Επίδειξη Testing

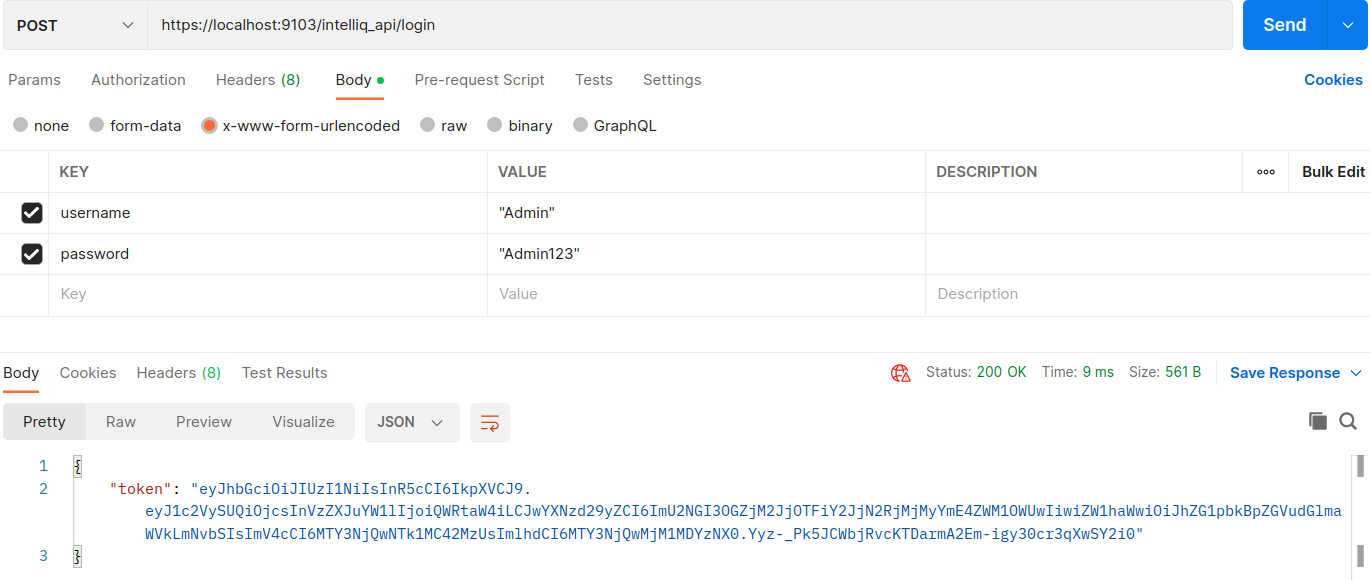
Τεχνολογία Λογισμικού

Ομάδα 24

Επειδή έχουμε υλοποιήσει διαπίστευση χρηστών, όλες οι κλήσεις πρέπει να γίνουν από κάποιον χρήστη που έχει κάνει login. Ακόμα και ο ανώνυμος χρήστης πρέπει να κάνει login με username: “AnonymousUser”, password : “”.

Για να ελέγξουμε την ορθότητα όλων των endpoints πρέπει να κάνουμε login ως admin με POST request στο https://localhost:9103/intelliq\_api/login

με username:”Admin”, password:”Admin123” όπως φαίνεται παρακάτω:

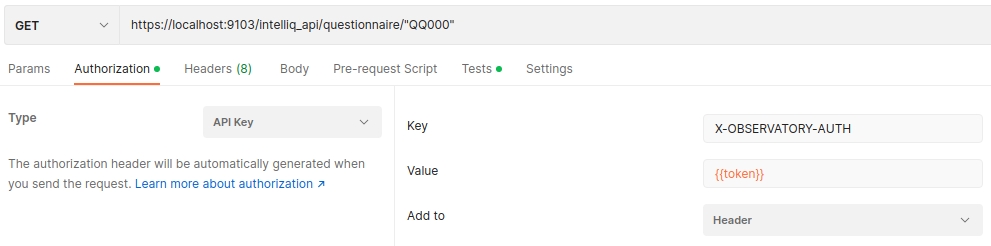


Άμα δώσουμε σωστό username και password θα μας επιστραφεί ένα token. Αυτό το token πρέπει να μπει στα headers κάθε request που κάνουμε. Για να μην το κάνουμε αυτό με το χέρι κάθε φορά, στο collection που έχουμε ανεβάσει στο github repo μας, έχουμε ορίσει στο login (το οποίο είναι το πρώτο request που τρέχουμε) να ορίζει μια collection variable:

var jsonData **=** pm.response.json();

pm.collectionVariables.**set**("token", jsonData.token);

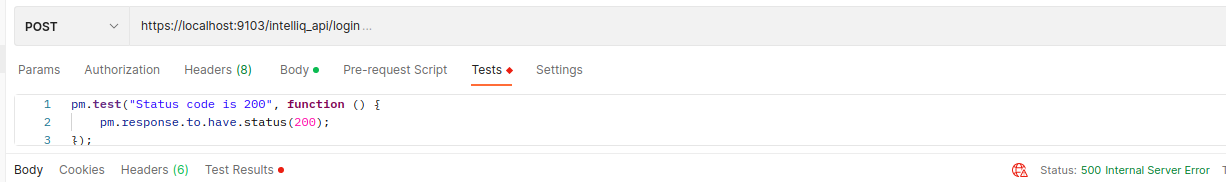
Συνεπώς στα άλλα requests το μόνο που έχουμε να κάνουμε είναι να προσθέσουμε header X-OBSERVATORY-AUTH με τιμή {{token}} όπως φαίνεται στην παρακάτω εικόνα:



τυχαίο request πρόσθεσε το token στα headers

Και αυτό έχει γίνει στο collection που έχουμε ανεβάσει.

Από την άλλη, άμα πάμε στο αρχείο connect.js του api-backend το οποίο είναι υπεύθυνο για τη σύνδεση της βάσης μας και αλλάξω το όνομα της βάσης από intelliqDB (έτσι ονομάσαμε τη βάση μας) σε gibberish θα πρέπει το η κλήση στο login να επιστρέψει internal server error όπως ακριβώς έχουμε ορίσει στο αρχείο της κλήσης login:



Συνεπώς και αυτό δουλεύει κατά τον επιθυμητό τρόπο.

Ας ξεκινήσουμε από τα endpoints των λειτουργιών αλληλεπίδρασης:

* **{baseURL}/questionnaire/:questionnaireID**

Στα mockdata που έχουμε δημιουργήσει υπάρχουν δύο ερωτηματολόγια με IDs QQ000 και QQ001 αντίστοιχα.

