**Documentație ‘LadyBug’**

**-Routes-**

***Categoria I – AUTHENTICATION***

**Test 1 – Register** : *API pentru crearea unui nou utilizator*

**Method**: POST

**URL**: <http://localhost:3000/api/auth/register>

**Body**:

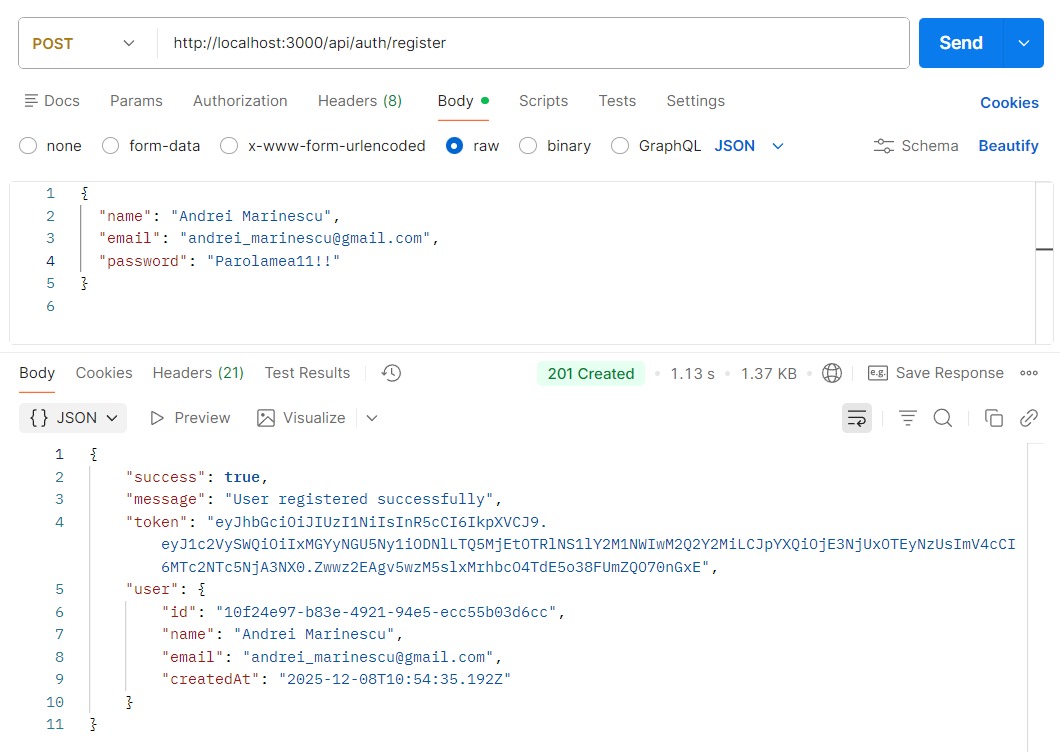
{

"name": "Andrei Marinescu",

"email": "Andrei\_marinescu@gmail.com",

"password": "Parolamea11!"

}



**Test 2 – Login** : *API pentru logarea utilizatorului în aplicație*

**Method** : POST

**URL** : <http://localhost:3000/api/auth/login>

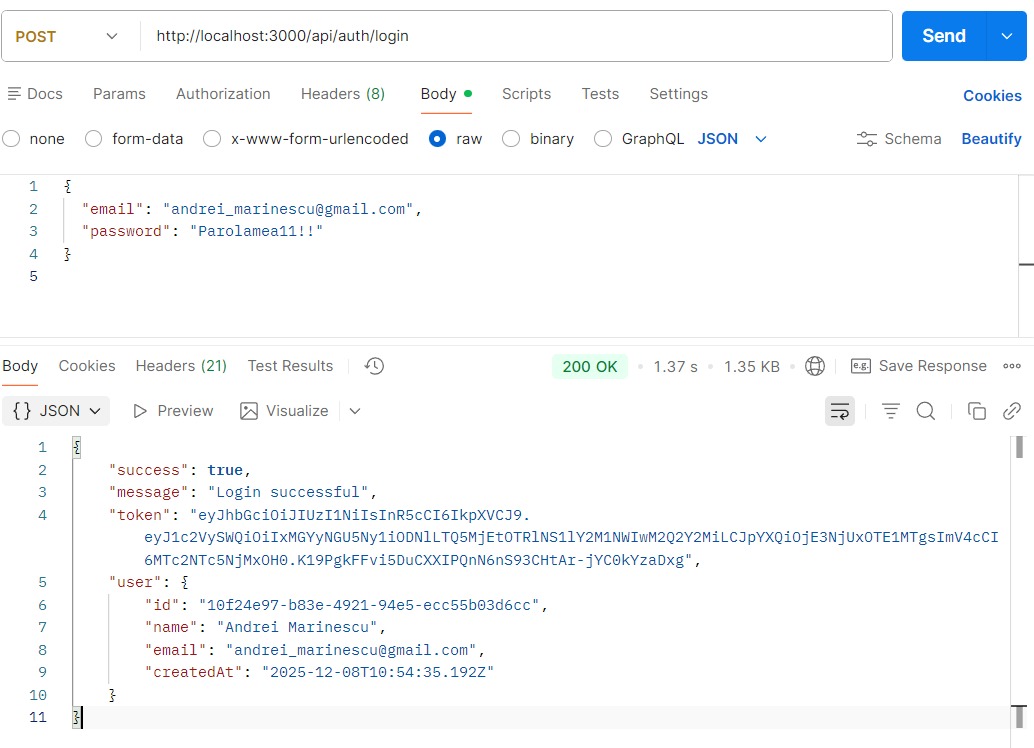
**Body** :

{

"email": "Andrei\_marinescu@gmail.com",

"password": "Parolamea11!"

}

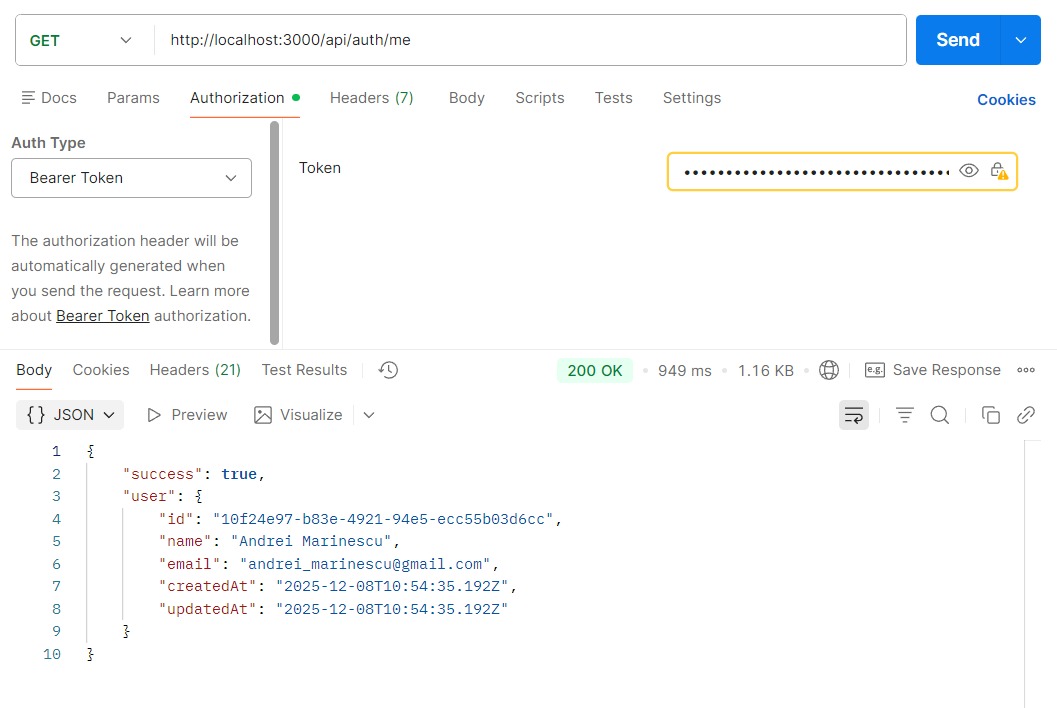


**Test 3 – Get current user** : *API pentru vizualizarea profilului personal*

**Method** : GET

**URL** : <http://localhost:3000/api/auth/me>

**Authorization** : token personal



**Test 4 – Update profil**e : *API pentru actualizarea profilului personal*

**Method** : PUT

**URL** : <http://localhost:3000/api/auth/profile>

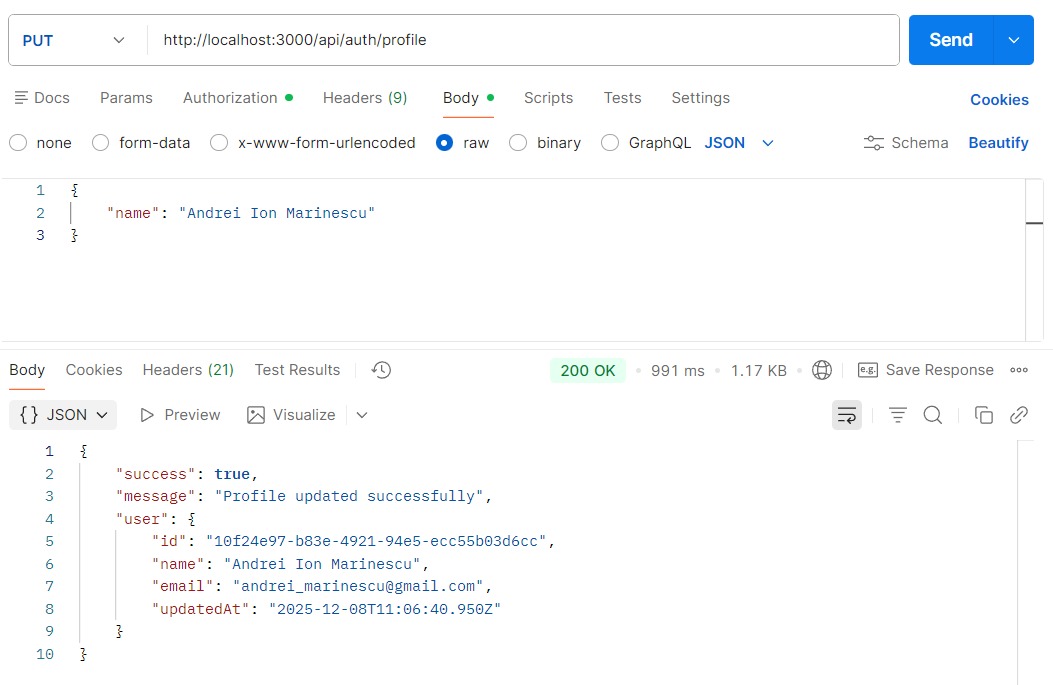
**Body**:

{

“name” : “Andrei Ion Marinescu”

