



Whisqr API

Use our API to integrate your Whisqr Loyalty or Membership program with your Mobile App, POS System or Web Application.

Overview

Create your Membership or Loyalty Program on our website (<https://whisqr.com> (<https://whisqr.com>)); where you will find detailed information about the capabilities of our programs. The website is also where you will be able to manage your program, view reports, change settings, and much more.

Punches and Redemptions

A Punch Action is the term we use for a single act of adding Punches and/or Redemptions to a Card. A Punch Action might consist of any number of Punches or none at all. The number of Punches actually awarded to the Customer or Member is calculated by the platform based on a combination of the business's settings and the parameters passed in the request. A Redemption is always associated with a Punch Action (i.e. you must initialize a Punch Action, even if it is for 0 Punches, before issuing a redemption).

If the program awards punches based on a purchase amount; then the Punch Initialize and Update endpoints will return a number of Punches based on a combination of the purchase value submitted and the settings from the Basic Punch Behaviour (in the Administrator console).

Users and Cards

Often physical cards are used that are not Registered. Therefore Cards do not necessarily have a User account. Also; multiple Cards can be associated with a single User account. The Punches and Redemptions are pooled together from all of the Cards to calculate the User's total. Punch Actions are always associated with Cards.

Authentication

Generate public/private key pairs using the API section of your Administrator Console. We encourage you to generate and replace key pairs frequently.

Most calls to the API require both a Public Key and a Content Hash. The Content Hash is generated using sha256 hash of the request body content, using the private key. In PHP generating a hash would look something like this:

```
$contentHash = hash_hmac('sha256', $content, $privateHash);
```

This page provides examples for generating hmac/sha256 hashes in a variety of languages: <https://www.jokecamp.com/blog/examples-of-creating-base64-hashes-using-hmac-sha256-in-different-languages/> (<https://www.jokecamp.com/blog/examples-of-creating-base64-hashes-using-hmac-sha256-in-different-languages/>)

Error Codes

All responses to JSON requests contain a "status" parameter which has three possible states: "success", "pending" or "failure". There is also a "message" parameter which contains human readable details regarding the status of the response.

POST App Key Create

```
https://whisqr.com/api/v1.2/apikeys
```

Generate keys for apps from within the API using an administrator secret key. We encourage you to generate and replace app keys frequently and to delete retired keys using the DELETE request type.

HEADERS

X-Public

pk_live_XX

Content-Type

application/json

Example Request

Create App Key

POST /api/v1.2/apikeys HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XX
Content-Type: application/json

Example Response

200 – OK

{
 "status": "success",
 "message": "generated API key pair for mobile app",
 "keyPublic": "pk_live_XX",
 "keySecret": "sk_live_XX",
 "businesscode": "coffeetime"
}

DEL App Key Delete

https://whisqr.com/api/v1.2/apikeys

We strongly encourage you to generate app keys frequently and delete keys that you know to be no longer in use.

HEADERS

X-Public

pk_live_XX

X-Hash

XX

) Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

BODY raw

```
{
  "keyPublicDelete": "pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
}
```

Example Request

Delete App Key

DELETE /api/v1.2/apikeys HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XX
X-Hash: XXX
Content-Type: application/json

{
 "keyPublicDelete": "pk_live_XX"
}

Example Response

200 – OK

```
{
  "status": "success",
  "message": "key pair deleted"
}
```

POST Customer Create

<https://whisqr.com/api/v1.2/user/customer>

Create a new customer (loyalty program)

If the user is created using only their email address; the platform will create a partial user record, send the user an email letting them know that they have been added to the program in question, and inviting them to set the name and password. This endpoint will return a status of "pending".

If the user is created using an email address, first name and password; the platform will create a complete user record and no follow-up action will be required by the user.

In both cases this endpoint will return a card code for the customer.

HEADERS

X-Public

pk_live_XX

X-Hash

XX

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

BODY raw

```
{
  "email": "doesnotexist@whisqr.com",
  "phone": "(234) 567-8900",
  "firstname": "testname",
  "password": "password",
  "token": "cus_FELksedIdQLMLM"
}
```

Example Request

Customer Create (Complete) ▼

```
POST /api/v1.2/user/customer HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-Hash: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json

{
  "email": "doesnotexist@whisqr.com",
  "firstname": "testname",
  "password": "password"
```

Example Response

200 – OK

```
{
  "status": "success",
  "message": "Complete customer account created. No further action required.",
  "email": "doesnotexist@whisqr.com",
  "firstname": "testname",
  "cardcode": "3NtnwRK"
}
```

POST Member Create

<https://whisqr.com/api/v1.2/user/member>

Create a new member (membership program)

If the user is created using only their email address; the platform will create a partial user record, send the user an email letting them know that they have been added to the program in question, and inviting them to pay for their membership and to set their name and password.

