Εργασία στο Μάθημα της Τεχνολογίας Λογισμικού - Προδιαγραφές του RESTful Web API του Παρατηρητηρίου

Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2018-19

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

Γενικές αρχές

Το διαδικτυακό παρατηρητήριο τιμών που θα αναπτύξετε θα πρέπει να υποστηρίζει ένα RESTful Application Programming Interface (REST API) για την καταχώριση, ανάκτηση και αναζήτηση των δεδομένων του.

Base URL

Το REST API θα είναι διαθέσιμο στο ακόλουθο base URL για όλες τις εργασίες:

https://localhost:8765/observatory/api

Τα επιμέρους Resources (REST endpoints) που θα διατίθενται μέσω του API θα είναι προσβάσιμα μέσω του παραπάνω base URL, ως εξής:

{baseURL}/{path-to-resource}

Για παράδειγμα, το endpoint παράθεσης των προϊόντων θα είναι το εξής:

{baseURL}/products

Ενώ το endpoint για την ανάκτηση των δεδομένων του προϊόντος με κωδικό 123 θα είναι το εξής:

{baseURL}/products/123

Μορφότυπος δεδομένων

Το REST API θα υποστηρίζει τον μορφότυπο JSON (content-type: application/json) για τις ομάδες 5 και 6 ατόμων και τον μορφότυπο XML (content-type: text/xml) για τις ομάδες 7 ατόμων. Η επιλογή του μορφότυπου θα καθορίζεται στην αίτηση ως εξής (query parameter):

{baseURL}/{path-to-resource}?format={json|xml}

Αν η παράμετρος format δεν παρέχεται σε κάποια αίτηση, μπορείτε να θεωρήσετε ότι το json θα είναι η default τιμή. Οι ομάδες των 5 και 6 ατόμων θα πρέπει να επιστρέφουν ένδειξη σφάλματος σε κάθε αίτηση που το format έχει την τιμή xml (π.χ. Κωδικός 400, Bad Request). Οι ομάδες των 7 ατόμων θα πρέπει να υποστηρίξουν πλήρως και τους δύο μορφότυπους.

Σε κάθε περίπτωση η κωδικοποίηση χαρακτήρων (character encoding) θα πρέπει να είναι UTF8.

Διαπίστευση και δικαιοδοσία χρηστών

Τα endpoints ανάκτησης και αναζήτησης δεδομένων του API θα είναι διαθέσιμα στο ευρύ κοινό ως ανοικτά δεδομένα (Open Data) και δεν θα απαιτούν κανένα στοιχείο διαπίστευσης (username, password, API key, token, κλπ.). Αντίθετα, τα endpoints προσθήκης, ενημέρωσης ή διαγραφής των δεδομένων θα είναι διαθέσιμα μόνο σε διαπιστευμένους χρήστες (authenticated χρήστες που έχουν κάνει login στην εφαρμογή), οι οποίοι θα πρέπει να έχουν και την ειδική δικαιοδοσία για καταγραφή δεδομένων (authorized χρήστες με το ρόλο του Εθελοντή). Προφανώς και οι χρήστες με το ρόλο του Διαχειριστή μπορούν να εκτελέσουν όλες τις ενέργειες που μπορούν να εκτελεστούν από τους Εθελοντές (με ειδικό χειρισμό σε κάποιες περιπτώσεις, όπως περιγράφεται στη συνέχεια).

Τα «διαπιστευτήρια» του χρήστη, κωδικοποιημένα με τον τρόπο που εσείς κρίνετε πιο συμβατό με τη σχετική βέλτιστη διεθνή πρακτική, θα πρέπει να παρέχονται σε ειδικό για το σκοπό αυτό custom HTTP Header στις αιτήσεις προσθήκης, αλλαγής ή διαγραφής δεδομένων. Το όνομα του custom HTTP header θα πρέπει να είναι X-OBSERVATORY-AUTH.

