Manual Test Plan for sp21-CS242-assignment 2.2

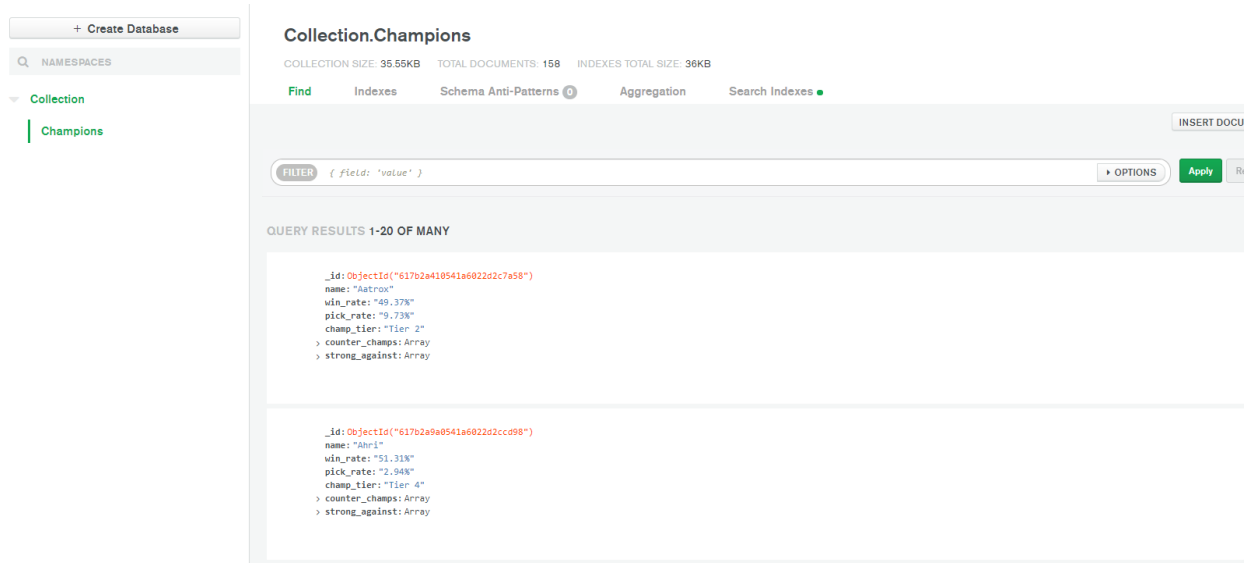
Table of Contents

Environment Setup

- **Python 3.0**
- **unittest**
- **Javascript**
- **HTML**
- **D3 and svg**
- **Eclipse - 4.20.0**
- **Windows 10**

Test : Scraping/Database Handling
Test 1.1 Basic Program Run

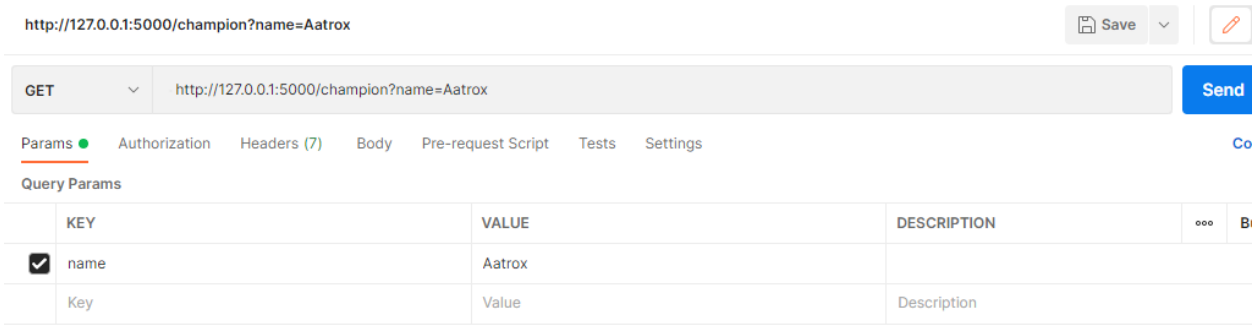
```python
from scraper import Scraper
import time
from database import get_key
from database import database_handler

"""
Implements the scraping of data of all the champions from op.gg
each champions data is based off their current meta role
e.g aatrox meta role is top lane so we scrape data based of his top lane stats
"""
def main():
    s = Scraper()
    arr = s.scrape_champion_links()
    c_counter = 0
    N = len(arr)
    while(c_counter < N):
        retArr = s.scrape_champion_page(arr[c_counter])
        if retArr is None:
            continue
        c_counter+=1;
        time.sleep(10)
        database_handler(retArr)

if __name__ == "__main__":
    main()
```

