

Voucher's - Online API

Contract Details for Balance Check & Redeem APIs for online providers

Version History of this document:

Updated Date	Details
11-Nov-2020	Initial document
4-Dec-2020	Added VOID and SAMPLE calls
22-Dec-2020	Adjusted copy with latest program details, fixed errors and inconsistencies.
15-Jan-2021	Updated to indicate short code length is 8-10 and has a signifier prefix
22-Feb-2021	Updated to include totalAmount field
18-Jun-2021	Update to include Accommodation Voucher.
18-Oct-2021	Update to indicate other voucher types
30-Nov-2021	Update to include BASC Voucher type
6-Dec-2021	Minor text corrections

Introduction

This document describes API endpoints that online platforms can use to redeem NSW Government's stimulus vouchers administered by Service NSW.

Getting Started

Service NSW provides a sandbox environment for you to perform development work with the APIs. Here is what you will need to integrate with the APIs.

Service Providers need to follow these steps to register as a provider.

To redeem vouchers online platforms and service providers need to:

Contact us (vouchersapi@service.nsw.gov.au) for the following:

- Base url:
 - Sandbox (Test/Staging) environment:
<https://api-psm.g.testservicensw.net/var-voucher-redemptions-external-api>
 - Production environment:
<https://api.g.service.nsw.gov.au/var-voucher-redemptions-external-api>
- Authorization
 - We will create a CLIENT_ID and CLIENT_SECRET for you when you come onboard.
 - You will need to use these details to authenticate using the OAuth protocol to generate an access token which will be used in the API.
 - These will be different for the sandbox and production environments.
- Business ID – a business ID will be provided for each business that is registered to redeem vouchers. This will need to be sent in API calls and identifies which business is redeeming the vouchers.
- IP address range:
 - This identifies your online platform back end which is authorized to make the calls to the voucher redemption API. Calls from other addresses will not be accepted.
 - We do not support a web front end to communicate directly with the vouchers API. The API should be called from your server. No CORS headers will be set on responses from the API.

Ready for Go Live

When you are ready for go live, you need to perform testing in the non-production environment to demonstrate your readiness. Following a successful demonstration, you will be given access to the production environment to go live.

Documentation

Authentication

Authentication to use the API is by the OAuth protocol. You will be issued with a CLIENT_ID and CLIENT_SECRET with which you need to call the authorization service to get an access token. This access token will have a limited life so you will need to refresh the token when it expires.

Permission to use various voucher types and for which businesses vouchers can be redeemed using this service will be restricted according to these credentials. The CLIENT_ID, CLIENT_SECRET and access token should not be exposed to customers web browsers and API traffic should be sent from your back-end systems so as not to disclose your API credentials.

Rate Limits

Rate limits will be applied to voucher balance checks and redemptions. We can work out rate limits to align with expected volumes for a provider.

Rate limits will be lower for manually entered short 8-to-10-character codes than for the full-length strong codes from the QR code (for fraud checks/enumeration prevention).

General API formatting

The API formats data in JSON and normal JSON rules and conventions should be followed.

Numbers are formatted as per the [JSON specification](#).

Versions

Now there is only one available version of this API. This is labelled as v2. There was a previous v1 of the API for Active Kids and Creative Kids vouchers. At this time Active Kids and Creative Kids vouchers cannot be used with this version of the API.

Transaction History

Currently the API does not supply transaction history. To see or download transaction history you need to log into your Business Profile on the Service NSW website.

Voucher Types

The following are the current voucher types available. You should configure your system to be able to easily add new voucher types when/if the government announces them. You should rely on the API balance check to tell you what amount is remaining on the voucher rather than assuming the value of a voucher based on the voucher type.

The voucher type code is encoded in the QR code that can be scanned. Alternatively, if the customer types in the code manually, you can determine the voucher type by the first letter in the manual code.