Ας τεστάρουμε το QQ000 το οποίο είναι ένα ερωτηματολόγιο με κάποιες ελληνικές ομάδες ποδοσφαίρου. Πιο συγκεκριμένα, η πρώτη ερώτηση ρωτάει τον χρήστη ποια είναι η αγαπημένη του ομάδα από τις Ολυμπιακός, Παναθηναϊκός και ΑΕΚ και η επόμενη ερώτηση τον ρωτάει ποιος είναι ο αγαπημένος του παίχτης δίνοντας του επιλογές από την ομάδα που διάλεξε στην πρώτη ερώτηση.

Κάνοντας GET request στο https://localhost:9103/intelliq\_api/questionnaire/"QQ000"

λαμβάνουμε το παρακάτω json:

{

"questionnaireID": "QQ000",

"questionnaireTitle": "Greek Teams",

"keywords": [

"Football",

"Olympiacos",

"Panathinaikos",

"AEK"

],

"questions": [

{

"qID": "Q00",

"qtext": "Which is your favourite Greek Football team?",

"required": "TRUE",

"type": "question"

},

{

"qID": "Q01",

"qtext": "Who is your favourite Olympiacos Player",

"required": "FALSE",

"type": "question"

},

{

"qID": "Q02",

"qtext": "Who is your favourite Panathinaikos Player",

"required": "FALSE",

"type": "question"

},

{

"qID": "Q03",

"qtext": "Who is your favourite AEK Player",

"required": "FALSE",

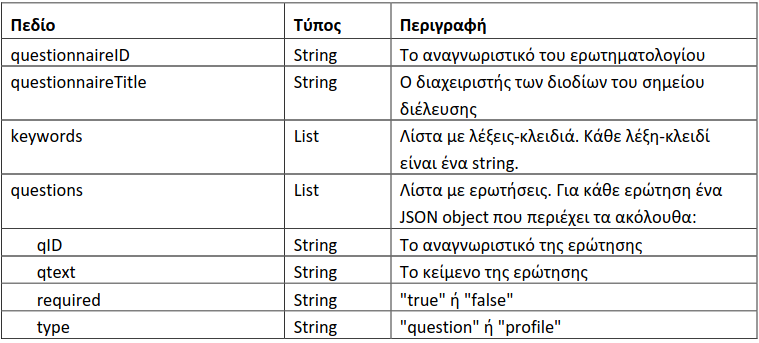
"type": "question"

}

]

}

Σύμφωνα με την εκφώνηση πρέπει το json που επιστρέφουμε να έχει τα παρακάτω πεδία:



Βλέπουμε ότι πράγματι λαμβάνουμε τα επιθυμητά πεδία. Μπορούμε αυτό να το ελέγξουμε και μέσα από τα test cases που έχουμε γράψει μέσω των βοηθητικών συναρτήσεων του postman.

Για παράδειγμα:

Ελέγχουμε άμα η κλήση μας παίρνει απάντηση 200 OK:

pm.**test**("Status code is 200", **function** () {

pm.response.to.have.status(200);

});

Ελέγχουμε άμα περιέχεται το πεδίο questionnaireID στην απάντηση:

pm.**test**("questionnaireID string", **function** () {

pm.expect(pm.response.text()).to.include("questionnaireID");

});

Ελέγχουμε άμα η τιμή του πεδίου questionnaireID είναι η επιθυμητή:

pm.**test**("Correct questionnaireID", **function** () {

var jsonData **=** pm.response.json();

pm.expect(jsonData.questionnaireID).to.eql("QQ000");

});

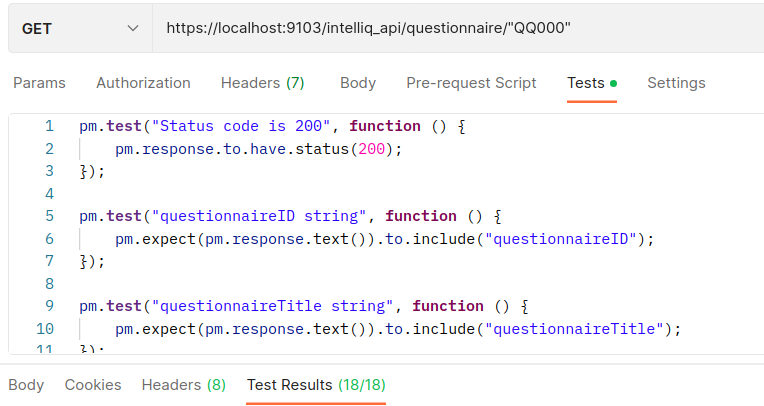
Ή μπορούμε να ελέγξουμε ολόκληρη την απάντηση γράφοντας με το χέρι το επιθυμητό επιστρεφόμενο json:

pm.**test**("Everything is correct", **function** () {

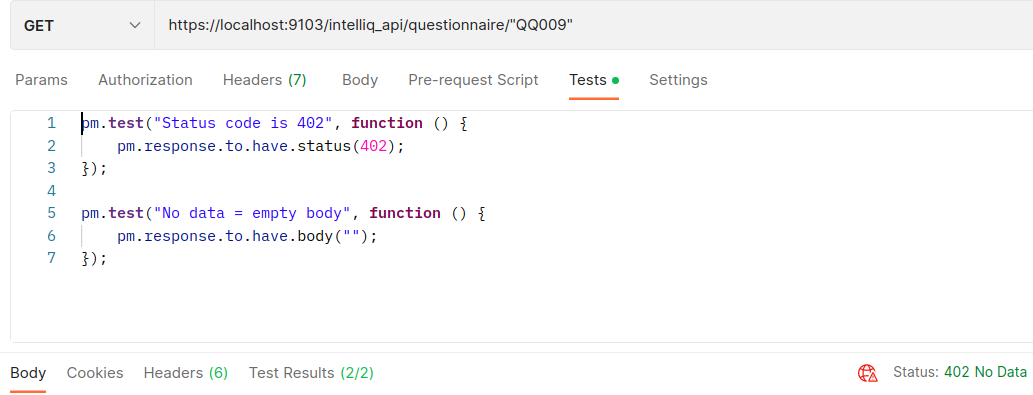
pm.response.to.have.body('{"questionnaireID":"QQ000","questionnaireTi tle":"Greek Teams","keywords":["Football","Olympiacos","Panathinaikos","AEK"],"q uestions":[{"qID":"Q00","qtext":"Which is your favourite Greek Football team?","required":"TRUE","type":"question"},{"qID":"Q01","qtext":"Wh o is your favourite Olympiacos Player","required":"FALSE","type":"question"},{"qID":"Q02","qtext":"W ho is your favourite Panathinaikos Player","required":"FALSE","type":"question"},{"qID":"Q03","qtext":"W ho is your favourite AEK Player","required":"FALSE","type":"question"}]}');

});

Γράφοντας διάφορα test cases, βλέπουμε ότι τα περνάει όλα.