}

**Authorization** : token personal



**Test 5 – Change Password** : *API pentru schimbarea parolei*

**Method** : PUT

**URL** : <http://localhost:3000/api/auth/password>

**Body**:

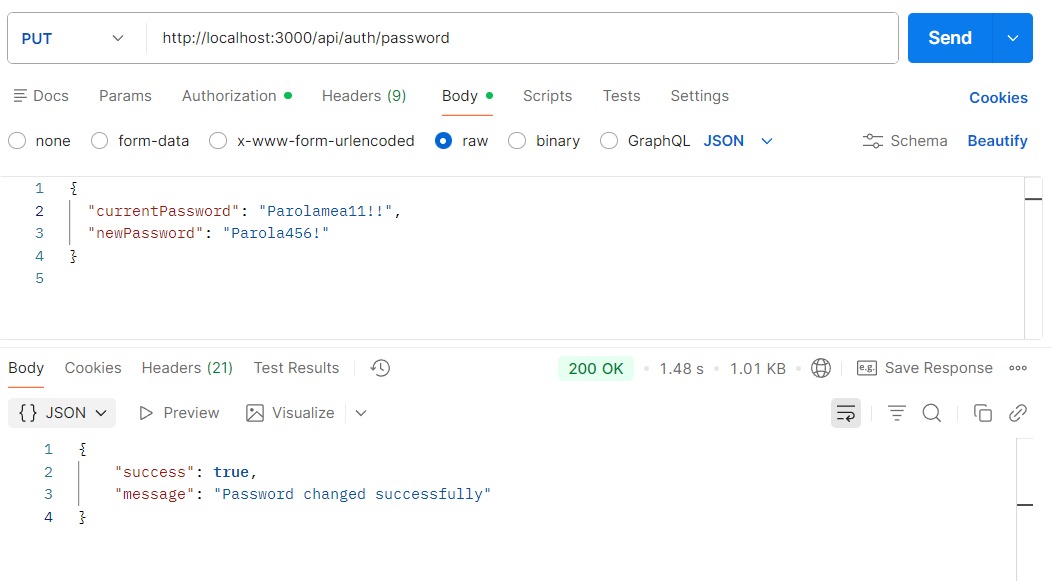
{

“currentPassword” : “Parolamea11!”,

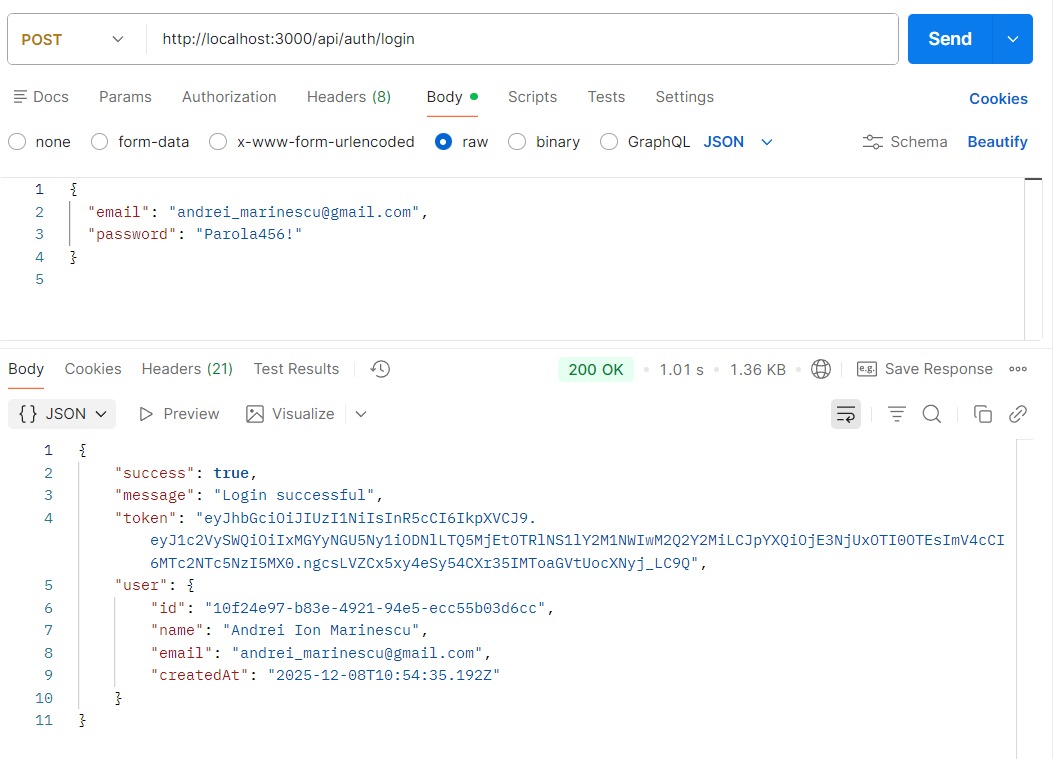
“newPassword” : “Parola456!”

}

**Authorization** : token personal



**Test** **pentru verificarea schimbării parolei** : relogarea utilizatorului



***Categoria II – PROJECTS***

**Test 1 – Create Project** : *API pentru crearea unui proiect nou*

**Method** : POST

**URL** : <http://localhost:3000/api/projects>

**Body** :

{

"name": "Aplicatie Restaurant",

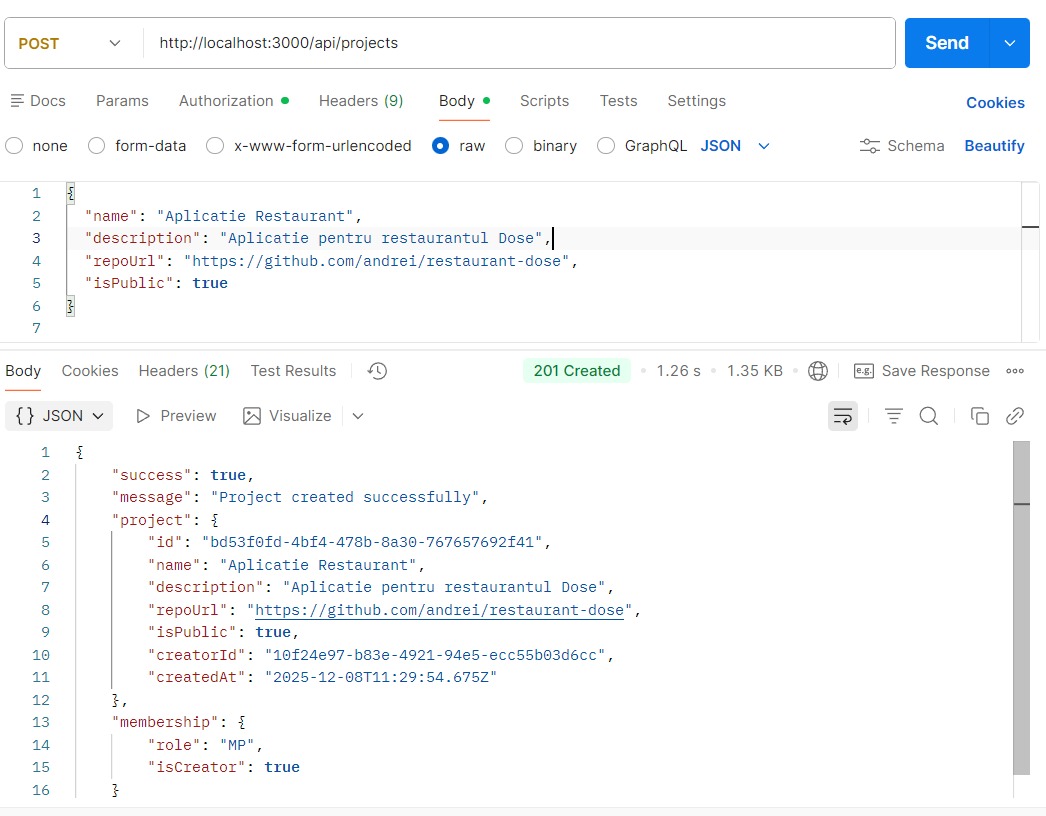
"description": "Aplicatie pentru restaurantul Dose",

