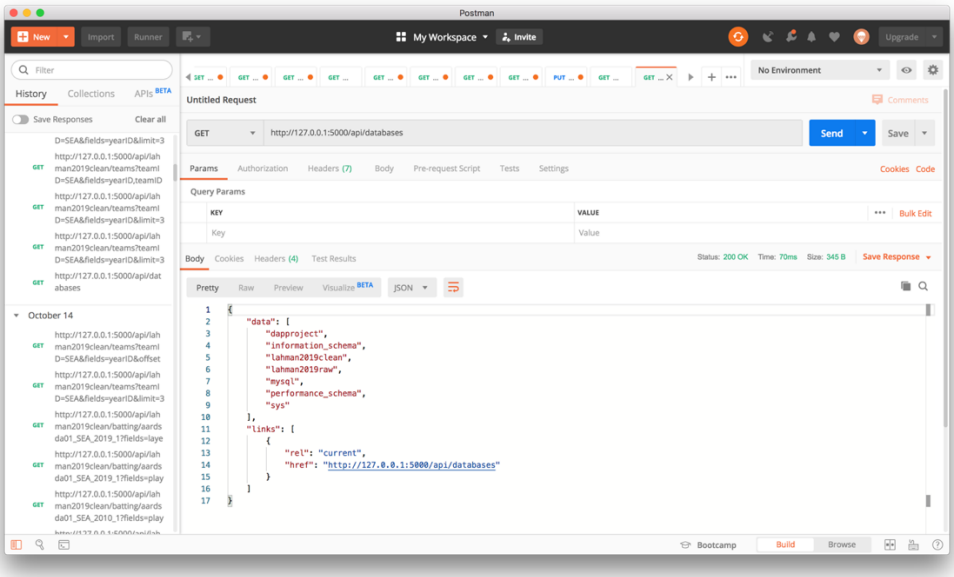
Homework2 – Part3 (Test cases for the REST APIs)
UNI – PKV2103

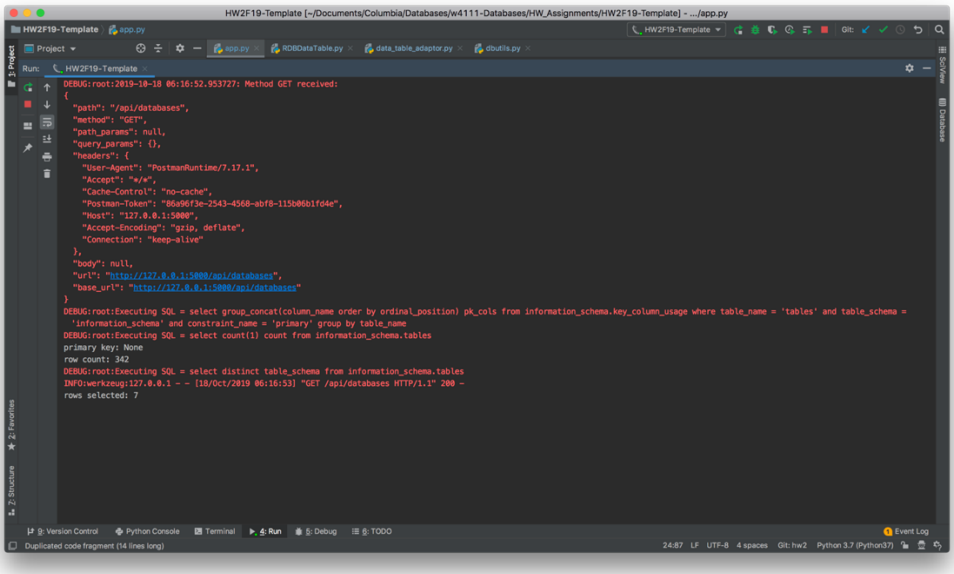
Implementation Notes:

- ❖ PUT and POST requires JSON objects to be defined in body
- ❖ GET and DELETE can be tested by URL with query params.

```
1. @application.route("/api/databases", methods=["GET"])
```

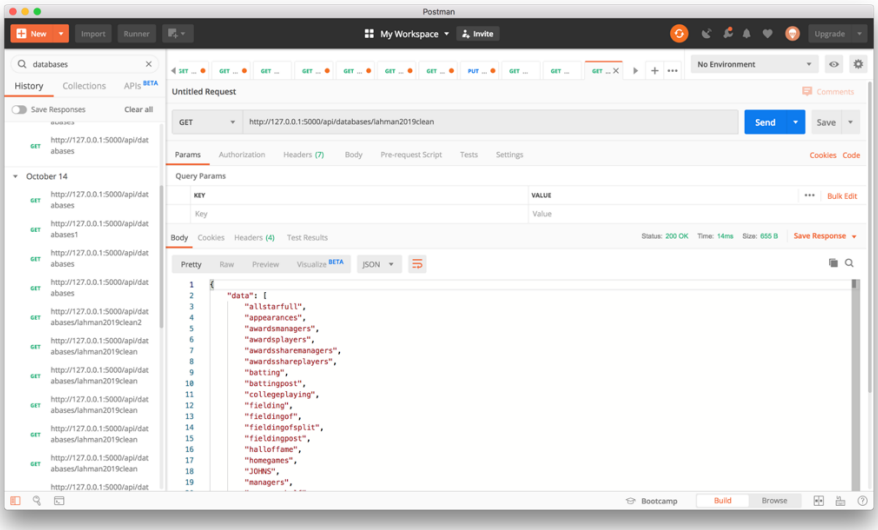
- GET list of databases in my local instance of mysql. No input parameters passed. The related return link is same as the original link. Please see below the screenshots of Postman and also console output. Test Successful

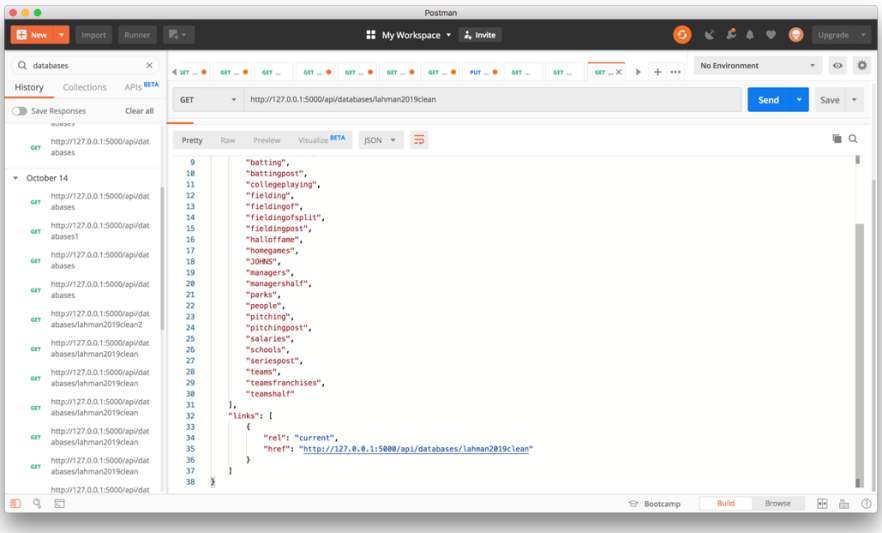


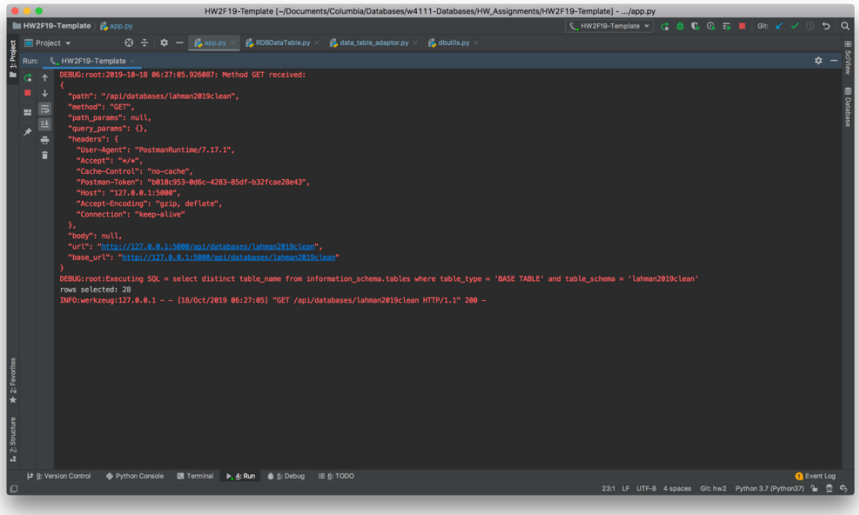



```
2. @application.route("/api/databases/<dbname>", methods=["GET"])
```