Voucher Type	Manual Codes start with	Program Name
CSG	g	COVID-19 Stimulus 'Discover NSW' voucher for entertainment venues
CSE	e	COVID-19 Stimulus 'Dine NSW' voucher for restaurants venues
CSA	a	COVID-19 Stimulus 'Stay NSW' vouchers.
DEMO	d	DEMO voucher Is often used in examples here but will be disabled in production. Normally it is best to test with the same voucher type you need to use in

		production. This may have different rules applied to it which make it useful for testing. (For instance, it has no day of week or public holiday redemption restrictions)
LTS	f	First Lap Program (Learn to swim vouchers)
BSC	ba	Before and After School Care vouchers
CSP	p	Parents NSW Voucher
BTS	BS	Back to School

Note. To prepare for future voucher program launches, you should integrate in a way that will allow you to handle additional “prefixes” and “voucher types” over time.

Authentication endpoint

To use the other parts of the API you will need to authorize with your CLIENT_ID and CLIENT_SECRET using the OAuth protocol and the authorization endpoint. This is only an indicative example and any implementation should use the appropriate and up-to-date OAuth libraries for the implementing platform.

OAUTH Endpoint: \$BASE_URL/v1/identity/oauth

NOTE: The OAUTH server does not include the path segment [var-voucher-redemptions-external-api](#) that the api endpoints do, see example below.

Example

```
IDENTITY_BASE_URL="https://api-psm.g.testservicensw.net/v1/identity/oauth"
curl "https://api-psm.g.testservicensw.net/v1/identity/oauth/client-credentials/token" \
-H "Content-Type: application/x-www-form-urlencoded" \
-H "Accept: application/json" \
--data-raw \
"grant_type=client_credentials&scope=&client_id=$VM2_CLIENT_ID&client_secret=$VM2_CLIENT_SECRET"
```

Example Response

```
{
  "access_token": "VPPFeX1oyhMJTEhHN1GgMfmf15XX",
  "expires_in": 7099,
  "token_type": "Bearer"
}
```

The access token returned can be added on the API calls as the Authorization header preceded by the token type as shown below in the Balance and Redeem API call examples.

The access token has a limited validity (noted by the expires_in result) and you must get a new access token when this expires.

Balance Endpoint

GET /v2/vouchers/{voucherTypeCode}/balance?code={voucherCode}

Request Headers

Key	Value
x-business-id	{the business id supplied from your registration}. This will be formatted as a 128 bit UUID.
Authorization	Bearer <access_token>

Request Parameters

Parameter	Description	Type	Example
code	Full voucher code (get from scanning QR) or short 8-12 alphanumeric printed on the voucher. All short codes manually typed are case-insensitive. Any code scanned from the QR code should have the case preserved. The long code will be a UUID or consist of base64 characters and will be of length <=40.	String	"20e405f1-f48c-4fee-bd85-cdcaec6fa057" or "gaB43cQ1c"
voucherType	CSE, CSG, CSA, LTS, BSC, BTS... See Voucher Type table above	String	"BTS"

Notes:

Body parameter names are case-sensitive.

Example

`BASE_URL="https://api-psm.g.testservicensw.net/var-voucher-redemptions-external-api"`

```
curl --location --request GET \  
'$BASE_URL/v2/vouchers/BTS/balance?code=20e405f1-f48c-4fee-bd85-cdcaec6fa057' \  
--header 'x-business-id: {the providers busines id, e.g 435a1d79-7124-45f9-aa3b-1d811b7a4bcc}' \  
--header 'Authorization: Bearer {access-token}'
```

Responses when the voucher is valid

Status Code: 200 Success

```
Body: {  
  "balance": 25.0  
}
```

Status Code: 200 Success

```
Body: {  
  "balance": 100  
}
```

For some drawdown voucher types, a maximum redemption may be in effect which is the maximum allowed redemption at one time.

