MyBeanJar JavaScript SDK v1.0

Introduction

MyBeanJar (MBJ) is a mobile commerce/game reward platform that distributes targeted digital coupons for game achievements. These rewards are redeemed straight from the MBJ mobile wallet, which prompts redemption via location-alerts, sharing, social networking, and a real-time leaderboard.
For more information about MyBeanJar, visit mybeanjar.com.

Table of contents :

1. About SDK

MyBeanJar JavaScript SDK is a javascript library for mobile html/javascript applications to get/send data from/to MyBeanJar Server. All available requests to Server and their helper functions are located in MBJRequest.js. Example code and description of parameters and return data are in specific chapter for each method.

2. How to add SDK to Project

To add **MyBeanJar JavaScript SDK** to mobile html/javascript project, place **MyBeanJarSDK.js** in sdk folder or another folder of your choise. Assume that you place MyBeanJarSDK.js in sdk folder now you need to add follow line to the head section of your html file.

```
<script type="text/javascript" charset="utf-8" src="sdk/MBJRequest.js"></script>
```

3. MBJRequest Functions

Each described function in next chapters return specific **array of objects, object** or **message** from server and **result** which is one of three description variables :

var INTERNAL_SERVER_ERROR = "Internal Server Error";

var STATUS_SUCCESS = "Status success";

var STATUS_FAIL = "Status fail";

Code below show example request to get data from server (in this example **result** and **array of objects**):

```
function test_winners() {

    callback = function(result,winners) {
        console.log("result of winners is ")
        console.log(result);
        console.log("winners is ");
        console.log(winners);
        console.log("winners - end of callback function");
    };


    get_winners("5", callback);
}
```

note that return data should be processing inside callback method.

Response from this request is :

```
result of winners is                          test.js:9
        Status success test.js:10
        winners is test.js:11
            [Object, Object, Object, Object, Object]
    1.  0: Object
        1.  appkey: "e4da3b7fbbce2345d7772b0674a318d5"
        2.  appname: "MyBeanJar"
        3.  sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/81"
        4.  username: "thedude11"
    5.  __proto__: Object
    2.  1: Object
        1.  appkey: "e4da3b7fbbce2345d7772b0674a318d5"
        2.  appname: "MyBeanJar"
        3.  sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/81"
        4.  username: "thedude11"
    5.  __proto__: Object
    3.  2: Object
        1.  appkey: "e4da3b7fbbce2345d7772b0674a318d5"
        2.  appname: "MyBeanJar"
        3.  sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/71"
        4.  username: "thedude11"
    5.  __proto__: Object
    4.  3: Object
        1.  appkey: "e4da3b7fbbce2345d7772b0674a318d5"
        2.  appname: "MyBeanJar"
        3.  sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/220"
        4.  username: "thedude11"
    5.  __proto__: Object
```

5. 4: Object
    1. appkey: "e4da3b7fbbce2345d7772b0674a318d5"
    2. appname: "MyBeanJar"
    3. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/220"
    4. username: "thedude11"
5. __proto__: Object
    6. length: 5
7. __proto__: Array[0]
    test.js:12
    winners - end of callback function

The result is "Status success". To log for example appkey of object 1 call this line :

```
console.log("appkey of second object is " + winners[1].appkey);
```

At the next points there are example code and their outputs, note that used usernames and passwords,beans and all data pass as parameters aren't correct or cannot be correct (can be change or if someone register them and set same passwords : can be correct). Also return data may be different. Output was tested using correct users and correct bean keys.

Before calls register new user.

3.1. authenticate_user

About authenticate_user function :

Parameters :

username – String

password – String (unencrypted)

callback – function(result,message);

Output :

result – "Status success" or "Status fail" or "Internal Server Error"

message – message from server

Example Code :

```
function test_authenticateuser() {
   log_result = function(which_callback,result,message){
      console.log("callback "+which_callback+". result of authenticateuser is ");
      console.log("callback "+which_callback+". " + result);
      console.log("callback "+which_callback+". message of authenticateuser is ");
      console.log("callback "+which_callback+". " +message);
   }

   callback0 = function(result,message) {
      log_result("0",result,message);
   };

   callback1 = function(result,message) {
      log_result("1",result,message);
   };

   callback2 = function(result,message) {
      log_result("2",result,message);
   };

   callback3 = function(result,message) {
      log_result("3",result,message);
   };

   callback4= function(result,message) {
      log_result("4",result,message);
   };

   authenticate_user("lukasz123", "lukasz123", callback0);
   authenticate_user("lukasz12asfas3", "lukasz123", callback1);
   authenticate_user("lukasz123", "lukaszasfas123", callback2);
   authenticate_user("", "lukasz123", callback3);
   authenticate_user("", "", callback4);
}
```

Output :

```
 callback 2. result of authenticateuser is test.js:134
callback 2. Status fail test.js:135
callback 2. message of authenticateuser is test.js:136
callback 2. Invalid Username and/or Password test.js:137
callback 1. result of authenticateuser is test.js:134
callback 1. Status fail test.js:135
callback 1. message of authenticateuser is test.js:136
callback 1. Invalid Username and/or Password test.js:137
callback 0. result of authenticateuser is test.js:134
callback 0. Status success test.js:135
callback 0. message of authenticateuser is test.js:136
callback 0. successfully authenticated test.js:137
callback 3. result of authenticateuser is test.js:134
callback 3. Status fail test.js:135
callback 3. message of authenticateuser is test.js:136
callback 3. Invalid Username and/or Password test.js:137
callback 4. result of authenticateuser is test.js:134
callback 4. Status fail test.js:135
callback 4. message of authenticateuser is test.js:136
callback 4. Invalid Username and/or Password
```

