

Space Link 1



CONTENTS

1. test	3
2. ALL BLOCKS	4
3. CODE DRAWERS	11
4. Untitled	18
5. EXTERNAL LINK 02	
6. EXTERNAL LINK 01	
7. Test	19
7.1. test 2	20
7.1.1. I changed this	20
8. Virtual Account Number API Specification	21
8.1. Create an Automatcher (Receivables) Account	22
8.2. Get Account Status By Bank Account	23
8.3. Batch Create	24
8.4. Set Account Status	25
8.5. Batch Status	26
8.6. Get Account Status By ClientUniqueld	27
8.7. securitySchemes	28
8.8. schemas	29
8.9. schemas	30
8.10. schemas	31
8.11. schemas	32
8.12. schemas	33
8.13. schemas	34
9. Virtual Account Number API Specification	35
9.1. Create an Automatcher (Receivables) Account	35
9.2. Get Account Status By Bank Account	36
9.3. Batch Create	37
9.4. Set Account Status	38
9.5. Batch Status	39
9.6. Get Account Status By ClientUniqueld	40
9.7. securitySchemes	41
9.8. schemas	42
9.9. schemas	43
9.10. schemas	44
9.11. schemas	45
9.12. schemas	46
9.13. schemas	47



1. test



2. ALL BLOCKS

Heading 1

▼ Expandable Heading 1

Expandable Heading 1 Content

Heading 2

▼ Expandable Heading 2

Expandable Heading 2 Content

Heading 3

 **Figma embed**

Add Embed

▼ Expandable Heading 3

Expandable Heading 3 Content

- ☐ Checklist
- ☐ Checklist 2
- ☐ Checklist 3
- ☐ ALL BLOCKS

TABLE	TABLE	TABLE
TABLE	TABLE	TABLE
TABLE	TABLE	TABLE
TABLE	TABLE	TABLE

BUTTON

Callout

VERTICAL SPLIT, LEFT SIDE

Doing

MINI TASK 1

Testing

MINI TASK 2

Done

MINI TASK 3

VERTICAL SPLIT, RIGHT SIDE



LINK GRID 1



LINK GRID 2



LINK GRID 3

1 Workflow 1



Content of workflow 1

2 Workflow 2

Content of workflow 2

TAB 1

Content of Tab 1

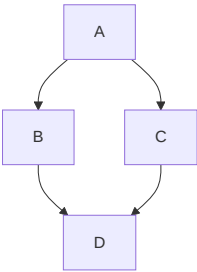
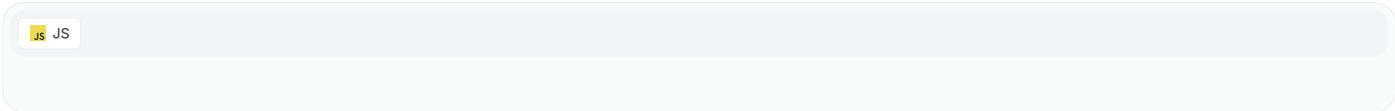
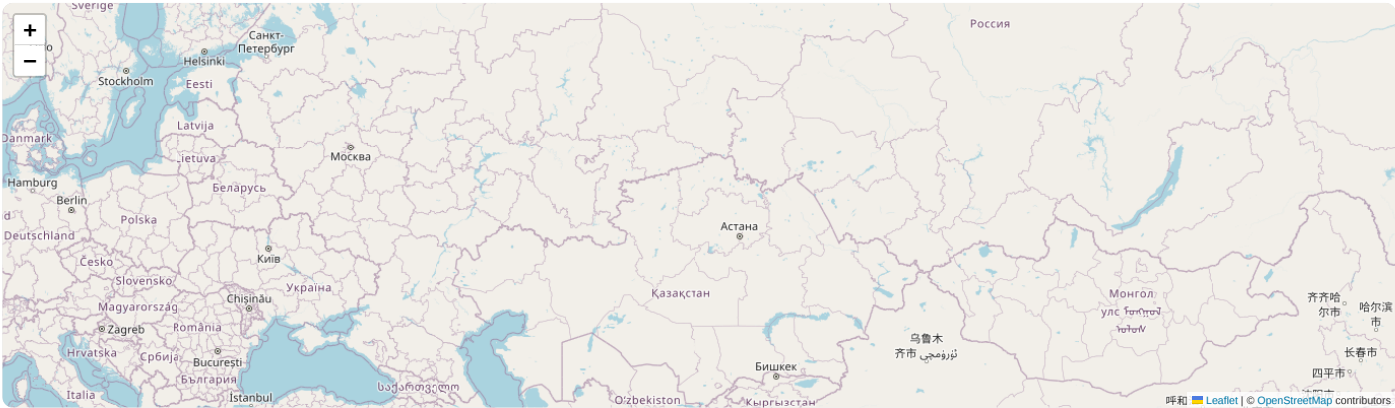
TAB 2