Διαχείριση σφαλμάτων

Κάθε κλήση στο API θα πρέπει να επιστρέφει τα κατάλληλα HTTP Status Codes σε περίπτωση σφάλματος. Ειδικότερα, σε περίπτωση που ο ζητούμενος πόρος δεν υπάρχει (π.χ. λάθος κωδικός προϊόντος), τότε θα πρέπει να επιστρέφεται το 404 – Not Found. Σε περίπτωση που ένας διαπιστευμένος χρήστης δεν έχει τη δικαιοδοσία να εκτελέσει μια ενέργεια, θα πρέπει να επιστρέφεται το 401 Not Authorized. Σε περίπτωση που ένας μη διαπιστευμένος χρήστης δεν επιτρέπεται να εκτελέσει μια ενέργεια θα πρέπει να επιστρέφεται το 403 – Forbidden. Τέλος, σε περίπτωση που οι παράμετροι που δίνονται από τον χρήστη σε μία κλήση δεν είναι έγκυροι (π.χ. κενό υποχρεωτικό πεδίο), το σύστημα θα πρέπει να επιστρέφει το 400 – Bad Request.

Προϊόντα

Για τη διαχείριση των προϊόντων, οι προδιαγραφές του API είναι οι εξής.

GET {baseURL}/products

Επιστρέφεται η λίστα των προϊόντων του παρατηρητηρίου.

Υποστηριζόμενες παράμετροι (ως μέρος του URL query)

start	Integer, default 0
count	Integer, default 20
status	String, με επιτρεπτές τιμές: ALL | WITHDRAWN | ACTIVE, default ACTIVE
sort	String, με επιτρεπτές τιμές: id|ASC, id|DESC, name|ASC, name|DESC, default id|DESC

Παράδειγμα:

GET {baseURL}/products?start=0&count=100&sort=id|ASC&status=ALL

Τα αποτελέσματα επιστρέφονται με την εξής κωδικοποίηση:

start	Integer
count	Integer
total	Long
products	List<Product>

Τα πεδία start, count, total αφορούν στη «σελιδοποίηση» των αποτελεσμάτων, ενώ το πεδίο products θα περιέχει τη λίστα με τα προϊόντα που περιέχονται στη συγκεκριμένη «σελίδα». Στη λίστα products, κάθε προϊόν περιέχει τα εξής ελάχιστα κοινά πεδία:

id	String ή Long	Μοναδικός προσδιοριστής προϊόντος.
name	String	Το όνομα του προϊόντος.
description	String	Η περιγραφή του προϊόντος.
category	String	Η κατηγορία του προϊόντος (π.χ. Laptop, Τηλεόραση, κλπ), η οποία καθορίζει την ειδική κλάση που θα περιέχει ενδεχόμενα πρόσθετα πεδία.
tags	List<String>	Λίστα από tags, πρόσθετα του category.
withdrawn	Boolean	Ένδειξη για το αν οι τιμές του προϊόντος έχουν πάψει να «καταγράφονται» στο παρατηρητήριο, default false.

Η έννοια «προϊόν» στην εφαρμογή σας μπορεί να αναφέρεται είτε σε προϊόντα, είτε σε υπηρεσίες. Σε κάθε περίπτωση, όμως, θα πρέπει να επιστρέφονται τα παραπάνω ελάχιστα κοινά πεδία.

Τα όποια πρόσθετα πεδία που μπορεί να υποστηρίζονται από την εφαρμογή σας (προαιρετικά), εφόσον υπάρχουν, θα πρέπει να παρέχονται ομαδοποιημένα σε ένα πεδίο extraData (τα πεδία που περιέχονται κάθε φορά στα extraData προκύπτουν από το category του κάθε προϊόντος).

Παράδειγμα κωδικοποίησης ενός προϊόντος σε μορφή JSON/XML.