3.2. delete_bean

About delete_bean function :

Parameters :
username – String

password – String (unencrypted)

beankey – id of beankey to delete.

callback – function(result,message);

Output :

result –   "Status success" or "Status fail" or "Internal Server Error"

message – message from server

Example Code :

```
function test_deletebean(){
  log_result = function(which_callback,result,message){
    console.log("callback "+which_callback+". result of deletebean is ");
    console.log("callback "+which_callback+". " + result);
    console.log("callback "+which_callback+". message of deletebean is ");
    console.log("callback "+which_callback+". " + message);
  }

  callback0 = function(result,message){
    log_result("0",result,message);
  };

  callback1 = function(result,message){
    log_result("1",result,message);
  };

  callback2 = function(result,message){
    log_result("2",result,message);
  };

  callback3 = function(result,message){
    log_result("3",result,message);
  };

  callback4 = function(result,message){
    log_result("4",result,message);
  };

  delete_bean("lukasz123", "lukasz123", "b4681a619cf012318eed690452faeb0e94f", callback0);
  delete_bean("lukasasz1233", "abasdcdef", "b4681a123619cf018eed690452faeb0e94f", callback1);
  delete_bean("123dsaa3", "abcddaef", "b4681213a619cf018eed690452faeb0e94f", callback2);
  delete_bean("asd123fsa", "abcdadef", "b4681adsa619cf018eed690452faeb0e94f", callback3);
}
```

Output :

```
this is deletebean json deleteMethods.js:11
     Object {status: 0, response: Object}
  1. response: Object
    1. message: "Invalid Username and/or Password"
  2. __proto__: Object
```

```
    2. status: 0
 3.  __proto__: Object
    deleteMethods.js:12
    callback 1. result of deletebean is test.js:289
    callback 1. Status fail test.js:290
    callback 1. message of deletebean is test.js:291
    callback 1. Invalid Username and/or Password test.js:292
    this is deletebean json deleteMethods.js:11
      Object {status: 0, response: Object}
 1. response: Object
   1. message: "Invalid Username and/or Password"
 2.  __proto__: Object
   2. status: 0
 3.  __proto__: Object
    deleteMethods.js:12
    callback 3. result of deletebean is test.js:289
    callback 3. Status fail test.js:290
    callback 3. message of deletebean is test.js:291
    callback 3. Invalid Username and/or Password test.js:292
    this is deletebean json deleteMethods.js:11
      Object {status: 0, response: Object}
 1. response: Object
   1. message: "Invalid Username and/or Password"
 2.  __proto__: Object
   2. status: 0
 3.  __proto__: Object
    deleteMethods.js:12
    callback 2. result of deletebean is test.js:289
    callback 2. Status fail test.js:290
    callback 2. message of deletebean is test.js:291
    callback 2. Invalid Username and/or Password test.js:292
    this is deletebean json deleteMethods.js:11
      Object {status: 0, response: Object}
 1. response: Object
   1. message: "Invalid Bean"
 2.  __proto__: Object
   2. status: 0
 3.  __proto__: Object
    deleteMethods.js:12
    callback 0. result of deletebean is test.js:289
    callback 0. Status fail test.js:290
    callback 0. message of deletebean is test.js:291
 callback 0. Invalid Bean
```

3.3. get_apps

About get_apps function :

Parameters :

username – String

password – String (unencrypted)

callback – function(result,apps);

Output :

result – "Status success" or "Status fail" or "Internal Server Error"

apps – array of objects , apps from server.

Example Code :

```javascript
function test_apps() {

  log_result = function(which_callback,result,apps){
    console.log("callback "+which_callback+". result of apps is ");
    console.log("callback "+which_callback+". "+result);
    console.log("callback "+which_callback+". apps is ");
    console.log("callback "+which_callback+". ");
    console.log(apps);
    console.log("callback "+which_callback+". apps - end of callback function");
  }

  callback0 = function(result,apps) {
    log_result("0",result,apps);
  };

  callback1 = function(result,apps) {
    log_result("1",result,apps);
  };

  callback2 = function(result,apps) {
    log_result("2",result,apps);
  };

  get_apps(tokenUser.user, tokenUser.password, "", callback0);
  get_apps(tokenUser.user, tokenUser.password, "3", callback1);
  get_apps(tokenUser.user, tokenUser.password, "1", callback2);
}
```

Output :

callback 1. result of apps is test.js:38
    callback 1. Status success test.js:39
    callback 1. apps is test.js:40
    callback 1. test.js:41
      [Object, Object, Object]
  1. 0: Object
  2. 1: Object
    1. appkey: "a87ff679a2f3e71d9181a67b7542122c"
    2. appstoreurl: "http://itunes.apple.com/us/app/theseus-beanjar-enabled/id475217016?ls=1&mt=8"

3. description: "Avoid the devil and escape the grid alive. **Note: Currently working with only iOS6 or earlier--will be available soon on iOS7."
4. iconurl: "http://api.mybeanjar.com/resources/appicon/4"
5. name: "BeanjarTheseus"
6. publisher: "Jason Fieldman"
7. publisherkey: "c81e728d9d4c2f636f067f89cc14862c"
8. publisherurl: ""
9. __proto__: Object
3. 2: Object
4. length: 3
5. __proto__: Array[0]
test.js:42
callback 1. apps - end of callback function test.js:43
callback 2. result of apps is test.js:38
callback 2. Status success test.js:39
callback 2. apps is test.js:40
callback 2. test.js:41
   [Object]
1. 0: Object
2. length: 1
3. __proto__: Array[0]
test.js:42
callback 2. apps - end of callback function test.js:43
callback 0. result of apps is test.js:38
callback 0. Status success test.js:39
callback 0. apps is test.js:40
callback 0. test.js:41
   [Object, Object, Object]
1. 0: Object
2. 1: Object
3. 2: Object
4. length: 3
5. __proto__: Array[0]
test.js:42
callback 0. apps - end of callback function test.js:43

3.4. get_award

About get_award function :

Parameters :

username – String

password – String (unencrypted)

appkey – id of application

callback – function(result,award);

Output :

result – "Status success" or "Status fail" or "Internal Server Error"

award – object, award from server.