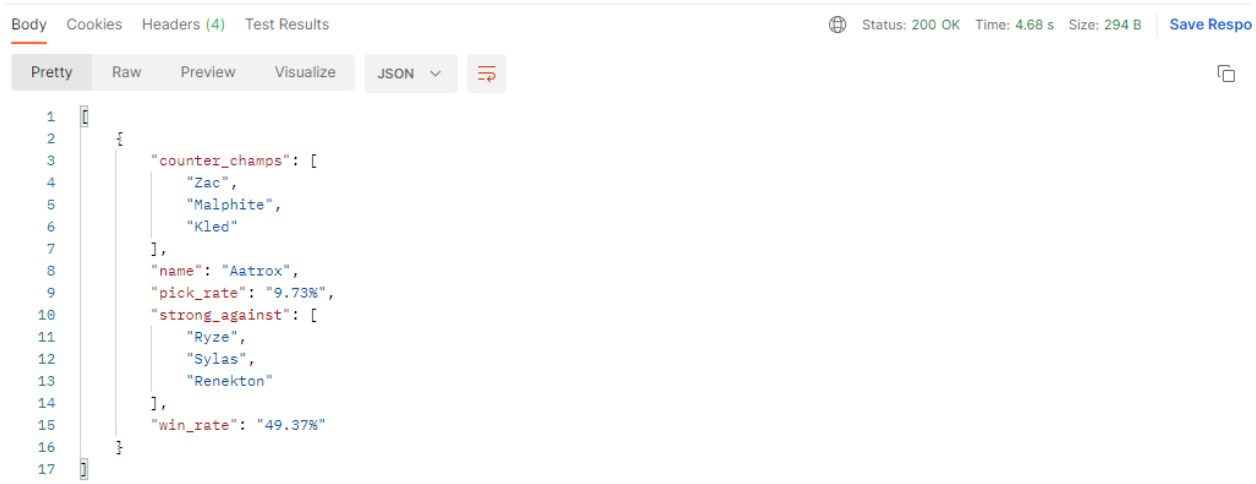


@Test 1.1

- **Users can run the python module ‘scraper_exe’ to execute the Scraper class functionality and push all champions listed on the website op.gg to the MongoDB database.**