Status Code: 200 Success

```
Body: {  
  "balance": 100.0,  
  "maximumRedemption": 25.00  
}
```

Notes:

- A successful Balance Check does not lock the voucher to this session or put it into a 'pending' state. If users e.g. several customer looking at registration platform(s) use several browser tabs or devices to use the same voucher code several times, then Balance Check will succeed everywhere until a first (and only one single one) Redeem call has succeeded no matter in which session.
- If rules for a voucher program permit using multiple for a customer per day this will only be effective in a balance check if those vouchers have already been redeemed. For instance, for Dine and Discover a customer cannot redeem more than one voucher per day with a vendor., but until one voucher is redeemed the balance check will give a success, so if you do 2 balance checks before redeeming a redemption could fail because of this rule and you might need to roll back.
- Don't unnecessarily hard code expectations about the available balance. Voucher program amounts may change over time.
- The maximumRedemption value in the response may or may not be present, if it is not present the maximum redemption amount is the whole balance. (This is the case for all current voucher programs)
- Don't unnecessarily code assumptions about the formatting on amount number (decimal places) you should follow the JSON number formatting conventions (allow integer or with decimals and cents) (Some programs may be drawdown voucher programs and so the remaining value might not be an integer, but may have cents)

For errors that occur on the balance check API see the list below the redeem API.

Redeem Endpoint

POST /v2/vouchers/{voucherTypeCode}/redeem

Request Headers

Key	Value
Content-Type	application/json
Authorization	Bearer <access_token>
x-business-id	{the business id supplied from your registration}. This will be formatted as a 128 bit UUID. e.g. "435a1d79-7124-45f9-aa3b-1d811b7a4bcc"

Request Body

Body Parameter	Description	Type	Example
amount	Amount to be redeemed from voucher	Float	25.0
voucherCode	Full voucher code (get from scanning QR) or short 8-12 alphanumeric printed on the voucher. All short codes manually typed are case-insensitive. Any code scanned from the QR code should have the case preserved. The long code will be a UUID or consist of <u>base64 characters</u> and will be of length <=40.	String	"20e405f1-f48c-4fee-bd85-cdcaec6fa057" or "dshdfS524aB+/1" or "aB43cQ1c"
voucherType	CSE, CSG, CSA, BSC, CSP, BTS... See voucher type codes list	String	"BTS"
externalReference	The online platform's reference number. (Optional) Max length: 20 chars	String	"CODE FROM SNSW"
providerIdentifier	Should be the same as the header 'x-business-id' on balance check and will be checked the authorization details.		"435a1d79-7124-45f9-aa3b-1d811b7a4bcc"
totalAmount	The total amount of the invoice the voucher is being used on before the application of the voucher. If there is a decimal point the value	Float	50.51

	<i>should be a whole number of cents. (i.e. A value of 20.125 would be invalid so a rounded value of 20.13 should be sent)</i>		
metadata	<p><i>(optional)</i> <i>This is a key value pair for extra data needed for a voucher program. Some programs may require data here.</i></p> <p><i>API users could put a value here that could be seen by SNSW support teams if needed, and the externalReference is not suitable.</i></p>	JSON object, key value pair	<pre>{ "key1": "test", "startDate": "2021-03-01", "endDate": "2021-03-05", "postcode": "2000" }</pre>

Note: It is recommended to incorporate all the above values into your response in order to assist with any redemption reconciliation/support.

Example:

`BASE_URL="https://api-psm.g.testservicensw.net/var-voucher-redemptions-external-api"`

`curl --location --request \`

`POST "$BASE_URL/v2/vouchers/DEMO/redeem" \`

`--header 'Authorization: Bearer {apigee-app-token}' \`

`--header 'x-business-id: {businessId: e.g 435a1d79-7124-45f9-aa3b-1d811b7a4bcc}' \`

`--header 'Content-Type: application/json' \`

`--data-raw '{`
 `"voucherCode": "20e405f1-f48c-4fee-bd85-cdcaec6fa057",`
 `"providerIdentifier": "435a1d79-7124-45f9-aa3b-1d811b7a4bcc",`
 `"amount": 25,`
 `"totalAmount": 124.0,`
 `"externalReference": "MyInvoice-1234"`
`}'`

Response for a successful redeem

Status Code: 200 OK

A successful redeem returns a transaction code to uniquely identify this transaction

```
{
  "transactionCode": "d1460b54-c614-4579-8170-aec9d30d5928",
  "status": "REDEEMED"
}
```

Status Code: 202 ACCEPTED

The voucher has not yet been redeemed but is awaiting confirmation.