Example Code :

```
function test_award() {

  log_result = function(which_callback,result,award){
    console.log("callback "+which_callback+". result of award is ");
    console.log("callback "+which_callback+". "+result);
    console.log("callback "+which_callback+". award is ");
    console.log(award);
    console.log("callback "+which_callback+". award - end of callback function");
  }

  callback0 = function(result,award) {
    log_result("0",result,award);
  };

  callback1 = function(result,award) {
    log_result("1",result,award);
  };

  callback2 = function(result,award) {
    log_result("2",result,award);
  };

  callback3 = function(result,award) {
    log_result("3",result,award);
  };

  callback4 = function(result,award) {
    log_result("4",result,award);
  };

  get_award("hujemuje", "abcdef", "e4da3b7fbbce2345d7772b0674a318d5", callback0);
  get_award("wrongpasswords", "abcdef", "e4da3b7fbbce2345d7772b0674a318d5", callback1);
  get_award("hujemuje", "abcdef", "wrongappkey", callback2);
```

```
  get_award("hujemuje", "abcdefsfsaf", "e4da3b7fbbce2345d7772b0674a318d5", callback3);
  get_award("hujemuje", "abcdef", "a87ff679a2f3e71d9181a67b7542122c", callback4);
}
```

Output : TODO

3.5. get_beans

About get_beans function :

Parameters :

username – String

password – String (unencrypted)

limit_value – limit value of return beans

sort_by – sort_by which field of beans

callback – function(result, beans);

Output :

result – "Status success" or "Status fail" or "Internal Server Error"

beans – beans from server, array of objects.

Example Code :

```
function test_beans() {
  log_result = function(which_callback,result,beans){
    console.log("callback "+which_callback+". result of beans is ");
    console.log("callback "+which_callback+". " + result);
    console.log("callback "+which_callback+". beans is ");
    console.log(beans);
    console.log("callback "+which_callback+". beans - end of callback function");
  }

  callback0 = function(result,beans) {
    log_result(0,result,beans);
  };
```

```javascript
callback1 = function(result,beans) {
   log_result(1,result,beans);
};


callback2 = function(result,beans) {
   log_result(2,result,beans);
};


callback3 = function(result,beans) {
   log_result(3,result,beans);
};


callback4 = function(result,beans) {
   log_result(4,result,beans);
};


get_beans("hujemuje", "abcdef", "0", "0", callback0);
get_beans("hujkjkemuje", "abcdef", "0", "0", callback1);
get_beans("hujemuje", "abcdekllkf", "0", "0", callback2);
get_beans("hujemuje", "abcdef", "0", "1", callback3);
get_beans("hujemuje", "abcdef", "0", "2", callback4);
}
```

Output :

callback 2. result of beans is test.js:80
     callback 2. Status fail test.js:81
     callback 2. beans is test.js:82
     [] test.js:83
     callback 2. beans - end of callback function test.js:84
     callback 1. result of beans is test.js:80
     callback 1. Status fail test.js:81
     callback 1. beans is test.js:82
     [] test.js:83
     callback 1. beans - end of callback function test.js:84
     callback 0. result of beans is test.js:80
     callback 0. Status success test.js:81
     callback 0. beans is test.js:82
      [Object, Object]

1. 0: Object
   1. beankey: "d25a34b9c2a87db380ecd7f7115882ec"
   2. days: 44
   3. expirationdate: "2014-07-31 00:00:00 -0700"
   4. game: "MyBeanJar"
   5. geocodekey: ""
   6. longdescription: "10% Off $75 or more; promocode-DISCOVER"