Test - API Requests
Test 1.2 - Get Request

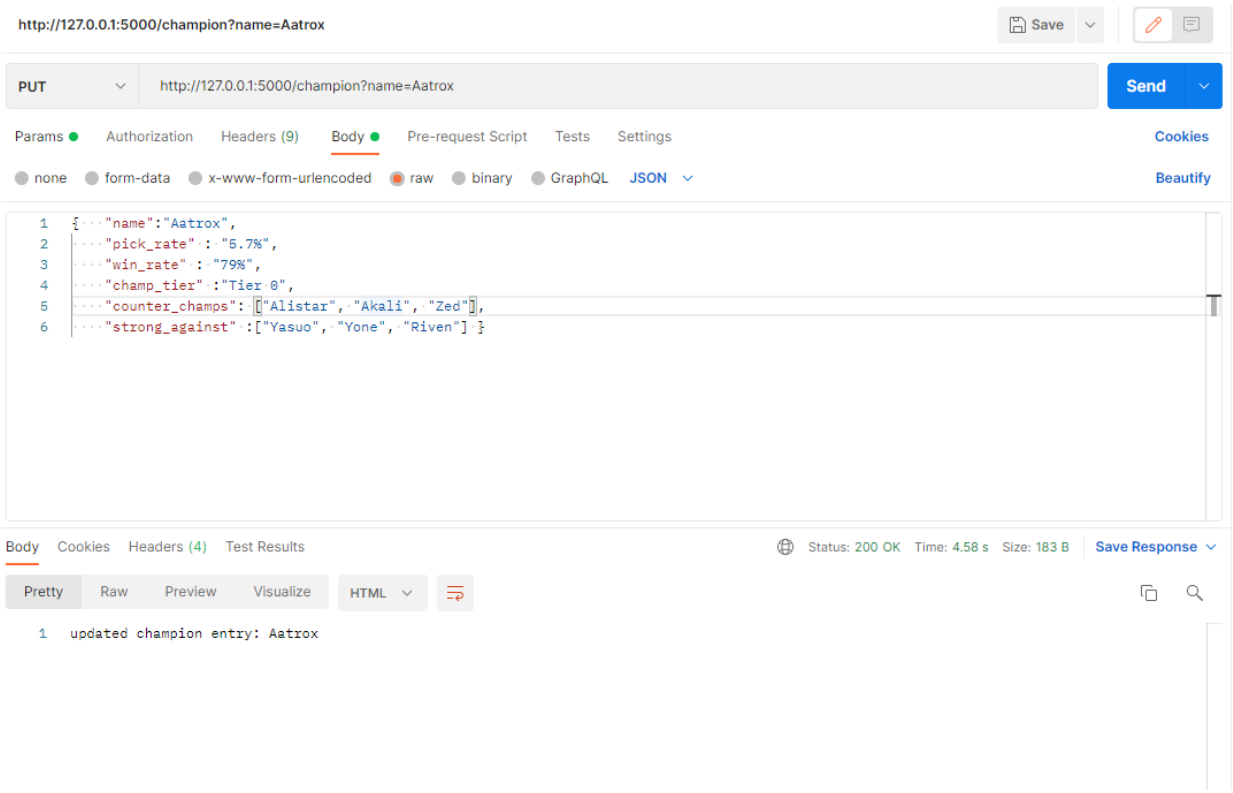




@Test 1.2

- **Users can use the GET API Request by first entering the command ‘set FLASK_APP=src/api’ and then flask run to start the local server.**
- **Specify the Request to be GET and then enter a champion’s name to start the request. Upon a valid name being given, the json object should contain all the fields that are in the MongoDB database.**

