

# Εργασία στο Μάθημα Τεχνολογία Λογισμικού 2024-2025

## Προδιαγραφές του REST web API και του Command-Line Interface - CLI

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Εθνικό Μετσόβιο Πολυτεχνείο

Χειμερινό εξάμηνο 2024-2025

Διδάσκων: Β. Βεσκούκης

### REST API: Γενικές αρχές

Το σύστημα λογισμικού διαχείρισης διαλειτουργικότητας διοδίων στους αυτοκινητόδρομους που θα αναπτύξετε θα πρέπει να υποστηρίζει ένα RESTful Application Programming Interface (REST API) για την αποθήκευση και ανάκτηση των δεδομένων που αποθηκεύονται στη Βάση που θα υλοποιήσετε. Οι λειτουργίες (κλήσεις) που περιγράφονται ακολούθως πρέπει να υλοποιηθούν όλες, ανεξάρτητα από το αν θα χρησιμοποιηθούν στις περιπτώσεις χρήσης που θα επιλέξετε.

#### Documentation

Το REST API θα πρέπει να περιλαμβάνει documentation και στις δύο ακόλουθες μορφές: α) σύμφωνα με το πρότυπο OpenAPI 3.0 (yaml ή json) και β) σε μορφή συλλογής (collection) του λογισμικού Postman (json).

#### Base URL

Το REST API θα είναι διαθέσιμο στο ακόλουθο base URL:

<https://{{host}}:9115/api>

όπου {{host}} είναι η διεύθυνση του server που θα εκτελείται το API.

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

`{baseURL}/{service}/{path-to-resource}`

Όπου {service} μία από τις υπηρεσίες που θα διατίθεται, όπως αναφέρονται παρακάτω. Για παράδειγμα, το endpoint για την ανάκτηση των διελεύσεων από το σταθμό NAO01 για το μήνα Νοέμβριο 2024 είναι το

`https://localhost:9115/api/tollStationPasses/NAO01/20241101/20241130`

Όλα τα αποτελέσματα που επιστρέφει το API θα είναι ταξινομημένα ως προς τη σήμανση χρόνου (timestamp) που περιέχουν με αύξουσα τάξη.

#### Μορφότυποι δεδομένων

Το REST API θα επιστρέφει δεδομένα σε κωδικοποίηση UTF8, στους μορφότυπους JSON (content-type: application/json) και CSV (content-type: text/csv). Η επιλογή του μορφότυπου θα καθορίζεται στην αίτηση με query parameter ως εξής:

`{baseURL}/{service}/{path-to-resource}?format={json|csv}`

Αν η παράμετρος `format` δεν παρέχεται σε κάποια κλήση του REST API, να θεωρήσετε ότι το JSON θα είναι η `default` τιμή. Για παράδειγμα, η προηγούμενη κλήση με αίτημα μορφότυπου δεδομένων "csv", έχει ως εξής:

```
https://localhost:9115/api/tollStationPasses/NA001/20241101/20241130?format=csv
```

Ο διαχωριστικός χαρακτήρας (delimiter) στα αρχεία csv θα είναι ο χαρακτήρας `,` (κόμμα).

## Διαχείριση σφαλμάτων

Κάθε κλήση στο API θα πρέπει να επιστρέφει κατάλληλα HTTP status codes σε περίπτωση επιτυχίας ή σφάλματος, όπως φαίνεται στον ακόλουθο πίνακα.

200	Success	Σε περίπτωση επιτυχούς κλήσης
204	No content	Σε περίπτωση που η κλήση είναι επιτυχής, αλλά απάντηση είναι κενή (δεν επέστρεψε δεδομένα)
400	Bad request	Σε περίπτωση που οι παράμετροι που δίνονται σε μία κλήση δεν είναι έγκυρες (π.χ. λείπει κάποιο υποχρεωτικό πεδίο)
401	Not authorized	Σε περίπτωση που η αίτηση γίνεται από μη διαπιστευμένο χρήστη (μόνο εφόσον υλοποιηθεί διαπίστευση χρηστών)
500	Internal server error	Σε περίπτωση οποιουδήποτε άλλου σφάλματος

## Ελεγχόμενη πρόσβαση (προαιρετική)

### Διαπίστευση χρηστών

Για λόγους ελέγχου πρόσβασης των διαφορετικών stakeholders στο σύστημα, για τη χρήση του API μπορείτε προαιρετικά να υλοποιήσετε μηχανισμό διαπίστευσης των χρηστών της επιλογής σας. Στην περίπτωση αυτή, οι λογαριασμοί χρηστών θα δημιουργούνται από το διαχειριστή μέσω του Command Line Interface (CLI), όπως θα περιγραφεί στη συνέχεια.