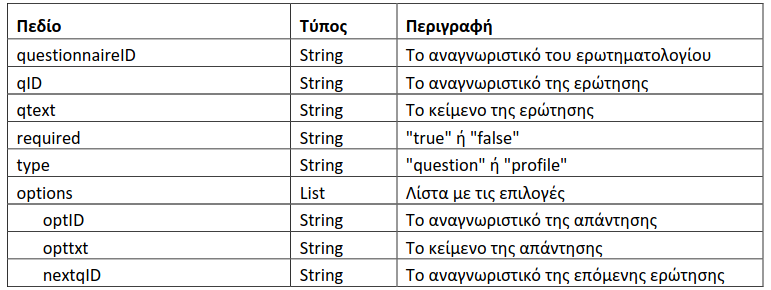


Κάνοντας μία άλλη κλήση για κάποιο questionnaireID που δεν υπάρχει επιστρέφει απάντηση “402 No Data” και περνάει τα αντίστοιχα test cases:



* **{baseURL}/question/:questionnaireID/:questionID**

Σύμφωνα με την εκφώνηση θέλουμε απάντηση που να περιέχει αυτά τα πεδία:



Κάνοντας GET request στο https://localhost:9103/intelliq\_api/question/"QQ000"/"Q01"

(η ερώτηση με ID Q01 ανήκει στο ερωτηματολόγιο με ID QQ000 όπως φαίνεται και από το προηγούμενο endpoint) παίρνουμε το παρακάτω αποτέλεσμα:

{

"questionnaireID": "QQ000",

"qID": "Q01",

"qtext": "Who is your favourite Olympiacos Player",

"required": "FALSE",

"type": "question",

"options": [

{

"optID": "Q01A1",

"opttxt": "Bouchalakis",

"nextqID": **null**

},

{

"optID": "Q01A2",

"opttxt": "El Arabi",

"nextqID": **null**

},

{

"optID": "Q01A3",

"opttxt": "Valbuena",

"nextqID": **null**

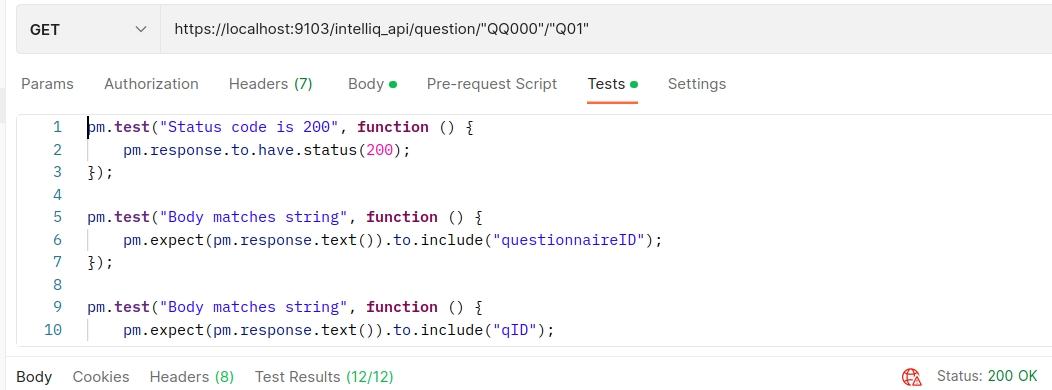
}

]

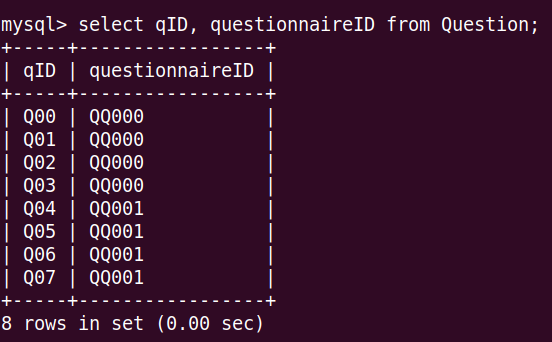
}

Βλέπουμε ότι τα πεδία του json είναι τα επιθυμητά (αν και πλέον ο Μπουχαλάκης δεν αγωνίζεται στον Ολυμπιακό).

Τρέχουμε διάφορα test cases και τα περνάει όλα:



Άμα κάνουμε request στο https://localhost:9103/intelliq\_api/question/"QQ000"/"Q04" όπου η ερώτηση με αναγνωριστικό Q04 δεν ανήκει στο ερωτηματολόγιο με αναγνωριστικό QQ000 αλλά σε αυτό με QQ001 όπως φαίνεται παρακάτω:



Λαμβάνουμε απάντηση “402 No Data”:

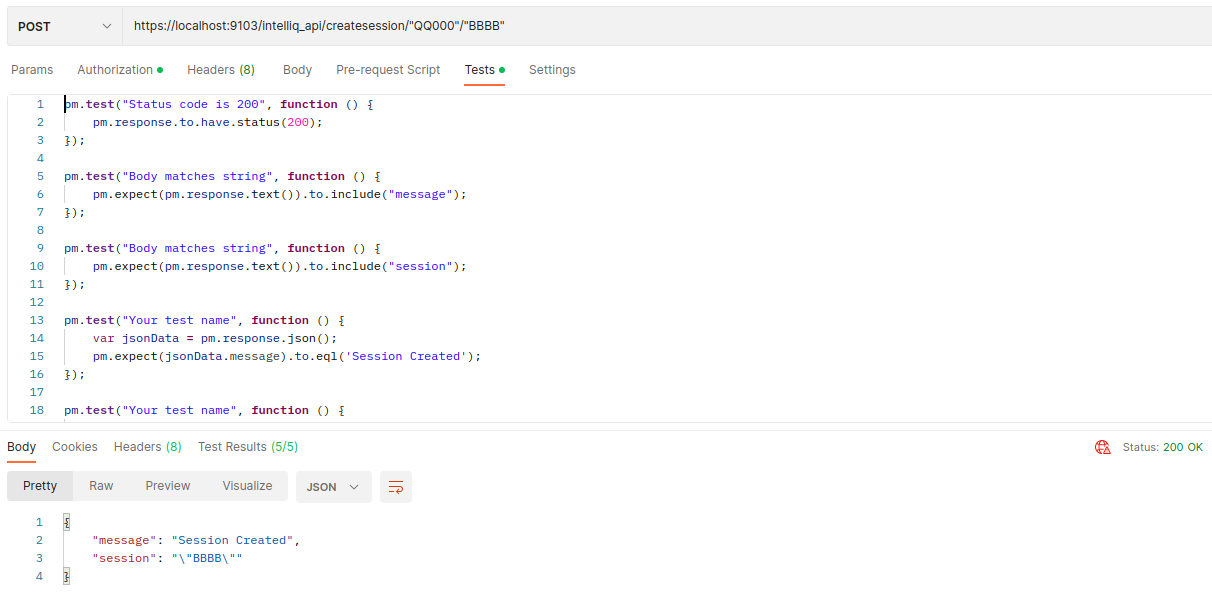


* **{baseURL}/doanswer/:questionnaireID/:questionID/:session/:optionID**

Για να κάνουμε αυτήν την κλήση πρώτα πρέπει να χρησιμοποιήσουμε το βοηθητικό endpoint https://localhost:9103/intelliq\_api/createsession/:questionnaireID/:session το οποίο δημιουργεί ένα νέο κενό Session με αναγνωριστικό session για το ερωτηματολόγιο με αναγνωριστικό questionnaireID (αυτό το endpoint δημιουργήθηκε καθώς όταν ένας χρήστης απαντάει ένα ερωτηματολόγιο στο frontend, δεν είναι λογικό να επιλέγει δικό του session οπότε το δημιουργούμε αυτόματα).

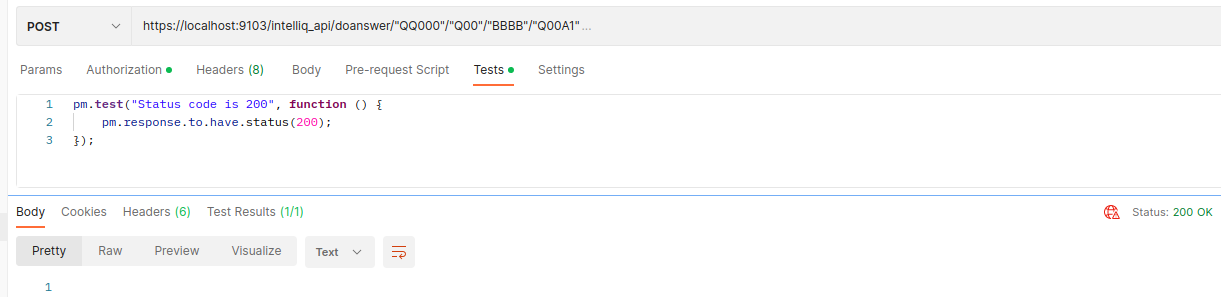
Κάνουμε λοιπόν κλήση στο create session δίνοντάς του questionnaireID και session:

https://localhost:9103/intelliq\_api/createsession/"QQ000"/"BBBB":



Πλέον δημιουργήσαμε το session BBBB για το ερωτηματολόγιο QQ000.

Κάνουμε do answer απαντώντας κάποια ερώτηση του QQ000:

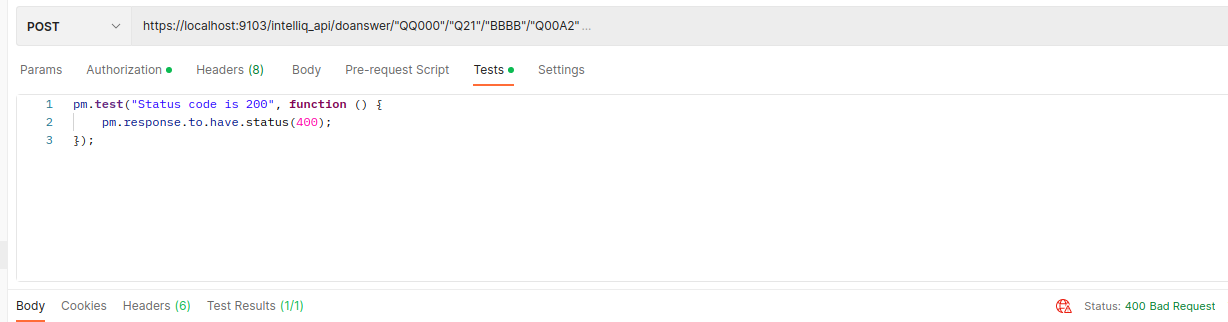


Και βλέπουμε ότι περνάει τα tests μας.

Κάνοντας κλήση getsessionanswers με session = BBBB πρέπει να δούμε την απάντηση που μόλις δώσαμε:



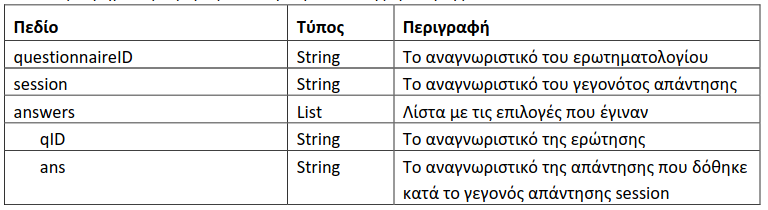
Από την άλλη, άμα κάνουμε doanswer βάζοντας για questionID μια ερώτηση που δεν ανήκει στο QQ000 θα πρέπει να πάρουμε απάντηση Bad Request.



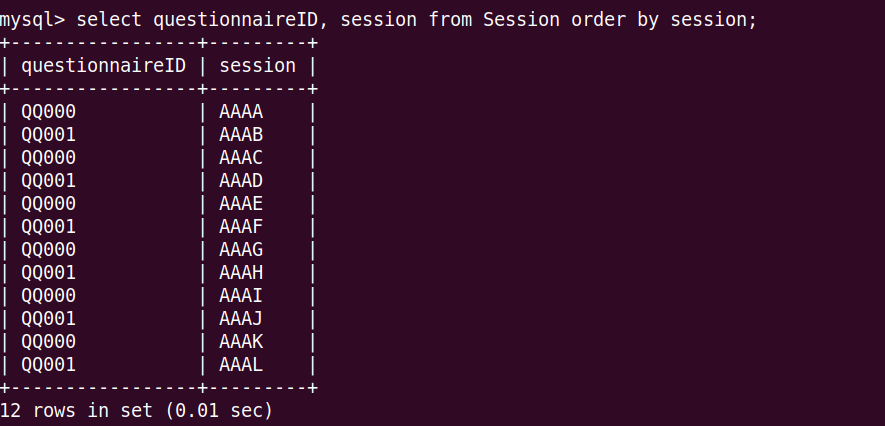
Βλέπουμε ότι είναι σωστό.

* **{baseURL}/getsessionanswers/:questionnaireID/:session**

Προχωρώντας στο getsessionanswers, θέλουμε να λάβουμε ως απάντηση τις απαντήσεις που δόθηκαν κατά το γεγονός απάντηση session του ερωτηματολογίου με αναγνωριστικό questionnaireID.