This is not expected for Dine and Discover vouchers.


```
{
  "transactionCode": "d1460b54-c614-4579-8170-aec9d30d5928",
  "status": "ACCEPTED"
}
```

Note 1 : **transactionCode** is a unique transaction code we return to registration platforms. You may supply your own reference code in the **externalReference** field, and it will be available in the Business Profile transaction summary on the Service NSW website.

Error codes (For balance check and redeem calls)

An example response for an invalid voucher code is:

Status Code: 400 Bad Request

Body:

```
{
  "message": "The voucher has been used",
  "error": "VOUCHER_HAS_BEEN_USED",
  "status": 400,
  "errorCode": 1007,
  "path": "/v2/vouchers/BTS/redeem",
  "timestamp": "2020-12-22T09:52:56.088+11:00"
}
```

When coding your system to display messages it is recommended you use the **http status** or **errorCode** to do this. The SNAKE_CASE error description is for developers to quickly understand the issue and may not always be provided in a production environment for all errors and could potentially change in future versions.

Error Code	Http Status	Error	Notes
1001	404	VOUCHER_NOT_FOUND	The voucher code could not be found for the specified voucher type.
1003	404	VOUCHER_SERVICE_PROVIDER_NOT_FOUND	<p>The Business ID identifying the provider who is redeeming the voucher was not matched with a valid business.</p> <p>This could mean the business ID you are using is not registered for the voucher type you are trying to redeem. If the business ID issued to you is not working, please contact SNSW support.</p>
1005	400	INVALID_AMOUNT	An amount outside the rules of the program was requested. Programs may apply minimum and maximum amounts.

1006	400	PROVIDER_IS_INACTIVE	The registration of the business redeeming the voucher is not Active.
1007	400	VOUCHER_HAS_BEEN_USED	For a voucher type that is a single use voucher that has been redeemed or if a drawdown voucher has been completely used.
1008	400	VOUCHER_HAS_EXPIRED	It is after the expiry date on the voucher.
1011	400	VOUCHER_REDEMPTION_NOT_REDEEMABLE_TODAY	The voucher type has restrictions on which days of the week it can be redeemed, and an attempt was made to redeem it when not allowed.
1013	409	CUSTOMER_IS_ALREADY_REDEEMED_FOR_THIS_BUSINESS_TODAY	The customer has already redeemed a voucher for this business today. Configurable by product
1014	400	VOUCHER_REDEMPTION_NOT_REDEEMABLE_ON_PUBLICHOLIDAY	The voucher type has restrictions on Public Holidays and an attempt was made to redeem it when not allowed.

VOID Redemption

GET /v2/vouchers/{voucherType}/redeem/{transactionCode}/void

This API will reverse a redemption that was made. Use this if other parts of the transaction were unable to complete such as a credit card co-payment, or seat allocation call. Note that there is a limited time to make this **void** call after the **redeem** call (which depends on the voucher program).

For the Dine & Discover NSW program this limit is 10 minutes.

Some voucher programs may not allow a redemption to be voided.

Request Header	Value		
Authorization	Bearer <access_token>		
x-business-id	{the business id supplied from your registration}. This will be formatted as a 128-bit UUID. e.g. "435a1d79-7124-45f9-aa3b-1d811b7a4bcc"		
Path Parameters	Description	Type	Example
transactionCode	The transaction code returned from the redeem API	String	"d1460b54-c614-4579-8170-aec9d30d5928"
voucherType	CSE, CSG, CSA, BSC, CSP or BTS See voucher type codes list Must be the same as the redeem call being voided.	String	"BTS"

Sample call

```
curl -X GET "https://api-psm.g.testservicensw.net/var-voucher-redemptions-external-api/v2/vouchers/CSE/redeem/d1460b54-c614-4579-8170-aec9d30d5928/void" -H "accept: application/json" -H "Authorization: Bearer XXXXX" -H "x-business-id: XXXXXXXX"
```

Sample Response

The response is the voucher redemption status following the void if successful will be "VOID"