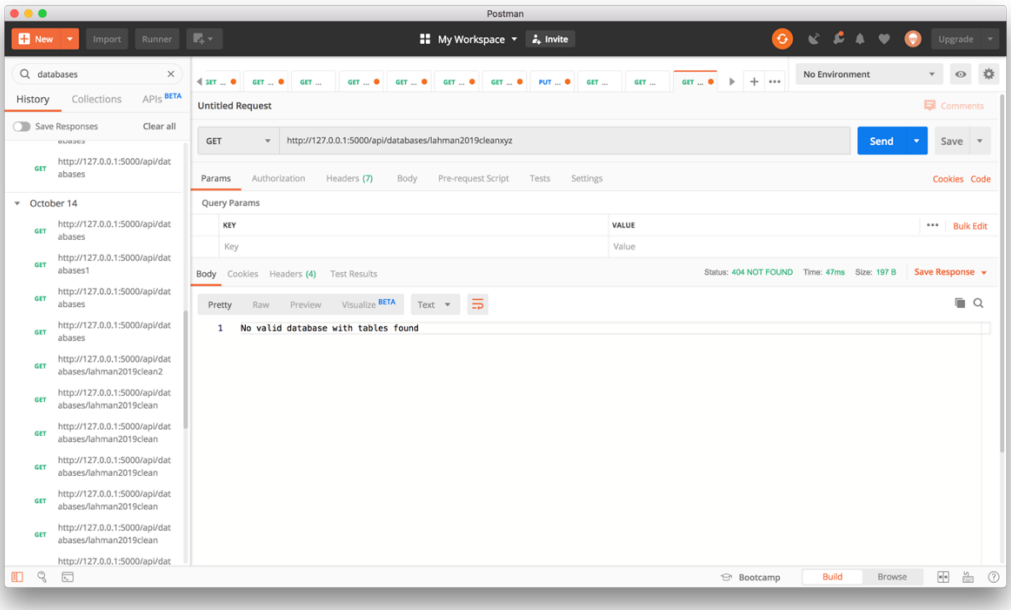
- GET list of tables in the given database. No query parameters passed. The related return link is same as the original link. Returns OK (200) status code. Please see below the screenshots of Postman and also console output







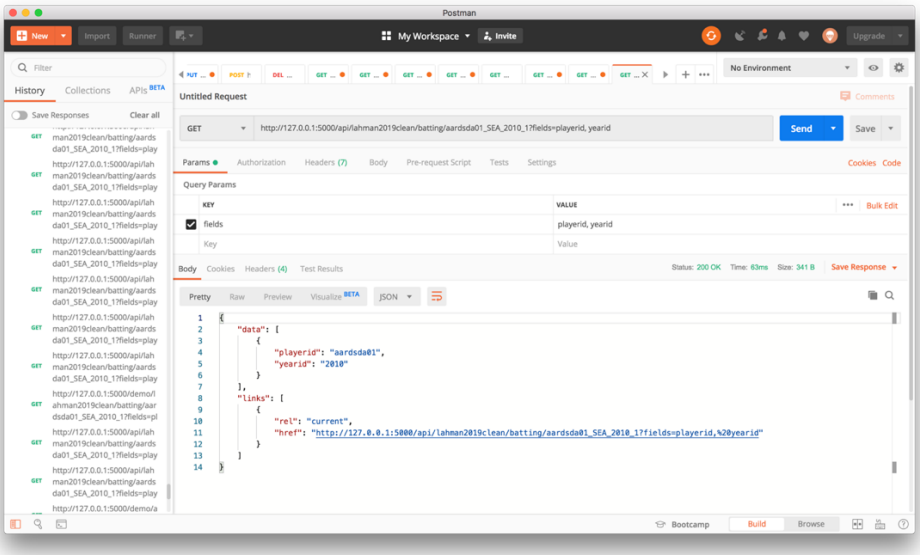
- **Edge Case**: Non-existent schema is passed to fetch the tables. Expectation is to return Not Found Error Message with 404 status code. Please see below the screenshots of Postman

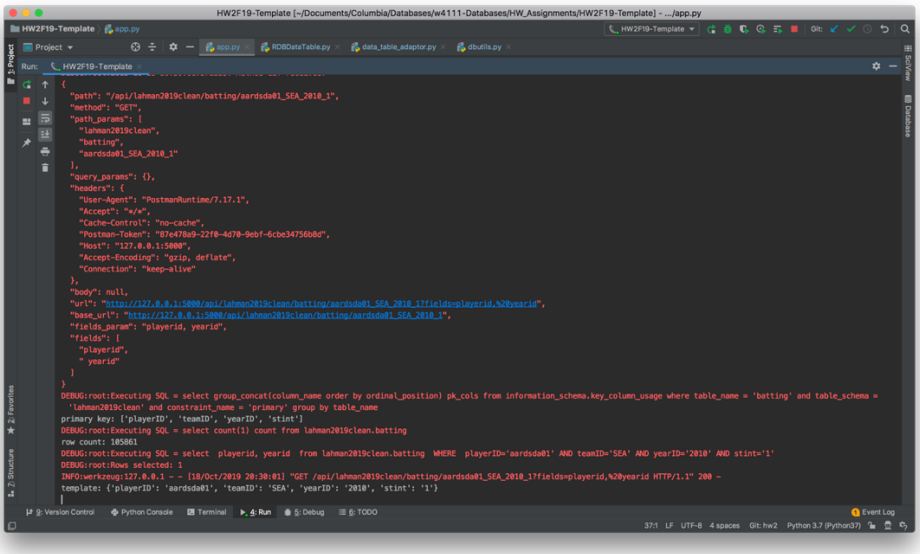



```
3. @application.route('/api/<dbname>/<resource>/<primary_key>', methods=['GET',
   'PUT', 'DELETE'])
```

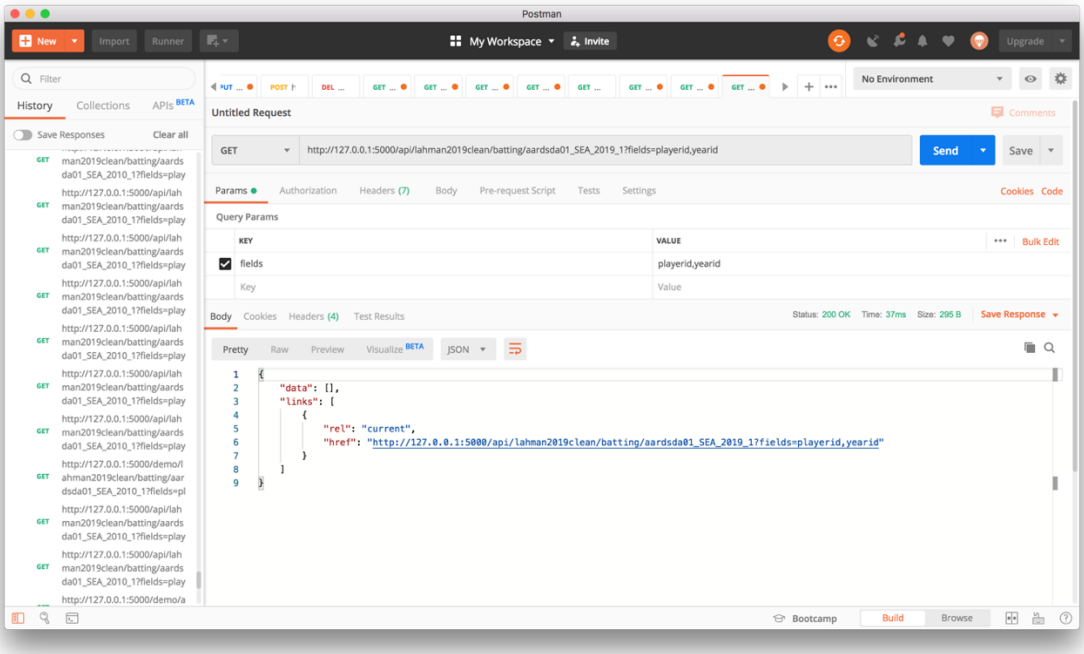
GET Method

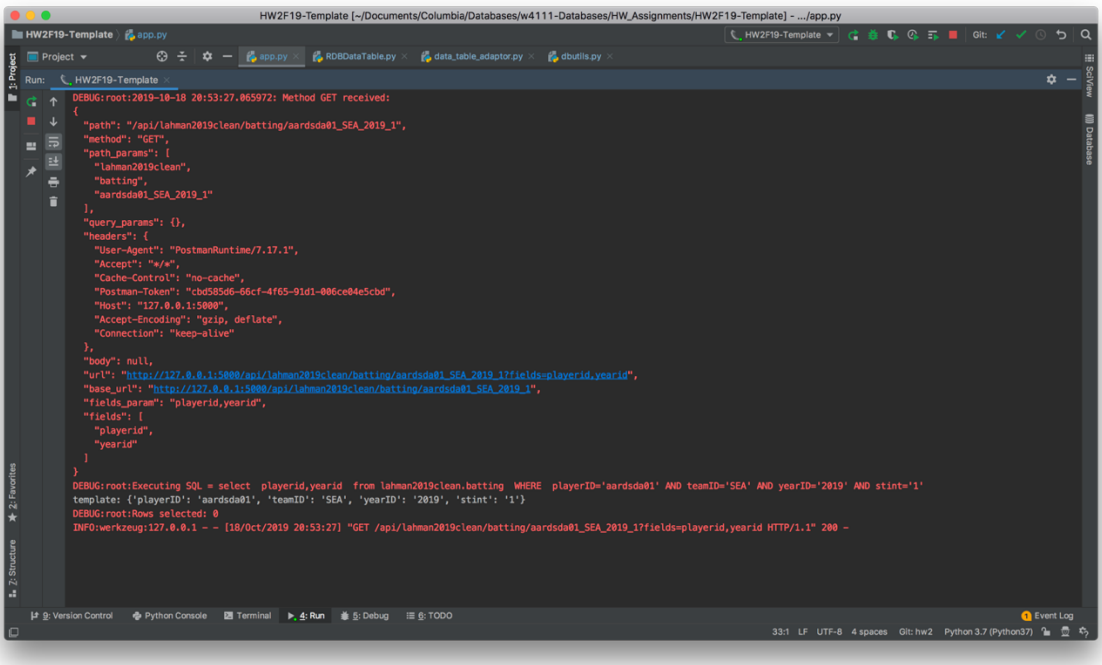
- GET by primary key (aardsda01_SEA_2010_1). Fields to be returned are playerid and yearid. No other query parameters passed. The related return link should be same as the original link. No prev and next links should be present. Returns OK (200) status code. Please see below the screenshots of Postman and also console output. *Test Successful.*