Content of Tab 2

New Tab

New Tab

- Bullet list 1
 - Bullet list 2
 - Bullet list 3
- 1. Numbered list 1
 - 1. Numbered list 2
 - 1. Numbered list 3



$\int_0^i n f t y x^2 d x$



Get Cakes

Get a cake by its ID

GET

https://api.cakes.com

Request

BODY PARAMETERS

id

String

required

ID of the cake to get

JS JS

```
var myHeaders = new Headers();
myHeaders.append("Accept", "application/json");
myHeaders.append("Content-Type", "application/json");

var raw = JSON.stringify({
  "id": "String"
});

var requestOptions = {
  method: 'GET',
  headers: myHeaders,
  body: raw,
  redirect: 'follow'
};

fetch("https://api.cakes.com", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Responses

200

```
{
  "name": "Cake's name",
}
```

1.0.7 OAS 2.0

Swagger Petstore

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [#swagger">irc.freenode.net, #swagger](irc.freenode.net). For this sample, you can use the api key **special-key** to test the authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Schemes

HTTPS

Authorize

pet Everything about your Pets

Find out more ^

POST

/pet/{petId}/uploadImage uploads an image



C

POST

/pet Add a new pet to the store





C

PUT **/pet** Update an existing pet  

C

GET **/pet/findByStatus** Finds Pets by status  

C

GET ~~**/pet/findByTags**~~ Finds Pets by tags  

C

Warning: Deprecated

GET **/pet/{petId}** Find pet by ID  

C

POST **/pet/{petId}** Updates a pet in the store with form data  

C

DELETE **/pet/{petId}** Deletes a pet  

C

store Access to Petstore orders 

GET **/store/inventory** Returns pet inventories by status  

C



POST **/store/order** Place an order for a pet

C

GET **/store/order/{orderId}** Find purchase order by ID

C

DELETE **/store/order/{orderId}** Delete purchase order by ID

C

user

Operations about user

[Find out more about our store](#)

POST **/user/createWithList** Creates list of users with given input array

C

GET **/user/{username}** Get user by user name

C

PUT **/user/{username}** Updated user

C

DELETE **/user/{username}** Delete user

C

GET **/user/login** Logs user into the system

C



GET

/user/logout Logs out current logged in user session

^

C

POST

/user/createWithArray Creates list of users with given input array

^

C

POST

/user Create user

^

C

Models

^

ApiResponse

▼

 {
 code > [...]
 type > [...]
 message > [...]
}

Category

▼

 {
 id > [...]
 name > [...]
}

Pet

▼

 {
 id > [...]
 category Category > {...}
 name* > [...]
 photoUrls* > [...]
 tags > [...]
 status > [...]
}

Tag

▼

 {
 id > [...]
 name > [...]
}

Order

▼

 {
 id > [...]
 petId > [...]
 quantity > [...]
 shipDate > [...]
 status > [...]
 complete > [...]
}