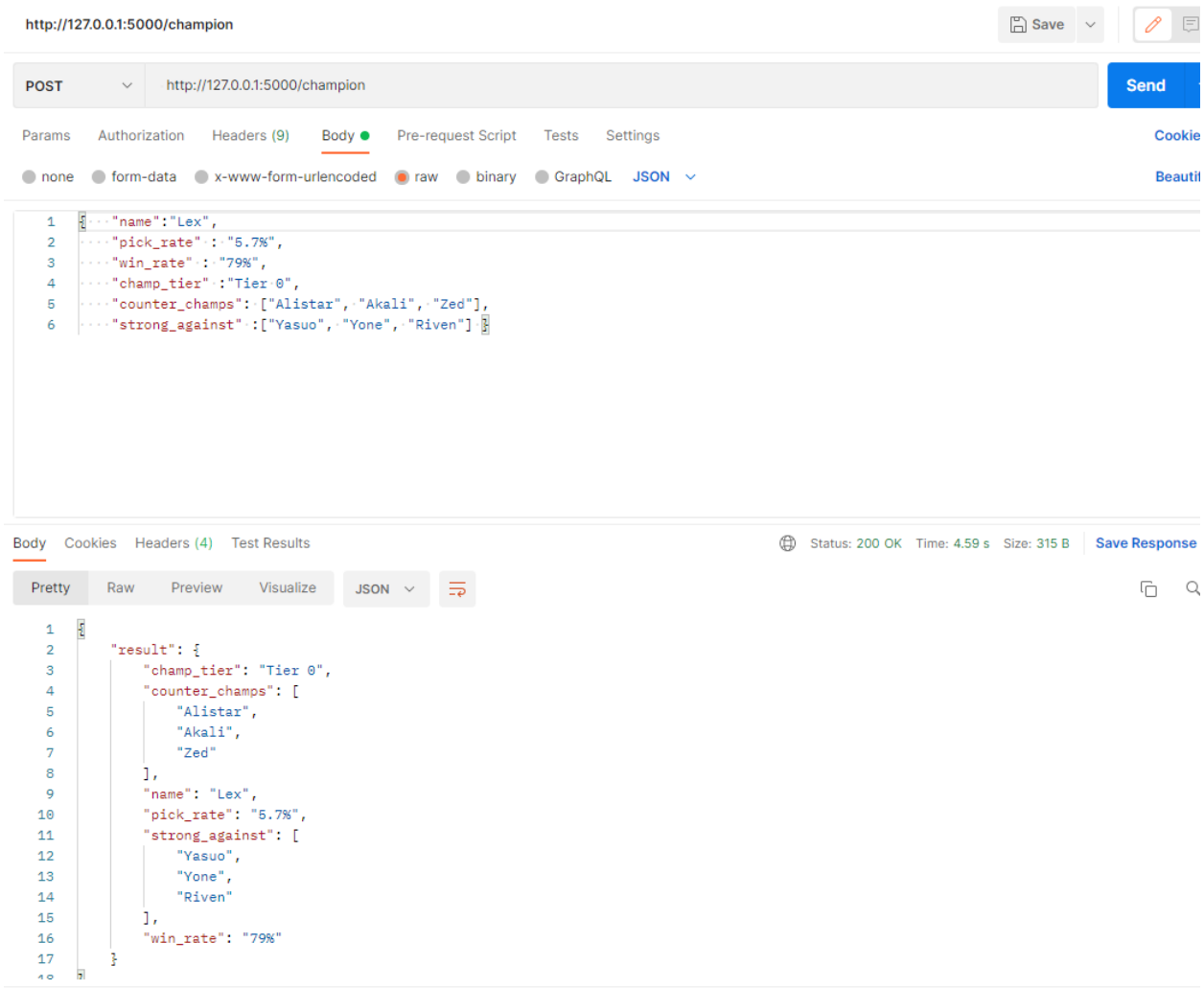
Test 1.3 - Put Request



@Test 1.3

- **Users can use the PUT API Request by first entering the command ‘set FLASK_APP=src/api’ and then flask run to start the local server.**
- **Specify the Request to be PUT and then enter a champion’s name to start the request. Upon a valid name being given along with a valid json object that updates any or all fields, the server will return a message depending on if the request was successful.**