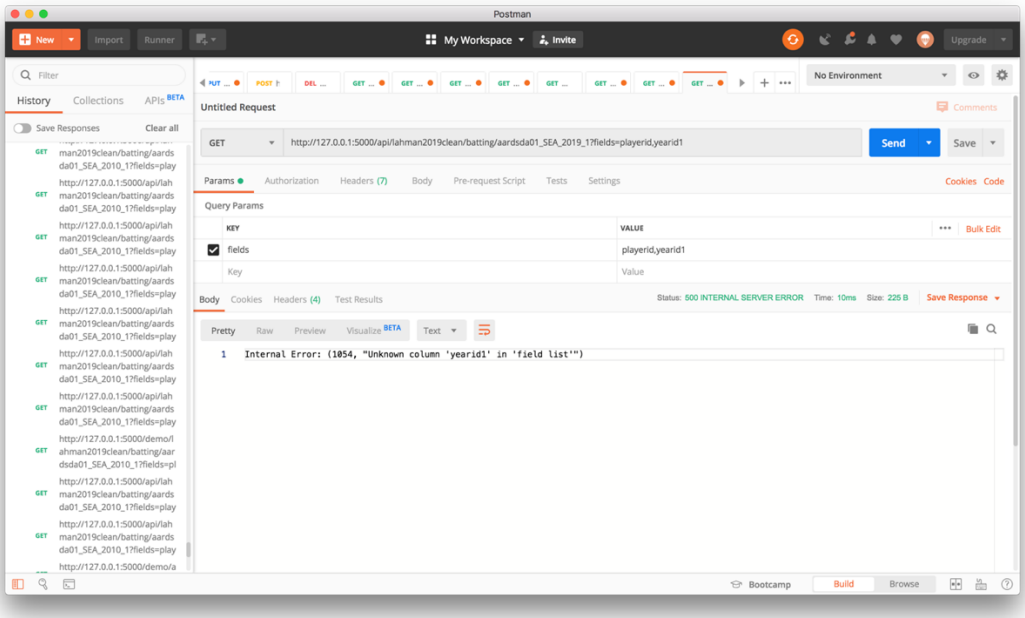


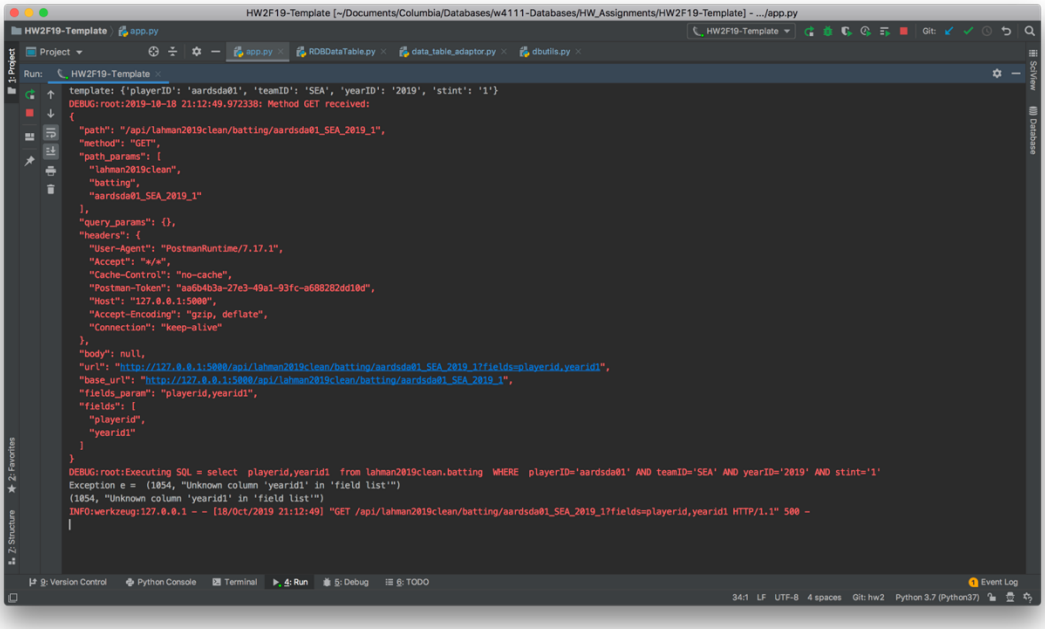
- Edge case: GET by primary key (aardsda01_SEA_2019_1). Zero rows returned by this key. Fields to be returned are playerid and yearid. No other query parameters passed. The related return link should be same as the original link. No prev and next links should be present. Returns OK (200) status code. Please see below the screenshots of Postman and also console output. *Test Successful.*



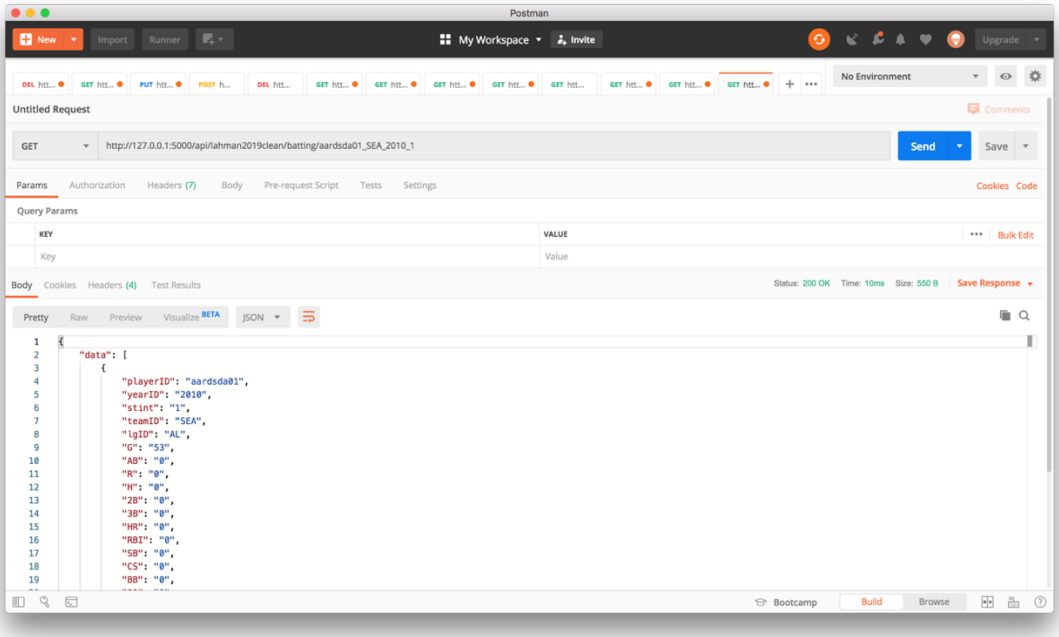


- Negative test case: GET by primary key (aardsda01_SEA_2010_1). Invalid field names to return. Returns Internal Error (500) status code along with the reason of incorrect column name. Please see below the screenshots of Postman and also console output. *Test Successful.*