Errors

1015	400	VOID_IS_NOT_ALLOWED_ON_THIS_PRODUCT	A Redemption cannot be voided on this product.
1016	422	VOID_IS_NOT_ALLOWED_AFTER_TIME_LIMIT	A Void was requested on a redemption after more than the time limit for voiding has passed. A redemption must usually be voided within the time limit for that voucher type. (e.g. 10 minutes for Dine and Discover NSW)
1010	404	VOUCHER_REDEMPTION_NOT_FOUND	No redemption with the specified code could be found

Sample Vouchers

`GET /v2/vouchers/{voucherTypeCode}/sample?tempCode=true`

This API call is only available in the non-production environments and allows developers to generate test vouchers for use in the other API calls. You can either generate a temporary code to simulate the 10-minute limited of the codes on the screen of the app, or a permanent code to simulate the code that will appear on a printed paper voucher.

A *temporary code* will have a **codeExpiry** property. Normally you will not see this code expiry, but the SNSW app will stop showing the QR code some time before it expires, and a new code will appear on the screen. User instructions could ask users to wait for a fresh code to appear on the screen. An error will be returned if attempting to use a voucher code that has expired. **Note: Temporary codes will not be available for the accommodation vouchers.**

Request Headers

Key	Value
x-business-id	{the business id supplied from your registration}. This will be formatted as a 128 bit UUID.
Authorization	Bearer <access_token>
Accept	Can be <code>*/*</code> or <code>application/json</code> to get a json response. If <code>image/png</code> is given you just get a PNG image version returned.

Request Parameters

Parameter	Description	Type	Example
tempCode	(optional) false=Get a permanent code with no time limit, equivalent to paper voucher. true=Get a temporary code with a 10 minute time limit (default) The format and length of the codes produced will differ slightly – see the voucher code description. You should test your implementation with both types of code and also try changing the case of alpha characters in code to see the impact of case on different code types.	Boolean (Query String)	true false
voucherType	CSE, CSG, CSA, BSC, CSP The voucher program type code from the table at the top of this documentation	String (Path)	"BTS"

Notes:

Body parameter names are case-sensitive.

Example – getting sample voucher details in JSON

```
curl --location --request GET 'https://api-psm.g.testservicensw.net/var-voucher-redemptions-external-api/v2/vouchers/DEMO/sample' \
--header 'Authorization: Bearer GmjHwNu4y5ttbJXQPKZqALZ6jpoA'
```

Sample Response

```
{
  "voucherType": "DEMO",
  "voucherCode": "malhIXp4aYYH1Bui",
  "codeExpiry": "2020-11-24T19:22:19",
  "shortCode": "malhIXp4",
  "amount": 100.0,
  "expiryDate": "2020-12-31",
  "nameOnVoucher": "",
  "voucherPeriod": "2020",
  "issue": 1,
  "status": "NEW",
  "statusTimestamp": "2020-11-24T19:12:19.77",
  "qrCodeContent": "eyJ2Ijp7ImQiOiJ1RTRLWGR4MEZVcm1lMWVxIiwidCI6IkkNTRSj9LCl0cyI6IjIwMjA6MzQ6MDluMDE0NjYyIiwibSI6Ikl1UaGpPVFV6WwVpNdFIUQTFAaTAwTURneExXSXdZamd0TVRCak0yTXhaRGxrWVdVeSj9",
  "qrCodePngBase64": "iVBORw0KGgoAAAANSUhEUgAAAPAAAADwAQAAAAAWLlQ/AAACuklEQVR4Xu2X3WorMRCDB/xahn1g1/L4KNPm2zJaehNO76K6c/ackFIgm0a+8cV/x+8rg/8dn3gt+vX8Iqlru+2ehv6yjlycZAH4Nyr955zCp97cC96cl4Pa9N11HXetnht3em+cQjmrPshissj8E5euTCmRxtiuEp+MsSXdGmkYcvS0phovhtvSb12/oj2EsBJPy52mhLucjruBpe2ueE4WMQU1OoSD6Z18JDaVAMCCMauQS6yVXD23pMKColEGzmdgaWFuHWUQvkAMccuJXD95hBHPTO2LHSDWw3tOSMkMrPAPLEGXcngQhTZO4sncAaZDsCyXk5IElyQJB4A9V7Ww4u2Fy8FAWF40YZr96JWCksX7a78DfquMZXdM1gOKwxKn1zw2875F7N7DCphb3nbgFNCOKeQuFXD8iSudzqeNC76Zf9MSyUMQL1TfhZl1xhBKFFD0v/KILMfHoG4ftfDOK/QixtZ6NyEGl6UwxSTeL3oHm6lGA9PaqWw2CRbVTwPbgCoMf/1sKXY9wt2cSnuZqqEyeLwVq1PDOZQNDyZ5TCPGoBbm8ezs1gOq+zFDW0onkHXXxV4AlaaNqoA1QAN7OcXWapg73VwOdIlg2vjCKzOQYo0OK1MuocPwGgQ4Zm3K2qe7si9AluGhFD2kx/BDMicW5ZKWKrgv4LfcIXa1xWrUg5LFDDeu6Lj3l9uez3cHYKmpvpvf4B3HQDdpvn4AnaigGnR6QPh4KRuMA7Kd5ZQJ+IB+FdABGEFcdHyftTSsS2swDMEvuN7PjYXoEG0NYDTP6yRWFb61vbfizcjilP3qkqU3FwdaAl8MqOy1B8Al2w5taNZxOAE50ckAN4MghWG8ZaZGUreYCdSBXDmOJOCUlo0wv/knfOHMAJn28XdXyCIIfnslHtUr4p/WB364P/HaVwv8A44TTHPWIArcAAAAASUVORK5CYII="
}
```