7. redeembarcodeurl: "http://api.mybeanjar.com/resources/beanredeemimg/329"
8. redemptionurl: "https://api.mybeanjar.com/resources/beanredeem/5578"
9. redemptionvalidation: 0
10. shortdescription: "10% Off $75 or more; promocode-DISCOVER"
11. sponsorkey: "fe9fc289c3ff0af142b6d3bead98a923"
12. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/83"
13. sponsorname: "Kohl's"
14. sponsorurl: "http://www.kohls.com"
15. wondate: "2014-05-26 00:23:23 -0700"
16. __proto__: Object
2. 1: Object
   1. beankey: "ac53fab47b547a0d47b77e424cf119ba"
   2. days: 197
   3. expirationdate: "2014-12-31 00:00:00 -0800"
   4. game: "MyBeanJar"
   5. geocodekey: ""
   6. longdescription: "20% Off Sitewide PROMOCODE 92586"
   7. redeembarcodeurl: "http://api.mybeanjar.com/resources/beanredeemimg/328"
   8. redemptionurl: "https://api.mybeanjar.com/resources/beanredeem/5577"
   9. redemptionvalidation: 0
   10. shortdescription: "20% Off Sitewide PROMOCODE 92586"
   11. sponsorkey: "13fe9d84310e77f13a6d184dbf1232f3"
   12. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/224"
   13. sponsorname: "Payless Shoe Source"
   14. sponsorurl: "http://www.payless.com"
   15. wondate: "2014-05-26 00:12:59 -0700"
16. __proto__: Object
   3. length: 2
4. __proto__: Array[0]
   test.js:83
   callback 0. beans - end of callback function test.js:84
   callback 3. result of beans is test.js:80
   callback 3. Status success test.js:81
   callback 3. beans is test.js:82
   [Object, Object]
1. 0: Object
   1. beankey: "d25a34b9c2a87db380ecd7f7115882ec"
   2. days: 44
   3. expirationdate: "2014-07-31 00:00:00 -0700"
   4. game: "MyBeanJar"
   5. geocodekey: ""
   6. longdescription: "10% Off $75 or more; promocode-DISCOVER"
   7. redeembarcodeurl: "http://api.mybeanjar.com/resources/beanredeemimg/329"
   8. redemptionurl: "https://api.mybeanjar.com/resources/beanredeem/5578"
   9. redemptionvalidation: 0
   10. shortdescription: "10% Off $75 or more; promocode-DISCOVER"
   11. sponsorkey: "fe9fc289c3ff0af142b6d3bead98a923"
   12. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/83"
   13. sponsorname: "Kohl's"
   14. sponsorurl: "http://www.kohls.com"
   15. wondate: "2014-05-26 00:23:23 -0700"
16. __proto__: Object
2. 1: Object
   1. beankey: "ac53fab47b547a0d47b77e424cf119ba"
   2. days: 197
   3. expirationdate: "2014-12-31 00:00:00 -0800"
   4. game: "MyBeanJar"
   5. geocodekey: ""
   6. longdescription: "20% Off Sitewide PROMOCODE 92586"
   7. redeembarcodeurl: "http://api.mybeanjar.com/resources/beanredeemimg/328"
   8. redemptionurl: "https://api.mybeanjar.com/resources/beanredeem/5577"
   9. redemptionvalidation: 0
   10. shortdescription: "20% Off Sitewide PROMOCODE 92586"

11. sponsorkey: "13fe9d84310e77f13a6d184dbf1232f3"
12. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/224"
13. sponsorname: "Payless Shoe Source"
14. sponsorurl: "http://www.payless.com"
15. wondate: "2014-05-26 00:12:59 -0700"
16. __proto__: Object
    3. length: 2
4. __proto__: Array[0]
test.js:83
callback 3. beans - end of callback function test.js:84
callback 4. result of beans is test.js:80
callback 4. Status success test.js:81
callback 4. beans is test.js:82
    [Object, Object]
1. 0: Object
    1. beankey: "ac53fab47b547a0d47b77e424cf119ba"
    2. days: 197
    3. expirationdate: "2014-12-31 00:00:00 -0800"
    4. game: "MyBeanJar"
    5. geocodekey: ""
    6. longdescription: "20% Off Sitewide PROMOCODE 92586"
    7. redeembarcodeurl: "http://api.mybeanjar.com/resources/beanredeemimg/328"
    8. redemptionurl: "https://api.mybeanjar.com/resources/beanredeem/5577"
    9. redemptionvalidation: 0
    10. shortdescription: "20% Off Sitewide PROMOCODE 92586"
    11. sponsorkey: "13fe9d84310e77f13a6d184dbf1232f3"
    12. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/224"
    13. sponsorname: "Payless Shoe Source"
    14. sponsorurl: "http://www.payless.com"
    15. wondate: "2014-05-26 00:12:59 -0700"
16. __proto__: Object
2. 1: Object
    1. beankey: "d25a34b9c2a87db380ecd7f7115882ec"
    2. days: 44
    3. expirationdate: "2014-07-31 00:00:00 -0700"
    4. game: "MyBeanJar"
    5. geocodekey: ""
    6. longdescription: "10% Off $75 or more; promocode-DISCOVER"
    7. redeembarcodeurl: "http://api.mybeanjar.com/resources/beanredeemimg/329"
    8. redemptionurl: "https://api.mybeanjar.com/resources/beanredeem/5578"
    9. redemptionvalidation: 0
    10. shortdescription: "10% Off $75 or more; promocode-DISCOVER"
    11. sponsorkey: "fe9fc289c3ff0af142b6d3bead98a923"
    12. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/83"
    13. sponsorname: "Kohl's"
    14. sponsorurl: "http://www.kohls.com"
    15. wondate: "2014-05-26 00:23:23 -0700"
16. __proto__: Object
    3. length: 2
4. __proto__: Array[0]
test.js:83
callback 4. beans - end of callback function

3.6. get_categories

About get_categories function :

Parameters :

username – String

password – String (unencrypted)

callback – function(result, beans);

Output :

result – "Status success" or "Status fail" or "Internal Server Error"

categories – categories from server, array of objects.

Example Code :

```javascript
function test_categories() {
  log_result = function(which_callback,result,categories){
    console.log("callback "+which_callback+". result of categories is ");
    console.log("callback "+which_callback+". "+result);
    console.log("callback "+which_callback+". categories is ");
    console.log(categories);
    console.log("callback "+which_callback+". categories - end of callback function");


  }

  callback0 = function(result,categories) {
    log_result("0",result,categories);
  };

  callback1 = function(result,categories) {
    log_result("1",result,categories);
  };

  callback2 = function(result,categories) {
    log_result("2",result,categories);
  };

  callback3 = function(result,categories) {
    log_result("3",result,categories);
  };

  callback4 = function(result,categories) {
    log_result("4",result,categories);
```

```
    };

    get_categories(tokenUser.user, tokenUser.password, callback0);

    get_categories("wrongusername", tokenUser.password, callback1);

    get_categories(tokenUser.user, "wrongpassword", callback2);

    get_categories("wrongusername", "wrongpassword", callback3);

}
```