"repoUrl": "https://github.com/andrei/restaurant-dose",

"isPublic": true

}

**Authorization** : token-ul utilizatorului care devine MP Creator

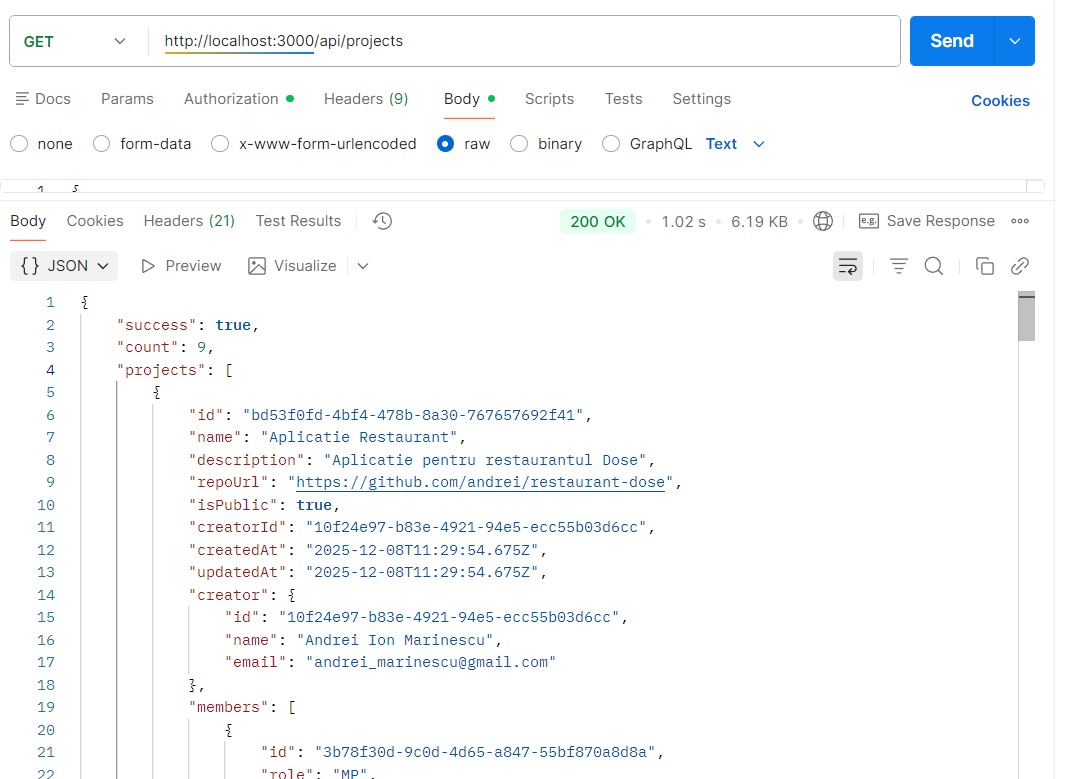


**Test 2 – Get All Projects** : *API pentru afișarea tuturor proiectelor din aplicație*

**Method** : GET

**URL** : <http://localhost:3000/api/projects>

**Authorization** : token-ul utilizatorului

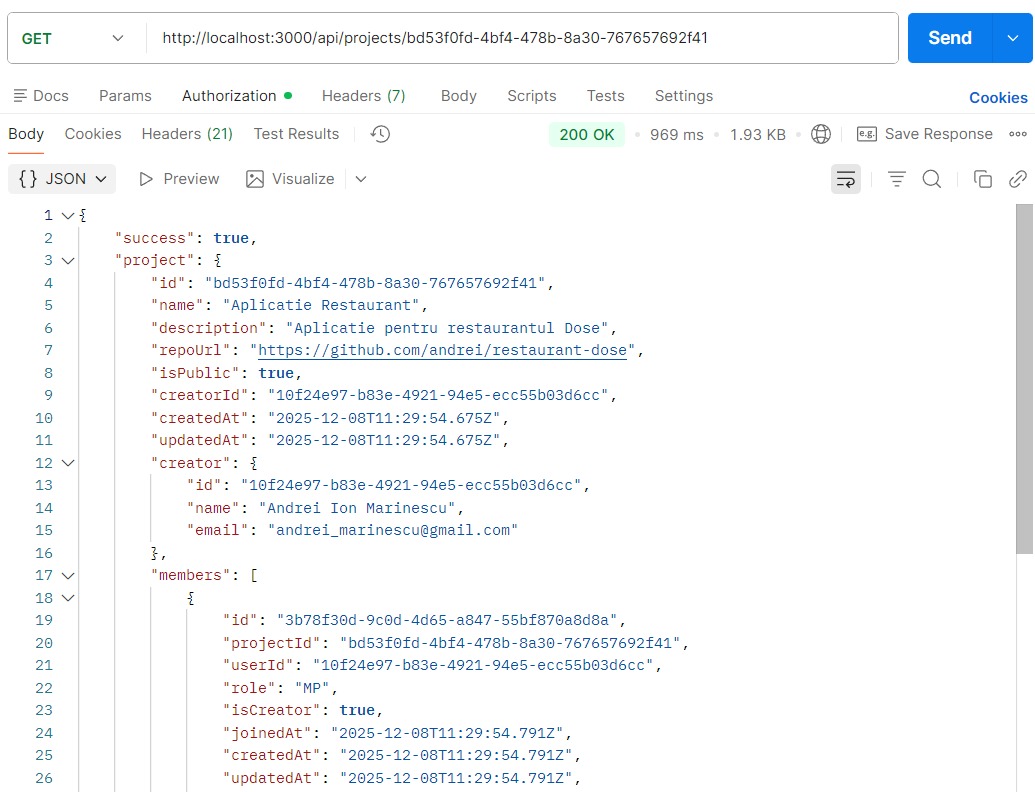


**Test 3 – Get project by ID** : *API pentru afișarea datelor unui proiect după ID-ul acestuia*

**Method** : GET

**URL** : <http://localhost:3000/api/projects/PROJECT_ID>

**Authorization**: token-ul creatorului proiectului

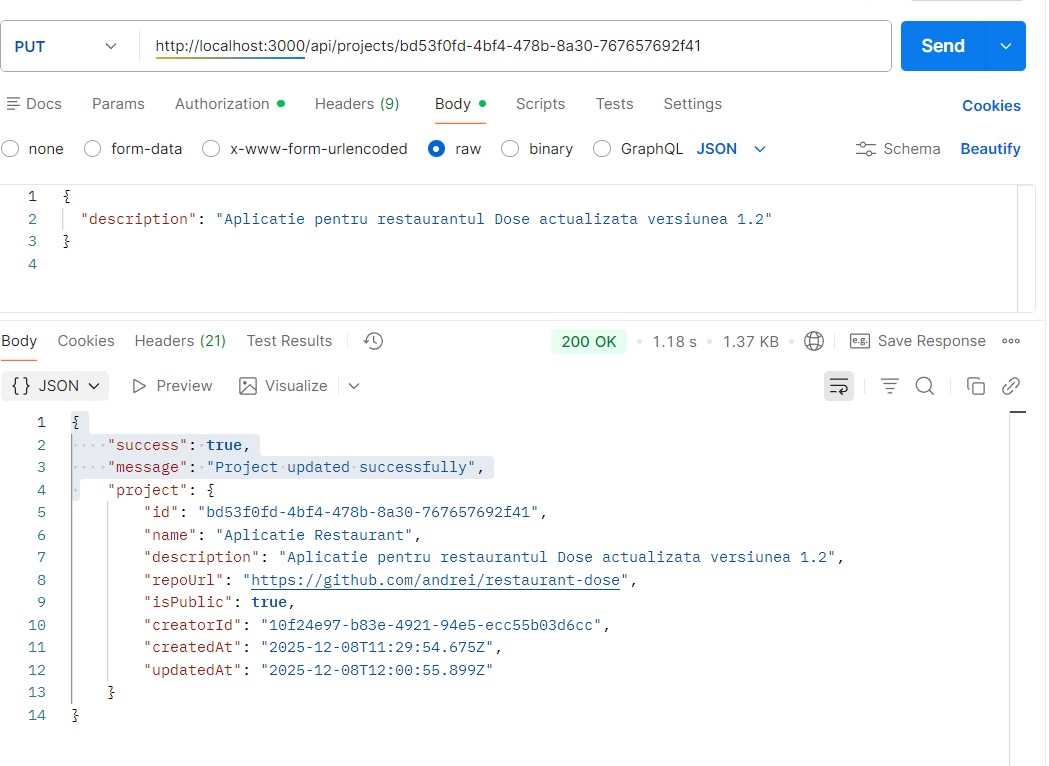


**Test 4 – Update Project** : *API pentru actualizarea datelor proiectului*

**Method** : PUT

**URL** : <http://localhost:3000/api/projects/PROJECT_ID>

**Authorization** : token-ul creatorului proiectului

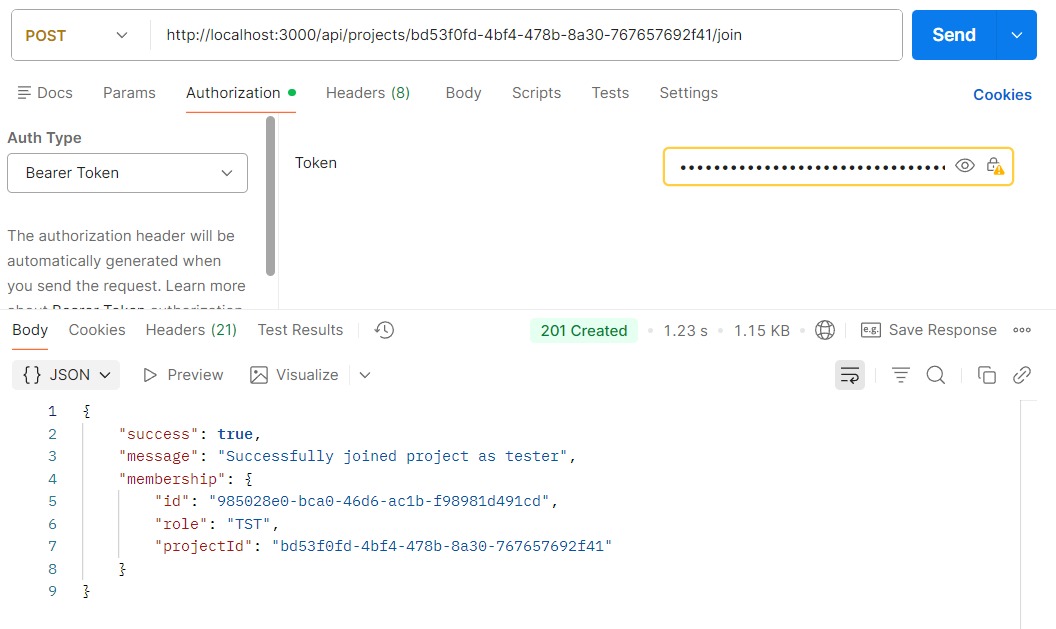


**Test 5 – Join as TST** : *API pentru înregistrarea ca tester în cadrul unui proiect*

**Method** : POST

**URL** : <http://localhost:3000/api/projects/PROJECT_ID/join>

**Authorization** : token-ul utilizatorului care dorește să se alăture ca tester proiectului, altul față de creatorul proiectului



**Test 6 – Add a member as MP** : *API pentru înregistrarea ca member project a unui utilizator de către creatorul proiectului*

**Method** : POST

**URL** : <http://localhost:3000/api/projects/PROJECT_ID/members>

**Body** :

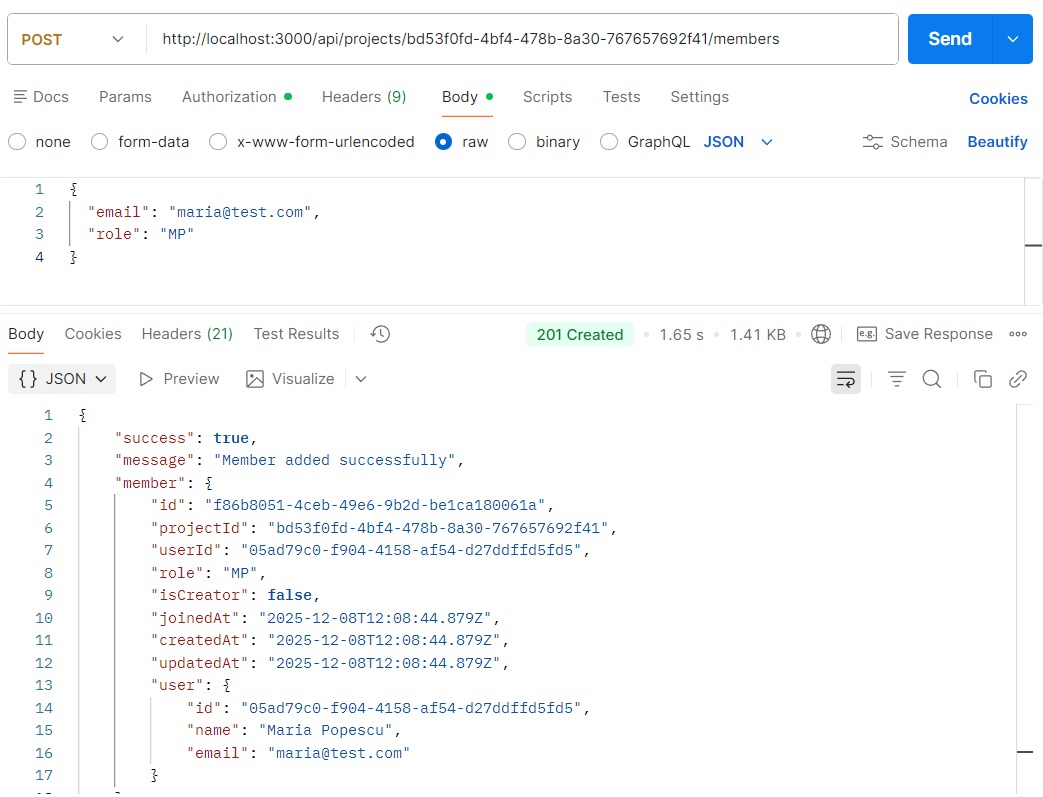
{

"email": "maria@test.com",

"role": "MP"

}

**Authorization** : token-ul utilizatorului care dorește să se alăture ca tester proiectului, altul față de creatorul proiectului

