**What all I have solved:**

1. Challenge 0 is solved as I have created an API which gives the list of Character.

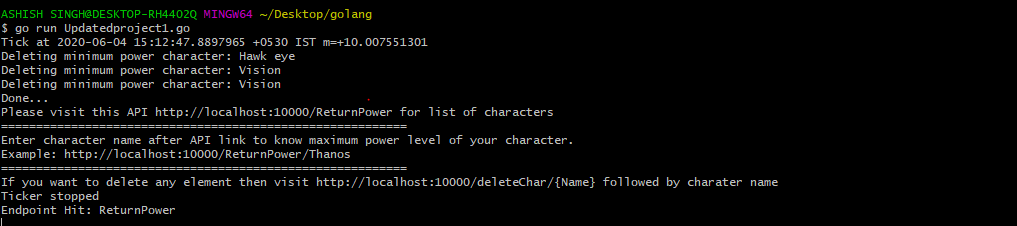
(<http://localhost:10000/ReturnPower>).

1. Called the APIs in parallel and store the list of characters and power level in map (mapdata) data structure. Where name of character is key and power level is set as value.
2. For maximum power level of character use [http://localhost:10000/ReturnPower/{Name}](http://localhost:10000/ReturnPower/%7bName%7d)
3. APIs will be refreshed after every 10 s. I have used Ticker from time package to implement this.
4. Challenge 1- Delete the Characters to free space. If there are more than 15 characters in map then user can delete the least used character. Use [http://localhost:10000/deleteChar/{Name}](http://localhost:10000/deleteChar/%7bName%7d)
5. Used gorilla/mux third party server.
6. Challenge 2- Delete the Characters to free space according to their minimum power level.

* For implementing this I have sort map by value (mapdata)
* For sorting I have used another map hack and after sorting stored values in hack.
* After this I have created a array to store all sorted hackkey.
* As I have calculate there are total 18 characters and our data structure allows only 15
* Therefore I have deleted 3 characters to maintain free space .

1. Deletion statement will also print the character’s name which is being deleted.

**Input and Output:**

****

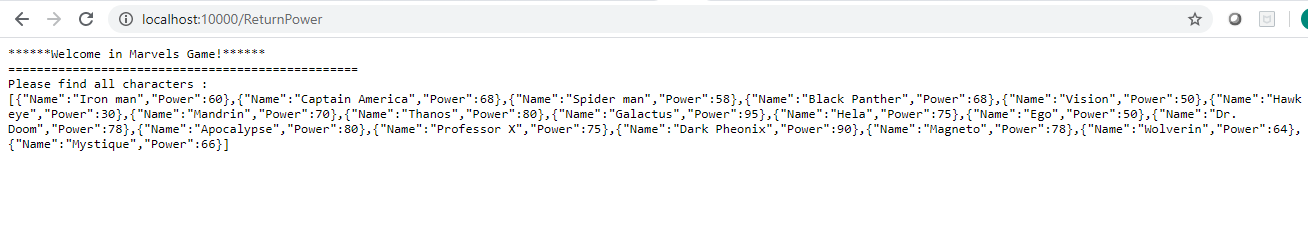
(Input.fig)

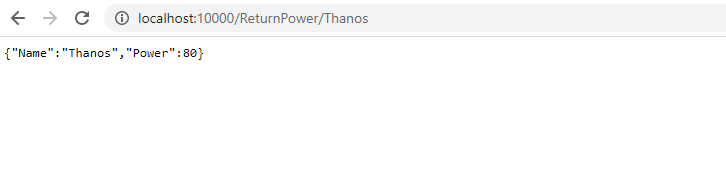
* As shown in (input.fig) first three lines are deleting minimum power characters
* Finally we get the API link which I have created to show the maximum power of character choose by user.

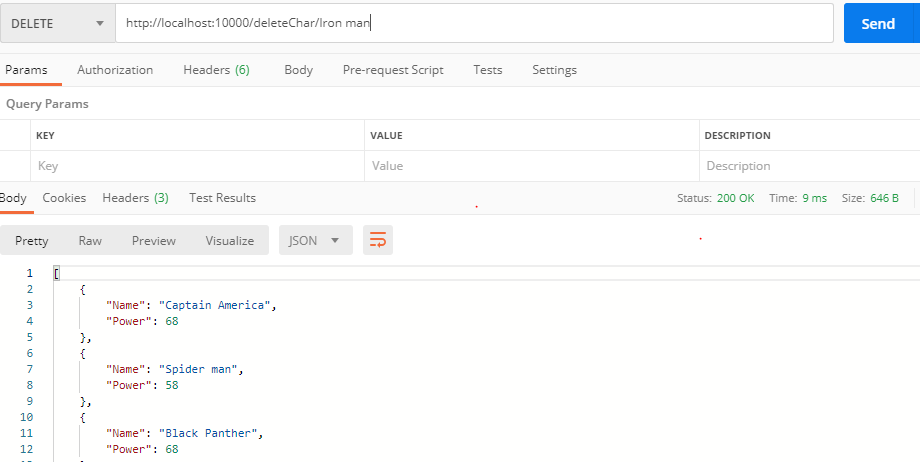
<http://localhost:10000/ReturnPower> - Return list of characters

<http://localhost:10000/ReturnPower/Thanos> - Maximum power

<http://localhost:10000/deleteChar/Thanos> - Deleting character to free space







(API output.fig)