Στα mock data μας έχουμε υλοποιήσει τα παρακάτω Sessions:



Κάνοντας GET request στο https://localhost:9103/intelliq\_api/getsessionanswers/"QQ000"/"AAAA"

λαμβάνουμε το παρακάτω json:

{

"questionnaireID": "QQ000",

"session": "AAAA",

"answers": [

{

"qID": "Q00",

"ans": "Q00A1"

},

{

"qID": "Q01",

"ans": "Q01A1"

}

]

}

Το αντικείμενο αυτό είναι σωστό και περνάει όλα τα test cases που του αναθέσαμε:



Αντιθέτως, άμα κάνουμε κλήση στο https://localhost:9103/intelliq\_api/getsessionanswers/"QQ000"/"AAAB"

, όπου το session AAAB δεν έγινε για το QQ000 αλλά για το QQ001 όπως φαίνεται από το αποτέλεσμα του παραπάνω query,

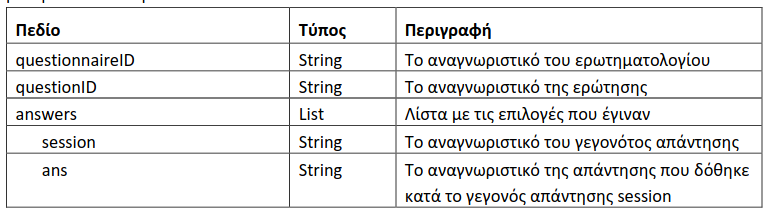
παίρνουμε απάντηση “402 No Data” και επιτυχημένα test cases:



Συνεπώς και αυτό το endpoint είναι σωστό.

* **{baseURL}/getquestionanswers/:questionnaireID/:questionID**

Θέλουμε το αποτέλεσμα μιας GET κλήσης στο παραπάνω URL να μας επιστρέφει object που περιέχει τις απαντήσεις που δόθηκαν σε όλα τα γεγονότα απάντησης, στην ερώτηση questionID.



Κάνοντας GET request στο https://localhost:9103/intelliq\_api/getquestionanswers/"QQ001"/"Q05"

παίρνουμε ως απάντηση:

{

"questionnaireID": "QQ001",

"questionID": "Q05",

"answers": [

{

"session": "AAAB",

"ans": "Q05A1"

},

{

"session": "AAAH",

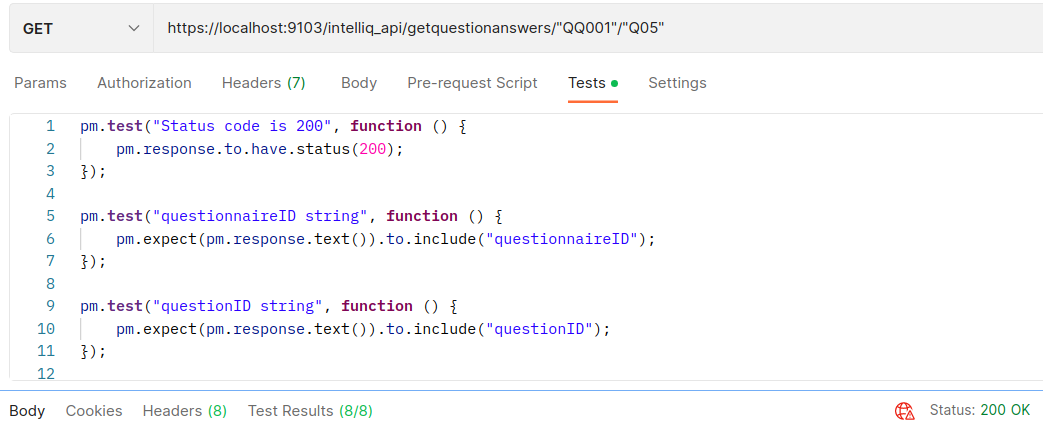
"ans": "Q05A2"

}

]

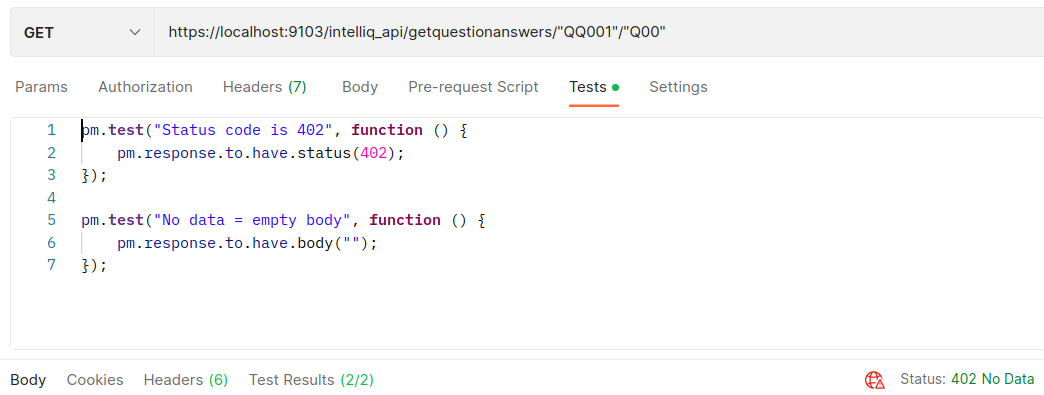
}

Βλέπουμε ότι πράγματι τα επιθυμητά πεδία βρίσκονται στο json της απάντησης και τρέχοντας διάφορα test cases έχουν και τις σωστές τιμές:



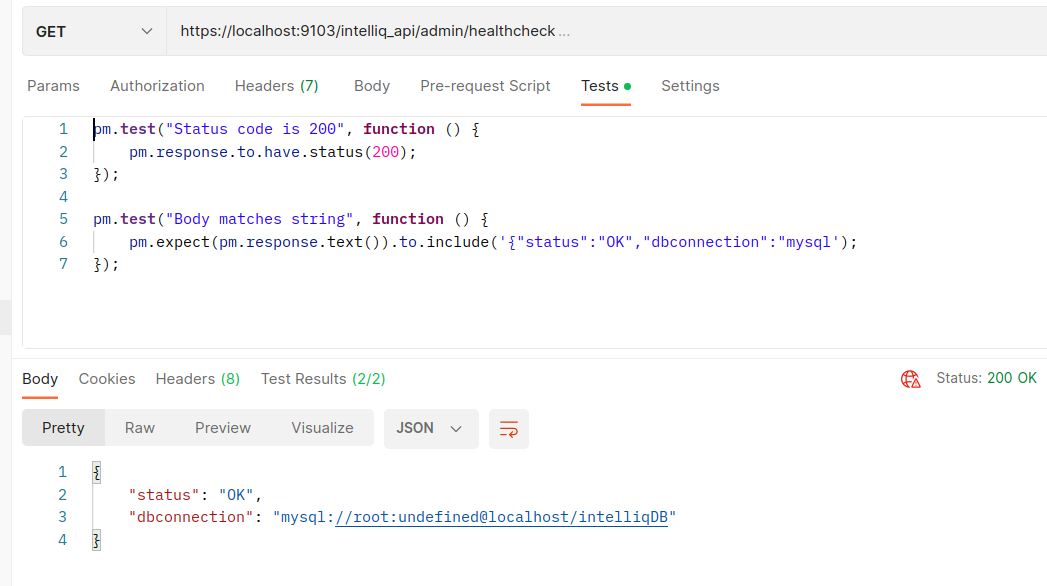
Άμα κάνουμε request στο https://localhost:9103/intelliq\_api/getquestionanswers/"QQ001"/"Q00"

