τα ακόλουθα στοιχεία:

- **Εννοιολογικό μοντέλο των δεδομένων**, για τα τρία σύνολα δεδομένων που περιγράφονται παραπάνω.
- **Σύνολα δεδομένων σε μορφή csv** (προς import στη ΒΔ), τα οποία θα σας διατεθούν σε 3 εκδόσεις:
 - ο δοκιμαστικά σύνολα: μικρής έκτασης για χρήση στην ανάπτυξη, αναφέρονται σε χρονικό εύρος 2-6 εβδομάδων,

- ο σύνολα παραγωγής: ικανής και διαχειρίσιμης για τις απαιτήσεις της εργασίας έκτασης, αναφέρονται σε χρονικό εύρος 2-6 μηνών
 - ο πλήρη σύνολα: αναφέρονται σε χρονικό εύρος πλέον των 2 ετών, ενδεχομένως να μην είναι πρακτικά εφικτή η χρήση τους ανάλογα με τους περιορισμούς του υπολογιστικού συστήματος που χρησιμοποιείται.
- **Αναλυτική προδιαγραφή των REST API endpoints**
 - **Έκθεση απαιτήσεων για τις λειτουργίες των τριών εφαρμογών - πελατών.**

Ομάδες εργασίας

Η εργασία θα υλοποιηθεί **από ομάδες των 3, 4 ή 5 ατόμων**, οι οποίες θα υλοποιήσουν τον πλήρη κύκλο ανάπτυξης του συστήματος (ανάλυση απαιτήσεων, σύνταξη προδιαγραφών, σχεδιασμός και αρχιτεκτονική, υλοποίηση και έλεγχοι αποδοχής, εγκατάσταση και λειτουργία).

Για τη συνεργατική διαχείριση των εκδόσεων της τεκμηρίωσης και του πηγαίου κώδικα, είναι **υποχρεωτική** η χρήση του Github, όπως αναφέρεται και ακολούθως. Η συγκρότηση των ομάδων και οι λογαριασμοί χρηστών των μελών τους στο Github θα πρέπει να μας γνωστοποιηθούν το αργότερο μέχρι τις **10.11.2019**, σύμφωνα με οδηγίες που θα ανακοινωθούν. Θα πρέπει, επίσης, **με τον ίδιο λογαριασμό χρήστη** (email), να εγγραφείτε στο συνεργατικό σύστημα ηλεκτρονικής μάθησης που χρησιμοποιείται στο μάθημα (moodle.softlab.ntua.gr), ώστε να έχετε άμεση πρόσβαση σε επικοινωνίες, ανακοινώσεις, υλικό, κ.ά.. Στο ίδιο σύστημα θα μπορείτε να στέλνετε τις όποιες απορίες σας σχετικά με την εργασία.

Ελάχιστες Κοινές Τεχνικές Προδιαγραφές

Το σύστημα που θα αναπτύξετε θα πρέπει να υποστηρίζει τα εξής χαρακτηριστικά (ελάχιστες κοινές προδιαγραφές):

1. Το σύστημα θα αποτελείται από (α) ένα back-end και (β) ένα ή περισσότερα front-end υποσυστήματα, ανάλογα με το πλήθος των μελών της ομάδας.
2. Οι γλώσσες προγραμματισμού που μπορείτε να χρησιμοποιήσετε είναι μία εκ των Java, JavaScript και Python. Δεν γίνεται δεκτή η χρήση άλλων γλωσσών προγραμματισμού.
3. Η γλώσσα των διεπαφών χρήστη στην εφαρμογή CLI θα πρέπει να είναι η αγγλική
4. Η γλώσσα των διεπαφών χρήστη για τις άλλες εφαρμογές, όπου απαιτείται, θα είναι η ελληνική.
5. Η εργασία θα υλοποιηθεί με χρήση **αποκλειστικά** της συστήματος διαχείρισης εκδόσεων git και της συνεργατικής υπηρεσίας Github (www.github.com). Δεν επιτρέπεται να χρησιμοποιηθούν άλλα παρεμφερή περιβάλλοντα (πχ Bitbucket, GitLab). Το repository θα πρέπει

να είναι ιδιωτικό (private) και θα έχουν πρόσβαση μόνο τα μέλη της ομάδας και οι διδάσκοντες.

6. Η ομάδα θα πρέπει να κάνει χρήση ενός εργαλείου αυτοματισμού του «χτισίματος» του λογισμικού (build automation), το οποίο στην περίπτωση της Java θα είναι το Gradle, ενώ στην περίπτωση της Javascript θα είναι της επιλογής σας. Στο πλαίσιο του μαθήματος θα γίνει φροντιστήριο / εργαστήριο για το Gradle.
7. Η ομάδα συντάξει σενάρια ελέγχου και θα ενσωματώσει την εκτέλεση των αντίστοιχων δοκιμών με αυτόματο τρόπο για τις λειτουργίες του back-end υποσυστήματος. Για το σκοπό αυτό, στην της Java θα χρησιμοποιηθεί το εργαλείο Spock, ενώ στην περίπτωση της Javascript, το εργαλείο θα είναι της επιλογής σας. Στο πλαίσιο του μαθήματος θα γίνει φροντιστήριο / εργαστήριο για το Spock.
8. Το back-end υποσύστημα θα παρέχει κατάλληλο REST API για τη διασύνδεσή του τις υπόλοιπες εφαρμογές, το οποίο θα πρέπει να είναι συμβατό με το πρότυπο OpenAPI 3.0. Τα endpoints που θα αναπτύξετε θα προδιαγραφούν λεπτομερώς από τους διδάσκοντες κατά τη διάρκεια υλοποίησης της εργασίας και θα είναι κοινά για όλες τις ομάδες.
9. Η πρόσβαση στο REST API θα πρέπει να γίνεται με λογαριασμούς χρηστών. Κάθε λογαριασμός χρήστη θα περιέχει εκτός από τα συνηθισμένα δεδομένα (username, password, email κλπ) και όριο χρήσης (quota) ώστε να περιορίζεται η κατανάλωση πόρων του API.
10. Θα πρέπει να υποστηρίζεται το πρωτόκολλο HTTPS για όλες τις χρηστικές ή προγραμματιστικές διεπαφές μέσω self-signed certificate.