The QR Code content fields should help you to test scanning the codes (or use the alternate example below to see the image). Scanning the code should be easier to use, quicker and more secure in a POS or web browser situation. It could also be done on the web if the browser device is fitted with a camera (there are many client and server side tools for processing of qr codes that could be used).

Example 2 – getting a QR code image (for test scanning)

```
curl --location --request GET 'https://api-psm.g.testservicensw.net/var-voucher-redemptions-external-api/v2/vouchers/DEMO/sample' \
--header 'Authorization: Bearer GmjHwNu4y5ttbJXQPKZqALZ6jpoA' \
--header 'Accept: image/png'
```

Sample Response (in postman)

Get Sample Voucher Image

GEThttp://localhost:8080/v2/vouchers/CSE/sample?tempCode=false

Params

Authorization

Headers (10)

Body

Pre-request Script

Tests

Settings

Headers <- 7 hidden

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> correlation-id	{{guid}}	
<input checked="" type="checkbox"/> x-app-name	local-postman	
<input checked="" type="checkbox"/> Accept	image/png	
Key	Value	Description

Body


Cookies

Headers (11)

Test Results

Status: 200 OK

Time: 35



QR Code for Voucher

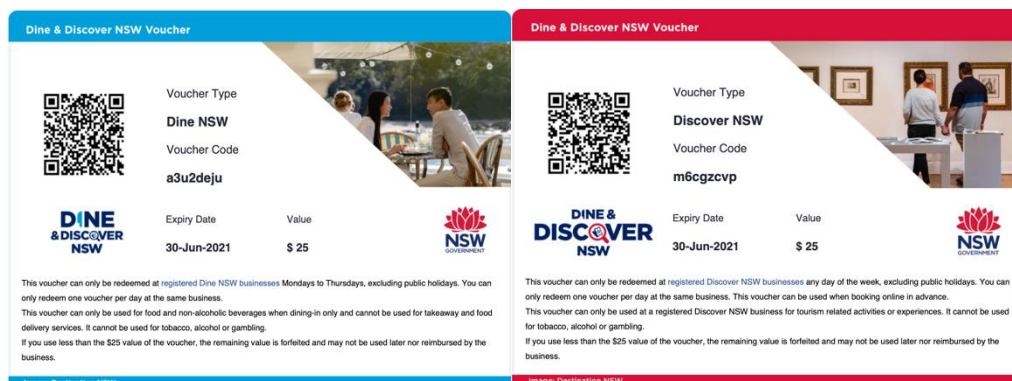
The voucher from Service NSW is either presented as a QR code printed on paper (as a PDF) on application, or as a QR code shown on the screen of the customer's phone in the MyService NSW App. The version on the phone uses a temporary code which will only be redeemable for a short time after being initially shown. Both codes have a short 8-character alphanumeric code at the bottom. The short code can be entered by hand, or the QR code can be scanned by a device camera and the fields extracted from it.