Κατά την κλήση του API, τα διαπιστευτήρια του χρήστη (πχ user access token), κωδικοποιημένα με τον τρόπο που εσείς κρίνετε πιο συμβατό με τη σχετική βέλτιστη πρακτική, θα πρέπει να παρέχονται σε ειδικό για το σκοπό αυτό custom HTTP header `X-OBSERVATORY-AUTH`.

### Login & logout

Το back-end σας θα υποστηρίζει δύο endpoints για το login και το logout των χρηστών:

1. **/login:** Υποστηρίζει την μέθοδο POST και λαμβάνει τις παραμέτρους `username`, `password` του χρήστη κωδικοποιημένους ως `"application/x-www-form-urlencoded"`. Σε περίπτωση επιτυχούς διαπίστευσης του χρήστη, επιστρέφει ένα json object με το token αυτού: `{πχ "token":"FOO"}`.
2. **/logout:** Υποστηρίζει τη μέθοδο POST και δεν λαμβάνει παραμέτρους (το token του χρήστη που πρέπει κάνει logout περιέχεται στον custom HTTP header `X-OBSERVATORY-AUTH`, όπως αναφέρθηκε παραπάνω). Σε περίπτωση επιτυχίας, επιστρέφει μόνο το status code 200 (empty response body).

## Διαχείριση

Το back-end σας θα υποστηρίζει τα παρακάτω διαχειριστικά endpoints, τα οποία θα υποστηρίζουν τον αυτοματοποιημένο έλεγχο κατά την παρουσίαση της εργασίας:

1. **/admin/healthcheck**: Υποστηρίζει τη μέθοδο GET και επιβεβαιώνει τη συνδεσιμότητα (end-to-end connectivity) μεταξύ του χρήστη και της βάσης δεδομένων. Σε περίπτωση επιτυχούς σύνδεσης επιστρέφεται κωδικός http status code **200** και body που περιέχει το json object:

```
{ "status": "OK",  
  "dbconnection": connection string,  
  "n_stations": n_of_toll_stations,  
  "n_tags" : n_of_tags,  
  "n_passes": n_of_passes }
```

Διαφορετικά επιστρέφεται http status code **401** και body που περιέχει το json object

```
{ "status": "failed",  
  "dbconnection": connection string }
```

Στο connection string περιέχεται ό,τι απαιτείται για τη ΒΔ της επιλογής σας.

2. **/admin/resetstations**: Υποστηρίζει τη μέθοδο POST και προβαίνει σε αρχικοποίηση του πίνακα σταθμών διοδίων με τις τιμές που περιέχονται στο αρχείο με όνομα **tollstations2024.csv**, το οποίο υπάρχει για κατέβασμα στο φάκελο "[Δεδομένα αρχικοποίησης σταθμών διοδίων](#)" στο helios. Σε περίπτωση επιτυχίας, επιστρέφεται το json object: {"status": "OK"} διαφορετικά επιστρέφεται {"status": "failed", "info": reason} με την τιμή του **reason** να αφορά πληροφορία σχετικά με την αποτυχία.
3. **/admin/resetpasses**: Υποστηρίζει τη μέθοδο POST και προβαίνει σε αρχικοποίηση των γεγονότων διέλευσης (διαγραφή όλων των γεγονότων), καθώς και όλων των πινάκων/συλλογών που διατηρούν δεδομένα που εξαρτώνται από τις διελεύσεις (πχ: tags, εφόσον υπάρχουν τέτοιοι πίνακες ή συλλογές).

Μόνο εφόσον υλοποιήσετε λογαριασμούς χρηστών, πρέπει επιπλέον να πραγματοποιεί και αρχικοποίηση του διαχειριστικού λογαριασμού με τιμές **username**: *admin* και **password**: *freepasses4all*. Σε περίπτωση επιτυχίας, επιστρέφεται το json object: {"status": "OK"} διαφορετικά επιστρέφεται {"status": "failed", "info": reason} με την τιμή του **reason** να αφορά πληροφορία σχετικά με την αποτυχία.