```
User {  
  id > [...]  
  username > [...]  
  firstName > [...]  
  lastName > [...]  
  email > [...]  
  password > [...]  
  phone > [...]  
  userStatus > [...]  
}
```

1 {
2 status,
3 people
4 }

▶

Variables

Headers

GraphQL

Chsadadsada

- AddedContent 1
- FixedContent 2



3. CODE DRAWERS

Heading 1

▼ **Expandable Heading 1**

Expandable Heading 1 Content

Heading 2

▼ **Expandable Heading 2**

Expandable Heading 2 Content

Heading 3

 **Figma embed**

Add Embed

▼ **Expandable Heading 3**

Expandable Heading 3 Content

- ☐ Checklist
- ☐ Checklist 2
- ☐ Checklist 3
- ☐ **ALL BLOCKS**

TABLE	TABLE	TABLE
TABLE	TABLE	TABLE
TABLE	TABLE	TABLE
TABLE	TABLE	TABLE

BUTTON

Callout

VERTICAL SPLIT, LEFT SIDE

Doing

MINI TASK 1

Testing

MINI TASK 2

Done

MINI TASK 3



LINK GRID 1



LINK GRID 2



LINK GRID 3

1 Workflow 1



Content of workflow 1

2 Workflow 2

Content of workflow 2

TAB 1

Content of Tab 1

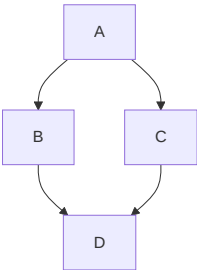
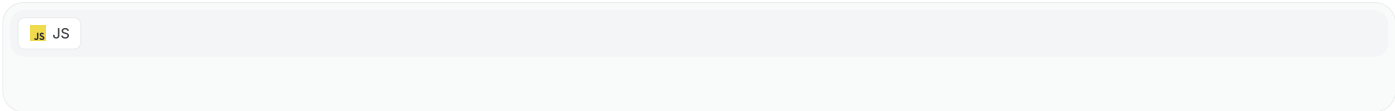
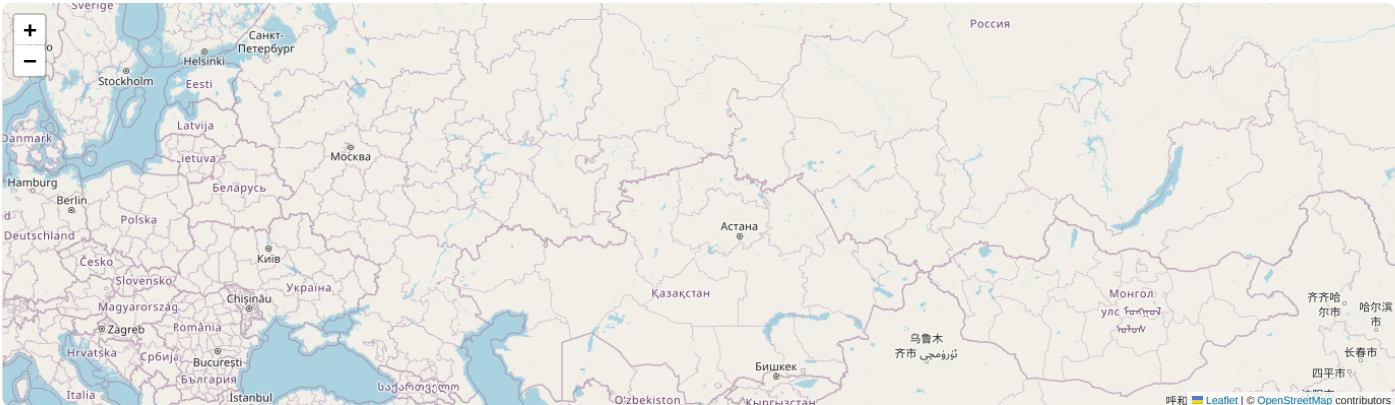
TAB 2

Content of Tab 2

New Tab

New Tab

- Bullet list 1
 - Bullet list 2
 - Bullet list 3
- 1. Numbered list 1
 - 1. Numbered list 2
 - 1. Numbered list 3



$\int_0^i n f t y x^2 d x$



Get Cakes

Get a cake by its ID

GET

https://api.cakes.com

Request

BODY PARAMETERS

id

String

required

ID of the cake to get

JS JS