Output :

callback 1. result of categories is test.js:116
 callback 1. Status fail test.js:117
 callback 1. categories is test.js:118
 [] test.js:119
 callback 1. categories - end of callback function test.js:120
 callback 3. result of categories is test.js:116
 callback 3. Status fail test.js:117
 callback 3. categories is test.js:118
 [] test.js:119
 callback 3. categories - end of callback function test.js:120
 callback 2. result of categories is test.js:116
 callback 2. Status fail test.js:117
 callback 2. categories is test.js:118
 [] test.js:119
 callback 2. categories - end of callback function test.js:120
 callback 0. result of categories is test.js:116
 callback 0. Status success test.js:117
 callback 0. categories is test.js:118
  [Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object]

1. 0: Object
2. 1: Object
3. 2: Object
   1. categorykey: "eccbc87e4b5ce2fe28308fd9f2a7baf3"
   2. name: "Food and Beverage"
3. __proto__: Object
4. 3: Object
5. 4: Object
6. 5: Object
7. 6: Object
8. 7: Object
9. 8: Object
10. 9: Object
11. 10: Object
12. 11: Object
13. 12: Object
14. 13: Object
15. 14: Object
16. 15: Object
17. 16: Object
18. 17: Object
19. 18: Object
 20. length: 19
21. __proto__: Array[0]
 test.js:119
 callback 0. categories - end of callback function

3.7. get_location_for_sponsor

About get_location_for_sponsor function : TODO
Example Code : TODO
Output : TODO

3.8. get_sponsors

About get_sponsors function :

Parameters :

username – String

password – String (unencrypted)

limit_value - limit value of return sponsors

callback – function(result, beans);

Output :

result – "Status success" or "Status fail" or "Internal Server Error"

sponsors – sponsors from server, array of objects.

Example Code :

```
function test_sponsors() {

  log_result = function(which_callback,result,sponsors){
    console.log("callback "+which_callback+". result of sponsors is ");
    console.log("callback "+which_callback+". "+result);
    console.log("callback "+which_callback+". sponsors is ");
    console.log(sponsors);
    console.log("callback "+which_callback+". sponsors - end of callback function");
  }


  callback0 = function(result,sponsors) {
```

```javascript
        log_result("0",result,sponsors);
    };


    callback1 = function(result,sponsors) {
        log_result("1",result,sponsors);
    };


    callback2 = function(result,sponsors) {
        log_result("2",result,sponsors);
    };


    callback3 = function(result,sponsors) {
        log_result("3",result,sponsors);
    };


    callback4 = function(result,sponsors) {
        log_result("4",result,sponsors);
    };


    get_sponsors(tokenUser.user, tokenUser.password, "8", callback0);
    get_sponsors(tokenUser.user, tokenUser.password, "25", callback1);
    get_sponsors(tokenUser.user, "wrongpassword", "20", callback2);
    get_sponsors(tokenUser.user, tokenUser.password, "-5", callback3);
    get_sponsors("wrongusername", tokenUser.password, "20", callback4);


}
```

Output :

```
undefined
        callback 2. result of sponsors is          test.js:67
        callback 2. Status fail test.js:68
        callback 2. sponsors is test.js:69
        [] test.js:70
        callback 2. sponsors - end of callback function test.js:71
        callback 4. result of sponsors is          test.js:67
        callback 4. Status fail test.js:68
        callback 4. sponsors is test.js:69
        [] test.js:70
        callback 4. sponsors - end of callback function test.js:71
        callback 0. result of sponsors is          test.js:67
        callback 0. Status success test.js:68
        callback 0. sponsors is test.js:69
          [Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object,
          Object, Object, Object, Object, Object]
    1.  0: Object
```

2. 1: Object
   1. category: undefined
   2. description: "unique new ice cream soda float to go! A new twist on the classic treat."
   3. geocodekey: "Box 713 Manhattan Beach, CA 90267"
   4. id: undefined
   5. logourl: "http://api.mybeanjar.com/resources/sponsorlogo/43"
   6. name: "Fizz Cup"
   7. offer: undefined
   8. siteurl: "http://www.icecreamfizz.com"
   9. sponsorkey: "17e62166fc8586dfa4d1bc0e1742c08b"
   10. __proto__: Object
3. 2: Object
4. 3: Object
5. 4: Object
6. 5: Object
7. 6: Object
8. 7: Object
9. 8: Object
10. 9: Object
11. 10: Object
12. 11: Object
13. 12: Object
14. 13: Object
15. 14: Object
16. 15: Object
17. 16: Object
18. 17: Object
19. 18: Object
   20. length: 19
21. __proto__: Array[0]
   test.js:70
   callback 0. sponsors - end of callback function test.js:71
   callback 1. result of sponsors is test.js:67
   callback 1. Status success test.js:68
   callback 1. sponsors is test.js:69
   [Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object]
1. 0: Object
2. 1: Object
3. 2: Object
   1. category: undefined
   2. description: "Bedding, Bath, Furniture, Dining, Storage, Gifts"
   3. geocodekey: "2270 Broadway New York, NY 10024"
   4. id: undefined
   5. logourl: "http://api.mybeanjar.com/resources/sponsorlogo/59"
   6. name: "Laytner's Linen"
   7. offer: undefined
   8. siteurl: "http://www.laytners.com"
   9. sponsorkey: "093f65e080a295f8076b1c5722a46aa2"
   10. __proto__: Object
4. 3: Object
5. 4: Object
6. 5: Object
7. 6: Object
8. 7: Object
9. 8: Object
10. 9: Object
11. 10: Object
12. 11: Object
13. 12: Object
14. 13: Object
15. 14: Object
16. 15: Object

