

[About](#)

[API versioning](#)

[Unavailable API message](#)

[API Methods](#)

[Login](#)

[Log out](#)

[ATM Locator](#)

[Cash reload locator](#)

[Transaction list](#)

[Gallery Category List](#)

[Card Gallery List](#)

[Get an Anti-CSRF token](#)

[Sign up for a card](#)

[Get Support Categories](#)

[Submit a Support Request](#)

[Get Modal Content](#)

[Get Modal Content for a Specific Campaign](#)

[Indicate CTA has been viewed:](#)

[Indicate a user likes/dislikes a CTA](#)

[Notification settings](#)

[API Features to document](#)

[Modal Content](#)

[Affinity Content](#)

[Get 1-time login link](#)

This document is ~90% abandoned. The new hotness is <https://github.com/cardcorp/slate>

About

This document describes the JSON API for interacting with CARD.com cardholder accounts. The API is currently only to be used by CARD.com's mobile phone applications, but it can theoretically be used by other clients. The API currently only accepts one POST which is for logging in. All other operations are read-only in nature.

API versioning

The API will not be explicitly versioned until we need it. Bare urls will be api v 1.0.0 and all new features/behavior will be backwards compatible. When a feature or behavior is introduced that

breaks the API we'll figure out how to version the API.

Unavailable API message

It's possible that any of these messages can fail and will simply not return results, so requests should include timeout logic (e.g. 15 seconds) that will present a reasonable message to the client.

API Methods

Login

URL: /api/ministatement

Arguments: name and pass.

Login is handled via POST parameters. Login happens via the url /api/ministatement which also returns the ministatement.

Error handling: A login failure will return a 403 and the client (e.g. an app) should display an appropriate error message. Possible reasons for a 403: incorrect username/password, account is blocked, account is temporarily prevented from logging in due to a "flood" restriction. If the client is not authenticated and tries to access the page they will get a 200 response and empty cardholder array. If the user has not yet activated their card the app will also return somewhat "empty" data.

On success, returns: balance, dda, routing_number, enrollment_date as DD/MM/YYYY, bank_name. Client is expected to store these values locally in non-permanent storage that is purged on a logout. Also returns information about a cardholders card including the campaign node id (nid). This campaign id is useful for the cta method.

Note: ministatement is mini and only shows the first 15 transactions. When the client needs to show more transactions, please see /api/transactions

```
{
  cardholder: {
    cards: {
      1: {
        campaign: "261",
        campaign_delta: "0",
        card_status: "B",
        friendly_card_status: "All
        transactions allowed",
        card_dda: "8260000917328",
        card_art:
"https://www.card.com/sites/default/
files/Card-flag-usstates-colorado.jpg"
      }
    }
  }
}
```

```
,
  last4: "7328",
  ordered: "2014 Jan 21",
  date_fulfilled: "2014 Jan 22",
  est_delivery_date: "2014 Jan 29"
},
2: {
  campaign: "209",
  campaign_delta: "5",
  card_status: "B",
  friendly_card_status: "All
transactions allowed",
  card_dda: "8260001141936",
  card_art:
"https://www.card.com/sites/default/
files/card\_images/Card-FridaKahlo-8.
jpg",
  last4: "1936",
  ordered: "2014 Mar 10",
  date_fulfilled: "2014 Mar 11",
  est_delivery_date: "2014 Mar 17"
}
},
bank_name: "The Bancorp",
enrollment_date: "01/21/2014",
balance: "918.32",
transactions: {
  T10208762: {
    amount: "-20.14",
    date: "06/20/2014",
    day_of_week: "Friday",
    long_date: "June 20, 2014",
    time: "09:11 pm",
    mcc: "7299",
    merchant_name:
"BAL*www.gittip.com",
    merchant_alternate_name:
"BAL*www.gittip.com ",
    description: "POS Signature
```

```
    "Purchase"  
  },  
  "T9802830": {
```

Log out

URL: /api/logout

Arguments: none.

It's not explicitly required to logout, but if a user does explicitly logout it is possible to destroy their session by calling api/logout.

This method returns a 200: "messages: "Logged out".

The only real failures are probably 500 or simply a failure to respond.

If a user doesn't manually log out their session will expire after 24 hours.