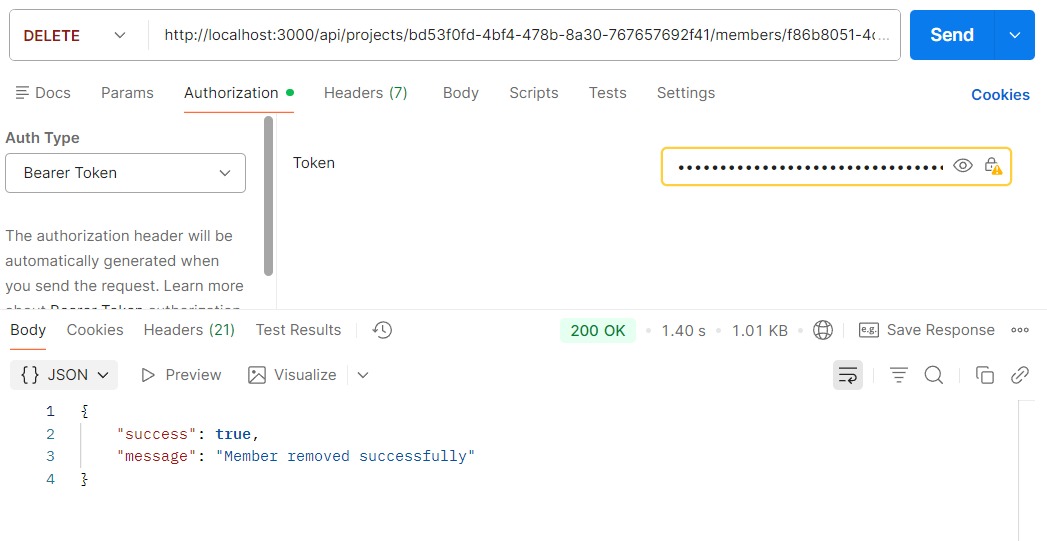


**Test 7 – Delete MP** : *API pentru ștergerea unui MP din funcție de către MP Creator*

**Method** : DELETE

**URL** : <http://localhost:3000/api/projects/PROJECT_ID/members/MEMBER_ID>

**Authorization** : token-ul utilizatorului care dorește să se alăture ca tester proiectului, altul față de creatorul proiectului



***CATEGORIA III – BUGS***

**Test 1 – Report Bug** : *API pentru raporatarea unui bug de către un TST sau MP al proiectului*

**Method** : POST

**URL** : <http://localhost:3000/api/bugs>

**Body** :

{

"projectId": "PROJECT\_ID”,

"title": "Bug test Silviu",

"description": "Meniul nu se deschide pe telefon",

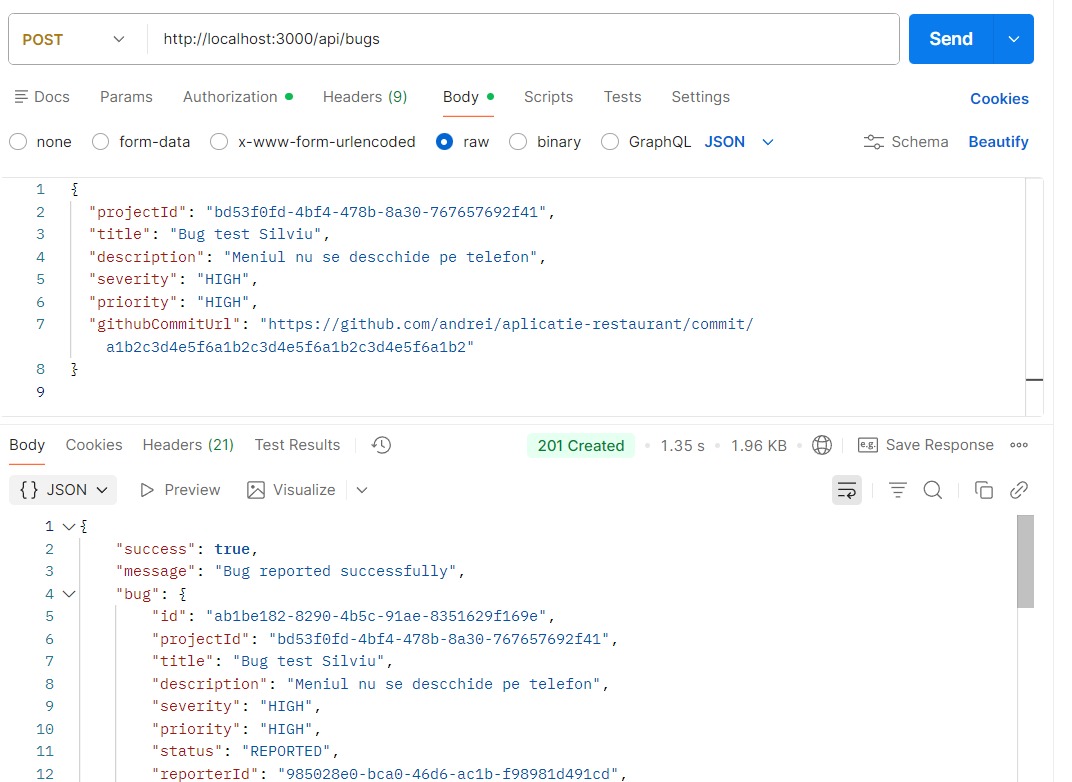
"severity": "HIGH",

"priority": "HIGH",

"githubCommitUrl": "https://github.com/andrei/aplicatie-restaurant/commit/a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5f6a1b2"