17. 16: Object
18. 17: Object
19. 18: Object
20. length: 19
21. __proto__: Array[0]
test.js:70
callback 1. sponsors - end of callback function test.js:71
callback 3. result of sponsors is test.js:67
callback 3. Status success test.js:68
callback 3. sponsors is test.js:69
[Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object]
1. 0: Object
2. 1: Object
3. 2: Object
4. 3: Object
5. 4: Object
6. 5: Object
7. 6: Object
8. 7: Object
9. 8: Object
10. 9: Object
11. 10: Object
12. 11: Object
13. 12: Object
14. 13: Object
1. category: undefined
2. description: "Modell's is the source for all your sporting goods needs. Support your favorite team and get everything you need to get on the field and get in shape."
3. geocodekey: "498 7th Ave, 20th Floor New York, NY 10018"
4. id: undefined
5. logourl: "http://api.mybeanjar.com/resources/sponsorlogo/220"
6. name: "Modell's Sporting Goods"
7. offer: undefined
8. siteurl: "http://www.modells.com"
9. sponsorkey: "ec8ce6abb3e952a85b8551ba726a1227"
10. __proto__: Object
15. 14: Object
16. 15: Object
17. 16: Object
18. 17: Object
19. 18: Object
20. length: 19
21. __proto__: Array[0]
test.js:70
callback 3. sponsors - end of callback function

3.9. get_winners

About get_winners function :

Parameters :

limit – limit of return winners

callback – function(result,message);

Output :

result – "Status success" or "Status fail" or "Internal Server Error"

winners – winners from server, array of objects.

Example Code :

```javascript
function test_winners() {

  log_result = function(which_callback,result,winners){
    console.log("callback "+which_callback+". result of winners is ")
    console.log("callback "+which_callback+". " + result);
    console.log("callback "+which_callback+". " + "winners is ");
    console.log("callback "+which_callback+". ");
    console.log(winners);
    console.log("callback "+which_callback+". winners - end of callback function");
  }

  callback0 = function(result,winners) {
    log_result("0",result,winners);
  };

  callback1 = function(result,winners) {
    log_result("1",result,winners);
  };

  callback2 = function(result,winners) {
    log_result("2",result,winners);
  };

  get_winners("3", callback0);
  get_winners("1", callback1);
  get_winners("-3", callback2);
}
```

Output :

callback 0. result of winners is <u>test.js:9</u>
    callback 0. Status success <u>test.js:10</u>
    callback 0. winners is <u>test.js:11</u>
    callback 0. <u>test.js:12</u>
      [Object, Object, Object]
  1. 0: Object
    1. appkey: "e4da3b7fbbce2345d7772b0674a318d5"
    2. appname: "MyBeanJar"
    3. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/81"
    4. username: "thedude11"
  5. __proto__: Object
  2. 1: Object
  3. 2: Object
    4. length: 3
  5. __proto__: Array[0]
  <u>test.js:13</u>
  callback 0. winners - end of callback function <u>test.js:14</u>
  callback 1. result of winners is <u>test.js:9</u>
  callback 1. Status success <u>test.js:10</u>
  callback 1. winners is <u>test.js:11</u>
  callback 1. <u>test.js:12</u>
    [Object]
  1. 0: Object
    1. appkey: "e4da3b7fbbce2345d7772b0674a318d5"
    2. appname: "MyBeanJar"
    3. sponsorlogourl: "http://api.mybeanjar.com/resources/sponsorlogo/81"
    4. username: "thedude11"
  5. __proto__: Object
    2. length: 1
  3. __proto__: Array[0]
  <u>test.js:13</u>
  callback 1. winners - end of callback function <u>test.js:14</u>
  callback 2. result of winners is <u>test.js:9</u>
  callback 2. Status fail <u>test.js:10</u>
  callback 2. winners is <u>test.js:11</u>
  callback 2. <u>test.js:12</u>
  [] <u>test.js:13</u>
 callback 2. winners - end of callback function

3.10. reedem_bean

About reedem_bean function :

Parameters :

username – String

password – String (unencrypted)

beankey – id of bean to reedem String

callback – function(result,message);

Output :

result   -   "Status success" or "Status fail" or "Internal Server Error"

message – message from server

Example Code :

```
function test_reedembean() {

  log_result = function(which_callback, result, message){
    console.log("callback "+which_callback+". result of reedembean is ");
    console.log("callback "+which_callback+". "+result);
    console.log("callback "+which_callback+". message of reedembean is ");
    console.log("callback "+which_callback+". "+message);
  }

  callback0 = function(result, message){
    log_result("0",result,message);
  };

  callback1 = function(result, message){
    log_result("1",result,message);
  };

  callback2 = function(result, message){
    log_result("2",result,message);
  };
  reedem_bean("lukasz123", "lukasz123", "6b49323020asfassf5f780e1bcasf26945df7481e5", callback0);
  reedem_bean("wrongusernamedsa14", "abcdef", "6b493230205f780e1bcasfas26945df7481e5", callback1);
  reedem_bean("lukasz123", "wrongpassword42ds", "6b493230205f780e1bc26945sfafsadf7481e5", callback2);
}
```