If the user is created using an email address, first name and password; the platform will create a complete user record. The member will still be required to pay for their membership before their membership status is set to active.

In both cases; the endpoint will return a card code.

HEADERS

X-Public

pk_live_XX

X-Hash

XX

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

BODY raw

```
{
  "email": "doesnotexist@whisqr.com",
  "phone": "(234) 567-8900",
  "firstname": "testname",
  "password": "password",
  "token": "cus_FELksedIdQLMLM"
}
```

) Example Request

Member Create (Partial) ▼

```
POST /api/v1.2/user/member HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-Hash: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json
```

```
{
  "email": "doesnotexist@whisqr.com",
  "firstname": "testname",
  "password": ""
}
```

Example Response

200 – OK

```
{
  "status": "pending",
  "message": "member account initialized. Enrollment email sent to customer.",
  "email": "doesnotexist@whisqr.com",
  "cardcode": "3Ntp2GB"
}
```

POST User Login

<https://whisqr.com/api/v1.2/user/login>

This method provides a means for applications to retrieve a app key using employee login credentials.

)

```
application/json
```

```
{
  "email": "apitest@whisqr.com",
  "password": "password",
  "businesscode": "coffeetime"
}
```

```
POST /api/v1.2/user/login HTTP/1.1
Host: whisqr.com
Content-Type: application/json
```

```
{
  "email": "apitest@whisqr.com",
  "password": "password",
  "businesscode": "coffeetime"
}
```

200 – OK

```
{
  "status": "success",
  "message": "generated API key for app",
  "keyPublic": "pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "keySecret": "sk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "businesscode": "coffeetime",
  "businesses": []
}
```

POST Punch Initialize (email)

```
https://whisqr.com/api/v1.2/punch
```

Initiates a "punch action". Use this method to punch a customer or member's card. Cards can be punched using a customer or member's email address or card code.

The request will return a punchcode which can be used to update the details of this punch; using the Punch Update endpoint.

Endpoint returns the punch details; including any automatic values applied as part of applicable behaviours (e.g. if the "First Customer of the Day" behaviour applies to this punch action, the details will be included in the response). When a behaviour is selected in the administrator console, and any of those active behaviours affect this punch action, then that punch action is applied and the details of that application is included as part of the response. Any subsequent update applied to the details of a punch action is an active arbiter in what settings can be updated and to what extent. For instance, an app could update the customer's friend count but not how many punches reward should be rewarded for that number. The number of punches a customer might be rewarded for bringing friends is determined by an administrator using the administrator console.

HEADERS

X-Public

```
pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

X-Hash

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

```
application/json
```

BODY raw

```
{
  "email": "apitest@whisqr.com",
  "locationcode": "xyz321"
}
```

Example Request

```
Punch Initialize (email)
```



```
POST /api/v1.2/punch HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-Hash: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json

{
  "email": "testlll3@qrloyalty.cards",
  "locationcode": "29LmZ"
}
```

Example Response

200 – OK

```
{
  "status": "success",
  "message": "card punched",
  "fullname": "Test Name",
  "punchcode": "4CWPG9b",
  "cardcode": "2Kso2AK",
  "punchcount": 3,
  "punchtotal": 146,
  "details": {
    "basicpunch": {
```

POST Punch Initialize (business user token)

<https://whisqr.com/api/v1.2/punch>

Initiates a "punch action". Use this method to punch a customer or member's card. Cards can be punched using a customer or member's email address or card code.

The request will return a punchcode which can be used to update the details of this punch; using the Punch Update endpoint.

Endpoint returns the punch details; including any automatic values applied as part of applicable behaviours (e.g. if the "First Customer of the Day" behaviour applies to this punch action, the details will be included in the response). When a behaviour is selected in the administrator console, and any of those active behaviours affect this punch action, then that punch action is applied and the details of that application is included as part of the response. Any subsequent update applied to the details of a punch action is an active arbiter in what settings can be updated and to what extent. For instance, an app could update the customer's friend count but not how many punches reward should be rewarded for that number. The number of punches a customer might be rewarded for bringing friends is determined by an administrator using the administrator console.

HEADERS

X-Public

pk_live_XX

X-Hash

XX

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

BODY raw

```
{
  "email": "apitest@whisqr.com",
  "locationcode": "xyz321"
}
```

)

Example Request

Punch Initialize (email)

```
POST /api/v1.2/punch HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-Hash: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json

{
  "email": "testlll3@qrloyalty.cards",
  "locationcode": "29LmZ"
}
```

Example Response

200 – OK