}

**Authorization** : token-ul celui care raportează bug-ul

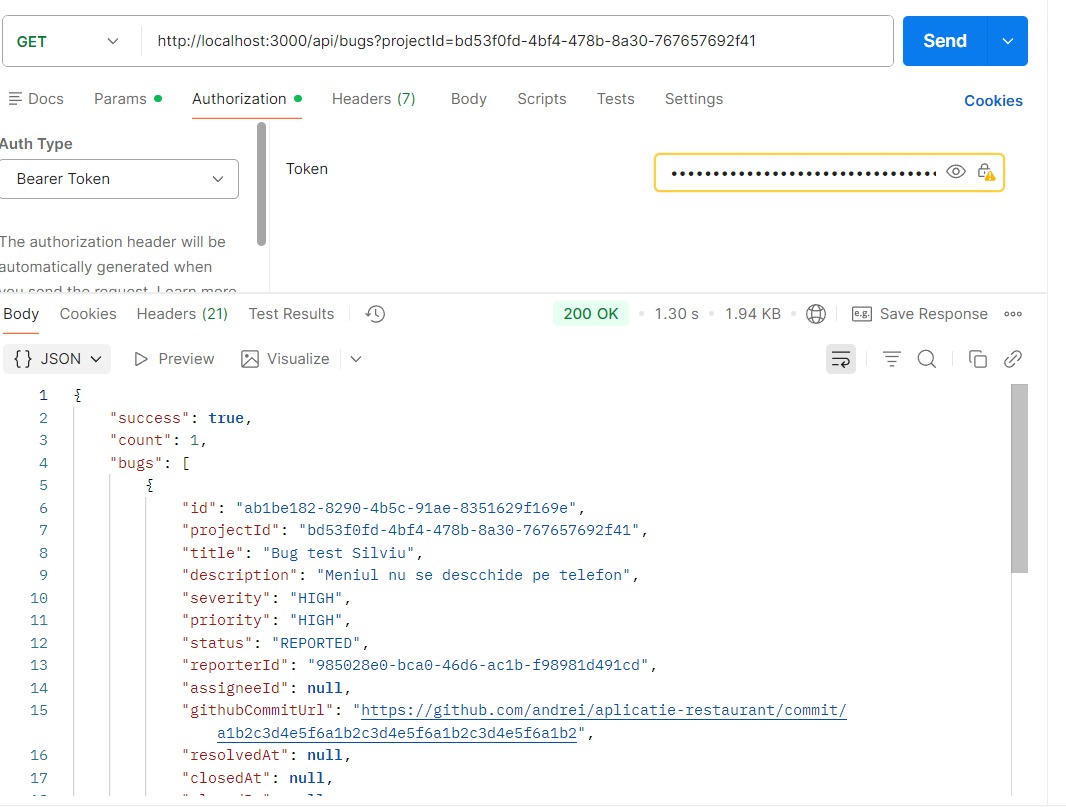


**Test 2 – Get All Bugs** : *API pentru afișarea tuturor bug-urilor dintr-un proiect*

**Method** : GET

**URL** : <http://localhost:3000/api/bugs?projectID=PROJECT_ID>

**Authorization** : token-ul creatorului de proiect

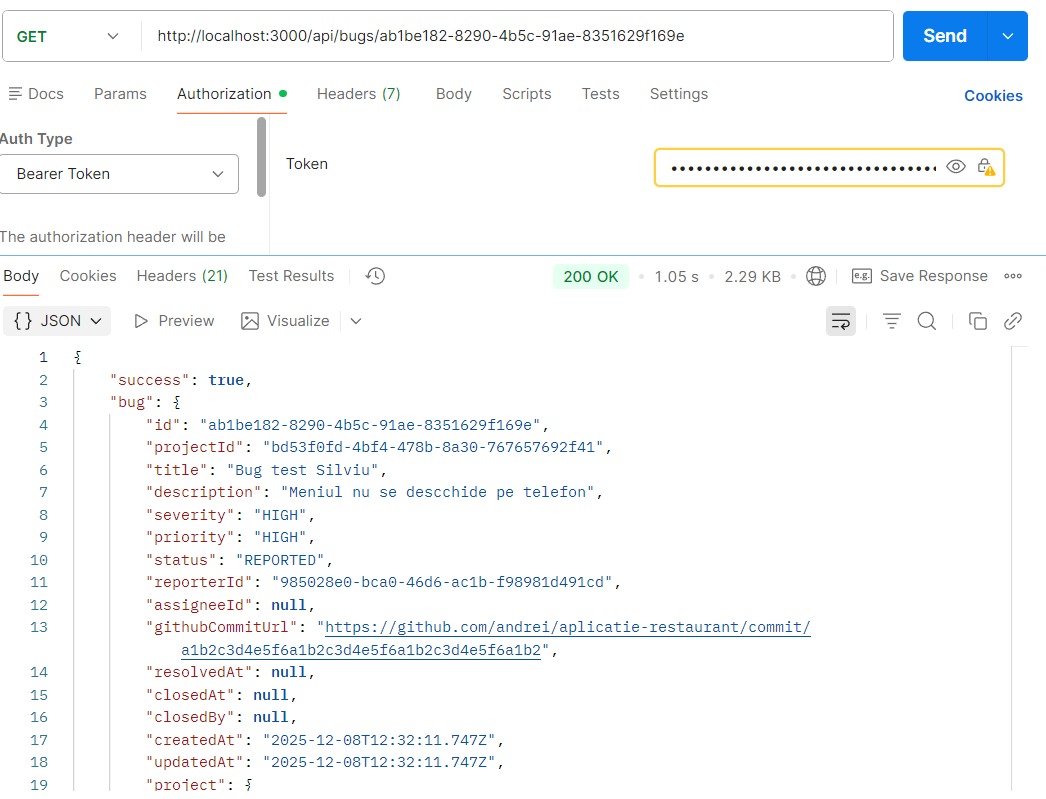


**Test 3 – Get Bug by ID** : *API pentru afișarea datelor unui bug după ID-ul acestuia*

**Method** : GET

**URL** : http://localhost:3000/api/bugs/BUG\_ID

**Authorization** : token-ul creatorului de proiect



**Test 4 – Assign Bug to MP** : *API pentru alocarea unui bug către un MP de către creatorul proiectului*

**Method** : POST

**URL** : <http://localhost:3000/api/bugs/MEMBER_ID/assign>

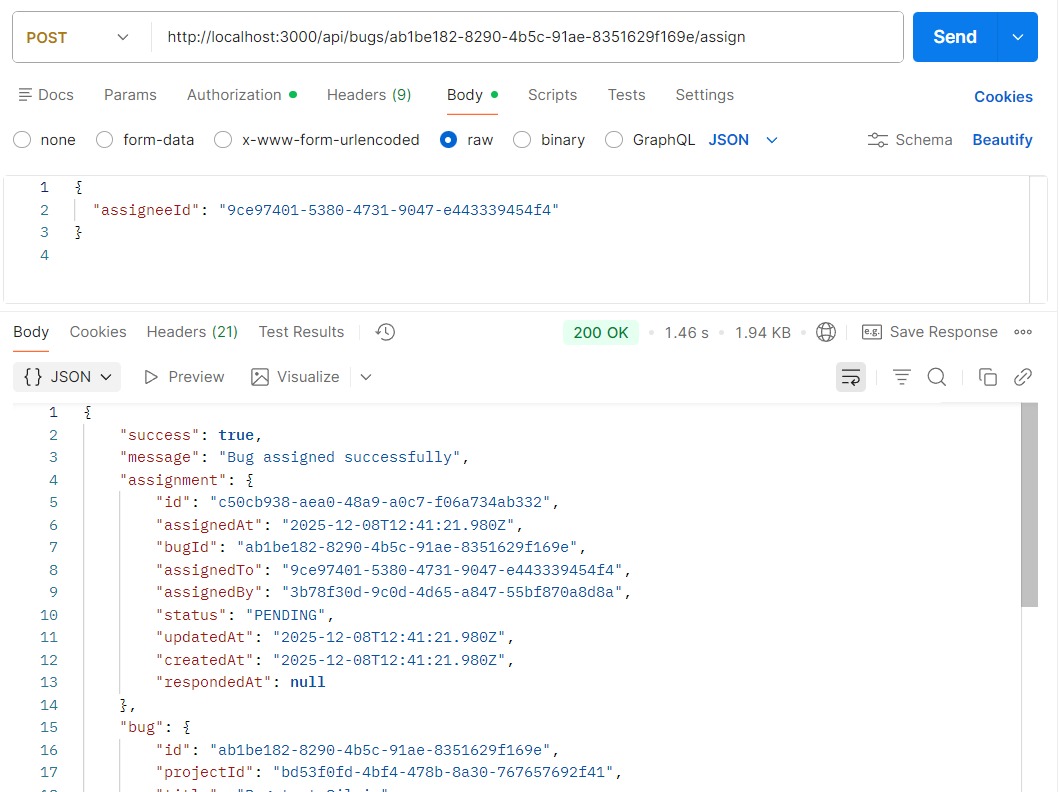
**Body** :

{

“asigneeId” : “PROJECT\_MEMBER\_ID”

}

**Authorization** : token-ul creatorului de proiect



**Test 5 – Self-Assign Bug** : *API pentru alocarea proprie a unui bug într-un proiect în care utilizatorul este membru*

**Method** : POST

**URL** : <http://localhost:3000/api/bugs/BUG_ID/self-assign>

**Authorization** : token-ul membrului care își alocă bug-ul



**Test 6 – Update Status Bug** : *API pentru actualizarea statusului bug-ului de către membrul căruia îi este alocat*

**Method** : PUT

**URL** : <http://localhost:3000/api/bugs/BUG_ID/status>

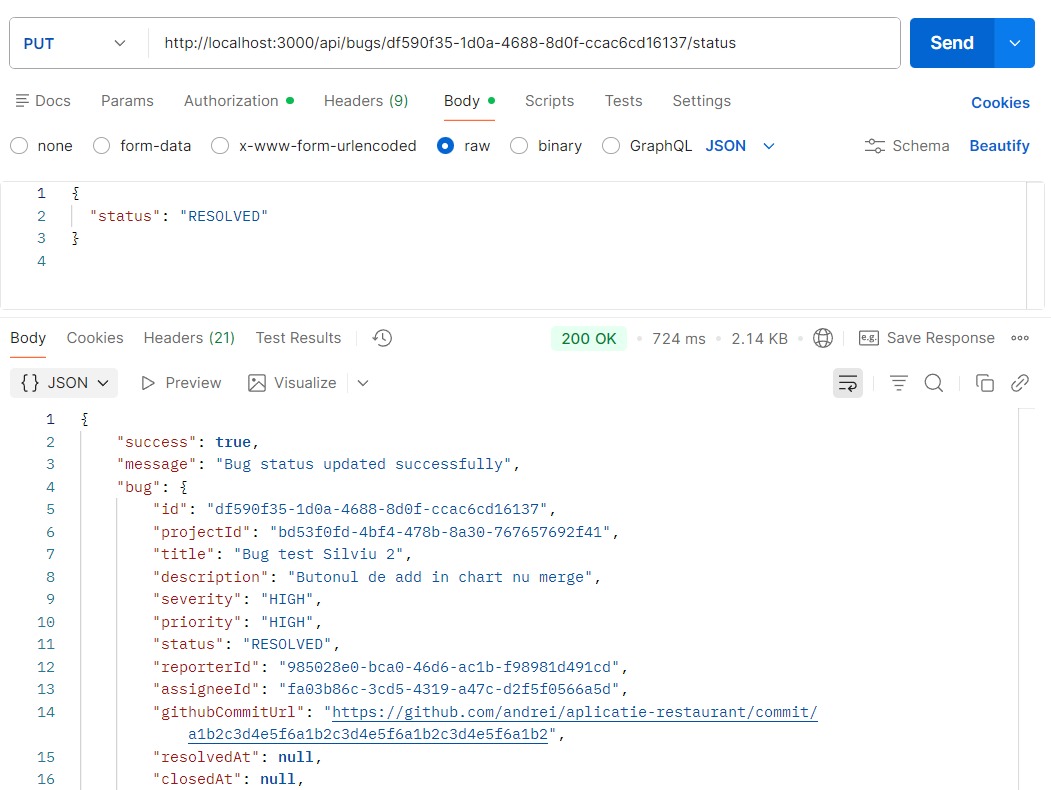
**Body** :

{

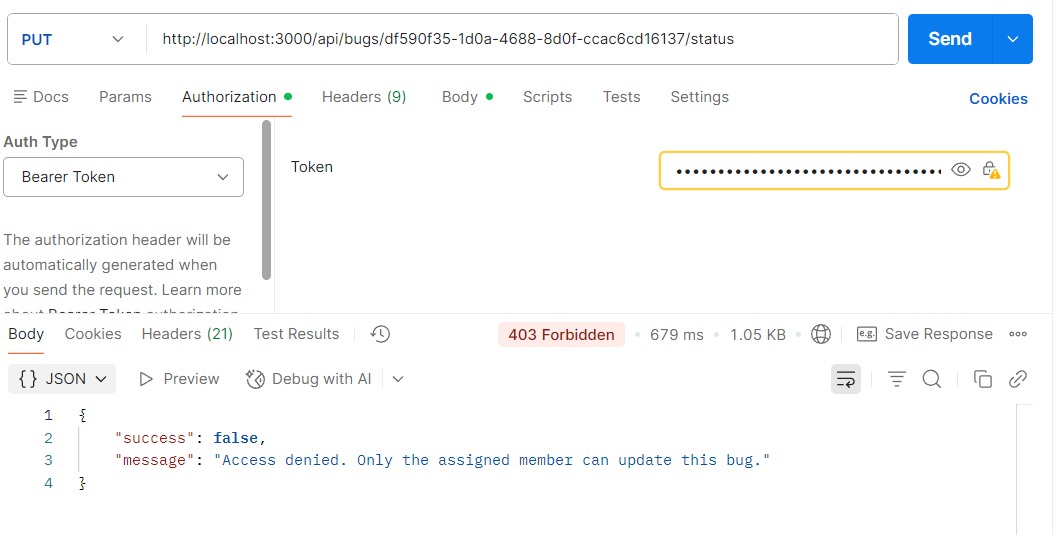
"status": "RESOLVED"

}

**Authorization** : token-ul membrului căruia îi este alocat bug-ul



**Test** **pentru verificarea permisiunilor** : încercarea altui membru de a schimba statusul unui bug care nu îi este alocat