JSON	XML

{
“id”: “123”,
“name”: “Vendor X Laptop Model 987”,
“description”: “….”,
“category”: “Laptop”,
“tags”: [“computing”, “laptops”],
“withdrawn”:false,
“extraData”: {
“RAM”: “8GB”,
“HD”: “1TB”,
…
}
}
	<?xml version="1.0" encoding="UTF-8"?>
	<product>
	<id>123</id>
	<name>Vendor X Laptop Model 987</name>
	<description>….</description>
	<category>Laptop</category>
	<tags>computing, laptop</tags>
	<withdrawn>false</withdrawn>
	<extraData>
	<RAM>8GB</RAM>
	<HD>1TB</HD>
	…
	</more>
	</product>

Παράδειγμα κωδικοποίησης των αποτελεσμάτων μιας επίκλησης της GET {baseURL}/products

JSON	XML
{ “start”: 0, “count”: 20, “total”: 3823, “products”: [	<?xml version="1.0" encoding="UTF-8"?> <results> <start>0</start> <count>20</count> <total>3823</total>

{…product 1…}, {…product 2…}, …, {…product 20…},] }	<products> <product>…product 1…</product> <product>…product 2…</product> … <product>…product 20…</product> </products> </results>

POST {baseURL}/products

Δημιουργία νέου προϊόντος.

Η αίτηση θα πρέπει να περιέχει στο body (και όχι στο query) τις τιμές για τα υποχρεωτικά πεδία του προς δημιουργία προϊόντος (name, description, category, tags).

Προαιρετικά, θα μπορούσατε να υποστηρίξετε και τα πρόσθετα πεδία του προϊόντος που υποστηρίζονται από το ανά περίπτωση category.

Το αποτέλεσμα της αίτησης, σε περίπτωση επιτυχούς δημιουργίας νέου προϊόντος, θα είναι η πλήρης κωδικοποίηση των δεδομένων του, όπως περιγράφηκε προηγουμένως.

GET {baseURL}/products/{id}

Επιστρέφονται τα δεδομένα του προϊόντος με το συγκεκριμένο id. Η κωδικοποίηση των δεδομένων θα γίνεται όπως έχει ήδη περιγραφεί προηγουμένως,.

Η μόνη υποστηριζόμενη παράμετρος είναι το format.

PUT {baseURL}/products/{id}

Ενημέρωση των στοιχείων του προϊόντος με το συγκεκριμένο id.

Τα πεδία, όπως και στην περίπτωση της POST, θα πρέπει να κωδικοποιηθούν στο body και όχι στο query της αίτησης. Η αίτηση θα πρέπει να περιλαμβάνει το σύνολο των υποστηριζόμενων πεδίων του κάθε προϊόντος, αντικαθιστώντας όλες τις προηγούμενες τιμές (FULL UPDATE).

Το αποτέλεσμα θα είναι η πλήρης κωδικοποίηση των δεδομένων του προϊόντος.

PATCH {baseURL}/products/{id}

Μερική ενημέρωση του προϊόντος με το συγκεκριμένο id.

Η κλήση αυτή χρησιμοποιείται μόνο όταν απαιτείται η αλλαγή ενός μόνο πεδίου (PARTIAL UPDATE) και τα δεδομένα της αίτησης κωδικοποιούνται κι αυτά στο body και όχι στο query αυτής.

Το αποτέλεσμα θα είναι η πλήρης κωδικοποίηση των δεδομένων του προϊόντος.

DELETE {baseURL}/products/{id}

Διαγραφή του προϊόντος με το συγκεκριμένο id.

Αν ο χρήστης που εκτελεί την αίτηση έχει το ρόλο του Εθελοντή, το σύστημα θα πρέπει να θέσει την τιμή withdrawn=true στο προϊόν. Αν ο χρήστης έχει το ρόλο του Διαχειριστή, το σύστημα θα προχωρά στη διαγραφή του προϊόντος και όλων των σχετικών του πληροφοριών (π.χ. τιμές, σημεία πώλησης).

Το αποτέλεσμα θα είναι απλό μήνυμα επιβεβαίωσης, όπως φαίνεται στον ακόλουθο πίνακα.

JSON	XML
{ “message”: “OK” }	<message>OK</message>

Καταστήματα / σημεία πώλησης

Για τη διαχείριση των καταστημάτων / σημείων πώλησης, οι προδιαγραφές του API είναι οι εξής.

GET {baseURL}/shops

Επιστρέφεται η λίστα των καταστημάτων του παρατηρητηρίου.

