JAVASCRIPT SDK DOCUMENTATION

MYBEANJAR

JULY 7, 2015

PREPARED FOR MYBEANJAR
BY CHRISTOPHER CHARLES, SUBPLEX

subplex
STRATEGIC
COMMUNICATION

TABLE OF CONTENTS

METHODS

authenticate_user()

DESCRIPTION

Determine whether the supplied credentials are valid for a specific username

PARAMETERS

username (STRING)

The username of the account to be authenticated.

password (STRING)

The password of the account to be authenticated.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and message.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

message (STRING)

A verbose description of the result of the request.

JSON DATA STRUCTURE

```
“response”:{
        “message”:
}
“status”:
```

ADDITIONAL NOTES

ℹ The password parameter is expected in plaintext. The SDK automatically hashes it before it is included in the request.

get_apps()

DESCRIPTION

Returns an array of apps that use MyBeanJar.

PARAMETERS

username (STRING)

The username of the account making the request.

password (STRING)

The password of the account making the request.

limit_value (NUMBER)

The maximum number of apps to return.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and apps.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

apps (ARRAY)

An array of apps, including key-value pairs for each app.

JSON DATA STRUCTURE

```
"response" : {
        "apps" : [{
                    "appkey" :,
                    "appstoreurl" :,
                    "description" :,
                    "iconurl" :,
                    "name" :,
                    "publisher" :,
                    "publisherkey" :,
                    "publisherurl" :
        }]
}
"status" :
```

ADDITIONAL NOTES

⚠ This method does not automatically hash the password parameter before submitting. However, the server expects an MD5 hash. Be sure to hash passwords before passing them to this function.

get_award()

DESCRIPTION

Adds a Bean to the user's wallet. Information about the awarded Bean is returned in the callback.

PARAMETERS

username (STRING)

The username of the account receiving the Bean.

password (STRING)

The password of the user receiving the Bean.

appkey (STRING)

The unique ID of the app making the request.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and award.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

award (ARRAY)

An array of award Beans, including key-value pairs for each Bean.

JSON DATA STRUCTURE

```
“response” : {
      “award” : [{
                “awarded” :,
                “beankey” :,
                “imageurl” :,
                “iconurl” :,
                “message”:
      }]
}
“status” :
```

ADDITIONAL NOTES

⚠ This method does not automatically hash the password parameter before submitting. However, the server expects an MD5 hash. Be sure to hash passwords before passing them to this function.

get_beans()

DESCRIPTION

Returns an array of Beans available to be awarded to users.

PARAMETERS

username (STRING)

The username of the account making the request.

password (STRING)

The password of the account making the request.

limit_value (NUMBER)

The maximum number of Beans to return.

sort_by (NUMBER)

The sort method by which results will returned. Valid sort methods include:

- 0: Sort by award date
- 1: Sort by expiration date
- 2: Sort by user proximity
- 3: Sort by sponsor name
- 4: Sort by category name

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and beans.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

beans (ARRAY)

An array of Beans, including key-value pairs for each Bean.

JSON DATA STRUCTURE

```
“response" : {
        “beans" : [{
                    “beankey" :,
                    “expirationdate" :,
                    “game" :,
                    “geocodekey" :,
                    “longdescription" :,
                    “redeembarcodeurl" :,
                    “redemptionurl" :,
                    “redemptionvalidation" :,
                    “shortdescription" :,
                    “sponsorkey" :,
                    “sponsorlogourl" :,
                    “sponsorname" :,
                    “sponsorurl" :,
                    “wondate" :
        }]
}
“status" :
```

ADDITIONAL NOTES

ℹ The password parameter is expected in plaintext. The SDK automatically hashes it before it is included in the request.

get_categories()

DESCRIPTION

Returns an array of categories in which Beans can be awarded.

PARAMETERS

username (STRING)

The username of the account making the request.

password (STRING)

The password of the account making the request.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and categories.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

categories (ARRAY)

An array of categories, including key-value pairs for each category.

JSON DATA STRUCTURE

```
“response”:{
        “categories”:[{
                “categorykey”:,
                “name”:
        }]
}
“status”:
```

ADDITIONAL NOTES

⚠ This method does not automatically hash the password parameter before submitting. However, the server expects an MD5 hash. Be sure to hash passwords before passing them to this function.

get_locations_for_sponsor()

DESCRIPTION

Returns an array of redemption locations for a given sponsor.

PARAMETERS

username (STRING)

The username of the account making the request.

password (STRING)

The password of the account making the request.

sponsorkey (STRING)

The unique ID of the sponsor for which redemption locations are sought.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and categories.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

locations (ARRAY)

An array of redemption location objects, including key-value pairs for each location.

JSON DATA STRUCTURE

```
“response” : {
        “locations” : [{}]
}
“status” :
```

ADDITIONAL NOTES

⚠ This method is still under development. Any calls made using this method will yield only an empty `locations` array

⚠ This method does not automatically hash the password parameter before submitting. However, the server expects an MD5 hash. Be sure to hash passwords before passing them to this function.