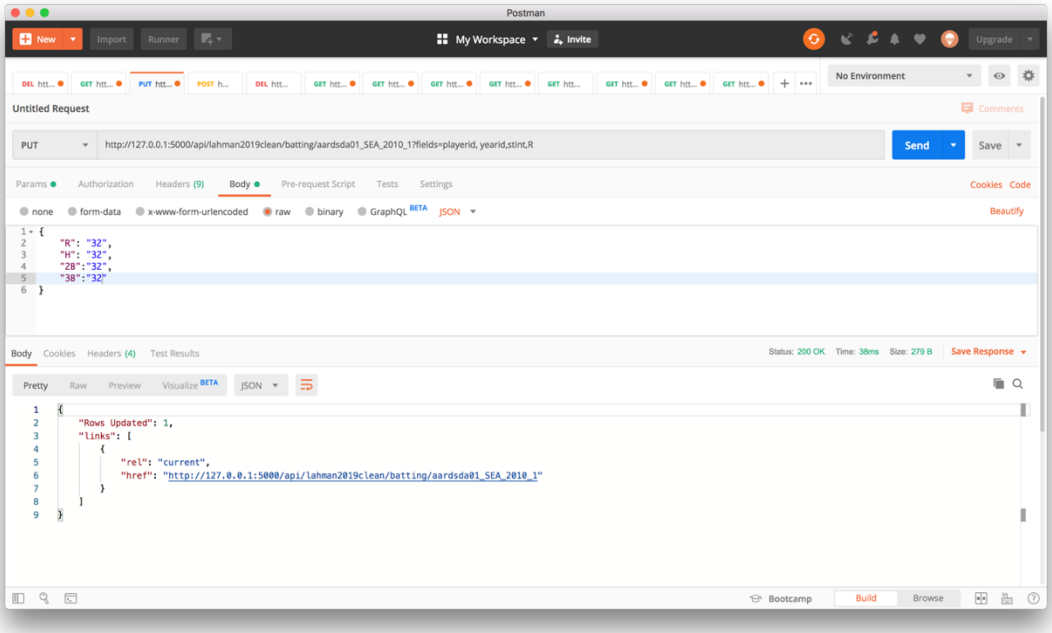


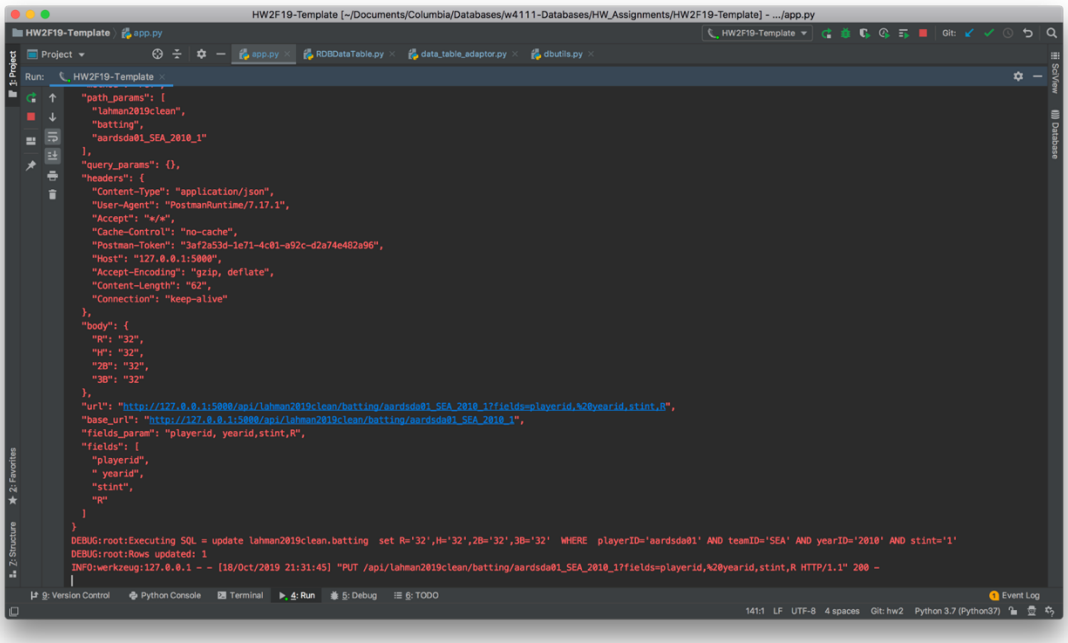
- GET by primary key (aardsda01_SEA_2010_1). No return fields passed explicitly. The related return link should be same as the original link. No prev and next links should be present. Returns OK (200) status code and the data for all fields. Please see below the screenshots of Postman and also console output. *Test Successful*



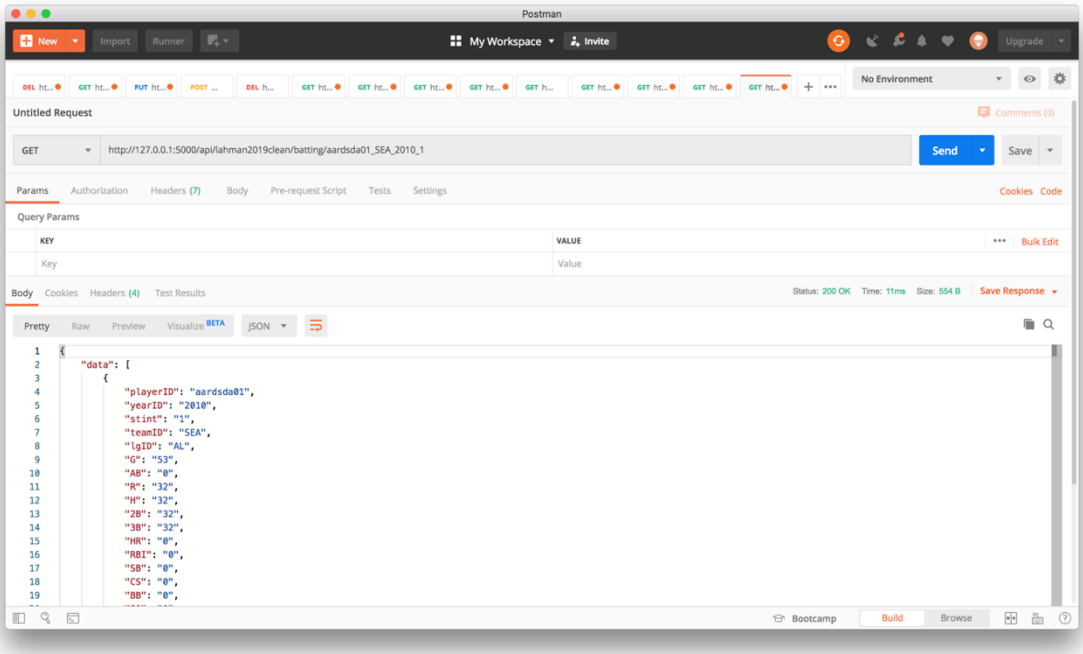
PUT Method

- PUT by primary key. I chose the same batting table. The values to be updated are passed in the body. It should return the number of records updated. The related return link should contain the link with which you can verify the updated (i.e., by primary_key). No prev and next links should be present. Returns OK (200) status code. Please see below the screenshots of Postman and also console output. *Test Successful.*

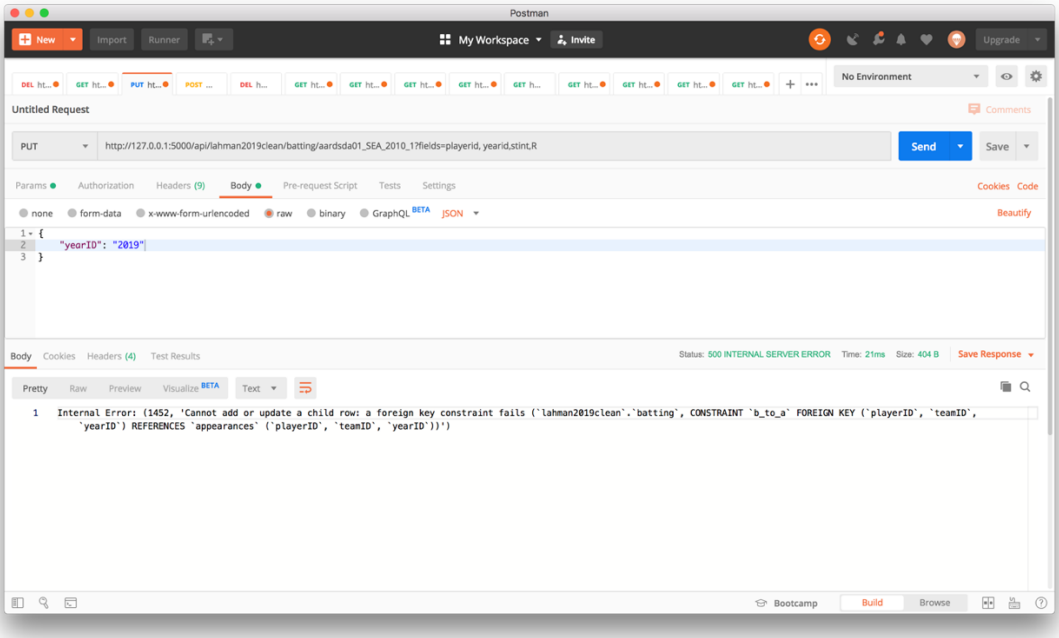


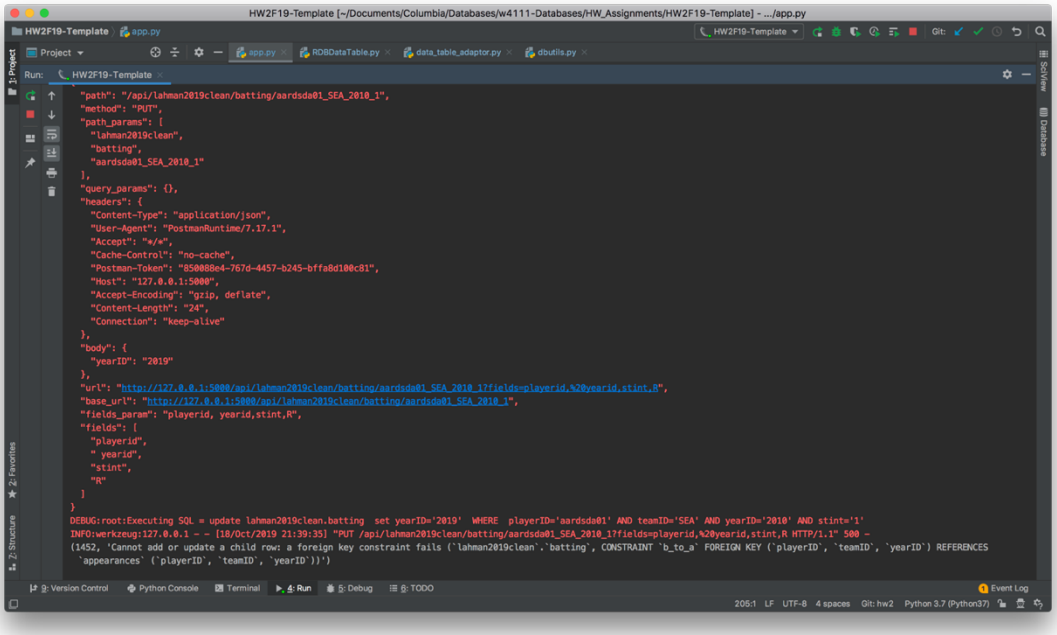


When you click on the related link provide, it will provide the data based on the primary key. Check that the updated values are reflected. *Test Successful*.