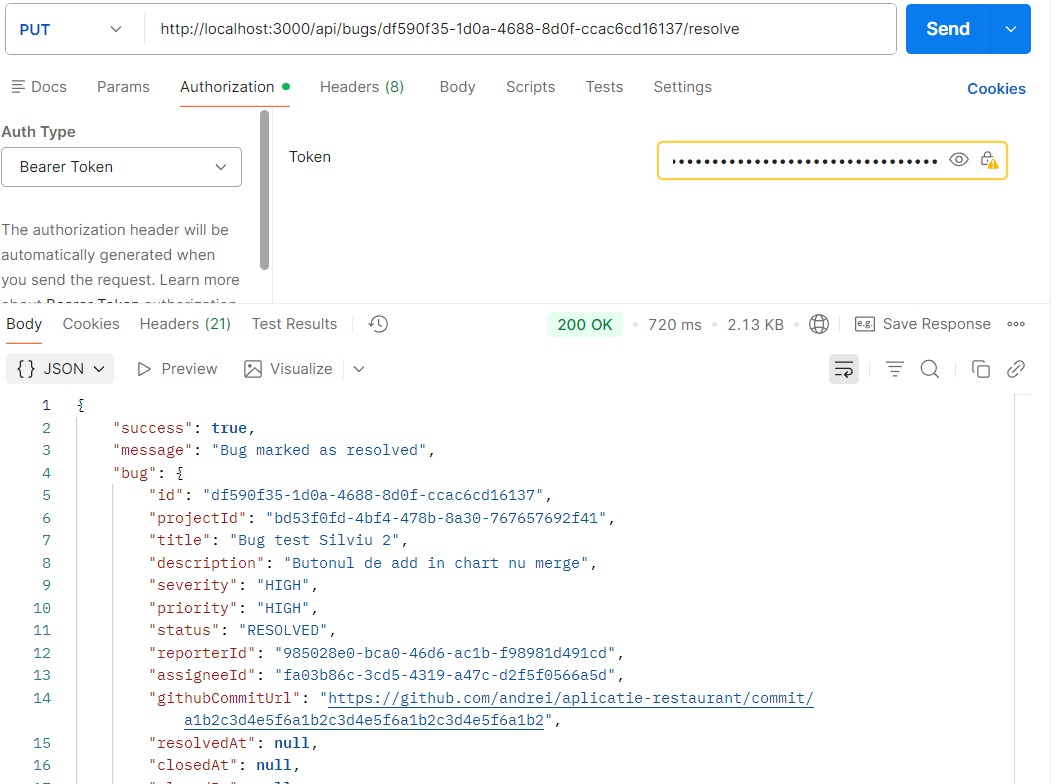


**Test 7 – Resolve Bug** : *API pentru declararea rezolvării unui bug*

**Method** : PUT

**URL** : <http://localhost:3000/api/bugs/BUG_ID/resolve>

**Authorization** : token-ul membrului căruia îi este alocat bug-ul

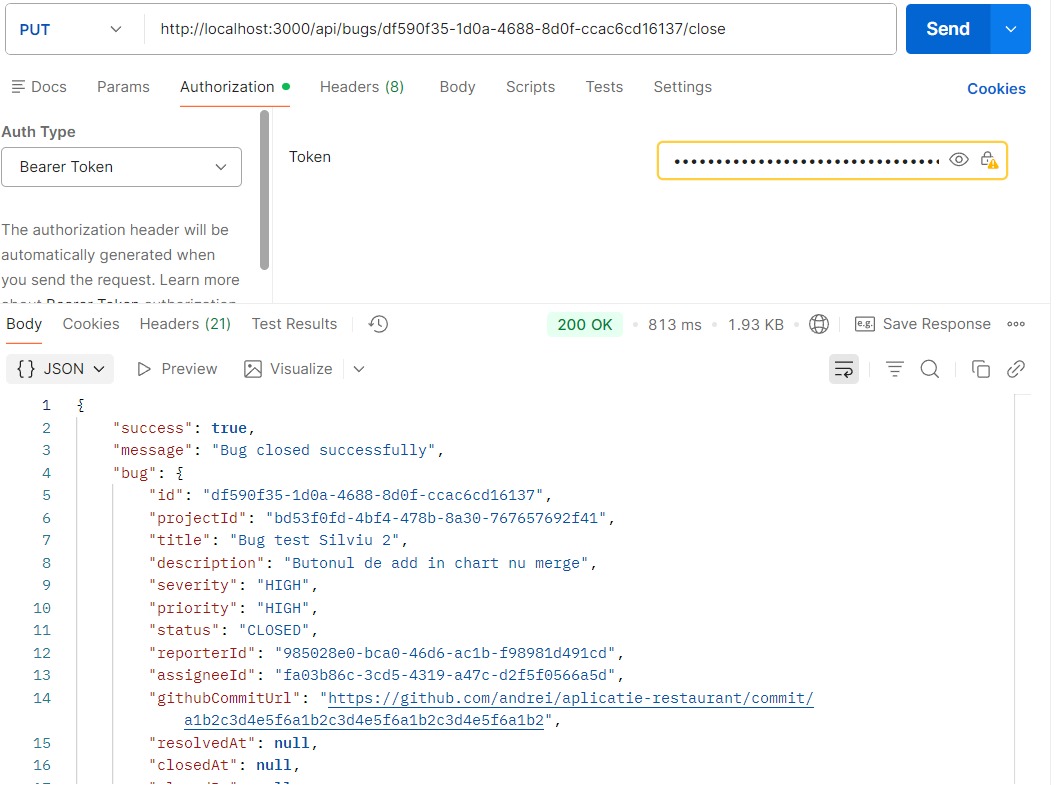


**Test 8 – Close Bug** : *API pentru închiderea unui bug rezolvat*

**Method** : PUT

**URL** : <http://localhost:3000/api/bugs/BUG_ID/close>

**Authorization** : token-ul creatorului de proiect

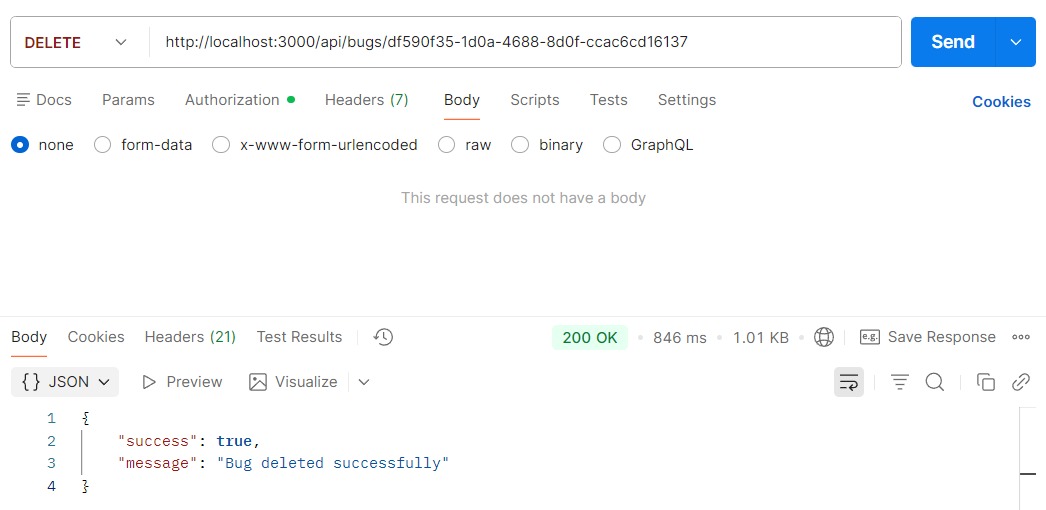


**Test 9 – Delete Bug** : *API pentru ștergerea unui bug*

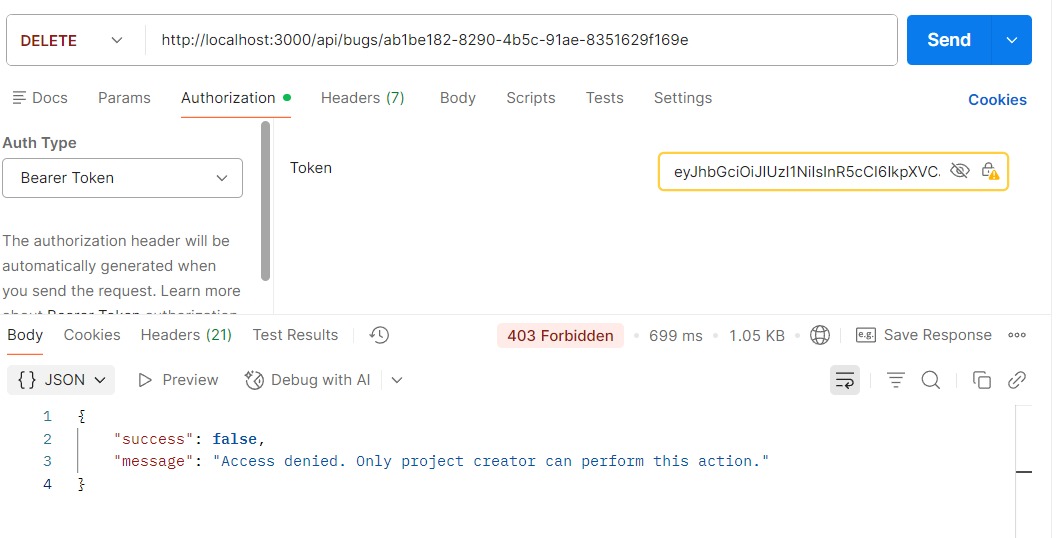
**Method** : DELETE

**URL** : <http://localhost:3000/api/bugs/BUG_ID>

**Authorization** : token-ul creatorului de proiect



**Test pentru verificarea permisiunilor** : încercarea altui membru, altul decât creatorul proiectului, de a șterge un bug



***CATEGORIA IV – BUG ASSIGNMENTS***

**Test 1 – Accepting Bug** : *API pentru acceptarea unui bug de către un membru, după ce i-a fost alocat de către creatorul proiectului*

**Method** : PUT

**URL** : <http://localhost:3000/api/bug-assignments/ASSIGNMENT_ID/accept>

**Authorization** : token-ul membrului căruia i s-a alocat bug-ul



**Test 2 – Rejecting Bug** : *API pentru refuzarea unui bug de către un membru, după ce i-a fost alocat de către creatorul proiectului*

**Method** : PUT

**URL** : http://localhost:3000/api/bug-assignments/ASSIGNMENT\_ID/reject

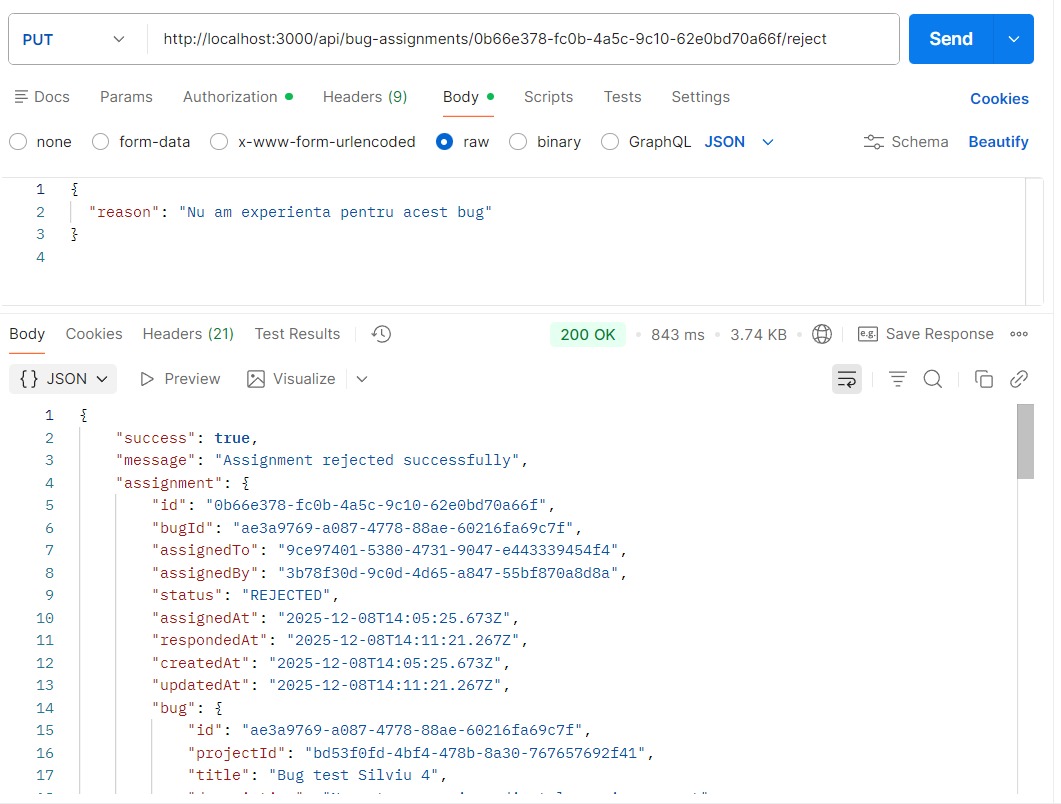
**Body** :

{

“reason” : “Nu am experinta pentru acest bug.”