```
var myHeaders = new Headers();
myHeaders.append("Accept", "application/json");
myHeaders.append("Content-Type", "application/json");

var raw = JSON.stringify({
  "id": "String"
});

var requestOptions = {
  method: 'GET',
  headers: myHeaders,
  body: raw,
  redirect: 'follow'
};

fetch("https://api.cakes.com", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

Responses

200

```
{
  "name": "Cake's name",
}
```

1.0.7 OAS 2.0

Swagger Petstore

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [#swagger](irc://freenode.net). For this sample, you can use the api key **special-key** to test the authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Schemes

HTTPS

Authorize



pet Everything about your Pets

Find out more ^

POST /pet/{petId}/uploadImage uploads an image



C

POST /pet Add a new pet to the store





C

PUT **/pet** Update an existing pet

C

GET **/pet/findByStatus** Finds Pets by status

C

GET **/pet/findByTags** Finds Pets by tags

C

Warning: Deprecated

GET **/pet/{petId}** Find pet by ID

C

POST **/pet/{petId}** Updates a pet in the store with form data

C

DELETE **/pet/{petId}** Deletes a pet

C

store Access to Petstore orders

GET **/store/inventory** Returns pet inventories by status

C



POST	/store/order	Place an order for a pet	^
C			
GET	/store/order/{orderId}	Find purchase order by ID	^
C			
DELETE	/store/order/{orderId}	Delete purchase order by ID	^
C			

user Operations about user

[Find out more about our store](#) ^

POST	/user/createWithList	Creates list of users with given input array	^
C			
GET	/user/{username}	Get user by user name	^
C			
PUT	/user/{username}	Updated user	^
C			
DELETE	/user/{username}	Delete user	^
C			
GET	/user/login	Logs user into the system	^
C			



GET

/user/logout Logs out current logged in user session

^

C

POST

/user/createWithArray Creates list of users with given input array

^

C

POST

/user Create user

^

C

Models

^

ApiResponse

▼

 {
 code > [...]
 type > [...]
 message > [...]
}

Category

▼

 {
 id > [...]
 name > [...]
}

Pet

▼

 {
 id > [...]
 category Category > {...}
 name* > [...]
 photoUrls* > [...]
 tags > [...]
 status > [...]
}

Tag

▼

 {
 id > [...]
 name > [...]
}

Order

▼

 {
 id > [...]
 petId > [...]
 quantity > [...]
 shipDate > [...]
 status > [...]
 complete > [...]
}



```
User {  
  id > [...]  
  username > [...]  
  firstName > [...]  
  lastName > [...]  
  email > [...]  
  password > [...]  
  phone > [...]  
  userStatus > [...]  
}
```

1 {
2 status,
3 people
4 }

▶

Variables

Headers

GraphQL

Chsadadsada

- AddedContent 1
- FixedContent 2



4. Untitled

zl,fddsl;kf msdf;kdsf l;sdfkksd l;kf sdk;if sdlk;fsdl; fel, fsdlf kl;dsfk dspf kso ks lfost lkdsif sk kdsif sdf sdf sdklfkds fkd ds fsdkl;f sdlk;f sd sdfsdfdf kl;sdl k;dskfsdlf dsfdfsdfs sf dsf dsfdfs
sf,elrw l;'sf,sdl ,e, rwrkl e;rw,k ;dlsk f;sdlf dsif ksfl;sd kdskf sd kf ;dsfl dskf ldsfkl;ekw kds fld;sk fdl;s fkdsf fkd;f;f sdlf sd

- a.
- b.