4. **/admin/addpasses**: Υποστηρίζει τη μέθοδο POST και προβαίνει σε ενημέρωση των γεγονότων διέλευσης, καθώς και όλων των πινάκων/συλλογών που διατηρούν δεδομένα που εξαρτώνται από τις διελεύσεις (πχ: tags, εφόσον υπάρχουν τέτοιοι πίνακες ή συλλογές), με προσθήκη των διελεύσεων που περιέχονται στο αρχείο **csv** που μεταφέρεται ως content type: *multipart/form-data* και mime type: *'text/csv'*. Ο μορφότυπος του αρχείου (ονόματα πεδίων) είναι **υποχρεωτικά** αυτός που δίνεται στο αρχείο-παράδειγμα με όνομα **passes-sample.csv** το οποίο υπάρχει για κατέβασμα στο φάκελο "[Ενδεικτικό csv αρχείο διελεύσεων](#)" στο helios.

## Λειτουργία

Ακολούθως δίνονται τέσσερα endpoints για την επιτέλεση των βασικών λειτουργιών, τα οποία **πρέπει να όλα υλοποιηθούν ακριβώς όπως δίνονται**, ώστε να μπορεί να γίνει αυτοματοποιημένος έλεγχος κατά την παρουσίαση της εργασίας. Επιπλέον μπορείτε να υλοποιήσετε και όσα δικά σας endpoints επιθυμείτε, όπως απαιτείται από τις περιπτώσεις χρήσης ή από τον τρόπο υλοποίησης που θα επιλέξετε.

Για όλες τις κλήσεις ισχύει ότι οι ημερομηνίες που δίνονται ως path parameters (date\_from, date\_to) πρέπει να έχουν τη μορφή "YYYYMMDD", ενώ οι χρονικές αναφορές (timestamps) που επιστρέφονται πρέπει να είναι της μορφής "YYYY-MM-DD hh:mm"

a. Διελεύσεις ανά σταθμό

/tollStationPasses/:tollStationID/:date\_from/:date\_to

Υποστηρίζει την μέθοδο GET και επιστρέφει αντικείμενο που περιέχει λίστα με την ανάλυση των διελεύσεων για τον σταθμό διοδίων και την περίοδο που δίνονται ως path parameters.

Πεδίο	Τύπος	Περιγραφή
stationID	String	Το μοναδικό ID του σημείου διέλευσης
stationOperator	String	Ο διαχειριστής του σταθμού διέλευσης tollStationID
requestTimestamp	String	Timestamp της κλήσης του endpoint
periodFrom	String	Η αιτούμενη περίοδος (από)
periodTo	String	Η αιτούμενη περίοδος (έως)
nPasses	Integer	Αριθμός γεγονότων διέλευσης στην περίοδο
passList:	List	(Περιέχει nPasses στοιχεία)
passIndex	Integer	A/A (1, 2, 3, ...)
passID	String	ID του γεγονότος διέλευσης
timestamp	String	Timestamp της διέλευσης
tagID	String	Η ταυτότητα του tag διέλευσης
tagProvider	String	Ο πάροχος του tag
passType	String	"home" ή "visitor"
passCharge	Float	Το κόστος διέλευσης

b. Ανάλυση διελεύσεων μεταξύ operators

/passAnalysis/:stationOpID/:tagOpID/:date\_from/:date\_to

Υποστηρίζει την μέθοδο GET και επιστρέφει αντικείμενο που περιέχει λίστα με την ανάλυση των γεγονότων διέλευσης που πραγματοποιήθηκαν με tag του **tagOpID** σε σταθμούς του **stationOpID** για την περίοδο που δίνονται ως path parameters

Πεδίο	Τύπος	Περιγραφή
stationOpID	String	Το ID του λειτουργού του σταθμού
tagOpID	String	Το ID του λειτουργού-παρόχου του tag
requestTimestamp	String	Timestamp της κλήσης του endpoint
periodFrom	String	Η αιτούμενη περίοδος (από)
periodTo	String	Η αιτούμενη περίοδος (έως)
nPasses	Integer	Αριθμός γεγονότων διέλευσης στην περίοδο
passList:	List	(περιέχει nPasses στοιχεία)
passIndex	Integer	A/A (1, 2, 3, ...)
passID	String	Το ID του γεγονότος διέλευσης
stationID	String	Το ID του σταθμού διέλευσης
timestamp	String	Timestamp της διέλευσης
tagID	String	Η ταυτότητα του tag διέλευσης
passCharge	Float	Το κόστος διέλευσης

### c. Κόστος διελύσεων μεταξύ operators

/passesCost/:tollOpID/:tagOpID/:date\_from/:date\_to

Υποστηρίζει την μέθοδο GET και επιστρέφει αντικείμενο με τον αριθμό των γεγονότων διέλευσης που πραγματοποιήθηκαν με tag του **tagOpID** σε σταθμούς του **tollOpID**, καθώς και το κόστος τους, δηλαδή το ποσό που ο **tagOpID** οφείλει στον **tollOpID**, για τη δοσμένη περίοδο.

