

Minions Vault Documentation

by William Lee

APIs:

Path: */vault/status*

Summary:

The vault status API retrieves all the coins a user owns in the server-side vault and returns it in the format seen below.

Method: GET

Request Header: Must contain user id and access token

Request Body: N/A

Response Body:

```
{
  coins: [
    {brandId: 5, coinAmt: 10, lastChangeDate: "Sat, 08 Oct 2016 11:08:28 GMT"},
    {brandId: 8, coinAmt: 1, lastChangeDate: "Sat, 08 Oct 2016 11:18:22 GMT"},
    ""
    {brandId: 95, coinAmt: 32, lastChangeDate: "Sat, 08 Oct 2016 10:12:54 GMT"}
  ]
}
```

Path: */vault/update*

Summary:

The vault update API takes a list of coins to update (in the format shown below) in the request body. A positive coin amount indicates a deposit into the user's server-side vault, and a negative coin amount indicates a withdrawal from the user's server-side vault. The response then contains a status code for each corresponding withdrawal/deposit transaction. A status code of 0 indicates the update was successful, and a status code of -1 indicates the update failed. An update failure is likely due to the withdrawal amount being greater than the amount the user owns of that coin in his vault.

Note: Whether or not a user is allowed to deposit a certain amount of coins can only be checked on the Frontend, since the Backend has no knowledge of the users client-side funds.

Method: POST

Request Header: Must contain user id and access token

Request Body:

```
{
  coins: [
    {brandId: 5, coinAmt: 10},
    {brandId: 8, coinAmt: -100},
    ""
    {brandId: 95, coinAmt: 5}
  ]
}
```

Response Body:

```
{
  coins: [
    {brandId: 5, coinAmt: 10, status: 0},
    {brandId: 8, coinAmt: -100, status: -1},
    ""
    {brandId: 95, coinAmt: 5, status: 0}
  ]
}
```