ATM Locator

URL: /api/atm-locations?location=90401

Arguments: location = for a string of location or a lat/lon pair in the form "latitude, longitude" like this location in Santa Monica: "34.0191539, -118.4907998"

Doesn't require any authentication.

If it returns something other than a 200 that's probably because no locations were found OR something really bad (failure of our stack to return valid information).

Example output:

```
[  
  • {  
    ○ agent_name: "BRITTANICA BAR PUB",  
    ○ address: "318 SANTA MONICA BLVD SANTA MONICA, CA  
      90401",  
    ○ deposit_taking: false,  
    ○ surcharge_free: true,  
    ○ loc: {  
      ■ lat: 34.0159329,  
      ■ lon: -118.495322  
    }  
  }  
]
```

```

    • },
    • {
      ○ agent_name: "U.S. Bank Santa Monica",
      ○ address: "201 SANTA MONICA BOULEVARD SUITE 101 SANTA MONICA, CA 90401",
      ○ deposit_taking: true,
      ○ surcharge_free: true,
      ○ loc: {
        ■ lat: 34.01533,
        ■ lon: -118.497035
      }
    }
  • },
  • {
    ○ agent_name: "ONEWEST BANK",
    ○ address: "401 WILSHIRE BLVD SANTA MONICA, CA 90401",
    ○ deposit_taking: false,
    ○ surcharge_free: true,
    ○ loc: {
      ■ lat: 34.0192289,
      ■ lon: -118.498189
    }
  }
• }
]

```

Cash reload locator

URL: /api/cash-reload-locations?location=Santa%20Monica

Arguments: location = for a string of location or a lat/lon pair in the form "latitude, longitude" like this location in Santa Monica: "34.0191539, -118.4907998"

Doesn't require any authentication.

If it returns something other than a 200 that's probably because no locations were found.

Returns location metadata with lat/lon pairs suitable for mapping

Example output:

```

[
(10)
{
  "agent_name": "A PLUS EXCHANGE INC",
  "address": "1454 4th St Santa Monica, CA 90401",
  "phone": "310.394.7211",

```

```

    "loc":
    {
      "lat": 34.0156975,
      "lon": -118.4943594
    }
  -
}
,
-
{
  "agent_name": "VONS #2262",
  "address": "710 Broadway Santa Monica, CA 90401",
  "phone": "310.260.0260",
  "loc":
  {
    "lat": 34.017175,
    "lon": -118.489981
  }
  -
}
,
-
{
... etc.

```

Transaction list

URL: <api/transactions?date=MM-YYYY>

List of transactions with an argument for the date (MM-YYYY - defaults to current month).

Requires a valid session.

This request returns all transactions in the month with some metadata including category, name of merchant and a bunch of other data that we will theoretically incorporate into a mobile phone app some day.

If there are no transactions in a requested month this will return an empty array. That doesn't mean to stop requesting transactions, it just means nothing happened in that month.

A client shouldn't request a month prior to enrollment date (as returned by ministatement), but if it does, the API will probably return zero transactions or return a code that means "requested a date before the enrollment date, please don't do that any more."

Error handling: the method will return a 403 if the user is anonymous. If they don't have an

active card it will return an empty array.

```
[
(15)
  {
    "amount": 4501.97,
    "transaction_amount": 450197,
    "date": "23-02-2013",
    "time": "12:34",
    "mcc": "mcc",
    "currency": "USD",
    "terminal_city": "AUSTIN",
    "terminal_state": "TX",
    "terminal_country": "USA",
    "description": "University Tuition For A Sweet School"
  }
,
-
  {
    "amount": 3939.14,
    "transaction_amount": 393914,
    "date": "30-09-2012",
    "time": "12:34",
    "mcc": "mcc",
    "currency": "USD",
    "terminal_city": "AUSTIN",
    "terminal_state": "TX",
    "terminal_country": "USA",
    "description": "Amazon.com's Cheap Knockoffs"
  }
,
-
  {
```