Πεδίο	Τύπος	Περιγραφή
tollOpID	String	Το ID του λειτουργού του σταθμού διέλευσης
tagOpID	String	Το ID του λειτουργού που χορήγησε το tag διέλευσης
requestTimestamp	String	Timestamp της κλήσης του endpoint
periodFrom	String	Η αιτούμενη περίοδος (από)
periodTo	String	Η αιτούμενη περίοδος (έως)
nPasses	Integer	Ο αριθμός γεγονότων διέλευσης στην περίοδο
passesCost	Float	Το συνολικό κόστος των διελύσεων

### 2d. Διελύσεις και χρεώσεις λοιπών λειτουργών

/chargesBy/:tollOpID/:date\_from/:date\_to

Υποστηρίζει την μέθοδο GET και επιστρέφει αντικείμενο που περιέχει λίστα με τον αριθμό των γεγονότων διέλευσης που πραγματοποιήθηκαν σε σταθμούς του **tollOpID**, από οχήματα όλων των υπολοίπων operators καθώς και το κόστος τους, δηλαδή το ποσό που καθένας από τους υπόλοιπους operators οφείλει στον **tollOpID** για τη δοσμένη περίοδο.

Πεδίο	Τύπος	Περιγραφή
tollOpID	String	Το ID του operator
requestTimestamp	String	Timestamp της κλήσης του endpoint
periodFrom	String	Η αιτούμενη περίοδος (από)
periodTo	String	Η αιτούμενη περίοδος (έως)
vOpList:	List	Περιλαμβάνει τόσα στοιχεία όσοι και οι toll operators των οποίων οχήματα πέρασαν από όλους τους σταθμούς διοδίων του <b>tollOpID</b> τη δοσμένη περίοδο (visiting operators)
visitingOpID	String	Το ID του visiting operator
nPasses	Integer	Ο αριθμός γεγονότων διέλευσης από οχήματα με tag του <b>visitingOpID</b> στην περίοδο
passesCost	Float	Το συνολικό κόστος των διελύσεων το οποίο οφείλεται από τον <b>visitingOpID</b> στον <b>tollOpID</b> για την περίοδο

## Command Line Interface - CLI

Οι προδιαγραφές του CLI περιλαμβάνουν κλήσεις οι οποίες είναι αντίστοιχες με εκείνες του REST API και μπορούν να εκτελεστούν μέσω shell, διαθέσιμου μόνο από την κονσόλα (command line, ssh) του συστήματος που φιλοξενεί την εφαρμογή. Λογαριασμοί χρηστών, εφόσον υλοποιηθούν, δημιουργούνται μόνο μέσω του CLI.

### Λειτουργία

Το CLI θα υποστηρίζει τον μορφότυπο JSON και τον μορφότυπο CSV, όπως ακριβώς και στο REST API. Η επιλογή του μορφότυπου θα καθορίζεται με παράμετρο όπως αναφέρεται πιο κάτω. Όλα τα αποτελέσματα που επιστρέφει το CLI θα είναι ταξινομημένα ως προς το χρόνο στον οποίο αναφέρονται (timestamp) με αύξουσα τάξη.

Το CLI θα καλείται από τη γραμμή εντολών με κλήσεις της μορφής:

```
$ se24XX scope --param1 value1 [--param2 value2 ...] --format fff
```

Όπου **XX** το αναγνωριστικό της ομάδας σας και **scope** όπως αναφέρεται στον πίνακα που ακολουθεί. Σε περίπτωση που δεν δίνονται παράμετροι, θα πρέπει να εμφανίζονται οι υποστηριζόμενες από το CLI παράμετροι για όλα τα scopes. Σε περίπτωση που δεν δίνεται η παράμετρος --format, η default τιμή για το CLI είναι **csv**. Για παράδειγμα, η κλήση για την ανάκτηση των γεγονότων διέλευσης από τον σταθμό NA001 για το μήνα Νοέμβριο σε μορφή csv είναι η εξής:

```
$ se24XX tollstationpasses --station NA001 --from 20241101 --to 20241130
```

Θα πρέπει να υποστηρίζονται οι ακόλουθες παράμετροι ανά scope.