, όπου η ερώτηση με ID Q00 δεν ανήκει στο ερωτηματολόγιο με ID QQ001 αλλά σε αυτό με ID QQ000 όπως δείξαμε στο endpoint {baseURL}/question/ , τότε παίρνουμε ως απάντηση “402 No Data” που περνάει τα test cases που ορίσαμε.



**Διαχειριστικά endpoints**

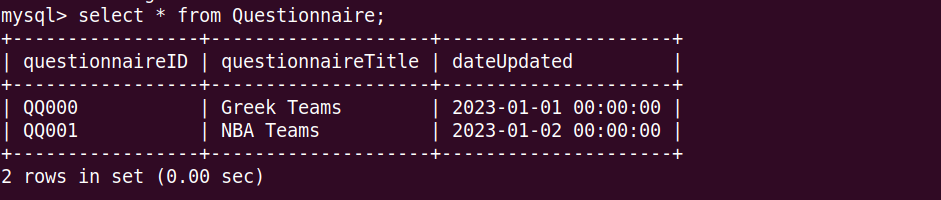
* **{baseURL}/admin/healthcheck**



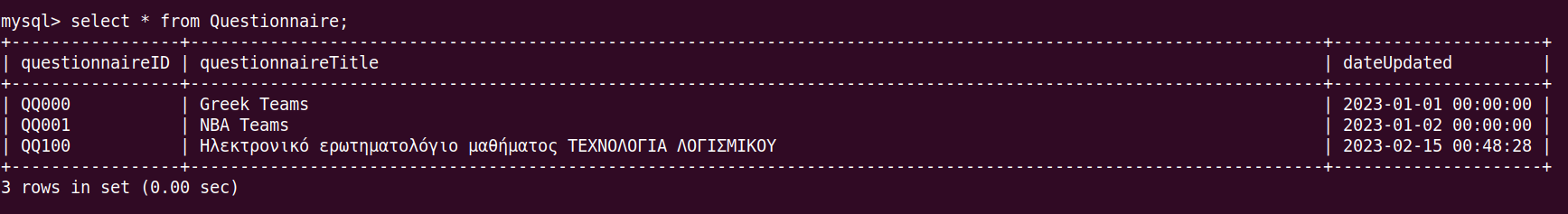
* **{baseURL}/admin/questionnaire\_upd**

Πηγαίνοντας στο Body του POST request επιλέγουμε form-data και μετά το αρχείο μέσα στο οποίο έχουμε το json του ερωτηματολογίου που θέλουμε να ανεβάσουμε. Στο github repo έχουμε ανεβάσει ένα .json αρχείο με όνομα “Questionnaire\_gia\_exetash.json” το οποίο τηρεί τις προδιαγραφές . Για να δοκιμάσουμε την questionnaire\_upd με αυτό το json αρχείο πρέπει να ενεργοποιήσουμε από τα Settings του Postman την επιλογή allow reading files outside working directory εκτός άμα βάλουμε το αρχείο στο working directory που έχουμε επιλέξει.

Τα ερωτηματολόγια στη βάση μας πριν από το POST request είναι:

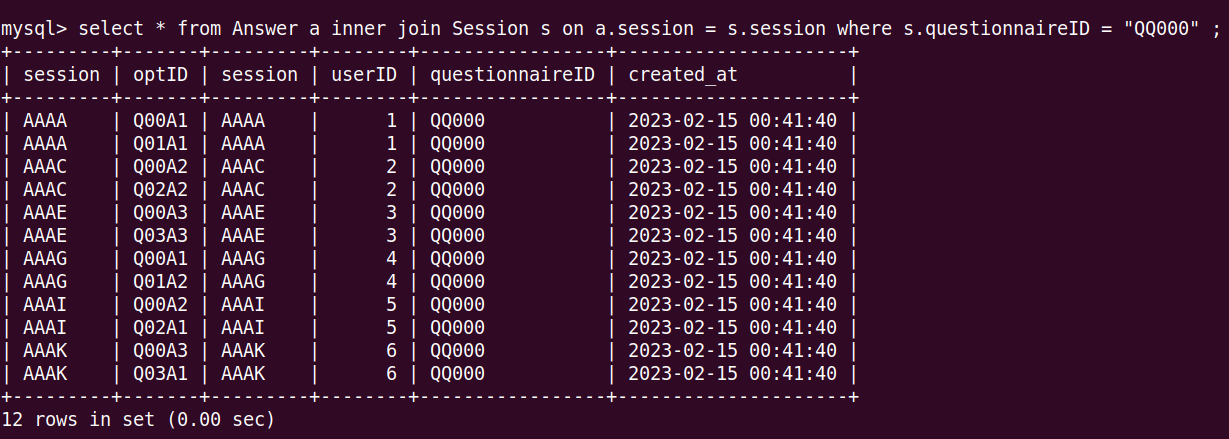


Μετά από το POST request:



* **{baseURL}/admin/resetq/:questionnaireID**

Οι απαντήσεις που έχουν δωθεί για το ερωτηματολόγιο με questionnaireID = “QQ000” πριν από το request είναι οι παρακάτω:



Μετά κάνουμε POST request στο https://localhost:9103/intelliq\_api/admin/resetq/"QQ000":



Πλέον τρέχοντας το ίδιο query με πριν βλέπουμε ότι έχουν σβηστεί οι απαντήσεις του ερωτηματολογίου.