Gallery Category List

URL: api/gallery/categories

Gives a complete, simple list of the top level categories on the site. These categories can be used to filter the list of cards.

```
[
  {"id":9, "name":"Animals"},
  {"id":21, "name":"Art & Culture"},
  {"id":22, "name":"Causes"},
  {"id":524, "name":"Cityscapes"},
  {"id":2309, "name":"Comics"},
  {"id":13, "name":"Fashion"},
```

```
{ "id": 18, "name": "Flags"},
{ "id": 20, "name": "Flowers"},
{ "id": 525, "name": "Food & Drink"},
{ "id": 2311, "name": "Gothic"},
{ "id": 523, "name": "Hipster"},
{ "id": 19, "name": "Nature"},
{ "id": 702, "name": "School"},
{ "id": 17, "name": "Seasonal"},
{ "id": 833, "name": "Sports"},
{ "id": 703, "name": "Symbols"},
{ "id": 2312, "name": "Tattoos"},
{ "id": 2310, "name": "TV"
}
```

Card Gallery List

URL: /api/gallery?categories=2310,2312

Provides a list of cards and can be filtered by the category= a comma separated list of ids from the Gallery Category list. Doesn't currently include pagination.

```
[
  {
    "nid": "209",
    "title": "The Frida Kahlo Card",
    "description": "Frida Kahlo's work was once described as a ribbon around a bomb. She, herself, said: pain, pleasure and death are no more than a process for...",
    "category": -1,
    "delta": [
      {
        "id": 0,
        "url":
          "https://d1bv8rh5xtz9w3.cloudfront.net/sites/default/files/styles/campaign_page_card/public/Card-FridaKahlo-3.jpeg?itok=ZpgU-M8U"
      },
      {
        "id": 2,
        "url":
          "https://d1bv8rh5xtz9w3.cloudfront.net/sites/default/files/styles/campaign_page_card/public/card_images/Card-FridaKahlo-4.jpg?itok=_e6BHGvt"
      },
      {
        "id": 5,
        "url":

```



```
"https://d1bv8rh5xtz9w3.cloudfront.net/sites/default/files/styles/campaign_page_card/public/card_images/Card-FridaKahlo-8.jpg?itok=ro6ona3l"  
  }  
 ]  
 },
```

<https://www.card.com/api/gallery/categories>

Get an Anti-CSRF token

URL: /api/session/token

Returns a token that is specific to the user session and the json api. This token can be used for any operations that change data to indicate the user's intent to perform the action (i.e. prevent CSRF).

Also returns the session api_key so that the user can keep their csrf token and the session persists. In the app we compare and set this if it doesn't exist, or replace one if currently does exist.

```
{"token": "dy2nyRukF0JxSEzCx-PPIet7quBPxJ-j1MORXqceoJg",  
 "api_key": "SESSb724669e9542a3a4640c33b25700475f=K6gx8C1NDXXe2qg5QJLXI0_NEaH6  
KxdMt5EDT10pwn8"  
}
```

Sign up for a card

URL: /api/signup

Requires a set of variables to be POSTed in order to sign up for a card. The following variables are possible in the POST. All are required except card_address2 which is optional.

- csrf_token from /api/session/token
- nid - integer
- card_delta
- first_name
- last_name
- email
- card_phone
- card_address
- card_address2
- card_zipcode

- card_city
- card_state
- dob_day - 2 digit integer from 1 to 31
- dob_month - 2 digit integer from 1 to 12
- dob_year - 4 digit integer - will get validated to the right age server side
- social - 9 digit social security number
- terms - 1 or 0 checkbox (people must accept the terms and conditions)

The server-side code will run a series of validations on the submitted values and respond with either an error or success. The response structure is:

```
{
  status: "success"
  message: "Congratulations! Your card will arrive in a white envelope in about 7 days."
  timestamp: 1378412774
}
```

There are dozens of potential error messages, but they will look roughly like:

```
{
  status: "error"
  error_type: "cardapi"
  errors:
  [
  2]
    0:
      {
        label: "first_name"
        message: "first_name is a required field"
      }
    -
    1:
      {
        label: "last_name"
        message: "last_name is a required field"
      }
  ]
}
```

```
-  
-  
  timestamp: 1378414623  
}
```

There is a special error message that the app could recognize and handle specially:

```
{  
  status: "error"  
  error_type: "cardapi"  
  errors:  
    [1]  
    0:  
      {  
        label: "csrf_token"  
        message: "There was an error processing your submission. Please close the app  
and try again."  
      }  
    timestamp: 1378414558  
}
```

When the CSRF token is invalid the app needs to fetch a new token and use it with the next request.