Output :

this is reedem data reedemMethods.js:9
{"status":0,"response":{"message":"Invalid Bean"}} reedemMethods.js:10
this is reedem json reedemMethods.js:12
Object {status: 0, response: Object}
1. response: Object
   1. message: "Invalid Bean"
2. __proto__: Object
   2. status: 0
3. __proto__: Object
reedemMethods.js:13
callback 0. result of reedembean is test.js:302
callback 0. Status fail test.js:303
callback 0. message of reedembean is test.js:304
callback 0. Invalid Bean test.js:305
this is reedem data reedemMethods.js:9
{"status":0,"response":{"message":"Invalid Username and\/or Password"}} reedemMethods.js:10
this is reedem json reedemMethods.js:12
Object {status: 0, response: Object}
1. response: Object
   1. message: "Invalid Username and/or Password"
2. __proto__: Object
   2. status: 0
3. __proto__: Object
reedemMethods.js:13
callback 2. result of reedembean is test.js:302
callback 2. Status fail test.js:303
callback 2. message of reedembean is test.js:304
callback 2. Invalid Username and/or Password test.js:305
this is reedem data reedemMethods.js:9
{"status":0,"response":{"message":"Invalid Username and\/or Password"}} reedemMethods.js:10
this is reedem json reedemMethods.js:12
Object {status: 0, response: Object}
1. response: Object
   1. message: "Invalid Username and/or Password"
2. __proto__: Object
   2. status: 0
3. __proto__: Object
reedemMethods.js:13
callback 1. result of reedembean is test.js:302
callback 1. Status fail test.js:303
callback 1. message of reedembean is test.js:304
callback 1. Invalid Username and/or Password

3.11. register_user

About register_user function :

Parameters :

username – String

password – String (unencrypted)

email – Strig email address

zipcode – String

categories – String array minimum 3 categories

callback – function(result, beans);

Output :

result – "Status success" or "Status fail" or "Internal Server Error"

beans – beans from server, array of objects.

Example Code :

```
function test_register() {

    log_result = function(which_callback,result,message){
    console.log("callback " + which_callback + ". result of register is ");
    console.log("callback " + which_callback + ". " + result);
    console.log("callback " + which_callback + ". register is ");
    console.log(message);
    console.log("callback " + which_callback + ". register - end of callback function");
  };

  callback0 = function(result,message){
    log_result("0",result,message);
  };

  callback1 = function(result,message){
    log_result("1",result,message);
  };

  callback2 = function(result,message){
    log_result("2",result,message);
  };

  callback3 = function(result,message){
    log_result("3",result,message);
  };

  callback4 = function(result,message){
    log_result("4",result,message);
```

```
    };

var categories = ["c4ca4238a0b923820dcc509a6f75849b",
"c81e728d9d4c2f636f067f89cc14862c",
"eccbc87e4b5ce2fe28308fd9f2a7baf3"];

    register_user("h34u78723je", "abcdef","asiadsf233s123iu532235i@f23as.pl","12414242",categories, callback0);
    register_user("lukasz123", "lukasz123","asiassf23ai34ui@fsadas.pl","12241242",categories, callback1);
    register_user("4353m34ujfs6e", "abcdef","asi323176a4s23523iui@fsadas.pl","12413242",categories, callback2);
    register_user("hu344655332uje", "abasdcdef","asia76s32132iu23i@fsadas.pl","1241242",categories, callback3);
    register_user("hu234u7s52331je", "abcdsd5ef","asia76s32132iu23i@fsadas.pl","1241242",categories, callback4);
}
```

Output :

```
this is register json registerMethods.js:10
      Object {status: 0, response: Object}
  1. response: Object
    1. code: "USER_ALREADY_EXISTS"
    2. message: "Username already exists."
  3. __proto__: Object
    2. status: 0
  3. __proto__: Object
  registerMethods.js:11
  callback 1. result of register is test.js:215
  callback 1. Status fail test.js:216
  callback 1. register is test.js:217
  Username already exists. test.js:218
  callback 1. register - end of callback function test.js:219
  this is register json registerMethods.js:10
      Object {status: 1, response: Object}
  1. response: Object
    1. message: "User Successfully Registered"
  2. __proto__: Object
    2. status: 1
  3. __proto__: Object
  registerMethods.js:11
  callback 3. result of register is test.js:215
  callback 3. Status success test.js:216
  callback 3. register is test.js:217
  User Successfully Registered test.js:218
  callback 3. register - end of callback function test.js:219
  this is register json registerMethods.js:10
      Object {status: 1, response: Object}
  1. response: Object
    1. message: "User Successfully Registered"
  2. __proto__: Object
    2. status: 1
  3. __proto__: Object
  registerMethods.js:11
  callback 0. result of register is test.js:215
  callback 0. Status success test.js:216
  callback 0. register is test.js:217
  User Successfully Registered test.js:218
  callback 0. register - end of callback function test.js:219
  this is register json registerMethods.js:10
      Object {status: 0, response: Object}
  1. response: Object
    1. code: "EMAIL_ALREADY_EXISTS"
    2. message: "Email already exists."
  3. __proto__: Object
```

```
    2. status: 0
  3. __proto__: Object
    registerMethods.js:11
    callback 4. result of register is test.js:215
    callback 4. Status fail test.js:216
    callback 4. register is test.js:217
    Email already exists. test.js:218
    callback 4. register - end of callback function test.js:219
    this is register json registerMethods.js:10
      Object {status: 1, response: Object}
  1. response: Object
    1. message: "User Successfully Registered"
  2. __proto__: Object
    2. status: 1
  3. __proto__: Object
    registerMethods.js:11
    callback 2. result of register is test.js:215
    callback 2. Status success test.js:216
    callback 2. register is test.js:217
    User Successfully Registered test.js:218
  callback 2. register - end of callback function
```

3.12. send_password

About send_password function:

Parameters :

username – String

callback – function(result,message);