Οι υποστηριζόμενες παράμετροι (ως μέρος του URL query), είναι ίδιες με την περίπτωση των προϊόντων, με την εξαίρεση της extraData, που δεν υποστηρίζεται για τα καταστήματα.

Παράδειγμα:

GET {baseURL}/shops?start=0&count=100&sort=id|ASC&status=ACTIVE

Τα αποτελέσματα επιστρέφονται με την εξής κωδικοποίηση:

start	Integer
count	Integer
total	Long
shops	List<Shop>

Τα πεδία start, count, total αφορούν στη «σελιδοποίηση» των αποτελεσμάτων, ενώ το πεδίο shops θα περιέχει τη λίστα με τα καταστήματα που περιέχονται στη συγκεκριμένη «σελίδα». Στη λίστα shops, κάθε κατάστημα περιέχει τα εξής ελάχιστα κοινά πεδία:

id	String ή Long	Μοναδικός προσδιοριστής καταστήματος.
name	String	Το όνομα του καταστήματος.
address	String	Η διεύθυνση του καταστήματος.
lng	Double	Το γεωγραφικό μήκος της θέσης του καταστήματος(*).
lat	Double	Το γεωγραφικό πλάτος της θέσης του καταστήματος(*).
tags	List<String>	Λίστα από tags.
withdrawn	Boolean	Ένδειξη για το αν το κατάστημα έχει πάψει να «καταγράφεται» στο παρατηρητήριο, default false.

(*) Στο σύστημα συντεταγμένων WGS84.

Η έννοια «κατάστημα» στην εφαρμογή σας μπορεί να έχει κι αυτή να προσαρμοστεί σε όποιες ειδικές απαιτήσεις απαιτούνται από την προσέγγισή σας, σε κάθε περίπτωση, όμως, θα πρέπει να επιστρέφονται τα παραπάνω ελάχιστα κοινά πεδία.

Παράδειγμα κωδικοποίησης ενός καταστήματος σε μορφή JSON/XML.

JSON	XML
{	<?xml version="1.0" encoding="UTF-8"?>

“id”: “321”, “name”: “Κατάστημα Ηλεκτρονικών Χ Ζωγράφου”, “address”: “Οδός 13, 12345, Ζωγράφου” “lng”: 32.12345345, “lat”: 33.02312412 “tags”: [“computing”, “laptops”], “withdrawn”: false }	<shop> <id>321</id> <name> Κατάστημα Ηλεκτρονικών Χ Ζωγράφου</name> <address> Οδός 13, 12345, Ζωγράφου</address> <lng> 32.12345345</lng> <lat>33.02312412</lat> <tags>computing, laptops</tags> <withdrawn>false</withdrawn> </shop>

Παράδειγμα κωδικοποίησης των αποτελεσμάτων μιας επίκλησης της GET {baseURL}/shops

JSON	XML

{
“start”: 0,
“count”: 20,
“total”: 187,
“shops”: [
{…shop 1…},
{…shop 2…},
…,
{…shop 20…},
]
}
	<?xml version="1.0" encoding="UTF-8"?>
	<results>
	<start>0</start>
	<count>20</count>
	<total>187</total>
	<shops>
	<shop>…shop 1…</shop>
	<shop>…shop 2…</shop>
	…
	<shop>…shop 20…</shop>
	</shops>
	</results>

POST {baseURL}/shops

Δημιουργία νέου καταστήματος.

Η αίτηση θα πρέπει να περιέχει στο body (και όχι στο query) τις τιμές για όλα τα πεδία του προς δημιουργία προϊόντος (όλα είναι υποχρεωτικά).

Το αποτέλεσμα της αίτησης, σε περίπτωση επιτυχούς δημιουργίας νέου προϊόντος, θα είναι η πλήρης κωδικοποίηση των δεδομένων του, όπως περιγράφηκε προηγουμένως.

GET {baseURL}/shops/{id}

Επιστρέφονται τα δεδομένα του καταστήματος με το συγκεκριμένο id. Η κωδικοποίηση των δεδομένων θα γίνεται όπως έχει ήδη περιγραφεί προηγουμένως,.