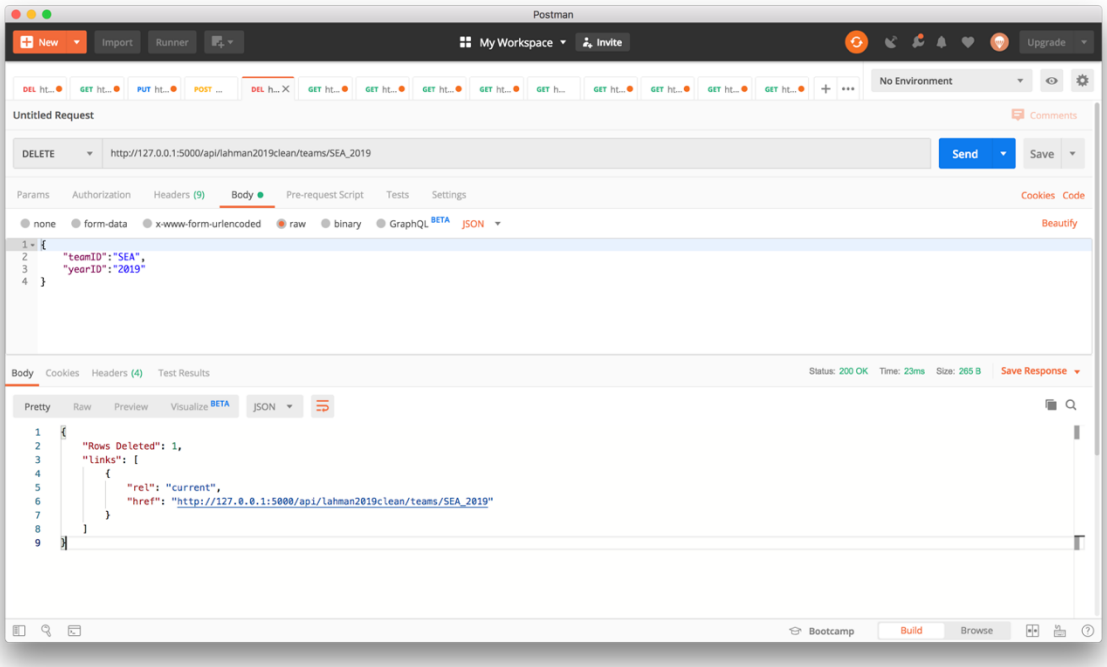
- Negative Test Case: PUT/Update by primary key. I chose the same batting table. Update the values such that it violates key constraints. Returns Internal Error (500) status code along with the reason of Key Violation. Please see below the screenshots of Postman and also console output. *Test Successful.*

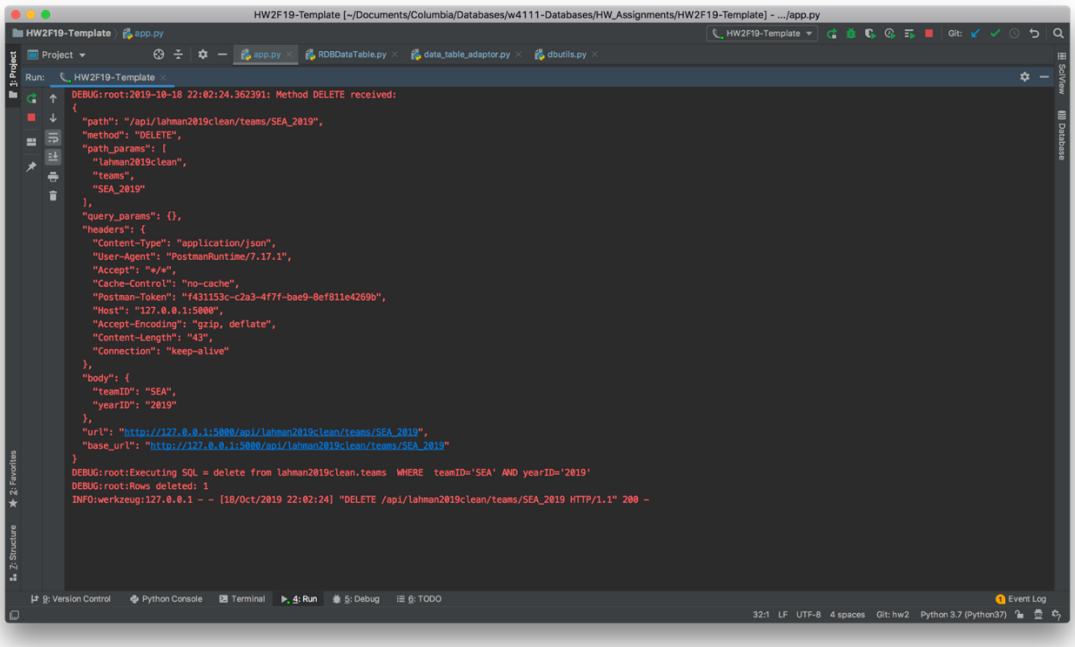




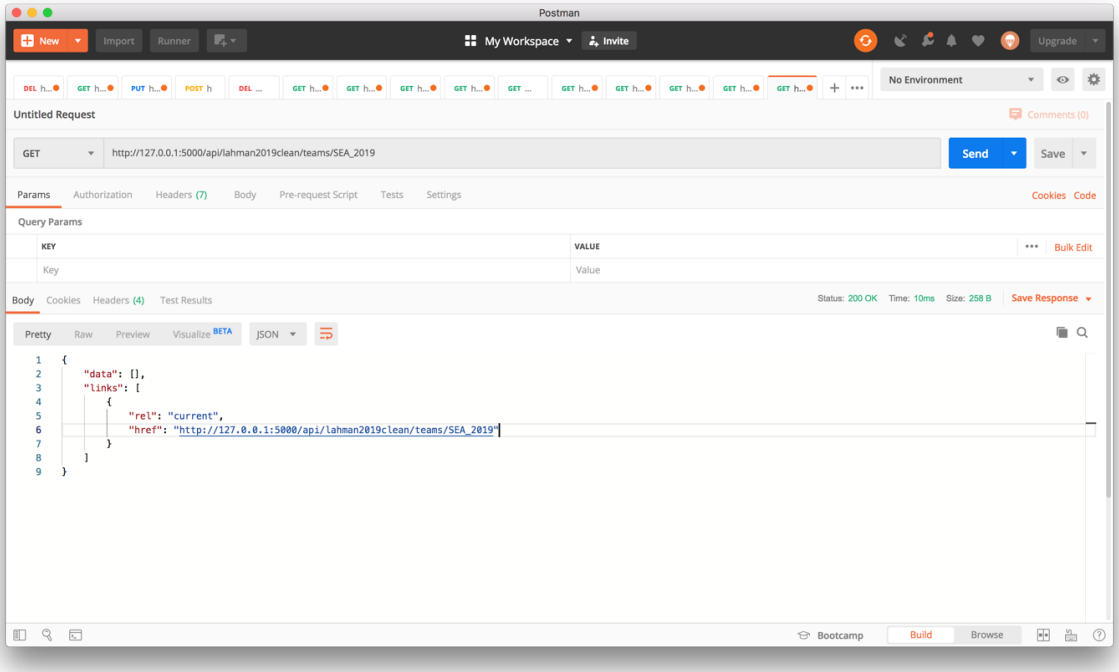
DELETE Method

- DELETE by primary key. The row should not be referenced by another child table. For this purpose, I inserted a record into teams table with teamID = SEA and yearID = 2019. And then deleted the record with this API. Return the number of records deleted (i.e., 1). And also return the link with which the data can validated for that key. *Test Successful.*