■ get_sponsors()

DESCRIPTION

Returns an array of sponsors associated with Beans

PARAMETERS

username (STRING)

The username of the account making the request.

password (STRING)

The password of the account making the request.

limit_value (NUMBER)

The maximum number of sponsors to return.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and sponsors.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

sponsors (ARRAY)

An array of sponsors, including key-value pairs for each sponsor.

JSON DATA STRUCTURE

```
“response” : {
        “sponsors” : [{
                “description” :,
                “geocodekey” :,
                “logourl” :,
                “name” :,
                “siteurl” :,
                “sponsorkey” :
        }]
}
“status” :
```

ADDITIONAL NOTES

⚠ This method does not automatically hash the password parameter before submitting. However, the server expects an MD5 hash. Be sure to hash passwords before passing them to this function.

get_winners()

DESCRIPTION

Returns the cumulative number of Beans awarded along with details on each Bean.

PARAMETERS

limit_value (NUMBER)

The maximum number of Beans to return.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and winners.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

winners (ARRAY)

An array of winners and their awarded Beans, including key-value pairs for each.

JSON DATA STRUCTURE

```
"response":{
        "totalbeans":,
        "winners":[{
                "appkey":,
                "appname":,
                "sponsorlogourl":,
                "username":
        }]
}
"status":
```

redeem_bean()

DESCRIPTION

Redeems a Bean contained within the user's wallet. On redemption, the Bean is removed from the user's wallet. This call returns a redemption code for the transaction.

PARAMETERS

username (STRING)

The username of the account redeeming the Bean.

password (STRING)

The password of the user redeeming the Bean.

beankey (STRING)

The unique ID of the Bean to be redeemed.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and message.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

message (ARRAY)

A verbose description of the result of the request.

JSON DATA STRUCTURE

```
“response”:{
        “message”:
        “data”:[{
                    “0”:,
                    “1”:
        }]
}
“status”:
```

ADDITIONAL NOTES

⚠ Earlier versions of the SDK contain a typo in the declaration of this method (i.e., “reedem_bean” versus “redeem_bean”).

ℹ The password parameter is expected in plaintext. The SDK automatically hashes it before it is included in the request.

ℹ This method is still under development. The unparsed JSON response to this request appears to include a malformed key-value pair. Within the “data” object is an array containing keys of “0” and “1” with values of “redemptioncode” and an actual redemption code.

delete_bean()

DESCRIPTION

Deletes a specified Bean from the user's wallet.

PARAMETERS

username (STRING)

The username of the account from which the Bean is to be deleted.

password (STRING)

The password of the account from which the Bean is to be deleted.

beankey (STRING)

The unique ID of the Bean to be deleted.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and message.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

message (ARRAY)

A verbose description of the result of the request.

JSON DATA STRUCTURE

```
"response":{
        "message":
        }
"status":
```

ADDITIONAL NOTES

ℹ The password parameter is expected in plaintext. The SDK automatically hashes it before it is included in the request.

register_user()

DESCRIPTION

Registers a new user account. Also triggers an activation e-mail message to be sent to the address supplied. Accounts remain inactive until they are activated by user.

PARAMETERS

`username` (STRING)

The desired username of the account to be registered.

`password` (STRING)

The desired password of the account to be registered.

`email` (STRING)

The e-mail address to be associated with the account to be registered.

`zipcode` (STRING)

The ZIP code to be associated with the account to be registered.

`categories` (ARRAY)

An array of category ID strings.

`callback` (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, `result` and `message`.

CALLBACK PARAMETERS

`result` (STRING)

The status of the request.

`message` (ARRAY)

A verbose description of the result of the request.

JSON DATA STRUCTURE

```
“response”:{
        “message”:
        }
“status”:
```

ADDITIONAL NOTES

ℹ The password parameter is expected in plaintext. The SDK automatically hashes it before it is included in the request.

ℹ The `categories` parameter should have no fewer than three category IDs included within it.

⚠ There appears to be no server-side checking of the size or contents of the `categories` array.

send_password()

DESCRIPTION

Sends an e-mailed password recovery link to the e-mail address associated with the specified username.

PARAMETERS

username (STRING)

The username of the account for which a password recovery message is to be sent.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and message.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

message (ARRAY)

A verbose description of the result of the request.

JSON DATA STRUCTURE

```
“response”:{
        “message”:
        }
“status”:
```

ADDITIONAL NOTES

ℹ Contrary to its name, this method does not actually send a password to the specified user. Rather, this method triggers the sending of a password recovery link to the e-mail address of the specified user.

validate_user()

DESCRIPTION

Returns the status of the account for a specified username (i.e., whether or not the account exists).

PARAMETERS

username (STRING)

The username of the account to be validated.

callback (FUNCTION)

The function to be performed on completion of the request. Accepts two parameters, result and message.

CALLBACK PARAMETERS

result (STRING)

The status of the request.

message (ARRAY)

A verbose description of the result of the request.

JSON DATA STRUCTURE

```
“response”:{
        “message”:
        }
“status”:
```

ADDITIONAL NOTES

ℹ Does not actually validate the account of the username supplied. This method only indicates whether the account has been validated.