}

**Authorization** : token-ul membrului căruia i s-a alocat bug-ul

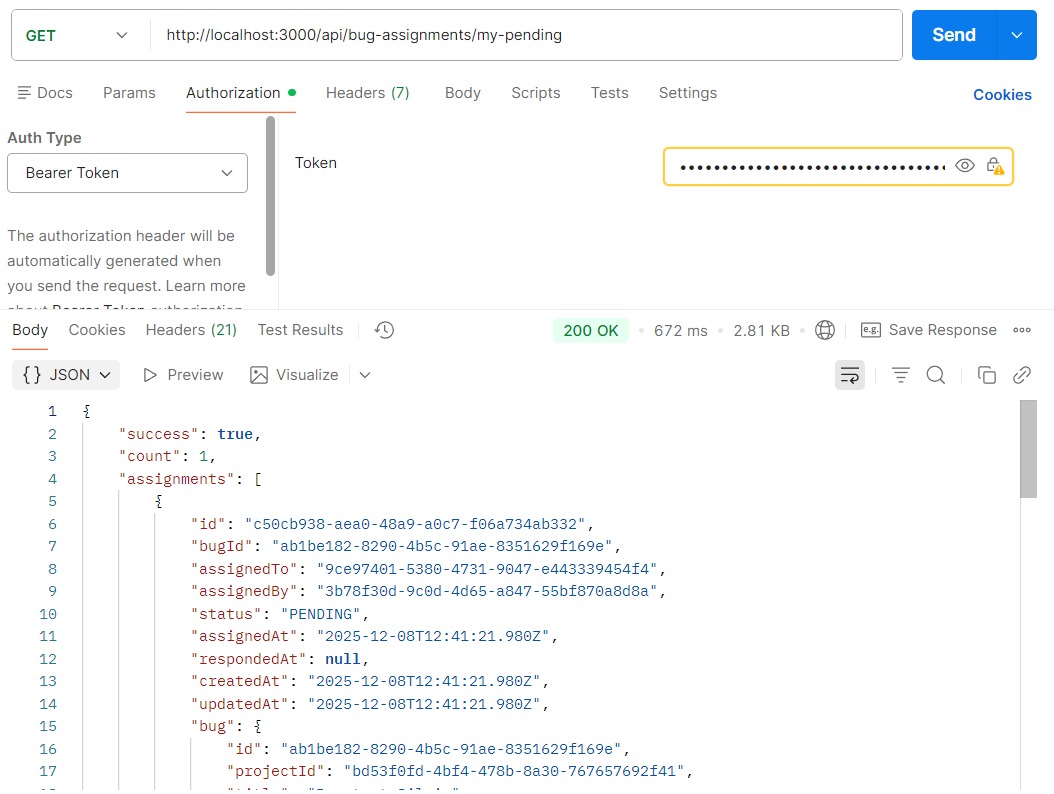


**Test 3 – Get My Pending Assignments**: *API pentru afișarea assigment-urilor care nu au primit încă răspuns*

**Method** : GET

**URL** : http://localhost:3000/api/bug-assignments/my-pending

**Authorization** : token-ul membrului căruia i s-au alocat bug-uri

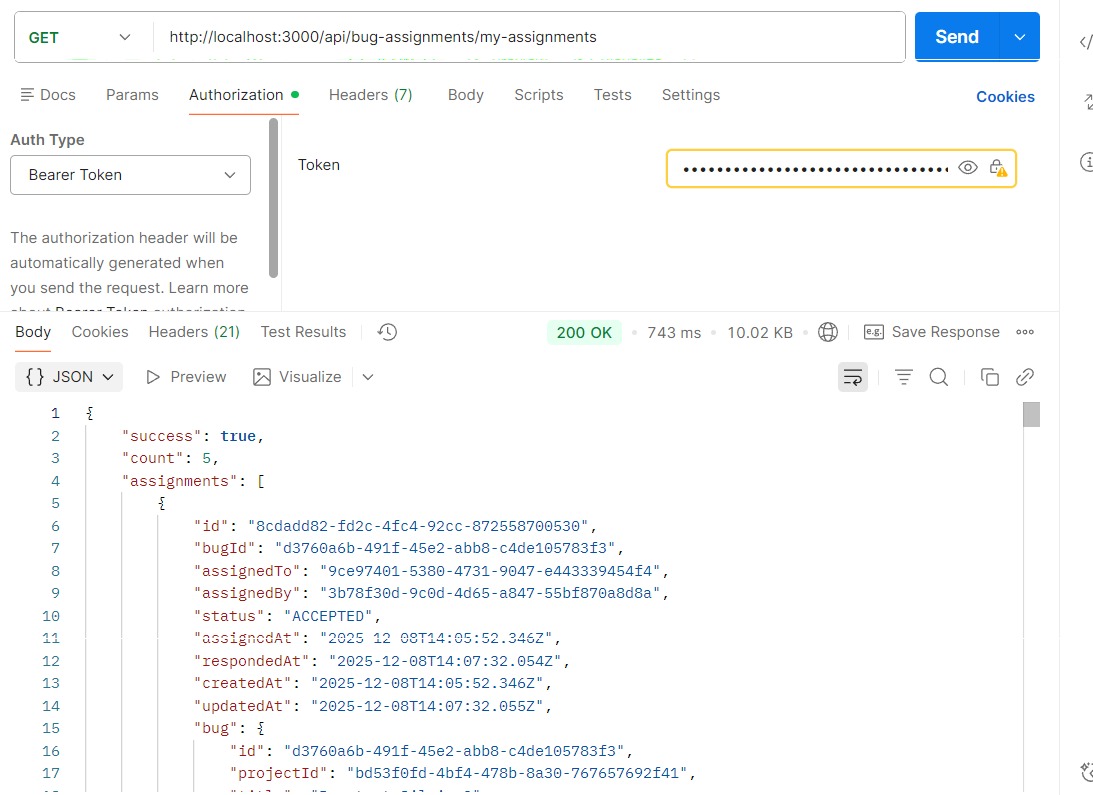


**Test 4 – Get All My Assignments**: *API pentru afișarea tuturor assignment-urilor, cu posibilitatea filtrării după status*

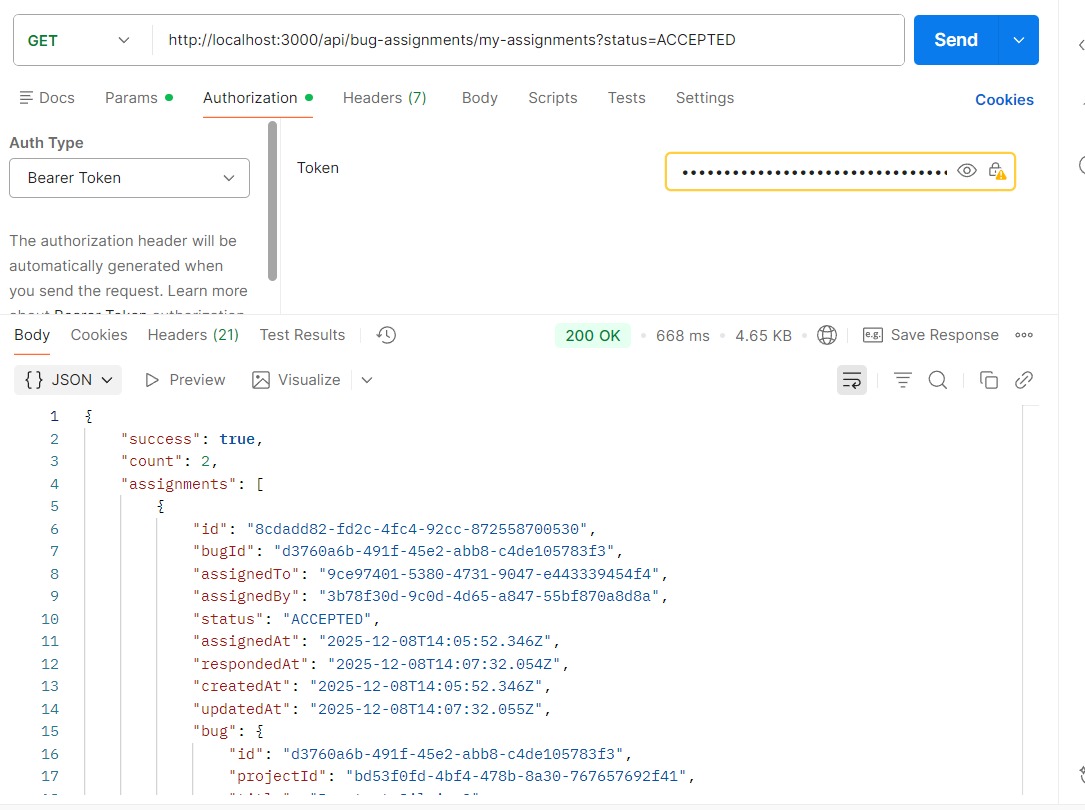
**Method** : GET

**URL** : <http://localhost:3000/api/bug-assignments/my-assignments>

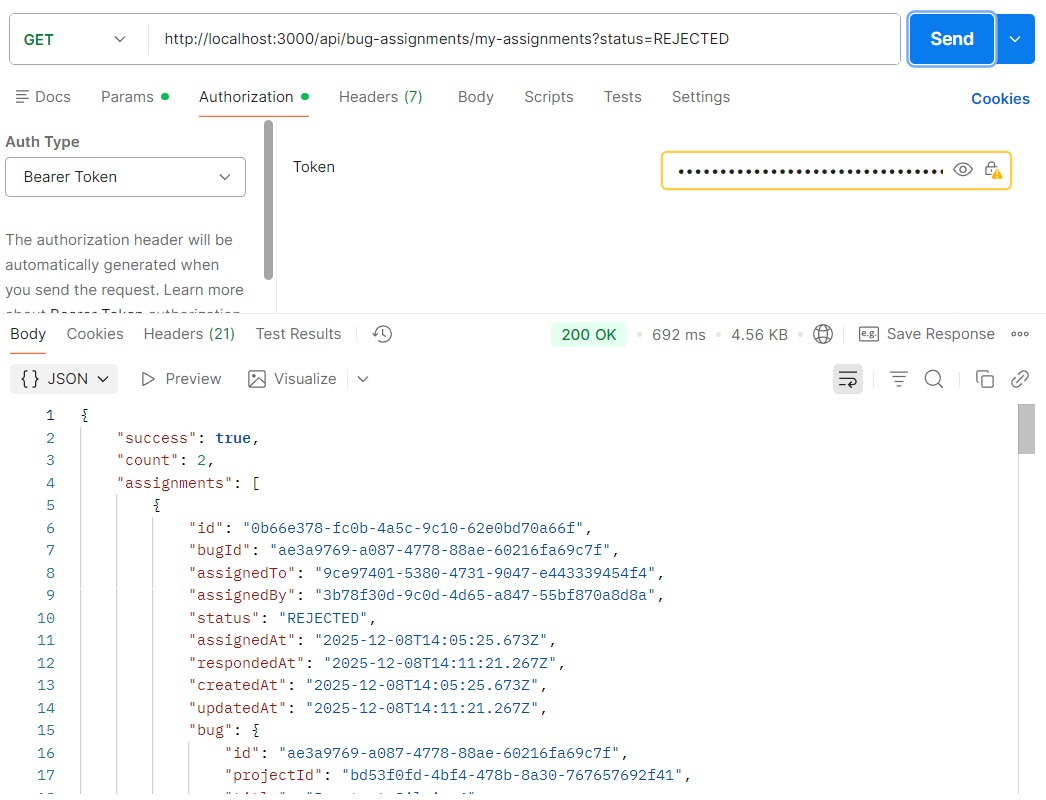
**Authorization** : token-ul membrului căruia i s-au alocat bug-uri