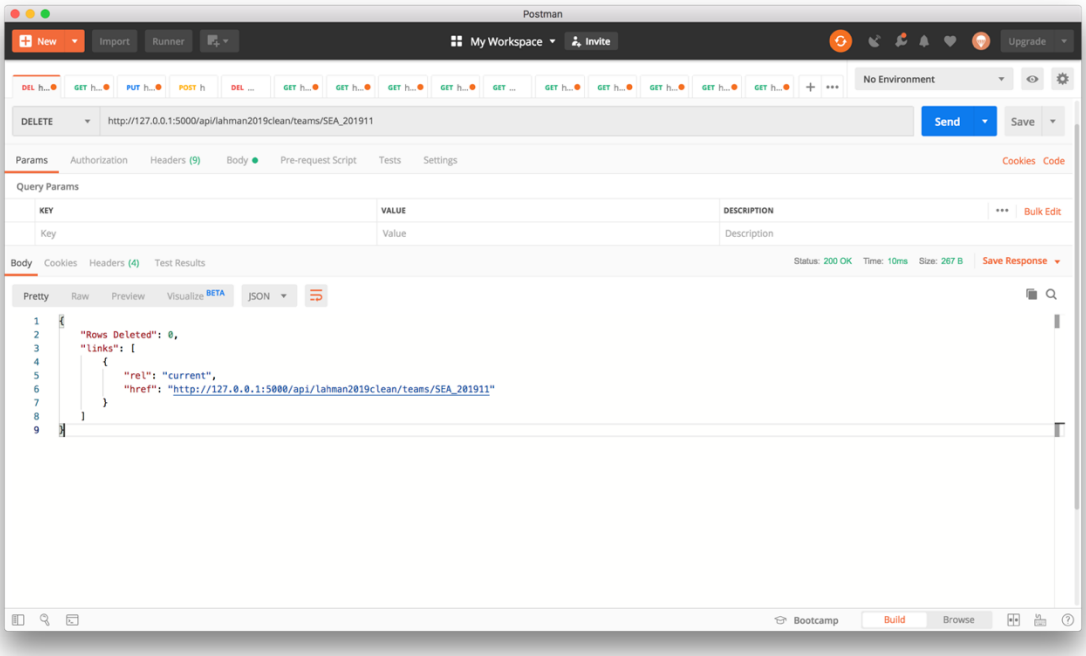




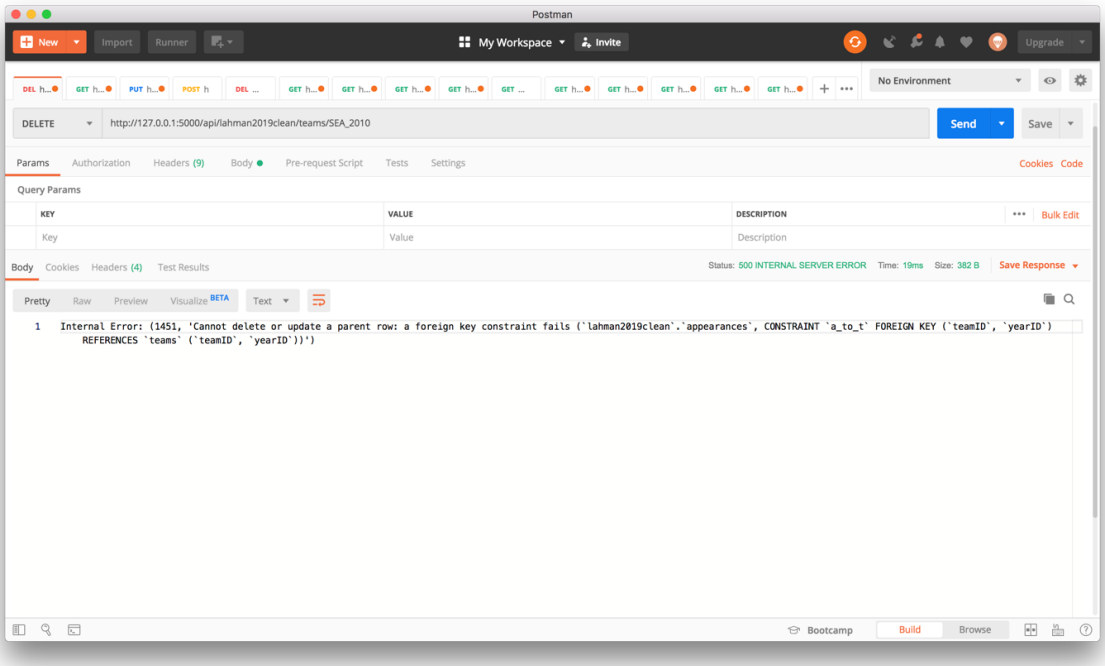
Click on the link to check that the data has been deleted and doesn't exist anymore. *Test Successful.*



- DELETE by non-existent key; It should give 0 rows deleted with OK status code. *Test Successful.*



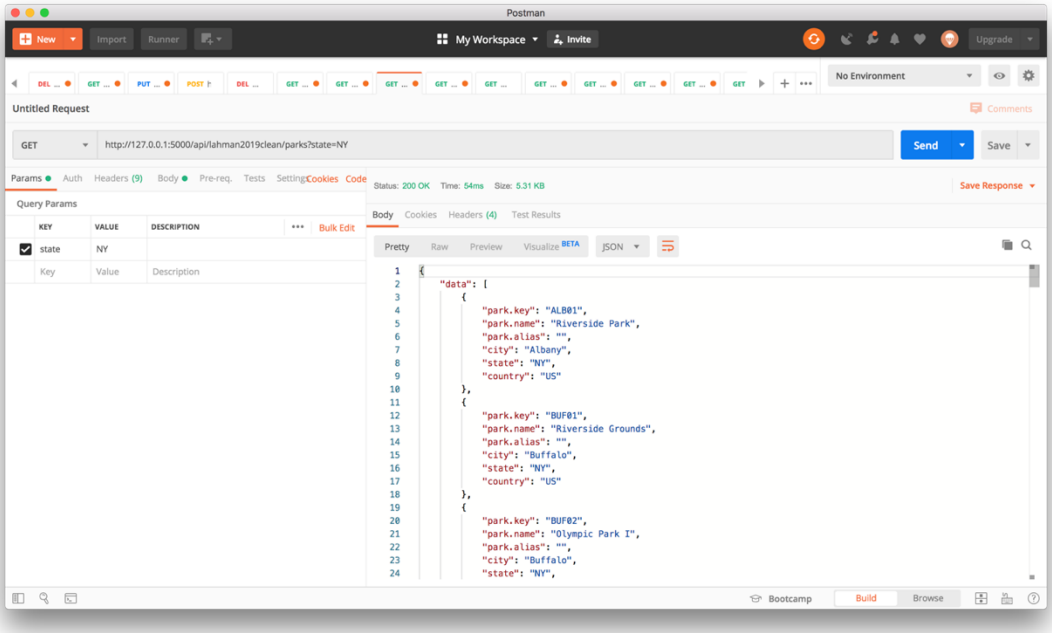
- DELETE a record from the parent table. It should give Internal Error (500) complaining that child is referencing this record. *Test Successful.*

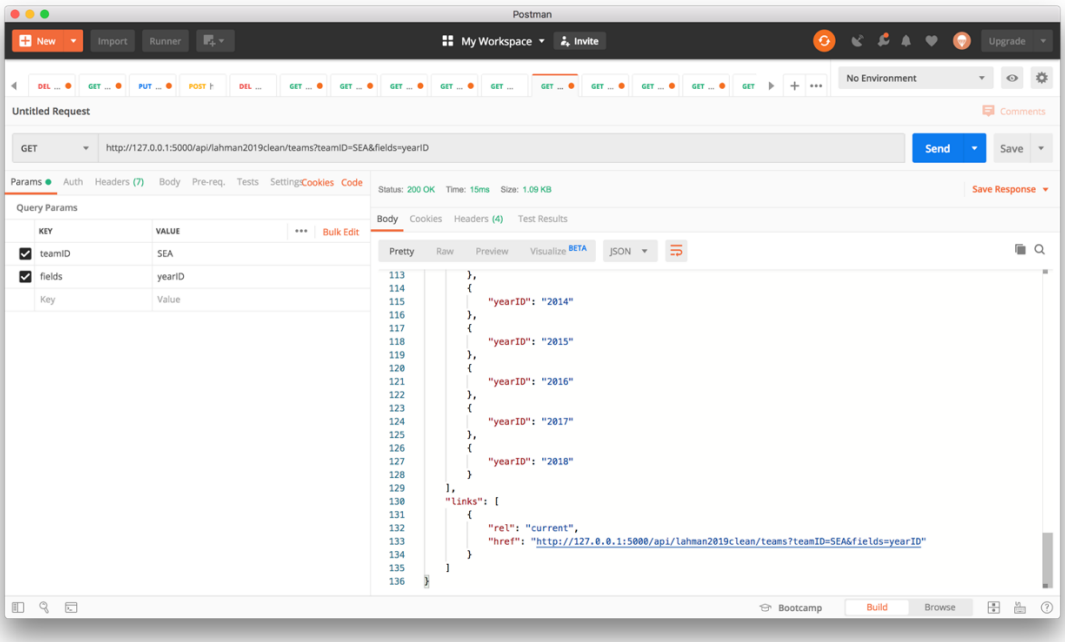


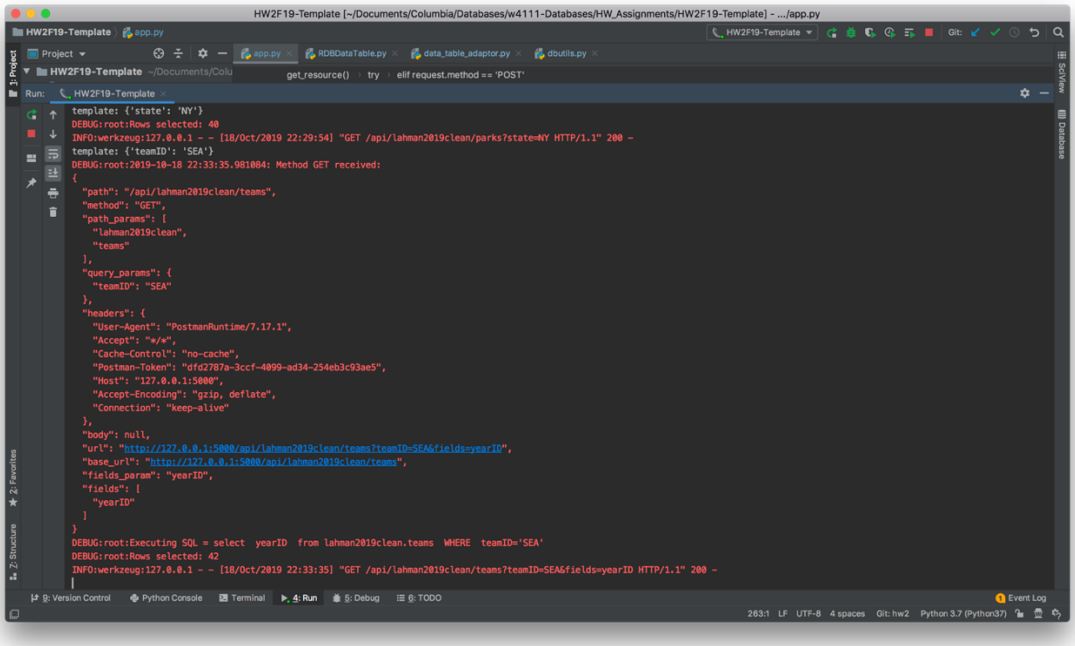

```
4. @application.route('/api/<dbname>/<resource_name>', methods=['GET', 'POST'])
```