Stay NSW vouchers examples below -



Dine & Discover examples -

The vouchers and QR codes appear like this:



The QR code(s) in the above screen shots are mock-up, and not examples of actual codes.

Please see the below QR code for an example. The code on a paper voucher will be slightly different. In the final version, a short 8-character voucher code will appear below the QR code for manual entry - e.g., **8aXb6yC5**. The user will need to tap the screen to get the short code, which can take the place of the voucherCode in the API. However, for short codes a lower rate limit will be applied due to the lower entropy.



The QR code will be like this containing a json string (encoded in base64), and the codes from the app must be used 10 minutes from being initially shown on the screen. It will also show a short 8-character code which will initially be masked and only revealed when the user touches it. The printed voucher will not expire in 10 minutes and have a slightly different format and will not have the *ts* or *m* elements in the json.

The above QR code will scan with the Base64 test

```
eyJ2Ijp7ImQiOiJmWVRjSUU2ZndQalY3eFpHIiwidCI6IkNTRSJ9LCJ0cyI6IjIwMjAtMTItMDNUMDQ6NTE6MjUuMTMxWiIsIm0iOiIxNWE3ZjlmNjc5YTk2YzY3OGJjNDRkn2M0Y2Q4OGZlYTU3ZWRhMzkzY2IzNTYyMzhjNDUzNzZlYmIwEwIn0=
```

Decode the base64 encoding to get this JSON:

```
{"v":{"d":"61c265bf-4c21-4f73-a1ac-f86f6e3aa02f","t":"CSG"},"ts":{"a timestamp"},"m":"/4Ci3HJCmHLhBXnU6gxTxA=="}
```

Extract the **voucher code** “v.d” and **voucher type** “v.t” from the JSON:

```
61c265bf-4c21-4f73-a1ac-f86f6e3aa02f  
with voucher type of ‘CSG’ – a Discover NSW voucher.
```

The voucher code and the voucher type may be used in the API. Use the voucher type to determine if you will accept the voucher and if it is the right voucher type for your product.

The voucher code may be a UUID or an encoding using the base64 character set of up to 40 characters.

The additional fields *m* and *ts* in the extracted json can be ignored for this API.

Errors

This section shows shared errors which could occur for any of the methods. Errors specific to a particular API will appear in the description of that method.

Most errors will be of this format:

Status Code: 400 Bad Request

Body:

```
{
  "message": "The voucher has been used",
  "error": "VOUCHER_HAS_BEEN_USED",
  "status": 400,
  "errorCode": 1007,
  "path": "/v2/vouchers/CSE/redeem",
  "timestamp": "2020-12-22T09:52:56.088+11:00"
}
```

Response when service (Balance Check & Redeem) is unavailable

Some error codes and messages are:

9001	VOUCHER_TYPE_DENIED_ACCESS	If voucher type is not registered (allowed) for the current client
9002	BUSINESS_DENIED_ACCESS	If business ID is not registered (allowed) for the current client

Status Code: 503 Service Unavailable

Body: {

```
  "message": "Service not available",
}
```

Response when Rate limit is violated

Status Code: 429 Too Many Requests

```
{
  "fault": {
    "faultstring": "Spike arrest violation. Allowed rate : MessageRate{messagesPerPeriod=1, periodInMicroseconds=1000000, maxBurstMessageCount=1.0}",
    "detail": {
      "errorCode": "policies.ratelimit.SpikeArrestViolation"
    }
  }
}
```

```
}
```

```
}
```

Response when the OAUTH access token has expired
Status Code: 401 Unauthorized

```
{
```

```
  "fault": {
```

```
    "faultstring": "Access Token expired",
```

```
    "detail": {
```

```
      "errorcode": "keymanagement.service.access_token_expired"
```

```
    }
```

```
  }
```

```
}
```

References

An OpenAPI specification file is available