**URL** : <http://localhost:3000/api/bug-assignments/my-assignments?status=ACCEPTED>



**URL** : <http://localhost:3000/api/bug-assignments/my-assignments?status=REJECTED>

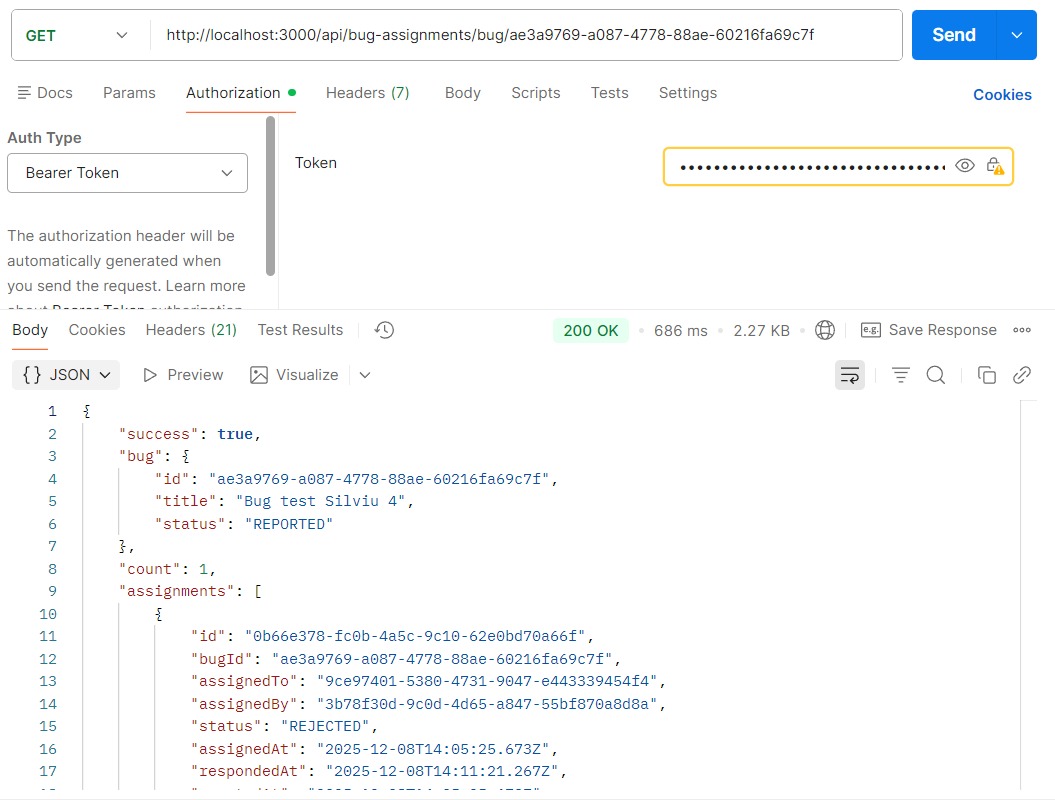


**Test 5 – Get History Bug** : *API pentru afișarea istoricului unui bug*

**Method** : GET

**URL** : <http://localhost:3000/api/bug-assignments/bug/BUG_ID>

**Authorization** : token-ul creatorului de proiect



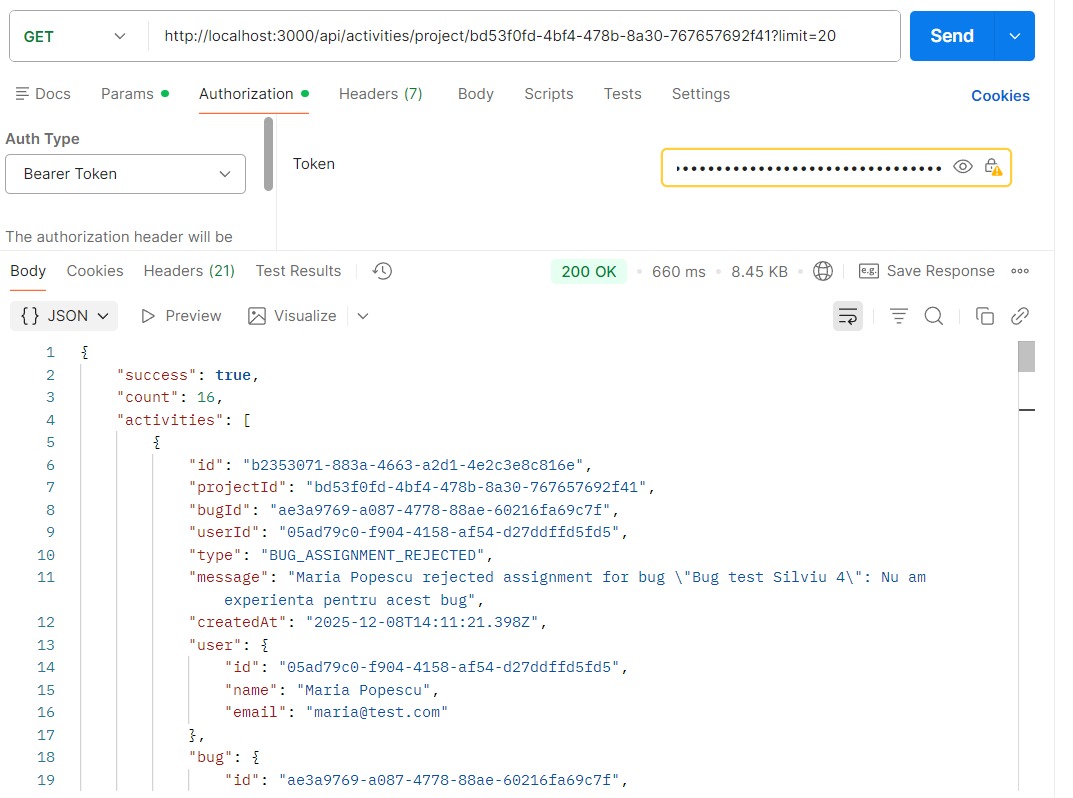
***CATEGORIA V – ACTIVITIES***

**Test 1 – Get Project Activities Feed** : *API pentru afișarea tuturor activităților unui proiect*

**Method** : GET

**URL** : <http://localhost:3000/api/activities/project/PROJECT_ID?limit=20>

**Authorization** : token-ul creatorului de proiectului

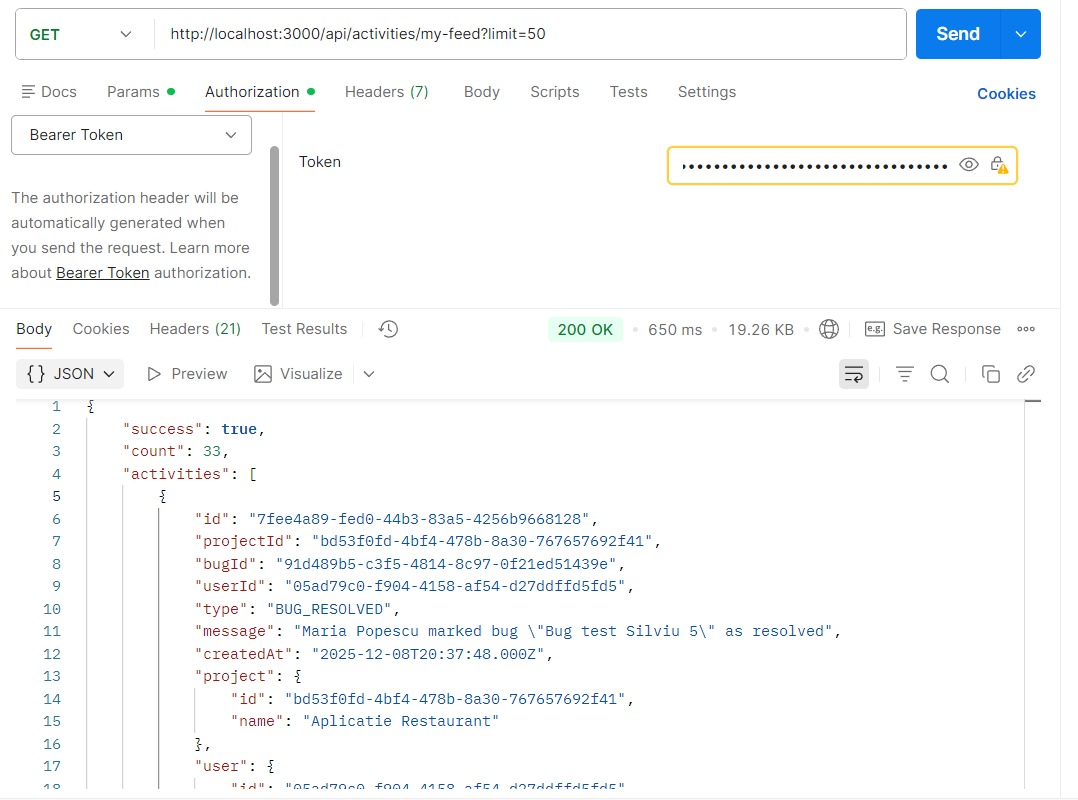


**Test 2 – Get My Activity Feed** : *API pentru afișarea tuturor activităților unui utilizator în cadrul tuturor proiectelor în care participă*

**Method** : GET

**URL** : http://localhost:3000/api/activities/my-feed?limit=50

**Authorization** : token-ul oricărui utilizator

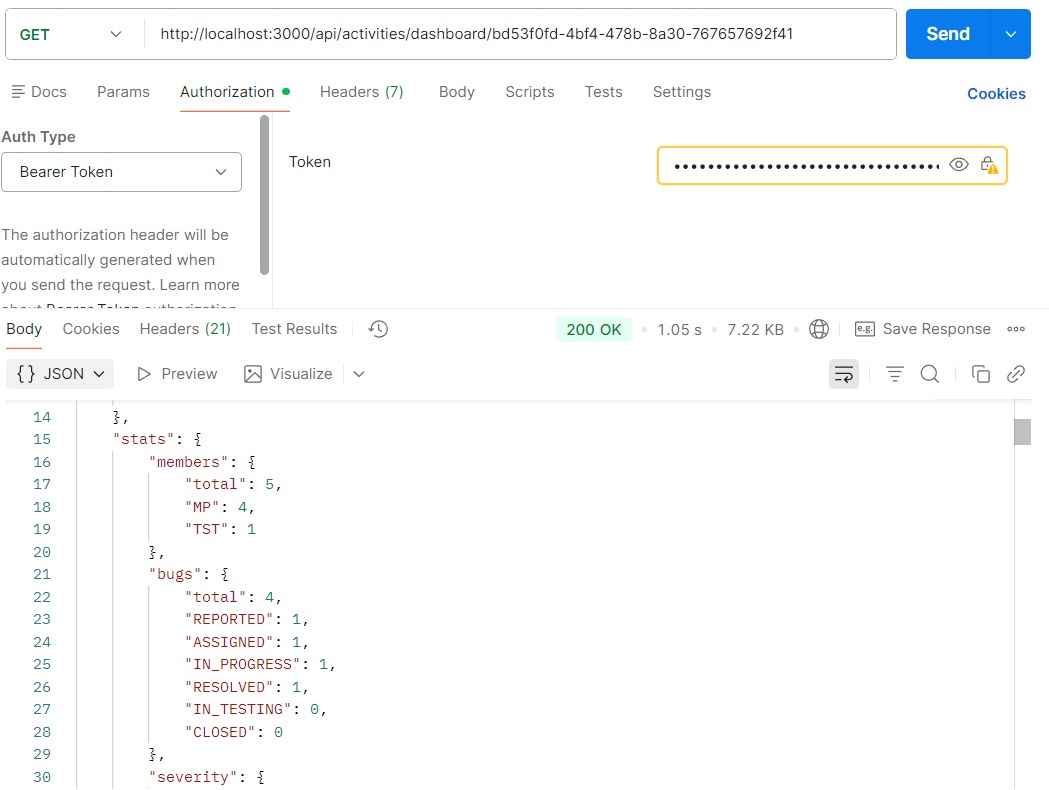


**Test 3 – Get Dashboard Statistics** : *API pentru afișarea tuturor statisticilor unui proiect*

**Method** : GET

**URL** : <http://localhost:3000/api/activities/dashboard/PROJECT_ID>

**Authorization** : token-ul creatorului de proiectului



**Test 4 – Get My Personal Statistics** : *API pentru afișarea tuturor statisticilor personale*

**Method** : GET

**URL** : <http://localhost:3000/api/activities/my-stats>

**Authorization** : token-ul oricărui utilizator