Get Support Categories

URL: /api/support/categories

Arguments: None

Request Type: GET

Returns an array of key-value pairs of different customer support categories that a user can select from when submitting a ticket or question.

Sample response:

```
{"late_card":"Card not received","direct_deposit":"Direct deposit","suggest_card":"Suggest a  
card","cardcom_issue":"General questions","other":"Other"}
```

Submit a Support Request

URL: /api/support/submit

Requires a set of variables to be POSTed so that a customer ticket can be injected into our customer relationship management system. The following variables are possible in the POST. The *mail*, *subject* and *message* variables are required.

- csrf_token from /api/session/token
- subject - subject line for the newly created support ticket (ie: "Help with login")
- message - contents or full body of the support request

Get Modal Content

Version 2 of the API will include one or more elements in each response for "modal" content that will be a string of HTML content to overlay on top of the upcoming page. An example use case for this is that the response to a successful login might include a message like "We see you used an ATM outside of our network. You can use the ATM locator in this app to find surcharge-free ATMs."

URL: /api/cta/

This returns generic information and should probably not be used. For sample content see the next method.

Get Modal Content for a Specific Campaign

URL: /api/cta/NID

Example: /api/cta/209

By asking for content about a specific campaign number (nid) the site can serve up more relevant content.

Sample response:

```
[
  {
    screen: "*",
    nid: 4359,
    body: "<article class=\"node-4359 node
node-blog-custom node-promoted node-teaser clearfix\">
<header class=\"with-avatar\"> <img class=\"avatar\"
src=\"https://d1bv...\"
....
```

```
]
    }
</article><!-- /.node --> "
```

Indicate CTA has been viewed:

URL: /api/cta/viewed/4359*/bank-screen

This callback will track which content has or has not been viewed which will help to fine tune what content to send to cardholders.

Indicate a user likes/dislikes a CTA

URL: api/cta/rank/4359*/1

The final parameter is either a 1 or a -1 to indicate like or dislike. Again, this is to help fine-tune the content sent to cardholders.

Track Remote Event

POST

URL: api/track-remote-event

This callback allows for remote event tracking to event tracker. POST values will be written to eventtracker for event_name, event_value, uuid (unique device id). Also returns the api_key associated with the session to make sure session persists.

Required Post Values: event_name, event_value, csrf_token

example response:

```
{
  "api_key": "SESSb724669e9542a3a4640c33b25700475f=K6gx8C1NDXXe2qg5QJLXI0_NEaH6KxdMt5EDT10pwn8"
}
```

SMS Settings

GET

URL: api/get-sms-settings

This callback returns the users current sms settings to populate the form in the app.

Optional URL Parmater: delta, decides which card to use. Default is 1.