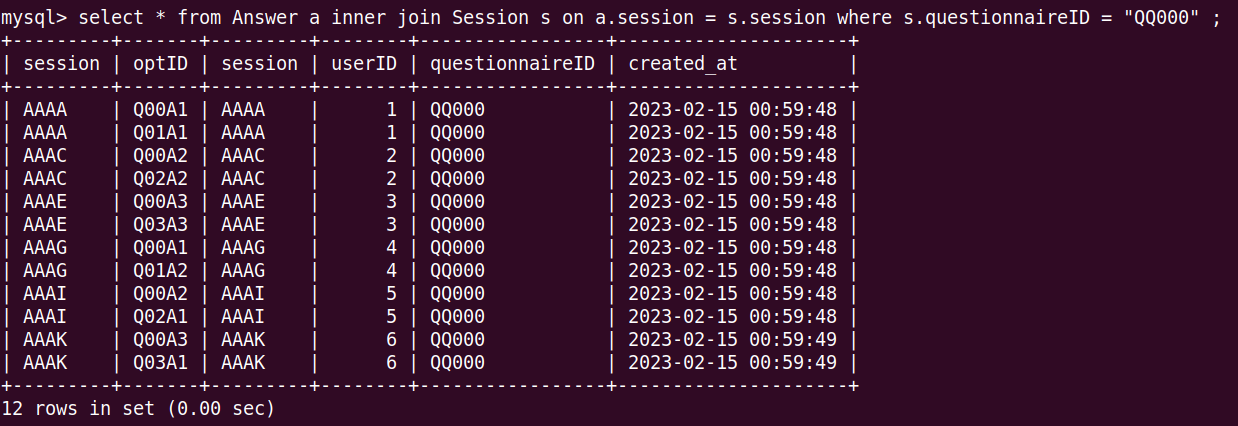
Screenshot from 2023-02-15 00-58-20

* **{baseURL}/admin/resetall**

Κάνουμε κλήση POST στο https://localhost:9103/intelliq\_api/admin/resetall.

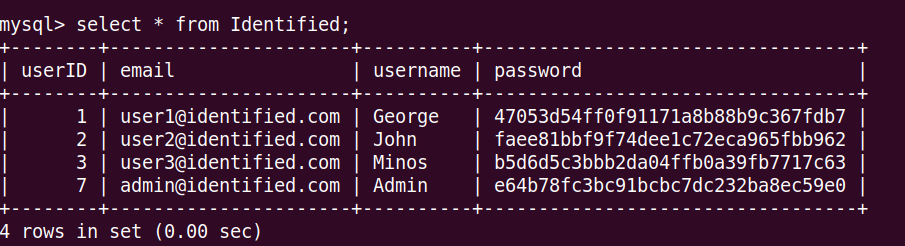


Άμα τρέξουμε το ίδιο query που είχαμε τρέξει πιο πάνω βλέπουμε ότι το ερωτηματολόγιο QQ000 ξαναγεμίζει με τις απαντήσεις που είχαμε βάλει στα mock data μας:



* **{baseURL}/admin/usermod/:username/:password**

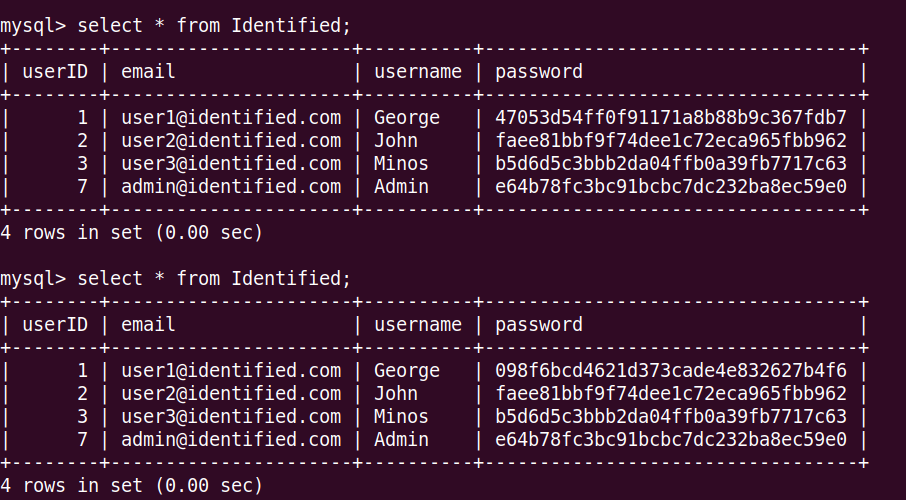
Πριν την κλήση, οι users που υπάρχουν στη βάση είναι (το password των χρηστών αποθηκεύεται hash-αρισμένο στη βάση με MD5):



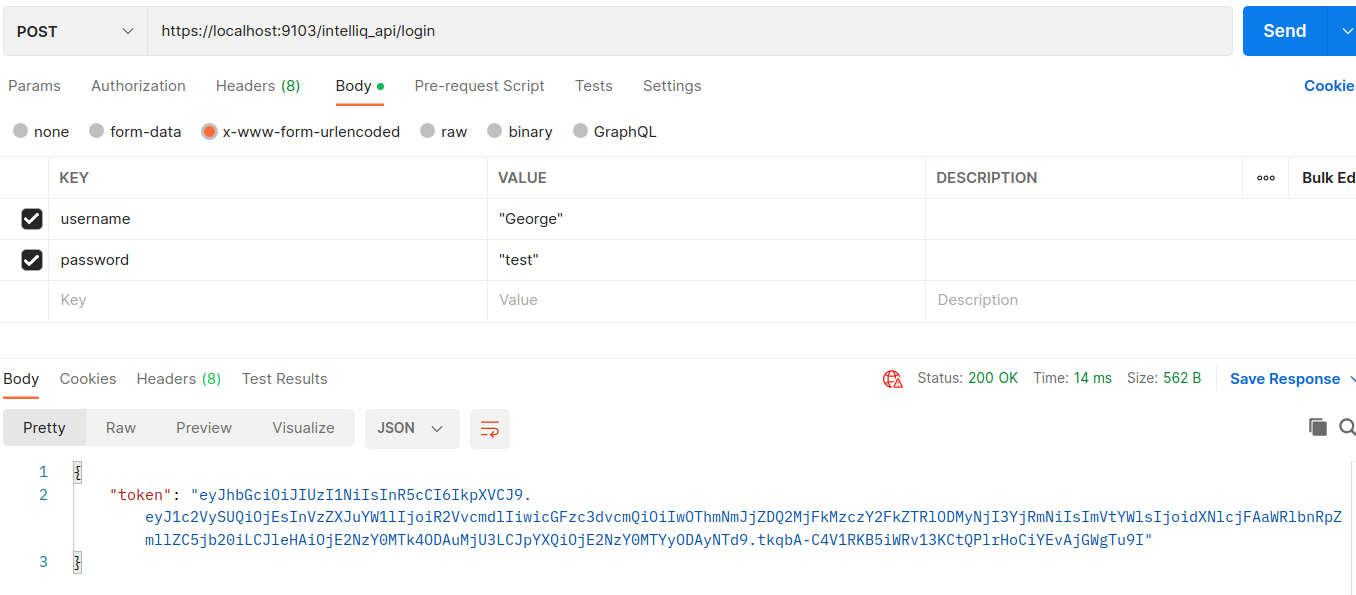
Θα χρησιμοποιήσουμε POST request στο https://localhost:9103/intelliq\_api/admin/usermod/"George"/"test"/"user1@identified.com"

για να αλλάξουμε τον κωδικό του χρήστη με username George σε test.