```
{
  "status": "success",
  "message": "card punched",
  "fullname": "Test Name",
  "punchcode": "4CWPG9b",
  "cardcode": "2Kso2AK",
  "punchcount": 3,
  "punchtotal": 146,
  "details": {
    "basicpunch": {
```

POST Punch Initialize (cardcode)

```
https://whisqr.com/api/v1.2/punch
```

Initiates a "punch action". Use this method to punch a customer or member's card. Cards can be punched using a customer or member's email address or card code.

The request will return a punchcode which can be used to update the details of this punch; using the Punch Update endpoint.

Endpoint returns the punch details; including any automatic values applied as part of applicable behaviours (e.g. if the "First Customer of the Day" behaviour applies to this punch action, the details will be included in the response).

HEADERS

X-Public

```
pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

) **X-Hash**

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

```
application/json
```

BODY raw

```
{
  "cardcode": "abc123",
  "locationcode": "xyz321"
}
```

Example Request

Punch Initialize (cardcode)

```
POST /api/v1.2/punch HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-Hash: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json

{
  "cardcode": "2Kso2AK",
  "locationcode": "29LmZ"
}
```

Example Response

```
200 – OK

{
  "status": "success",
  "message": "card punched",
  "fullname": "Test Name",
  "punchcode": "4CWLyCP",
  "cardcode": "2Kso2AK",
  "punchcount": 3,
  "punchtotal": 137,
  "details": {
    "basicpunch": {
```

PUT Punch Update

```
)
https://whisqr.com/api/v1.2/punch/4CcnEZs
```

Updates a "punch action". Use this method to change or add to the details of a punch action that has been initialized using either a customer's email address or card code.

Endpoint returns a revised set of details; including any automatic values applied as part of applicable behaviours (e.g. if the "First Customer of the Day" behaviour applies to this punch action, the details will be included in the response).

HEADERS

X-Public

pk_live_XX

X-Hash

XX

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

Event-Counter

123

Optional: An Integer value that should be incremented with each call to this endpoint. If the Event-Counter value is less than the value used in any of the previous requests; the request is ignored.

BODY raw

```
{
  "details": {
    "basicpunch": {
      "punches": 8
    },
    "broughtfriend": {
      "friendcount": 3
    },
    "mytreat": {
      "treatspurchasedlist": [
```

Example Request

Punch Update (Simple) ▼

```
PUT /api/v1.2/punch/4CVYEwz HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-Hash: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json
```

```
{
  "details": {
    "basicpunch": {
      "punchcount": 6
```

Example Response

200 – OK

```
{
  "status": "success",
  "message": "card punch updated",
  "fullname": "Test Name",
  "punchcode": "4CVYEwz",
  "cardcode": "2Kso2AK",
  "punchcount": 7,
  "punchtotal": 143,
  "details": {
    "dailyrepeater": {
```

PUT Punch Update (with purchase value)

<https://whisqr.com/api/v1.2/punch/4CcohSn>

Updates a "punch action". Use this method to change or add to the details of a punch action that has been initialized using either a customer's email address or card code.

Endpoint returns a revised set of details; including any automatic values applied as part of applicable behaviours (e.g. if the "First Customer of the Day" behaviour applies to this punch action, the details will be included in the response).

HEADERS

X-Public

pk_live_XX

X-Hash

XX

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

Event-Counter

123

Optional: An Integer value that should be incremented with each call to this endpoint. If the Event-Counter value is less than the value used in any of the previous requests; the request is ignored.

BODY raw

```
{
  "details": {
    "basicpunch": {
      "purchasevalue": 750,
      "purchasetax": 91,
      "purchasedetails": "{ \"objects\": [ { \"type\": \"ITEM\", \"id\": \"W62UWFY35CWMYGVWK6T\" } ] }",
    },
    "broughtfriend": {
      "friendcount": 3
    },
  },
}
```

Example Request

Punch Update (detailed)

```
PUT /api/v1.2/punch/4CVYEwz HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-Hash: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json
Event-Counter: 123

{
  "details": {
    "basicpunch": {
```

Example Response

```
200 – OK

{
  "status": "success",
  "message": "card punch updated",
  "fullname": "Test Name",
  "punchcode": "4CVYEwz",
  "cardcode": "2Kso2AK",
  "punches": 21,
  "punchtotal": 194,
  "details": {
    "basicpunch": {
```

PUT Redemption Update

```
https://whisqr.com/api/v1.2/redeem/4CVYEwz
```

Updates or creates redemption details for a "punch action".

Use this method to submit a redemption. Redemptions must always be associated with a punch action; even if the punch action is for zero punches.