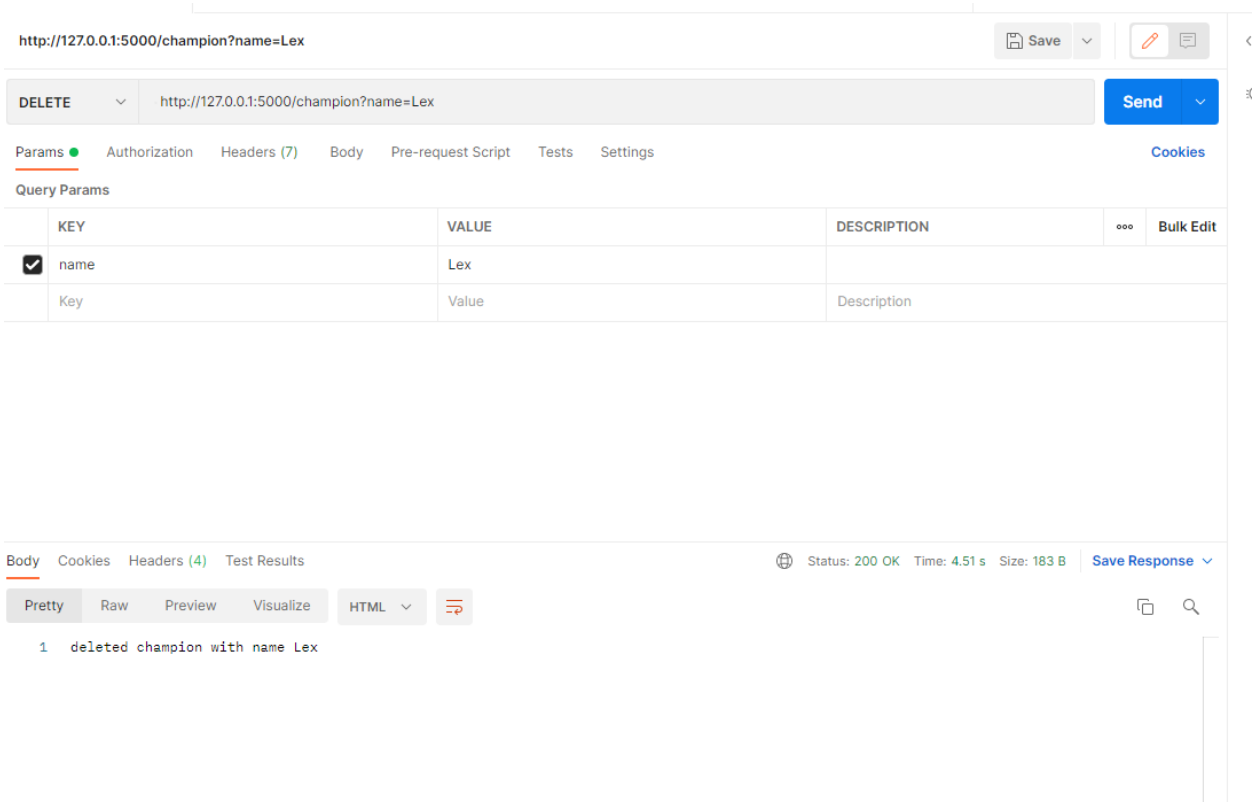
Test 1.4 - Post Request



@Test 1.4

- **Users can use the POST API Request by first entering the command ‘set FLASK_APP=src/api’ and then flask run to start the local server.**
- **Specify the Request to be DELETE and then enter a champion’s name to start the request. Upon a valid name being given, the**

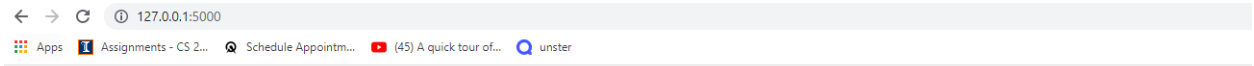
Test 1.4 - Delete Request



@Test 1.4

- **Users can use the DELETE API Request by first entering the command ‘set FLASK_APP=src/api’ and then flask run to start the local server.**
- **Specify the Request to be DELETE and then enter a champion’s name to start the request. Upon a valid name being given, the MongoDB database will delete the specified entry by name and return a message depending on if the request was successful.**

Test - UI
Test 1.5 - Home Page



A League of Legends Statistics Website

Top Champions by Pick Rate | Top Champions by Win Rate | CRUD operations

Champion Name:

Get Champ

Query (Ex : champion.win_rate: > 55):