GET Method with Pagination

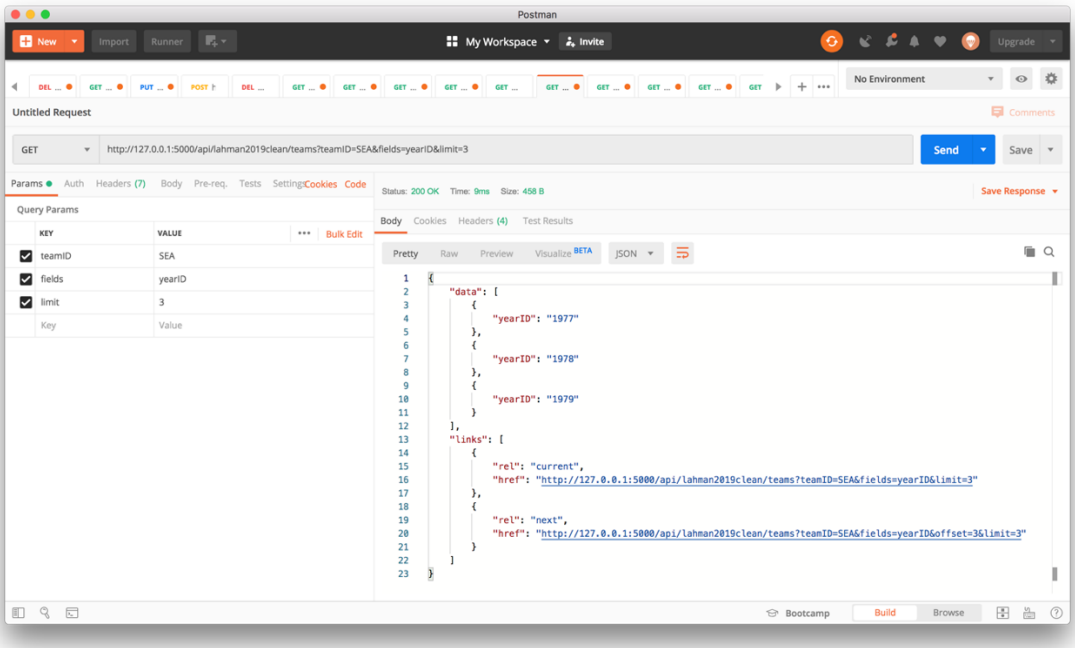
- GET data by template: No pagination/limits. Should return all records and only has one current link (url). Below are two snapshots. One on table parks with no return fields specified. The other on teams with return fields as yearID. *Test Successful.*


Postman
New
Import
Runner
My Workspace
Invite
Upgrade
No Environment
Untitled Request
Comments
GET
http://127.0.0.1:5000/api/lahman2019clean/parks?state=NY
Send
Save
Params
Auth
Headers (9)
Body
Pre-req.
Tests
Settings
Cookies
Code
Status: 200 OK
Time: 54ms
Size: 5.31 KB
Save Response
Query Params
KEY
VALUE
DESCRIPTION
Bulk Edit
state
NY
Key
Value
Description
Body
Cookies
Headers (4)
Test Results
Pretty
Raw
Preview
Visualize BETA
JSON
{
"data": [
{
"park.key": "ALB01",
"park.name": "Riverside Park",
"park.alias": "",
"city": "Albany",
"state": "NY",
"country": "US"
},
{
"park.key": "BUF01",
"park.name": "Riverside Grounds",
"park.alias": "",
"city": "Buffalo",
"state": "NY",
"country": "US"
},
{
"park.key": "BUF02",
"park.name": "Olympic Park I",
"park.alias": "",
"city": "Buffalo",
"state": "NY",
Bootcamp
Build
Browse

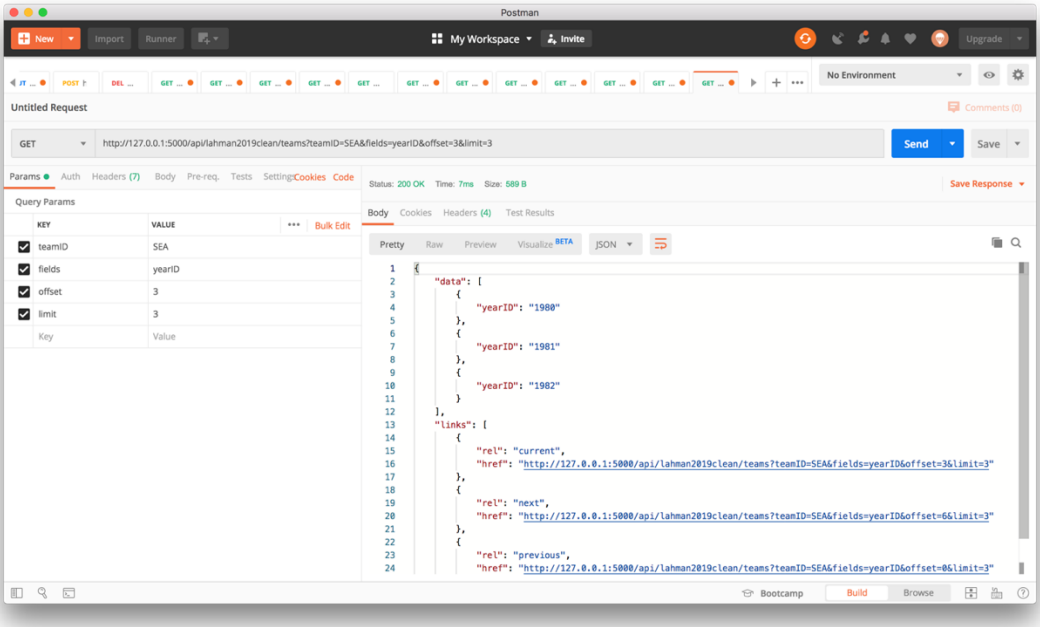

Postman
New
Import
Runner
My Workspace
Invite
Upgrade
No Environment
Untitled Request
Comments
GET
http://127.0.0.1:5000/api/lahman2019clean/teams?teamID=SEA&fields=yearID
Send
Save
Params
Auth
Headers (7)
Body
Pre-req.
Tests
Settings
Cookies
Code
Status: 200 OK
Time: 15ms
Size: 1.09 KB
Save Response
Query Params
KEY
VALUE
Bulk Edit
teamID
SEA
fields
yearID
Key
Value
Body
Cookies
Headers (4)
Test Results
Pretty
Raw
Preview
Visualize BETA
JSON
},
{
"yearID": "2014"
},
{
"yearID": "2015"
},
{
"yearID": "2016"
},
{
"yearID": "2017"
},
{
"yearID": "2018"
}
],
"links": [
{
"rel": "current",
"href": "http://127.0.0.1:5000/api/lahman2019clean/teams?teamID=SEA&fields=yearID"
}
]
}
Bootcamp
Build
Browse

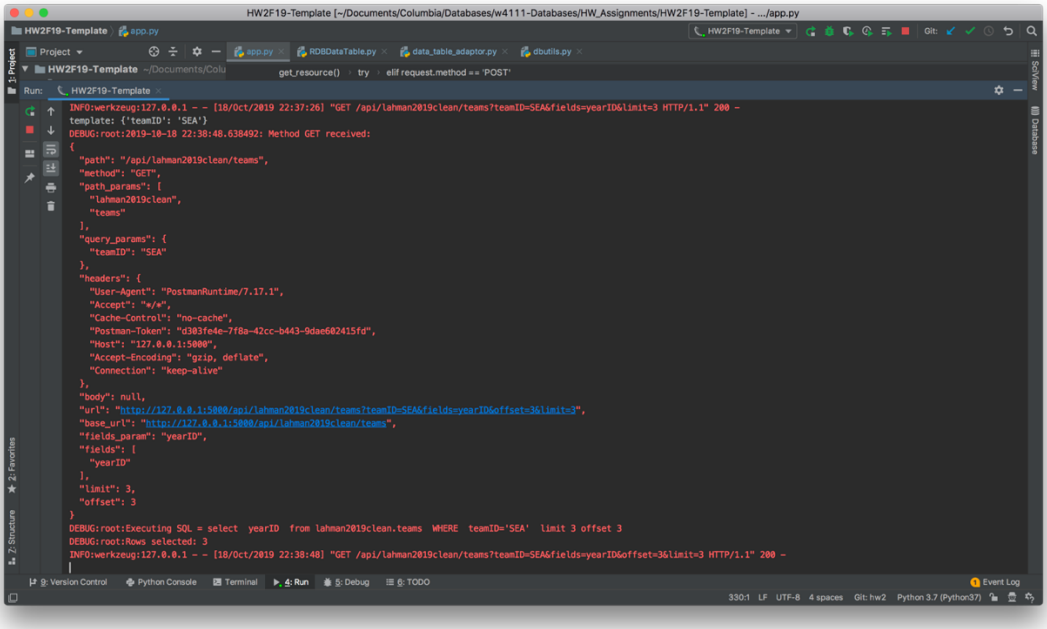


- GET Data by template. This time with a limit of 3 records. Should return only 3 rows and should show the link for next set of data. Should not have previous link. *Test Successful.*

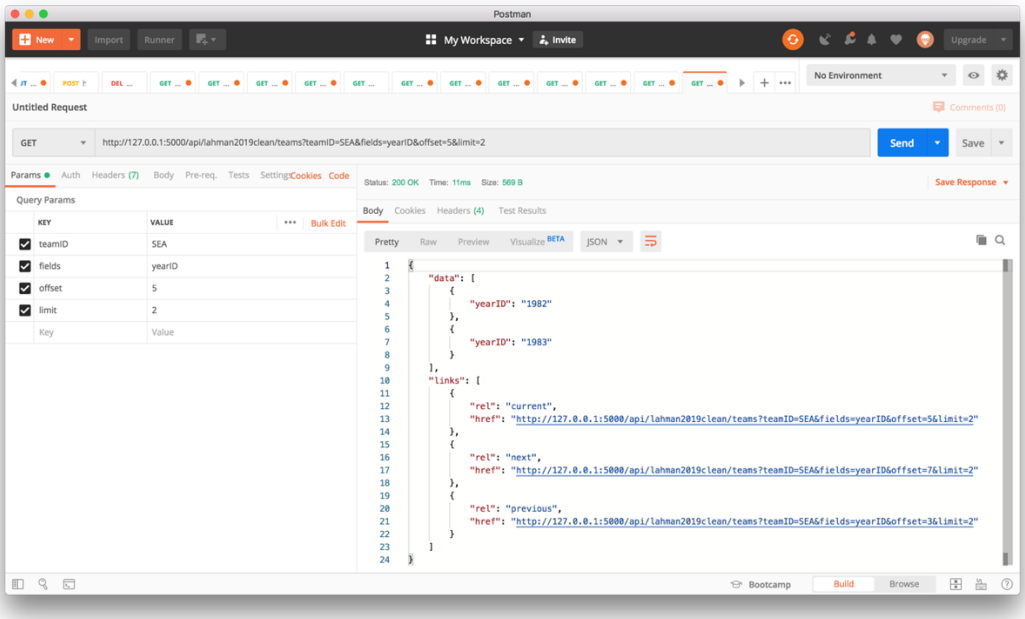


- Now click on the next url link. It should give next three records along with previous and next urls. *Test Successful.*

