**Modules added:**

* Added “loadash” module for functionality on object array
* Added “node-cron” module for scheduling
* Added “node-fetch” module for fetching data from CricAPI

**Date: 15-Aug-2020 and later:**

* Common mongoose connection with trigger when connected and disconnected in app.js. “db\_connection” global variable available to check if connection exists as true/false. All route modules will user this to check for connection
* Give error status 400 if connection not available
* Heavily optimised for time performance.
* Noticed that Connection to MongoDB taken 5-10 seconds. This need to be investigated.

1. async handler for mongoose “connected” and “disconnected” added
2. The above two handler update “db\_connection”
3. It has been noticed that “connected” gets triggered after 10-15 seconds
4. **Support provided in match module**
5. All matches

/match

1. Match of a given team with others

/match/csk

1. Match between two teams

/match/mi/rr

1. Match on given date

/match/date/<date>

e.g. /match/date/2020-09-11

1. **Support in user module**

Installed “loadash” module in backend. Very useful for object array

**Note: Still not filtering password. Will do it later on**

1. All users

/user

1. Users of group (currently only Group 1 supported)

/user/group or /user/group/1

1. Login, Signup and password reset

/user/login/username/password

/user/sign/username/password

/user/reset/username/password

1. Set captain

/user/captain/userId/PlayerId

1. Set Vice Captain

/user/vicecaptain/userId/PlayerId

**Note:** captain / vice-captain can be selected only once and should be selected before 19th Sep 2020 (IPL start date)

1. Balance amount available to user for auction (only group 1)

/user/balance/userId

1. My teams (players purchased by user)

/user/myteam/<userId> (for single user)

Example: Players purchased by user 4 (command /user/myteam/4)

{"4":

[

{"\_id":"5f338a923d21bb0d50e1ccf4","uid":4,"pid":5334,"gid":1,"bidAmount":400,"\_\_v":0},

{"\_id":"5f36244e4783b416f4814e1a","uid":4,"pid":30045,"gid":1,"bidAmount":230,"\_\_v":0}

]

}

/user/myteam/all (for all users – to be used by admin)

Example: Players purchased by user 4 and 6 (command /user/all)

{"4":

[

{"\_id":"5f338a923d21bb0d50e1ccf4","uid":4,"pid":5334,"gid":1,"bidAmount":400,"\_\_v":0},

{"\_id":"5f36244e4783b416f4814e1a","uid":4,"pid":30045,"gid":1,"bidAmount":230,"\_\_v":0}

],

"6":

[

{"\_id":"5f3675f442a27e0958f8049a","uid":6,"pid":7656,"gid":1,"bidAmount":344,"\_\_v":0}

]

}

1. **Support in group module**
2. Provide user Id of group owner/admin (only group 1 by default supported)

/group/admin (provides user record of admin/owner)

/group/owner (------ as above ------------)

1. List all groups

/group

1. Group close (called by owner once tournament is closed)

/group/<groupNo>/<OwnerId>

e.g. /group/1/9 (close of group 1 by Apurva)

**Note:** This will be called after tournament is over. Only after tournament is over maximum runs and maximum wickets are

1. **Support in auction module**
2. Add auction of player

/auction/add/1/userId/playerId/bidAmount

Auction will be permitted only if user has balance more than “bidAmount”.

1. Show details of players purchased by (all users)

/auction

1. **Support provided in player module**
2. Show all players

/player

1. Show all sold players

/player/sold

1. Show all unsold players

/player/unsold

1. Player available

/player/available/playerId

Returns true if available (not yet sold) else false

1. **Support provided in Statistics (STAT) module**
2. Points scored by each user (based on player performance)

/stat

Dispa

1. Points scored by each user (based on player performance)

/stat/score

Sample output

**[{"uid":8**,"pid":28081,"totalScore":0,"stat":[]},

**{"uid":4**,"pid":927119,"totalScore":0,"stat":[]},

{"**uid":4**,"pid":30045,"totalScore":0,"stat":[]},

{**"uid":6**,"pid":7656,"totalScore":0,"stat":[]},

{"**uid":6**,"pid":823509,"totalScore":0,"stat":[{"\_id":"5f428926f9917f1bc0d5fc23","mid":1227872,"pid":823509,"score":0,"inning":0,"playerName":"GD Phillips  ","run":0,"four":0,"six":0,"fifty":0,"hundred":0,"ballsPlayed":7,"wicket":0,"wicket3":0,"wicket5":0,"hattrick":0,"maiden":0,"oversBowled":0,"manOfTheMatch":false,"\_\_v":0}]},

**{"uid":11**,"pid":230558,"totalScore":9,"stat":[{"\_id":"5f428926f9917f1bc0d5fbe1","mid":1227873,"pid":230558,"score":9,"inning":0,"playerName":"SP Narine","run":8,"four":1,"six":0,"fifty":0,"hundred":0,"ballsPlayed":16,"wicket":0,"wicket3":0,"wicket5":0,"hattrick":0,"maiden":0,"oversBowled":0,"manOfTheMatch":false,"\_\_v":0}]}]

Will provide records of all the players purchased by user.

* uid 🡪 User Id
* pid 🡪 Player Id (purchased by uid during auction)
* totalScore 🡪 Point scored by player (in all the matches)
* stat 🡪 Statistics of all the matches played by player

User player can by adding totalScore of its record

**E.g. var userScore = \_.sumBy(statArray, x => x.totalScore);**

1. There is a scheduler incorporated which will keep on getting match details and statistics from Cricapi. This can also be tested manually calling **/stat/test**.
2. Scheduler is currently set for 15 minuted. Will be change for 2 or 5 minutes after testiing
3. Match statistics will be read once every **3 hours** ( configured in MATCHREADINTERVAL )
4. In match records it sets the end time as “Start Time + 9 hours” for ODI and “Start time + 4 hours” for match which is neither Test nor ODI (assuming it is T20).
5. Match record has start time and end time
6. It will fetch details of match only if
   1. Current time greater than start time
   2. Current time less than end time

Assuming that man of the match will get declared within the given time

**Details for CricApi**: (in playerstat.js)

* Have generated 15 set of API Keys to fetch data from CricApi
* These will be used in rotation whenever the next data is to be fetched from CricAPI