Η μόνη υποστηριζόμενη παράμετρος είναι το format.

PUT {baseURL}/shops/{id}

Ενημέρωση των στοιχείων του καταστήματος με το συγκεκριμένο id. Οι προδιαγραφές είναι ίδιες με την PUT {baseURL}/products/{id}.

PATCH {baseURL}/shops/{id}

Μερική ενημέρωση του καταστήματος με το συγκεκριμένο id. Οι προδιαγραφές είναι ίδιες με την PATCH {baseURL}/products/{id}.

DELETE {baseURL}/shops/{id}

Διαγραφή του καταστήματος με το συγκεκριμένο id.

Αν ο χρήστης που εκτελεί την αίτηση έχει το ρόλο του Εθελοντή, το σύστημα θα πρέπει να θέσει την τιμή withdrawn=true στο κατάστημα. Αν ο χρήστης έχει το ρόλο του Διαχειριστή, το σύστημα θα προχωρά στη διαγραφή του καταστήματος και όλων των σχετικών του πληροφοριών (π.χ. τιμές των προϊόντων που πωλούνται σε αυτό).

Το αποτέλεσμα θα είναι απλό μήνυμα επιβεβαίωσης, όπως φαίνεται στον ακόλουθο πίνακα.

JSON	XML
{ “message”: “OK” }	<message>OK</message>

Προσθήκη και Αναζήτηση τιμών

Το API θα παρέχει ένα endpoint για την αναζήτηση τιμών των προϊόντων και ένα endpoint για την προσθήκη τιμών προϊόντων.

GET {baseURL}/prices

Αναζήτηση τιμών προϊόντων σε καταστήματα.

Υποστηριζόμενες παράμετροι, κωδικοποιημένες στο query του αιτήματος

start	Integer, default 0
count	Integer, default 20
geo.dist	Integer, απόσταση σε χιλιόμετρα από το σημείο ενδιαφέροντος
geo.lng	Double, το γεωγραφικό μήκος του σημείου ενδιαφέροντος
geo.lat	Double, το γεωγραφικό πλάτος του σημείου ενδιαφέροντος
date.from	Date, με τη μορφή EEEE-MM-HH
date.to	Date, με τη μορφή EEEE-MM-HH
shops	List<String>, με κωδικούς καταστημάτων
products	List<String>, με κωδικούς προϊόντων
tags	List<String>
sort	List<String>, με επιτρεπτές τιμές πολλαπλούς συνδυασμούς των τιμών geo.dist, price, date και ASC, DESC, default price|ASC.

Τα πεδία geo.dist, geo.lng, geo.lat θα πρέπει είτε όλα να έχουν τιμές, είτε κανένα να μην έχει τιμή. Αν δοθούν τιμές σε κάποια από τα πεδία αυτά (αλλά όχι σε όλα), το σύστημα θα πρέπει να απαντήσει με το κατάλληλο μήνυμα λάθους. Αν δοθούν και οι τρεις τιμές, το σύστημα θα πρέπει να υπολογίσει στα αποτελέσματά του μόνο τις τιμές των προϊόντων εκείνων που διατίθενται σε καταστήματα που απέχουν κατά μέγιστο geo.dist χιλιόμετρα από τη θέση (geo.lng, geo.lat). Αν τα πεδία αυτά δεν λάβουν τιμές, το κριτήριο της απόστασης θα πρέπει να αγνοηθεί από το σύστημα για τον υπολογισμό των αποτελεσμάτων.

Το ίδιο ισχύει και για τα πεδία date.from, date.to. Αν κάποιο από τα δύο δεν έχει τιμές, το σύστημα θα πρέπει να απαντήσει με το κατάλληλο μήνυμα λάθους. Για να ζητηθεί μια συγκεκριμένη ημ/νία (και όχι το χρονικό διάστημα από date.from έως date.to), θα πρέπει να δοθεί η ίδια τιμή και στα δύο πεδία. Αν τα πεδία αυτά δεν λάβουν τιμές, το σύστημα θα πρέπει αυτόματα να θεωρήσει ως επιθυμητή ημ/νία βάση την ημέρα του χρόνου εκτέλεσης (today).