Οι απαιτήσεις αυτές εξειδικεύονται ανάλογα με το πλήθος των μελών της ομάδας, όπως φαίνεται στον ακόλουθο πίνακα:

ΑΠΑΙΤΗΣΗ	Ομάδες 3 ατόμων	Ομάδες 4 ατόμων	Ομάδες 5 ατόμων (*)
1. Εισαγωγή (import) και διαχείριση δεδομένων	Back-end CLI		
2. Διαχείριση χρηστών και περιορισμών	Back-end CLI	Back-end CLI Web-based front-end	Back-end CLI Web-based front-end Mobile App (*)
3. Πρόσβαση σε πρωτογενή (RAW) δεδομένα	Back-end CLI		
4. Παρουσίαση δεδομένων σε περιβάλλον browser		Web-based front-end	Web-based front-end
5. Παρουσίαση δεδομένων σε περιβάλλον smartphone (*)			Mobile app (*)
6. Λειτουργίες χρέωσης του REST API (billing) (*)			Back-end CLI Web-based front-end

7. Ανάλυση επιδόσεων του REST API (benchmarking) (*)			Back-end CLI
------------------------------------------------------	--	--	-----------------

(*) Οι ομάδες των 5 ατόμων πρέπει να επιλέξουν **ένα** από τα 5, 6 και 7.

Τεχνικοί περιορισμοί

Επιπλέον όσων αναφέρθηκαν ισχύουν και ορισμένοι άλλοι τεχνικοί περιορισμοί για την εκπόνηση της εργασίας. Το σύνολο των περιορισμών συνοψίζεται ως ακολούθως:

- Περιβάλλον διαχείρισης εκδόσεων: **github**
- Γλώσσες προγραμματισμού: **Java, Javascript, Python**
- Εργαλείο παραγωγής διαγραμμάτων UML: **Visual Paradigm 16, community edition**
- Εργαλείο παραγωγής OpenAPI κώδικα και τεκμηρίωσης (*): **Visual Paradigm 16 Professional, one-month trial version**
- Πρότυπο υλοποίησης και τεκμηρίωσης REST API: **OpenAPI 3**
- Διαχείριση δεδομένων: ένα εκ των **MySQL, MariaDB, PostgreSQL, Mongo, Elastic Search**

(*) Δεν είναι υποχρεωτικό, αν και ενθαρρύνεστε να χρησιμοποιήσετε εργαλείο για το σκοπό αυτό.

Τεχνικές επιλογές οι οποίες δεν περιλαμβάνονται στους παραπάνω περιορισμούς (πχ βιβλιοθήκες JS για διαγράμματα / γραφήματα στο web), είναι ελεύθερες.

Παράδοση εργασίας

Η εργασία απαιτεί την κατάθεση ενδιάμεσων παραδοτέων στο σύνολο της διάρκειας του εξαμήνου, οι προθεσμίες των οποίων θα ανακοινώνονται κατά τη διάρκεια του μαθήματος. Τα παραδοτέα αυτά είναι στοιχεία τεκμηρίωσης του συστήματος (documentation), σχεδίασης (architecture, design), πηγαίου κώδικα, περιπτώσεων ελέγχου, κ.ά.. Το σύνολο των στοιχείων αυτών (όχι μόνο ο πηγαίος κώδικας) θα "ανεβαίνει" στο github κάθε ομάδας.

Το τελικό προϊόν, ως ολοκληρωμένο και πλήρως λειτουργικό λογισμικό θα πρέπει να **έχει ολοκληρωθεί και παραδοθεί έως την ημερομηνία εξέτασης του μαθήματος** στην εξεταστική περίοδο του Φεβρουαρίου. Η παράδοση της εργασίας γίνεται αυτόματα μέσω του git repository της ομάδας και θα ληφθούν υπόψη μόνο τα περιεχόμενα του repository σας που δεν είναι μεταγενέστερα της ημερομηνίας εξέτασης, με βάση τη χρονοσήμανση των καταχωρήσεων (commits). Οι λεπτομέρειες που αφορούν την τελική εξέταση της εργασίας, την εγκατάστασή της (deployment) και την (ενδεχομένως δημόσια) επίδειξή της θα ανακοινωθούν στη διάρκεια του μαθήματος.

Πηγές πληροφοριών

- <https://transparency.entsoe.eu/>

- https://transparency.entsoe.eu/content/static_content/Static%20content/knowledge%20base/SFTP-Transparency_Docs.html αναφορικά με όσα περιγράφονται στις ενότητες 6.1.1, 6.1.2 και 6.2.3
- <https://www.openapis.org/>
- <https://swagger.io/specification/>
- <https://www.visual-paradigm.com/>
- <https://www.visual-paradigm.com/features/visual-api-designer/>
- <https://www.elastic.co/>
- <https://www.mongodb.com/>