Requires an api_key

example response:

```
{
  "client_config": [
    {
      "MsgTriggerId": "0f7b4819-2a11-4965-ab87-0bff3f8a396c",
      "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
      "MsgType": "SMS",
      "MsgId": "MT7",
      "MsgDescription": "Authorization Approvals",
      "ClientConfigStatus": "1",
      "AddiParam": {},
      "AccessLv": "1",
      "Reliable": "0",
      "ExpirationHrs": {},
      "DoNotDisturbFrom": {},
      "DoNotDisturbTo": {},
      "AccessLevelConfigData": {
        "AccessLevelConfig": {
          "MsgOptionId": "0",
          "AccessLvOverride": {},
          "Relation": "0"
        }
      }
    },
    {
      "MsgTriggerId": "21cbccb5-e9a1-46fd-8b36-2ef659202ace",
      "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
      "MsgType": "SMS",
      "MsgId": "MT8",
      "MsgDescription": "Authorization Declines",
      "ClientConfigStatus": "1",
      "AddiParam": {},
      "AccessLv": "1",
      "Reliable": "0",
      "ExpirationHrs": {},
      "DoNotDisturbFrom": {},
      "DoNotDisturbTo": {},
      "AccessLevelConfigData": {
        "AccessLevelConfig": {
          "MsgOptionId": "0",
```

```
    "AccessLvIOverride": {},
    "Relation": "0"
  }
}
},
{
  "MsgTriggerId": "459f801f-2a08-48e2-b005-5ce80c811a9a",
  "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
  "MsgType": "SMS",
  "MsgId": "MT9",
  "MsgDescription": "Low Balance",
  "ClientConfigStatus": "1",
  "AddIParam": "10",
  "AccessLvI": "1",
  "Reliable": "0",
  "ExpirationHrs": {},
  "DoNotDisturbFrom": {},
  "DoNotDisturbTo": {},
  "AccessLevelConfigData": {
    "AccessLevelConfig": {
      "MsgOptionId": "0",
      "AccessLvIOverride": {},
      "Relation": "0"
    }
  }
}
},
{
  "MsgTriggerId": "abaeb071-d193-495e-8193-d14a0721456e",
  "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
  "MsgType": "SMS",
  "MsgId": "MT6",
  "MsgDescription": "Value Load",
  "ClientConfigStatus": "1",
  "AddIParam": {},
  "AccessLvI": "1",
  "Reliable": "0",
  "ExpirationHrs": {},
  "DoNotDisturbFrom": {},
  "DoNotDisturbTo": {},
  "AccessLevelConfigData": {
    "AccessLevelConfig": {
      "MsgOptionId": "0",
      "AccessLvIOverride": {},
    }
  }
}
```

```
    "Relation": "0"
  }
}
],
"client_enrollment": {
  "Message": [
    {
      "MsgTriggerId": "0f7b4819-2a11-4965-ab87-0bff3f8a396c",
      "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
      "MsgId": "MT7",
      "MsgDescription": "Authorization Approvals",
      "ClientConfigStatus": {},
      "AccessLvl": {},
      "Reliable": {},
      "ExpirationHrs": {},
      "AccessLevelConfigData": {}
    },
    {
      "MsgTriggerId": "21cbccb5-e9a1-46fd-8b36-2ef659202ace",
      "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
      "MsgId": "MT8",
      "MsgDescription": "Authorization Declines",
      "ClientConfigStatus": {},
      "AccessLvl": {},
      "Reliable": {},
      "ExpirationHrs": {},
      "AccessLevelConfigData": {}
    },
    {
      "MsgTriggerId": "459f801f-2a08-48e2-b005-5ce80c811a9a",
      "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
      "MsgId": "MT9",
      "MsgDescription": "Low Balance",
      "ClientConfigStatus": {},
      "AccessLvl": {},
      "Reliable": {},
      "ExpirationHrs": {},
      "AccessLevelConfigData": {}
    },
    {
      "MsgTriggerId": "abaeb071-d193-495e-8193-d14a0721456e",
      "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
```



```

    "MsgId": "MT6",
    "MsgDescription": "Value Load",
    "ClientConfigStatus": {},
    "AccessLvl": {},
    "Reliable": {},
    "ExpirationHrs": {},
    "AccessLevelConfigData": {}
  }
],
"EnrollmentUID": "22155f8e-59bd-e411-92cd-00215a5b911c",
"MsgAddress": "3108823642"
}
}

```

POST

URL: api/set-sms-settings

This callback will update the users sms preferences based on what options are selected.

Requires an api_key

Required Post Values: delta

Optional Post Values: the msgTriggerIds of the options that have been selected

Example Response:

```

{
  "client_config": [
    {
      "MsgTriggerId": "0f7b4819-2a11-4965-ab87-0bff3f8a396c",
      "ConfigLevelid": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
      "MsgType": "SMS",
      "MsgId": "MT7",
      "MsgDescription": "Authorization Approvals",
      "ClientConfigStatus": "1",
      "AddlParam": {},
      "AccessLvl": "1",
      "Reliable": "0",
      "ExpirationHrs": {},
      "DoNotDisturbFrom": {},
      "DoNotDisturbTo": {},
      "AccessLevelConfigData": {
        "AccessLevelConfig": {
          "MsgOptionId": "0",
          "AccessLvlOverride": {},