Τα πεδία shops, products, tags είναι προαιρετικά και λαμβάνουν μηδέν ή περισσότερες τιμές.

Σε ψευδοκώδικα, το ερώτημα που θα πρέπει να εκτελείται από το σύστημα για τον υπολογισμό των αποτελεσμάτων της αναζήτησης είναι το εξής:

SELECT price, product.id, product.name, product.tags, shop.id, shop.name, shop.tags, shop.address, distanceOf(shop.lng, shop.lat, geo.lng, geo.lat) as dist, price.date

WHERE

dist < geo.dist

AND price.date in [date.from, date.to]

AND shop.id in [shops]

AND product.id in [products]

AND (

product.tags.containsAny(tags) OR shop.tags.containsAny(tags)

)

order by [sort]

Παραδείγματα:

1. Εμφάνιση των τιμών ενός προϊόντος σε δύο διαφορετικά καταστήματα σήμερα από τη φθηνότερη στην ακριβότερη:

   GET {baseURL}/prices?shops=321&shops=322&products=123& sort=price|ASC

2. Εμφάνιση των τιμών ενός προϊόντος στα καταστήματα που απέχουν 5χμλ από το σημείο (X,Y) σήμερα από το πιο κοντινό στο πιο απομακρυσμένο κατάστημα:

   GET {baseURL}/prices?geo.dist=5&geo.lng=X&geo.lat=Y&products=123&sort=dist|ASC

3. Εμφάνιση των τιμών ενός προϊόντος σε όλα τα καταστήματα για συγκεκριμένο χρονικό διάστημα:

   GET {baseURL}/prices?date.from=2018-11-15&date.to=2018-11-30 &products=123&sort=date|ASC

Τα αποτελέσματα επιστρέφονται με την εξής κωδικοποίηση:

Start	Integer
Count	Integer
Total	Long

Prices	List<Price>

Τα πεδία start, count, total αφορούν στη «σελιδοποίηση» των αποτελεσμάτων, ενώ το πεδίο prices θα περιέχει τη λίστα με τις τιμές που περιέχονται στη συγκεκριμένη «σελίδα», όπου κάθε στοιχείο της λίστας αντιστοιχεί σε μια δομή που περιέχει τα εξής πεδία:

price	Double	Η τιμή του προϊόντος σε ευρώ στο συγκεκριμένο κατάστημα τη συγκεκριμένη ημ/νία.
date	Date, με τη μορφή EEEE-MM-HH	Η ημερομηνία.
productName	String	Το όνομα του προϊόντος.
productId	String or Long	Ο προσδιοριστής του προϊόντος.
productTags	List<String>	Τα tags του προϊόντος.
shopId	String	Ο προσδιοριστής του καταστήματος.
shopName	String	Το όνομα του καταστήματος.
shopTags	List<String>	Τα tags του καταστήματος.
shopAddress	String	Η διεύθυνση του καταστήματος.
shopDist	Integer	Η απόσταση του καταστήματος από το σημείο ενδιαφέροντος.

POST {baseURL}/prices

Προσθήκη τιμής προϊόντος.

Η αίτηση θα πρέπει να περιέχει στο body (και όχι στο query) τις εξής τιμές υποχρεωτικά.

price	Double	Η τιμή του προϊόντος σε ευρώ.
date.from	Date, με τη μορφή EEEE-MM-HH	Το χρονικό διάστημα (ημερομηνίες από – έως) που το προϊόν έχει / είχε την τιμή.
date.to	Date, με τη μορφή EEEE-MM-HH	
productId	String or Long	Ο προσδιοριστής του προϊόντος.
shopId	String	Ο προσδιοριστής του καταστήματος.

Το αποτέλεσμα της αίτησης, σε περίπτωση επιτυχούς δημιουργίας νέου προϊόντος, θα είναι η πλήρης κωδικοποίηση των δεδομένων του, όπως περιγράφηκε προηγουμένως.