Output :

result - "Status success" or "Status fail" or "Internal Server Error"

message – message from server

Example Code :

```
function test_sendpassword(){

  log_result = function(which_callback,result,message){
    console.log("callback "+which_callback+". result of sendpassword is ");
    console.log("callback "+which_callback+". " + result);
    console.log("callback "+which_callback+". message of sendpassword is ");
    console.log("callback "+which_callback+". " + message);
  }


  callback0 = function(result,message){
    log_result("0",result,message);
```

```
    };

    callback1 = function(result,message){
       log_result("1",result,message);
    };


    callback2 = function(result,message){
       log_result("2",result,message);
    };


    callback3 = function(result,message){
       log_result("3",result,message);
    };


    callback4 = function(result,message){
       log_result("4",result,message);
    };


    send_password("hujemuje", callback0);
    send_password("lukasz123", callback1);
    send_password("lukasz1assa23", callback2);
    send_password("", callback3);
    send_password("1429012998dj288", callback4);
}
```

Output :

```
callback 2. result of sendpassword is test.js:229
callback 2. Status fail test.js:230
callback 2. message of sendpassword is test.js:231
callback 2. undefined test.js:232
callback 4. result of sendpassword is test.js:229
callback 4. Status fail test.js:230
callback 4. message of sendpassword is test.js:231
callback 4. undefined test.js:232
callback 3. result of sendpassword is test.js:229
callback 3. Status fail test.js:230
callback 3. message of sendpassword is test.js:231
callback 3. undefined test.js:232
this is data sendpasswordMethods.js:16
{"status":1,"response":{"message":"Instructions for changing your password have been mailed
to your registered email address."}} sendpasswordMethods.js:17
this is json from send password sendpasswordMethods.js:19
  Object {status: 1, response: Object}
1. response: Object
  1. message: "Instructions for changing your password have been mailed to your registered
    email address."
  2. __proto__: Object
  2. status: 1
3. __proto__: Object
```

sendpasswordMethods.js:20
callback 1. result of sendpassword is test.js:229
callback 1. Status success test.js:230
callback 1. message of sendpassword is test.js:231
callback 1. Instructions for changing your password have been mailed to your registered
email address. test.js:232
this is data sendpasswordMethods.js:16
{"status":1,"response":{"message":"Instructions for changing your password have been mailed
to your registered email address."}} sendpasswordMethods.js:17
this is json from send password sendpasswordMethods.js:19
    Object {status: 1, response: Object}
  1. response: Object
    1. message: "Instructions for changing your password have been mailed to your registered
       email address."
  2. __proto__ : Object
    2. status: 1
  3. __proto__: Object
  sendpasswordMethods.js:20
  callback 0. result of sendpassword is test.js:229
  callback 0. Status success test.js:230
  callback 0. message of sendpassword is test.js:231
  callback 0. Instructions for changing your password have been mailed to your registered email
  address.

3.13. validate_user

About validate_user function :

Parameters :

username – String

callback – function(result,message);

Output :

result -   "Status success" or "Status fail" or "Internal Server Error"

message – message from server

Example Code :

```
function test_validateuser() {

  log_result = function(which_callback, result,message){

    console.log("callback "+which_callback+". result of validateuser is ");

    console.log("callback "+which_callback+". " + result);

    console.log("callback "+which_callback+". message of validateuser is ");

    console.log("callback "+which_callback+". " + message);

  };


  callback0 = function(result,message) {
```

```
        log_result("0",result,message);
    };


    callback1 = function(result,message) {
        log_result("1",result,message);
    };


    callback2 = function(result,message) {
        log_result("2",result,message);
    };


    callback3 = function(result,message) {
        log_result("3",result,message);
    };


    callback4 = function(result,message) {
        log_result("4",result,message);
    };


    validate_user("hujemuje", callback0);
    validate_user("lukasz123", callback1);
    validate_user("wrongusername", callback2);
    validate_user("", callback3);
    validate_user("-2103021809321", callback4);
}
```

Output :

```
Object {status: 1, response: Object}
    1.  response: Object
        1.  message: "This is a valid user"
    2.  __proto__: Object
        2.  status: 1
    3.  __proto__: Object
        validateuserMethods.js:11
        callback 1. result of validateuser is test.js:294
        callback 1. Status success test.js:295
        callback 1. message of validateuser is test.js:296
        callback 1. This is a valid user test.js:297
        this is validate json validateuserMethods.js:10
            Object {status: 1, response: Object}
    1.  response: Object
        1.  message: "This is a valid user"
    2.  __proto__: Object
        2.  status: 1
    3.  __proto__: Object
```

validateuserMethods.js:11
callback 0. result of validateuser is test.js:294
callback 0. Status success test.js:295
callback 0. message of validateuser is test.js:296
callback 0. This is a valid user test.js:297
this is validate json validateuserMethods.js:10
   Object {status: 0, response: Object}
1. response: Object
   1. message: "This is not a valid user"
2. __proto__: Object
   2. status: 0
3. __proto__: Object
validateuserMethods.js:11
callback 2. result of validateuser is test.js:294
callback 2. Status fail test.js:295
callback 2. message of validateuser is test.js:296
callback 2. This is not a valid user test.js:297
this is validate json validateuserMethods.js:10
   Object {status: 0, response: Object}
1. response: Object
   1. message: "This is not a valid user"
2. __proto__: Object
   2. status: 0
3. __proto__: Object
validateuserMethods.js:11
callback 4. result of validateuser is test.js:294
callback 4. Status fail test.js:295
callback 4. message of validateuser is test.js:296
callback 4. This is not a valid user test.js:297
this is validate json validateuserMethods.js:10
   Object {status: 0, response: Object}
1. response: Object
   1. message: "This is not a valid user"
2. __proto__: Object
   2. status: 0
3. __proto__: Object
validateuserMethods.js:11
callback 3. result of validateuser is test.js:294
callback 3. Status fail test.js:295
callback 3. message of validateuser is test.js:296
callback 3. This is not a valid user test.js:297