7. Test



7.1. test 2

7.1.1. I changed this

Yeast output will be calculated by taking a sample of the fermentation broth when the fermentation has ended, and determining the (dry) yeast content per volume of the broth. This then needs to be multiplied with the final volume of broth in the fermenter when the fermentation ended, to obtain the total amount of yeast.

sadsada

$$m_{\text{yeast}} = \left(\frac{m_{\text{sample,dry}}}{V_{\text{sample}}} \right) \cdot V_{\text{final}}$$

The fermenter yield is a measure of how efficiently the carbon source (molasses, typically) has been "converted" to yeast biomass. This will be calculated by dividing the yeast output by the fed molasses, on dry weight basis.

$$Y_{\text{fermenter}} = \frac{m_{\text{yeast}}}{m_{\text{molasses}}}$$

Let's GO GYYYYYMMMMMM

/



8. Virtual Account Number API Specification



8.1. Create an Automatcher (Receivables) Account

POST<https://api.mpay.com.au/receivables/v1/create>

Try it

Production URL

Creates a new virtual account number. These account numbers can be used along with the returned BSB to receive payments from any bank in Australia.

Payments to an invalid bank-account number, i.e. one that has not been created, will be returned by Monoova to the remitting institution. Each account number has a check digit to minimize common payer errors such as adjacent transposition errors or single-digit typos.

An optional unique ID can be linked to each account number for your reference. If a unique ID is not provided, Monoova will generate and return one automatically.

If `isActive` is omitted, it will default to `false`. This can be changed to `true` via the `/receivables/v1/status` endpoint.

For the creation of batches of account numbers, please see `/receivables/v1/batchCreate`.

Body Parameters

body `automatcherCreateBody_V1` *

Responses

✓ ● 200 successful validation

`automatcherStatusResponse_V1`

Request

Curl

```
curl --request POST \
  --url https://api.mpay.com.au/receivables/v1/create \
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation
{}
```



8.2. Get Account Status By Bank Account

GET

<https://api.mpay.com.au/receivables/v1/statusByBankAccount/{bankAccountNumber}>

Try it

Production URL

This API is used for checking the status of a bank account using the bank account number. This API will only return statuses for virtual accounts belonging to the parent mAccount.

Path Parameters

> **bankAccountNumber** `string` *

Query Parameters

bsb `string`

Responses

▼ ● 200 successful validation

`automatcherStatusResponse_V1`

Request

Curl

```
curl --request GET \  
  --url https://api.mpay.com.au/receivables/v1/statusByBankAccount/{bankAccountNumber} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation  
{}
```



8.3. Batch Create

POST<https://api.mpay.com.au/receivables/v1/batchCreate>

Try it

Production URL

The batch create API has the same functionality as the `/receivables/v1/create` endpoint however it allows you to create up to 50,000 accounts in a single call by submitting a file. We recommend this for circumstances where more than 100 accounts need to be created within a short time period.

Creation of accounts with `isActive=true` is restricted to 1000. Multiple batches cannot be requested at the same time, you have to wait for the completion of a batch before requesting a new one.

The Batch Create API also supports a lightweight option whereby many accounts can be created by submitting a single line. This option automatically sets `isActive = true`, hence you can only request the creation of 1000 accounts in one request. For this method, all accounts will be created with the same specified name, and `clientuniqueid`'s cannot be specified will be automatically generated.

Responses

 • **200** successful validation`string<binary>`

Request

Curl

```
curl --request POST \
  --url https://api.mpay.com.au/receivables/v1/batchCreate \
  --header 'accept: application/json'
```

Responses

• **200**

```
// successful validation
""
```



8.4. Set Account Status

POST<https://api.mpay.com.au/receivables/v1/status>

Try it

Production URL

Update the status of a virtual account. Accounts can be active or inactive. If an account is inactive, all inbound payments will be rejected and returned to the payer.

Body Parameters

body [automatcherStatusBody_V1](#) *

Responses

▼ ● 200 successful validation

[automatcherStatusResponse_V1](#)

Request

Curl

```
curl --request POST \  
  --url https://api.mpay.com.au/receivables/v1/status \  
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation  
{}
```



8.5. Batch Status

POST`https://api.mpay.com.au/receivables/v1/batchStatus`

Try it

Production URL

Get the status of a request to the batchCreate endpoint.