Το πάνω query είναι πριν το POST request ενώ το κάτω είναι μετά. Παρατηρούμε ότι πλέον το hash του password του χρήστη George έχει αλλάξει.

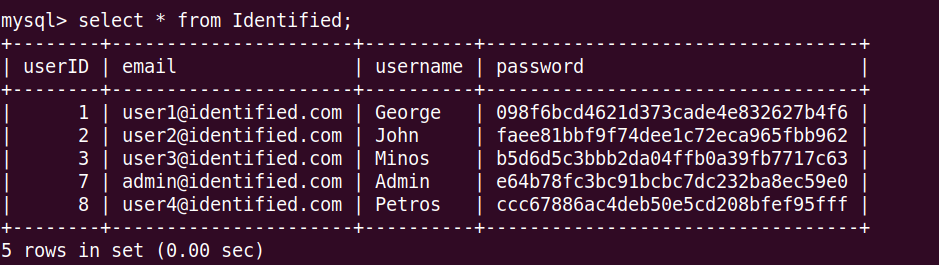


Πλέον μπορούμε να κάνουμε log in με username George και password test:



Τώρα θα κάνουμε POST request στο https://localhost:9103/intelliq\_api/admin/usermod/"Petros"/"Rapto"/"user4@identified.com"

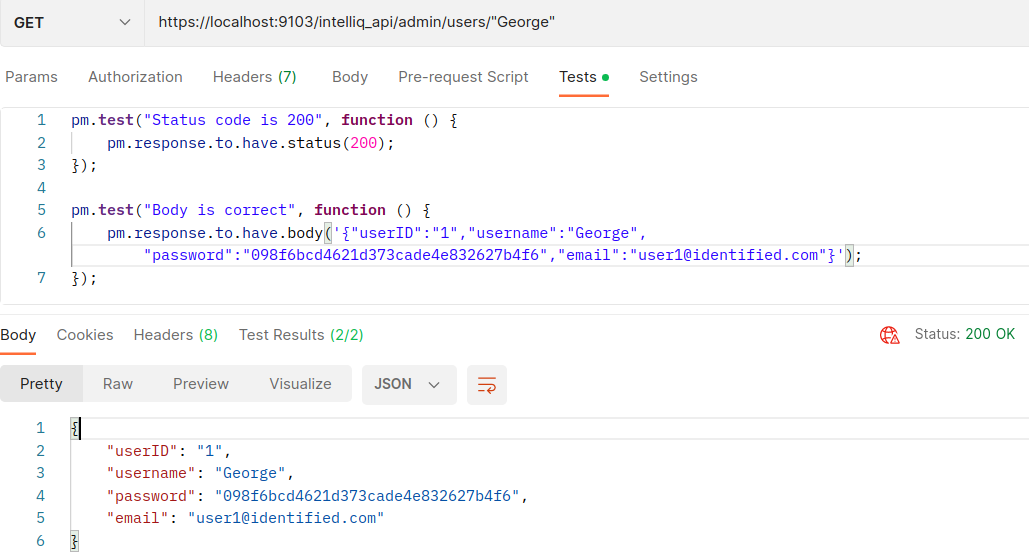
για να προσθέσουμε τον user με username Petros, password Rapto και email user4@identified.com (το email πρέπει να είναι @identified.com καθώς επιτρέπουμε μόνο χρήστες με αυτήν την μάσκα email)



* **{baseURL}/admin/users/:username**

Κάνουμε GET request στο https://localhost:9103/intelliq\_api/admin/users/"George"

για να πάρουμε τις πληροφορίες του χρήστη George:

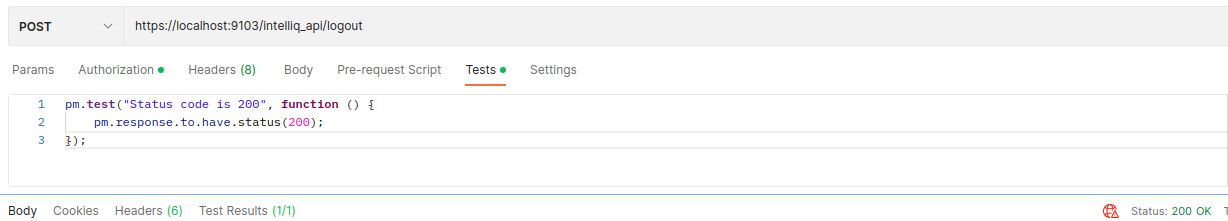


Βλέπουμε ότι η κλήση ήταν επιτυχημένη και περνάει τα tests.

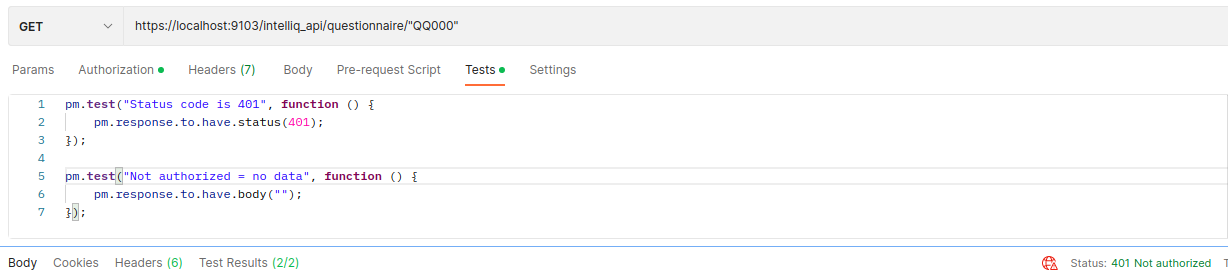
Τέλος μας έχει μείνει το endpoint **{baseURL}/logout**.

Κάνουμε την κλήση στο https://localhost:9103/intelliq\_api/logout

δίνοντάς του φυσικά το token στα headers, το οποίο έτσι κι αλλιώς πρέπει να περνάμε στα headers κάθε κλήσης:



Για να ελέγξουμε εάν έχει γίνει επιτυχές logout κάνουμε μια τυχαία κλήση σε κάποιο endpoint περνώντας στα headers του το token που μας δόθηκε κατά το login. Θέλουμε να λάβουμε απάντηση “401 Not Authorized”:



Αποδεικνύεται λοιπόν η σωστή λειτουργία και αυτού του endpoint.

Τρέχουμε όλο το collection και παρατηρούμε ότι περνάει όλα τα test cases μας:

