



Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Εργασία στο Μάθημα «Τεχνολογία Λογισμικού» Ελάχιστες Κοινές Προδιαγραφές για το REST API και το CLI Application

Εαρινό εξάμηνο 2019-20

Δρ. Κώστας Σαΐδης (saiko@di.uoa.gr)

REST API

Το back-end υποσύστημα (back-end web app) του Κέντρου Ελέγχου (ΚΕ) που θα αναπτύξετε στο πλαίσιο της εργασίας θα πρέπει να υποστηρίζει ένα RESTful Application Programming Interface (API) για τη διαχείριση των δεδομένων του ΚΕ.

Για τις ανάγκες εκτέλεσης κοινών αυτοματοποιημένων ελέγχων στο back-end υποσύστημα κάθε ομάδας, το παρόν έγγραφο προδιαγράφει ένα ελάχιστο κοινό υποσύνολο του RESTful API, το οποίο θα πρέπει να υποστηρίζεται με τον ίδιο ακριβώς τρόπο απ' όλες τις ομάδες. Κάθε ομάδα είναι ελεύθερη να επεκτείνει τις ελάχιστες αυτές προδιαγραφές κατά το δοκούν (με την προσθήκη περισσότερων endpoints, με την ανάπτυξη πιο πλούσιας μοντελοποίησης για τα δεδομένα, κ.ά), μεριμνώντας όμως ότι η υλοποίηση που θα προκύψει θα πληρεί τα όσα αναφέρονται στη συνέχεια.

Base URL

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

<https://localhost:9000/control-center/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}

Μορφότυποι δεδομένων

Το REST API θα υποστηρίζει τον μορφότυπο JSON (content-type: application/json) και τον μορφότυπο XML (content-type: text/xml). 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, να θεωρήσετε ότι το json θα είναι η default τιμή. Σε κάθε περίπτωση η κωδικοποίηση χαρακτήρων (character encoding) θα πρέπει να είναι UTF8.

Ελάχιστα κοινά Endpoints

Τα ελάχιστα κοινά endpoints που θα πρέπει να υποστηρίζονται απ' όλες τις ομάδες παρουσιάζονται στον ακόλουθο πίνακα. Η λεπτομερής τους προδιαγραφή είναι διαθέσιμη στο [RestAPISpecification.groovy test](#) του rest-api-client στο [repository του μαθήματος](#).

#	Path	Περιγραφή
1	/health-check	Endpoint που ελέγχει την «υγεία» του back-end σε σχέση με τα «εσωτερικά» του υποσυστήματα / συστατικά. Για παράδειγμα, στο repository του μαθήματος, η αντίστοιχη υλοποίηση επαληθεύει την ορθή επικοινωνία του back-end με τη ΒΔ. Εσείς θα πρέπει να προσαρμόσετε την υλοποίηση ώστε να καλύπτονται όλα τα υποσυστήματα που επιλέξατε. Μέθοδος GET: επιστρέφεται {"status": "ok"} ή <status>ok</status>
2	/reset	Endpoint που αρχικοποιεί το back-end, διαγράφοντας όλα τα υφιστάμενα δεδομένα απ' όλα τα υποσυστήματά του (ΔΕΝ πρέπει να υποστηρίζεται σε πραγματικό παραγωγικό σύστημα, υπάρχει εδώ για τις ανάγκες εκτέλεσης των κοινών αυτοματοποιημένων σεναρίων ελέγχου). Με το πέρας της κλήσης, το back-end θα έχει μόνο τα στοιχεία που απαιτούνται ώστε να μπορεί να συνδεθεί ο default administrator λογαριασμός του συστήματος (username: admin, password: pass123!). Μέθοδος POST: επιστρέφεται {"status": "ok"} ή <status>ok</status>
3	/login	Endpoint που εκτελεί τη σύνδεση ενός χρήστη, επιστρέφοντας ένα νέο user access token. Μέθοδος POST: επιστρέφεται {"token": "XYZ"} ή <token>XYZ</token>
4	/logout	Endpoint που εκτελεί την αποσύνδεση ενός χρήστη, «ακυρώνοντας» το δοθέν user access token. Μέθοδος POST: επιστρέφεται ένα άδειο JSON ή XML document
5	/incidents	Endpoint που επιτρέπει σε διαπιστευμένους χρήστες να: <ul style="list-style-type: none">• Ανακτήσουν σελιδοποιημένα τα συμβάντα του ΚΕ.• Προσθέσουν ένα νέο συμβάν. Μέθοδοι, παράμετροι και αποτελέσματα προδιαγράφονται στο RestAPISpecification.groovy test του rest-api-client.