Scope	Επίπεδο χρήστη	Υποχρεωτικές λοιπές παράμετροι	Αντίστοιχη κλήση REST API
healthcheck			/admin/healthcheck
resetpasses			/admin/resetpasses
resetstations			/admin/resetstations
login (*)		--username --passw	/login
logout (*)	Διαπιστευμένος Χρήστης (*)	Καμία	/logout
tollstationpasses	Διαπιστευμένος χρήστης, εφόσον υλοποιήσετε μηχανισμό διαπίστευσης.  Οποιοσδήποτε, σε περίπτωση μη υλοποίησης μηχανισμού διαπίστευσης	--station --from --to	/tollStationPasses
passanalysis		--stationop --tagop --from --to	/passAnalysis
passescost		--stationop --tagop --from --to	/passesCost
chargesby		--opid --from --to	/chargesBy
admin	Διαχειριστής	Βλ. παρακάτω	

## Διαχείριση

Για τις διαχειριστικές λειτουργίες (scope: admin) θα πρέπει να υποστηρίζονται οι ακόλουθες παράμετροι.

Παράμετρος	Επιτρεπτές τιμές	Υποχρεωτικές λοιπές παράμετροι	Λειτουργία	Αποτέλεσμα
--usermod (*)		--username --passw	Δημιουργία νέου χρήστη ή αλλαγή password	"success" ή "error"
--users (*)			Εμφάνιση λίστας χρηστών	Λίστα με τα usernames
--addpasses		--source	Πρόσθεση νέων διελεύσεων από αρχείο CSV	Εισαγωγή στη ΒΔ των διελεύσεων που περιέχονται στο αρχείο csv. Αντίστοιχη κλήση API /admin/addpasses
--source	όνομα αρχείου με διελεύσεις (csv)			

(\*) μόνο εφόσον υλοποιήσετε μηχανισμό διαπίστευσης χρηστών

## Αντιστοίχιση κλήσεων API με CLI

ΔΙΑΧΕΙΡΙΣΗ	API	CLI
Αρχικοποίηση και έλεγχος συστήματος	<code>/admin/resetpasses</code> <code>/admin/resetstations</code> <code>/admin/healthcheck</code>	<code>se24xx resetpasses</code> <code>se24xx resetstations</code> <code>se24xx healthcheck</code>
Είσοδος - έξοδος από το σύστημα (προαιρετικά)	<code>/login</code> <code>/logout</code> σε συνδυασμό με κατάλληλα header	<code>se24xx login --username [user] --passw [password]</code> <code>se24xx logout</code>
Διαχείριση χρηστών (προαιρετικά)	-	<code>se24xx admin --usermod --username [user] --passw [newpassw]</code> <code>se24xx admin --users</code>
Εισαγωγή διελεύσεων από αρχείο	<code>/admin/addpasses</code>	<code>se24xx admin --addpasses --source ./newpassesXXXXXX.csv</code>
ΛΕΙΤΟΥΡΓΙΑ	API	CLI
Διελύσεις ανά σταθμό	<code>/tollStationPasses/:tollStationID/:date_from/:date_to</code>	<code>se24xx tollstationpasses --station [stationID] --from [datefrom] --to [dateto] --format json</code>
Ανάλυση διελεύσεων μεταξύ 2 λειτουργιών	<code>/passAnalysis/:stationOpID/:tagOpID/:date_from/:date_to</code>	<code>se24xx passanalysis --stationop [op1] --tagop [op2] --from [datefrom] --to [dateto] --format json</code>
Κόστος διελεύσεων μεταξύ 2 λειτουργιών	<code>/passesCost/:tollOpID/:tagOpID/:date_from/:date_to?format=csv</code>	<code>se24xx passescost --stationop [op1] --tagop [op2] --from [datefrom] --to [dateto]</code>
Διελύσεις από οχήματα άλλων λειτουργιών	<code>/chargesBy/:tollOpID/:date_from/:date_to?format=csv</code>	<code>se24xx chargesby --opid [op] --from [datefrom] --to [dateto]</code>