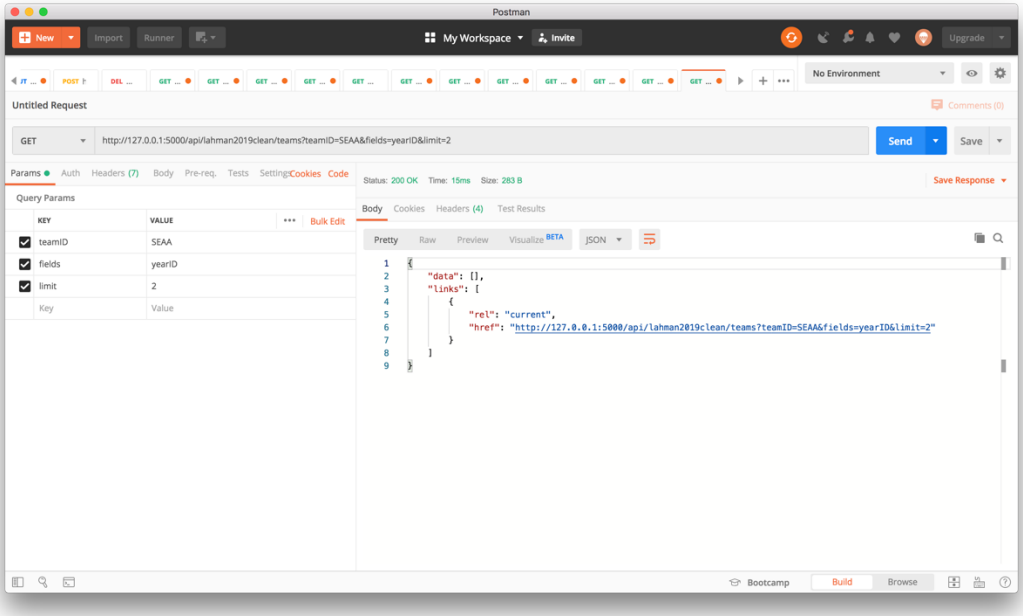




- Edge Case: Now test the GET API with an input on offset and limit. The previous, current and next links should be generated automatically. *Test Successful.*

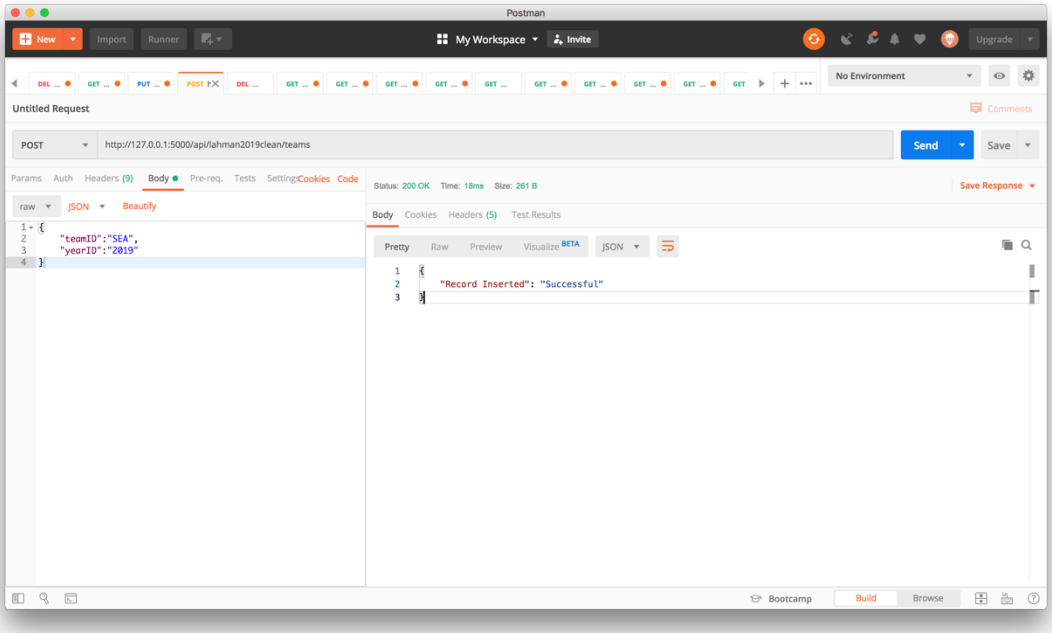


- Edge Case: GET API when the query params provided doesn't retrieve any data. Status code 200, with no data. There shouldn't be next and previous URLs. *Test Successful.*

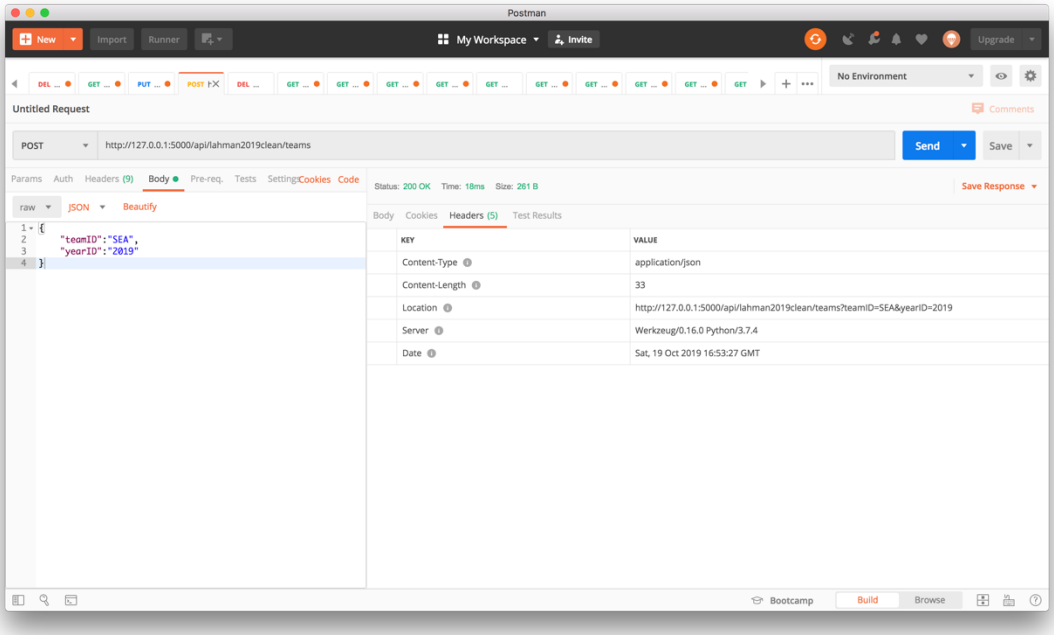


POST Method

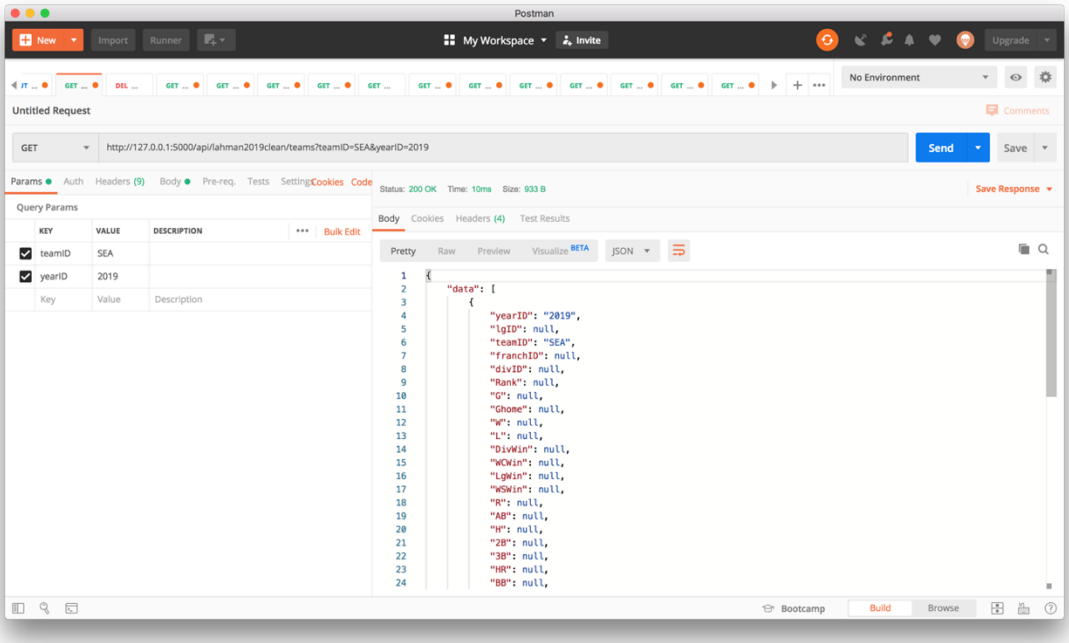
- Insert new record. Json captured in body for new record attributes. Return a message saying Row inserted successfully and also give a url for the inserted record in response header. Return status code is 200 (OK), as mentioned in the HW2 specifications (this could also be done as return code 201) – *Test Successful*.



Now check the response header for the URL. *Test Successful.*



Retrieve data using URL to confirm the data. *Test Successful.*



- Insert a new record which violates the primary key. Should return internal error with the error message. *Test Successful.*