Endpoint accepts a details list indicating the number of redemptions (count) and the reward's index, as it appears in the reward settings. Endpoint returns information about the punch action, the total punches redeemed for the punch action, and a details list of the redemptions, corresponding to the submitted list, including the total punches deducted for each reward.

And reward not submitted is assumed to have a redemption count of zero.

HEADERS

X-Public

```
pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

X-Hash

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

Event-Counter

123
Optional: An Integer value that should be incremented with each call to this endpoint. If the Event-Counter value is less than the value used in any of the previous requests; the request is ignored.

BODY raw

```
{
  "details": [
    {
      "count": 2,
      "rewardindex": 2
    }
  ]
}
```

Example Request

Redemption Update

PUT /api/v1.2/redeem/4CVYEwz HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XX
X-Hash: XX
Content-Type: application/json
Event-Counter: 123

{
 "details": [
 {

Example Response

200 – OK


```
{
  "status": "success",
  "message": "card redemption updated",
  "fullname": "Test Name",
  "punchcode": "4CVYEwz",
  "cardcode": "2Kso2AK",
  "details": [
    {
      "count": 2,
      "rewardindex": 2,
```

GET Card Punch Total Get (email)

```
https://whisqr.com/api/v1.2/punch/total
```

-) Get the current number of punches associated with a registered customer using their email address.

HEADERS

X-Public

pk_live_XX

X-Hash

XX

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

BODY raw

```
{
  "email": "apitest@whisqr.com"
}
```

Example Request

```
Get Card Punch Total (email)
```

```
GET /api/v1.2/punchtotal HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-Hash: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json

{
  "email": "apitest@whisqr.com"
}
```

Example Response

```
200 – OK

{
  "status": "success",
  "message": "user has 10 punches.",
  "punchtotal": 10
}
```

GET Card Punch Total Get (cardcode)

```
https://whisqr.com/api/v1.2/punch/total
```

Get the current number of punches associated with a customer (registered) or card (unregistered) using a cardcode.

HEADERS

X-Public
pk_live_XX

X-Hash
XX

Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type
application/json

BODY raw

```
{
  "cardcode": "abc123"
}
```


Generate a keyed hash of the Body using the HMAC method; using the corresponding secret key as a key

Content-Type

application/json

BODY raw

```
{
  "email": "apitest@whisqr.com"
}
```

Example Request

Check Active Membership

GET /api/v1.2/ismember HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XX
X-Hash: XX
Content-Type: application/json

{
 "email": "stephenakins@gmail.com"
}

Example Response

200 – OK

{
 "status": "success",
 "message": "matching active membership",
 "ismember": true
}

GET Behaviour Settings Get

```
https://whisqr.com/api/v1.2/business/behaviours
```

Return the details of active behaviours rewarded by the business associated with the public key.

HEADERS

X-Public

pk_live_XX

Content-Type

application/json

Example Request

Retrieve Behaviour Settings

```
GET /api/v1.2/business/behaviours HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json
```

Example Response

200 – OK

```
{
  "status": "success",
  "message": "returned behaviours settings for Coffee Time",
  "shortcode": "coffeetime",
  "settings_behaviours": {
    "basicpunch": {
      "active": "true",
      "fields": {
        "punches": "2",
        "nomorethan": "fivetimes",
```

GET Reward Settings Get

https://whisqr.com/api/v1.2/business/rewards

Return the list of active rewards offered by the business associated with the public key.

HEADERS

X-Public

pk_live_XX

Content-Type

application/json

Example Request

Retrieve Reward Settings

```
GET /api/v1.2/business/rewards HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json
```

Example Response

200 – OK

```
{
  "status": "success",
  "message": "returned rewards settings for Coffee Time",
  "shortcode": "coffeetime",
  "settings_rewards": [
    {
      "fields": {
        "rewardname": "A Free Cup of Coffee",
        "punchcost": "10",
        "retail": "",

```

GET Location List Get (for a business)

https://whisqr.com/api/v1.2/business/locations

Returns a list of locations for the business for which a public key was issued.

HEADERS

X-Public

pk_live_XX

Content-Type

application/json

Example Request

Retrieve Location List (for a business)

```
GET /api/v1.2/business/locations HTTP/1.1
Host: whisqr.com
X-Public: pk_live_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Content-Type: application/json
```

Example Response

200 – OK

```
{
  "status": "success",
  "message": "4 locations returned",
  "locations": [
    {
      "locationcode": "29LmZ",
      "locationname": "Test Location 1",
      "address": "Test Address 1",
      "lat": "48.879166",
      "lng": "-123.321297"
    }
  ]
}
```