Responses

▼ ● 200 successful validation

`string<binary>`

Request

Curl

```
curl --request POST \  
  --url https://api.mpay.com.au/receivables/v1/batchStatus \  
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation  
""
```



8.6. Get Account Status By ClientUniqueld

GET

`https://api.mpay.com.au/receivables/v1/statusByClientID/{clientUniqueld}`

Try it

Production URL

This API is used for checking the status of a bank account using the clientUniqueld provided when the account was created.

Path Parameters

> `clientUniqueld string *`

Query Parameters

`bsb string`

Responses

✓ ● 200 successful validation

`automatcherStatusResponse_V1`

Request

Curl

```
curl --request GET \  
  --url https://api.mpay.com.au/receivables/v1/statusByClientID/{clientUniqueId} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation  
{}
```



8.7. securitySchemes

BASICAUTH



Try it

Production URL

Responses

● 200

Request

Curl

```
curl --request BASICAUTH \  
  --url https://api.mpay.com.ausecuritySchemes \  
  --header 'accept: application/json'
```

Responses

● 200



8.8. schemas

AUTOMATCHERCREATEBODY_V1

Production URL

<https://api.mpay.com.auschemas>

Try it

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERCREATEBODY_V1 \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



8.9. schemas

AUTOMATCHERBATCHDETAILSCLASS



Try it

Production URL

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERBATCHDETAILSCLASS \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



8.10. schemas

AUTOMATCHERBATCHRESPONSE_V1

<https://api.mpay.com.auschemas>

Try it

Production URL

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERBATCHRESPONSE_V1 \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



8.11. schemas

DEBUGRESPONSE



Try it ▶

Production URL

Responses

● 200

Request

Curl

```
curl --request DEBUGRESPONSE \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



8.12. schemas

AUTOMATCHERSTATUSBODY_V1

<https://api.mpay.com.auschemas>

Try it

Production URL

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERSTATUSBODY_V1 \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



8.13. schemas

AUTOMATCHERSTATUSRESPONSE_V1

<https://api.mpay.com.auschemas>

Try it ▶

Production URL

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERSTATUSRESPONSE_V1 \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



9. Virtual Account Number API Specification

9.1. Create an Automatcher (Receivables) Account

POST<https://api.mpay.com.au/receivables/v1/create>

Try it

Production URL

Creates a new virtual account number. These account numbers can be used along with the returned BSB to receive payments from any bank in Australia.

Payments to an invalid bank-account number, i.e. one that has not been created, will be returned by Monoova to the remitting institution. Each account number has a check digit to minimize common payer errors such as adjacent transposition errors or single-digit typos.

An optional unique ID can be linked to each account number for your reference. If a unique ID is not provided, Monoova will generate and return one automatically.

If `isActive` is omitted, it will default to `false`. This can be changed to `true` via the `/receivables/v1/status` endpoint.

For the creation of batches of account numbers, please see `/receivables/v1/batchCreate`.

Body Parameters

body `automatcherCreateBody_V1` *

Responses

✓ ● 200 successful validation

`automatcherStatusResponse_V1`

Request

Curl

```
curl --request POST \
  --url https://api.mpay.com.au/receivables/v1/create \
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation
{}
```



9.2. Get Account Status By Bank Account

GET

`https://api.mpay.com.au/receivables/v1/statusByBankAccount/{bankAccountNumber}`

Try it

Production URL

This API is used for checking the status of a bank account using the bank account number. This API will only return statuses for virtual accounts belonging to the parent mAccount.

Path Parameters

> `bankAccountNumber` `string` *

Query Parameters

`bsb` `string`

Responses

▼ ● 200 successful validation

`automatcherStatusResponse_V1`

Request

Curl