```

```
    "Relation": "0"
  }
}
},
{
  "MsgTriggerId": "21cbccb5-e9a1-46fd-8b36-2ef659202ace",
  "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
  "MsgType": "SMS",
  "MsgId": "MT8",
  "MsgDescription": "Authorization Declines",
  "ClientConfigStatus": "1",
  "AddiParam": {},
  "AccessLvl": "1",
  "Reliable": "0",
  "ExpirationHrs": {},
  "DoNotDisturbFrom": {},
  "DoNotDisturbTo": {},
  "AccessLevelConfigData": {
    "AccessLevelConfig": {
      "MsgOptionId": "0",
      "AccessLvlOverride": {},
      "Relation": "0"
    }
  }
}
},
{
  "MsgTriggerId": "459f801f-2a08-48e2-b005-5ce80c811a9a",
  "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
  "MsgType": "SMS",
  "MsgId": "MT9",
  "MsgDescription": "Low Balance",
  "ClientConfigStatus": "1",
  "AddiParam": "10",
  "AccessLvl": "1",
  "Reliable": "0",
  "ExpirationHrs": {},
  "DoNotDisturbFrom": {},
  "DoNotDisturbTo": {},
  "AccessLevelConfigData": {
    "AccessLevelConfig": {
      "MsgOptionId": "0",
      "AccessLvlOverride": {},
      "Relation": "0"
    }
  }
}
```

```

    }
  }
},
{
  "MsgTriggerId": "abaeb071-d193-495e-8193-d14a0721456e",
  "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
  "MsgType": "SMS",
  "MsgId": "MT6",
  "MsgDescription": "Value Load",
  "ClientConfigStatus": "1",
  "AddiParam": {},
  "AccessLvl": "1",
  "Reliable": "0",
  "ExpirationHrs": {},
  "DoNotDisturbFrom": {},
  "DoNotDisturbTo": {},
  "AccessLevelConfigData": {
    "AccessLevelConfig": {
      "MsgOptionId": "0",
      "AccessLvlOverride": {},
      "Relation": "0"
    }
  }
}
},
],
"client_enrollment": {
  "Message": [
    {
      "MsgTriggerId": "0f7b4819-2a11-4965-ab87-0bff3f8a396c",
      "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
      "MsgId": "MT7",
      "MsgDescription": "Authorization Approvals",
      "ClientConfigStatus": {},
      "AccessLvl": {},
      "Reliable": {},
      "ExpirationHrs": {},
      "AccessLevelConfigData": {}
    },
    {
      "MsgTriggerId": "21cbccb5-e9a1-46fd-8b36-2ef659202ace",
      "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
      "MsgId": "MT8",
      "MsgDescription": "Authorization Declines",

```

```

    "ClientConfigStatus": {},
    "AccessLv": {},
    "Reliable": {},
    "ExpirationHrs": {},
    "AccessLevelConfigData": {}
  },
  {
    "MsgTriggerId": "459f801f-2a08-48e2-b005-5ce80c811a9a",
    "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
    "MsgId": "MT9",
    "MsgDescription": "Low Balance",
    "ClientConfigStatus": {},
    "AccessLv": {},
    "Reliable": {},
    "ExpirationHrs": {},
    "AccessLevelConfigData": {}
  },
  {
    "MsgTriggerId": "abaeb071-d193-495e-8193-d14a0721456e",
    "ConfigLevelId": "06749ba8-6ab4-4ba1-b6a4-4f46b553fa4a",
    "MsgId": "MT6",
    "MsgDescription": "Value Load",
    "ClientConfigStatus": {},
    "AccessLv": {},
    "Reliable": {},
    "ExpirationHrs": {},
    "AccessLevelConfigData": {}
  }
],
"EnrollmentUID": "22155f8e-59bd-e411-92cd-00215a5b911c",
"MsgAddress": "3108823642"
}
}

```

Notification settings

URL: api/notification-settings

Sample Response.

```

[
  {
    "rpId": "212000011111",
    "balance": "3",
    "big_debit": "2",

```

```
        "big_credit": "2",  
        "cardcom": "2"  
    }  
]
```

RPID: Unique Identifier for card

Other fields:

3 = both SMS and Notifications

2 = Notifications

1 = SMS

0 = None

Get 1-time login link

URL: `api/one-time-login`

Returns an array (`'url'` => `https://www.card.com...etc./`) for one-time login. The app can add `/login?destination=foo` to send someone to `card.com/foo`

API Features to document

Affinity Content

The "Affinity tab" needs a callback to gather its content. This can contain content relevant to all customers or specific to the affinity

- * Designate blog posts as being relevant to an affinity
- * Designate blog posts as being relevant to all customers