Get Query

Clear Results

Results Below

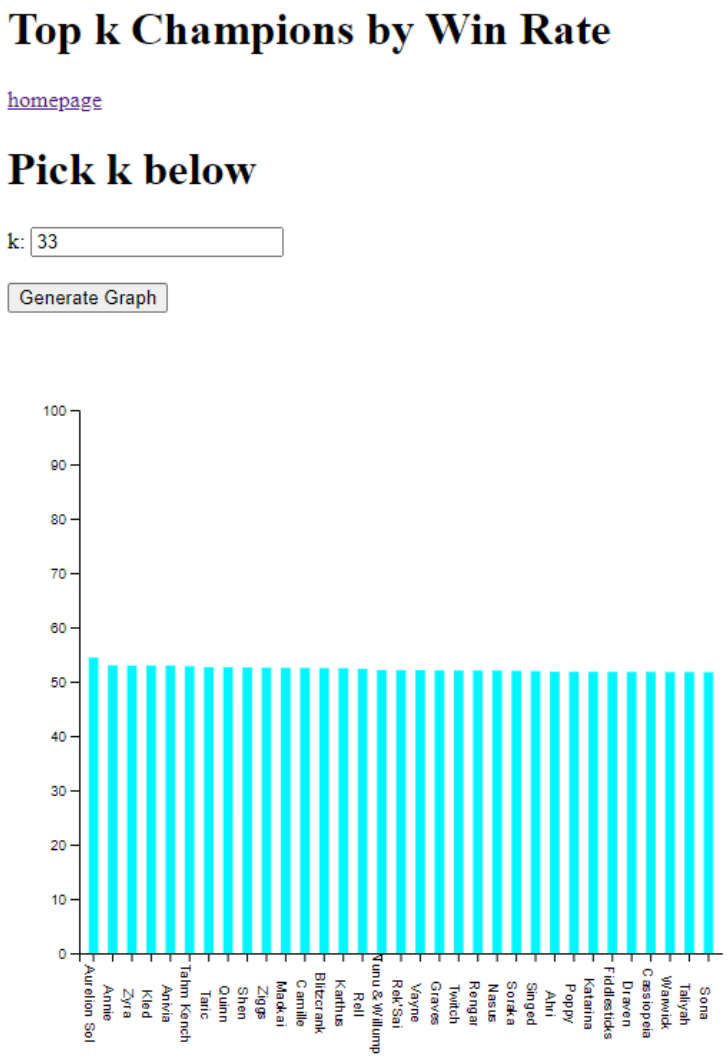
Results Below

name	pick_rate	win_rate	champ_tier	counter_champs	strong_against
Anivia	3.03%	52.94%	Tier 2	Aurelion Sol,Corki,Diana	Ryze,Lucian,Nunu & Willump
Annie	1.09%	52.98%	Tier 3	Neeko,Singed,Diana	Renekton,Garen,Sett
Aurelion Sol	0.67%	54.41%	Tier 3	Fizz,Malzahar,Katarina	Jayce,Azir,Ryze
Blitzcrank	14.65%	52.49%	Tier 1	Leona,Shaco,Taric	Vex,Jarvan IV,Yuumi
Camille	14.22%	52.52%	Tier 1	Teemo,Vayne,Shen	Ryze,Jayce,Yasuo
Graves	18.07%	52.06%	Tier 1	Poppy,Warwick,Sejuani	Qiyana,Zed,Rengar
Karthus	3.29%	52.46%	Tier 2	Nunu & Willump,Rengar,Graves	Udyr,Olaf,Qiyana
Kled	1.86%	52.94%	Tier 2	Singed,Quinn,Urgot	Pantheon,Viktor,Sylas
Maokai	6.95%	52.52%	Tier 1	Taliyah,Rell,Braum	Jarvan IV,Vex,Pantheon
Nasus	1.96%	52.04%	Tier 3	Lillia,Garen,Camille	Pantheon,Zac,Viktor
Nunu & Willump	3.29%	52.13%	Tier 2	Vi,Zac,Warwick	Udyr,Qiyana,Rumble
Quinn	1.14%	52.66%	Tier 2	Maokai,Sion,Singed	Jayce,Trundle,Kled
Rek'Sai	3.8%	52.12%	Tier 2	Shyvana,Karthus,Zac	Ivern,Rengar,Qiyana
Rell	2.02%	52.34%	Tier 2	Veigar,Swain,Soraka	Jarvan IV,Sett,Gragas
Rengar	1.59%	52.04%	Tier 3	Maokai,Pantheon,Zac	Ryze,Jayce,Vayne
Shen	3.34%	52.6%	Tier 2	Viktor,Quinn,Vayne	Cassiopeia,Gragas,Pantheon
Tahm Kench	4.9%	52.81%	Tier 1	Cho'Gath,Kled,Sion	Ryze,Jayce,Gragas
Taric	0.79%	52.66%	Tier 4	Brand,Shaco,Braum	Seraphine,Bard,Karma
Twitch	1.5%	52.05%	Tier 4	Swain,Yasuo,Xerath	Kog'Maw,Aphelios,Sivir
Vayne	8.54%	52.11%	Tier 2	Twitch,Swain,Kog'Maw	Yasuo,Ezreal,Aphelios
Ziggs	5.3%	52.55%	Tier 2	Twitch,Xerath,Sivir	Aphelios,Lucian,Kog'Maw
Zyra	2.03%	52.94%	Tier 3	Shen,Shaco,Blitzcrank	Anivia,Seraphine,Zac

@Test 1.5

- **Home page with default route- User can enter champion by Name or enter a specific query to return a list of champions. The output is in a readable table format. Users can also clear any previous entries with the Clear Results button if needed.**

Test 1.6 - Visualization Pages



@Test 1.6

- **Visualization pages prompt the user to enter a number k (less than 150) to generate a graph of k champions by win rate or pick rate)**

Test 1.7 - CRUD Handler

homepage

CRUD

Champion Name:

Win Rate:

Pick Rate:

Champion Tier:

Counter Champs(e.g Maokai, Yasuo, Zed):

Strong Against Champs(e.g Yone,Yasuo,Malphite):

UPDATE CREATE DELETE SCRAPE

Results

@Test 17

- **Users can specify fields to update, create, or delete. After processing the request, the results will notify whether the request was successful or not.**