```
curl --request GET \  
  --url https://api.mpay.com.au/receivables/v1/statusByBankAccount/{bankAccountNumber} \  
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation  
{}
```



9.3. Batch Create

POST<https://api.mpay.com.au/receivables/v1/batchCreate>

Try it

Production URL

The batch create API has the same functionality as the `/receivables/v1/create` endpoint however it allows you to create up to 50,000 accounts in a single call by submitting a file. We recommend this for circumstances where more than 100 accounts need to be created within a short time period.

Creation of accounts with `isActive=true` is restricted to 1000. Multiple batches cannot be requested at the same time, you have to wait for the completion of a batch before requesting a new one.

The Batch Create API also supports a lightweight option whereby many accounts can be created by submitting a single line. This option automatically sets `isActive = true`, hence you can only request the creation of 1000 accounts in one request. For this method, all accounts will be created with the same specified name, and `clientuniqueid`'s cannot be specified will be automatically generated.

Responses

 • **200** successful validation`string<binary>`

Request

Curl

```
curl --request POST \
  --url https://api.mpay.com.au/receivables/v1/batchCreate \
  --header 'accept: application/json'
```

Responses

• **200**

```
// successful validation
""
```



9.4. Set Account Status

POST`https://api.mpay.com.au/receivables/v1/status`

Try it

Production URL

Update the status of a virtual account. Accounts can be active or inactive. If an account is inactive, all inbound payments will be rejected and returned to the payer.

Body Parameters

body `automatcherStatusBody_V1` *

Responses

▼ ● 200 successful validation

`automatcherStatusResponse_V1`

Request

Curl

```
curl --request POST \  
  --url https://api.mpay.com.au/receivables/v1/status \  
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation  
{}
```



9.5. Batch Status

POST`https://api.mpay.com.au/receivables/v1/batchStatus`

Try it

Production URL

Get the status of a request to the batchCreate endpoint.

Responses

▼ ● 200 successful validation

`string<binary>`

Request

Curl

```
curl --request POST \  
  --url https://api.mpay.com.au/receivables/v1/batchStatus \  
  --header 'accept: application/json'
```

Responses

● 200

```
// successful validation  
""
```



9.6. Get Account Status By ClientUniqueld

GET`https://api.mpay.com.au/receivables/v1/statusByClientID/{clientUniqueld}`

Try it

Production URL

This API is used for checking the status of a bank account using the clientUniqueld provided when the account was created.

Path Parameters

> **clientUniqueld** `string` *

Query Parameters

bsb `string`

Responses

✓ ● **200** successful validation`automatcherStatusResponse_V1`

Request

Curl

```
curl --request GET \  
  --url https://api.mpay.com.au/receivables/v1/statusByClientID/{clientUniqueId} \  
  --header 'accept: application/json'
```

Responses

● **200**

```
// successful validation  
{}
```



9.7. securitySchemes

BASICAUTH



Try it

Production URL

Responses

● 200

Request

Curl

```
curl --request BASICAUTH \  
  --url https://api.mpay.com.ausecuritySchemes \  
  --header 'accept: application/json'
```

Responses

● 200



9.8. schemas

AUTOMATCHERCREATEBODY_V1

Production URL

<https://api.mpay.com.auschemas>

Try it ▶

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERCREATEBODY_V1 \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



9.9. schemas

AUTOMATCHERBATCHDETAILSCLASS



Try it

Production URL

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERBATCHDETAILSCLASS \  
--url https://api.mpay.com.auschemas \  
--header 'accept: application/json'
```

Responses

● 200



9.10. schemas

AUTOMATCHERBATCHRESPONSE_V1

<https://api.mpay.com.auschemas>

Try it

Production URL

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERBATCHRESPONSE_V1 \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



9.11. schemas

DEBUGRESPONSE



Try it ▶

Production URL

Responses

● 200

Request

Curl

```
curl --request DEBUGRESPONSE \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



9.12. schemas

AUTOMATCHERSTATUSBODY_V1

<https://api.mpay.com.auschemas>

Try it ▶

Production URL

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERSTATUSBODY_V1 \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200



9.13. schemas

AUTOMATCHERSTATUSRESPONSE_V1

<https://api.mpay.com.auschemas>

Try it ▶

Production URL

Responses

● 200

Request

Curl

```
curl --request AUTOMATCHERSTATUSRESPONSE_V1 \  
  --url https://api.mpay.com.auschemas \  
  --header 'accept: application/json'
```

Responses

● 200