6	/incidents/{id}	<p>Endpoint που επιτρέπει σε διαπιστευμένους χρήστες να:</p> <ul style="list-style-type: none"> • Ανακτήσουν τις πληροφορίες ενός συμβάντος. • Ενημερώσουν τις πληροφορίες ενός συμβάντος. • Να διαγράψουν ένα συμβάν. <p>Μέθοδοι, παράμετροι και αποτελέσματα προδιαγράφονται στο RestAPISpecification.groovy test του rest-api-client.</p>
7	/admin/users	<p>Endpoint που επιτρέπει σε χρήστες με το ρόλο administrator να:</p> <ul style="list-style-type: none"> • Ανακτήσουν σελιδοποιημένους τους λογαριασμούς των χρηστών του ΚΕ. • Προσθέσουν νέο λογαριασμό. <p>Μέθοδοι, παράμετροι και αποτελέσματα προδιαγράφονται στο RestAPISpecification.groovy test του rest-api-client.</p>
8	/admin/users/{id}	<p>Endpoint που επιτρέπει σε χρήστες με το ρόλο administrator να:</p> <ul style="list-style-type: none"> • Ανακτήσουν τα στοιχεία ενός λογαριασμού. • Ενημερώσουν τα στοιχεία ενός λογαριασμού. • Διαγράψουν έναν λογαριασμό. <p>Μέθοδοι, παράμετροι και αποτελέσματα προδιαγράφονται στο RestAPISpecification.groovy test του rest-api-client.</p>

Διαπίστευση και δικαιοδοσία χρηστών

Κατά την κλήση του API, τα διαπιστευτήρια του χρήστη (user access token), θα πρέπει να παρέχονται σε ειδικό για το σκοπό αυτό custom HTTP Header, το όνομα του οποίου θα είναι X-CONTROL-CENTER-AUTH.

Διαχείριση σφαλμάτων

Κάθε κλήση στο API θα πρέπει να επιστρέφει τα κατάλληλα HTTP Status Codes σε περίπτωση σφάλματος, όπως, ενδεικτικά φαίνονται στον ακόλουθο πίνακα.

401	Not authorized	Σε περίπτωση που ο χρήστης δεν έχει τη δικαιοδοσία να εκτελέσει μια ενέργεια.
404	Not found	Σε περίπτωση που ο ζητούμενος πόρος δεν υπάρχει.
400	Bad request	Σε περίπτωση που τα δεδομένα εισόδου δεν είναι έγκυρα (π.χ. κενό υποχρεωτικό πεδίο).

CLI Application

Το ΚΕ θα αναπτύξετε θα πρέπει να υποστηρίζει ένα Command Line Interface (CLI) application, για τη διαχείριση των λογαριασμών του συστήματος. Το CLI application που θα αναπτύξετε θα πρέπει να επικοινωνεί με τα δύο /admin/users REST endpoints που αναφέρθηκαν προηγουμένως, ώστε να επιτρέπει σε χρήστες με ρόλο administrator να εκτελέσουν όλες τις υποστηριζόμενες λειτουργίες διαχείρισης λογαριασμών.

Το CLI application θα δέχεται εντολές της ακόλουθης μορφής:

control-center COMMAND --option1=value1 --option2=value2 ... --optionN=valueN

Τα υποστηριζόμενα commands και οι σχετικές τους παράμετροι είναι τα ακόλουθα:

COMMAND	Επίπεδο διαπίστευσης χρήστη	Παράμετροι	Αντίστοιχη κλήση REST API
health-check	Κανένα	--format	GET /health-check Το status εκτυπώνεται στην οθόνη
login	Κανένα	--format --username --password	POST /login
list-users	Administrator	--format --start --count	GET /admin/users Η λίστα των χρηστών εκτυπώνεται στην οθόνη
add-user	Administrator	--format --username --password --firstName --lastName --role --agency	POST /admin/users Τα στοιχεία του νέου χρήστη εκτυπώνονται στην οθόνη
get-user	Administrator	--format --id --username --password --firstName --lastName --role --agency	GET /admin/users/{id} Τα στοιχεία του χρήστη εκτυπώνονται στην οθόνη
update-user	Administrator	--format --id	PUT /admin/users/{id} Τα νέα στοιχεία του χρήστη εκτυπώνονται στην οθόνη
delete-user	Administrator	--format --id	DELETE /admin/users/{id} Το status εκτυπώνεται στην οθόνη

Το token που επιστρέφεται από το /login endpoint θα πρέπει να αποθηκεύεται σε κάποιο προσωπικό αρχείο του χρήστη και να επαναχρησιμοποιείται στις εντολές που απαιτούν επίπεδο διαπίστευσης. Αν το αρχείο δε βρεθεί, θα πρέπει να εμφανίζεται αντίστοιχο μήνυμα λάθους καθώς και η προτροπή για την εκ νέου σύνδεση του χρήστη.