

public.qa.archbee.co

public.qa.archbee.co

1. Untitled
2. First doc with mentions
3. dasdasdsadasdsadasdasdas
4. dasdasdsadsadsasdadddddsssada
5. sadsadasdsaaaaaas
6. Untitled
7. sadsadasd
8. Untitled
9. Untitled
10. sdsadsadas
 - 10.1. Untitled
 - 10.2. Untitled
 - 10.3. Untitled
11. Untitled
12. adasdsadsa
 - 12.1. s`dasdaasdasdsads
13. Untitled
14. Test test testerino
15. Inline Images
16. Untitled
17. Untitled
18. 1
19. Virtual Account Number API Specification
 - 19.1. Create an Automatcher (Receivables) Account
 - 19.2. Get Account Status By Bank Account
 - 19.3. Batch Create
 - 19.4. Set Account Status
 - 19.5. Batch Status

19.6. Get Account Status By ClientUniqueld

19.7. securitySchemes

19.8. schemas

19.9. schemas

19.10. schemas

19.11. schemas

19.12. schemas

19.13. schemas

1. **Untitled**

2. First doc with mentions

Inline Images

Test test testerino

Test test testerino

First doc with mentions

Inline Images

Test test testerino

Test test testerino

aabbcc 112233

dsadsa

dsa

das

d

sad

sa

das

Heading 1 kek

dasdsad s{*dasdsad Heading 1 dasdsadas*}

fsdfsdfsdfsdf {{}}

3. dasdasdsasadasdsadasdasdas

sadasdasdsadsa

4. dasdasdsadsadsadaddddddsada

saas

5. sadsadasdsaaaaaas

sadsadsa

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

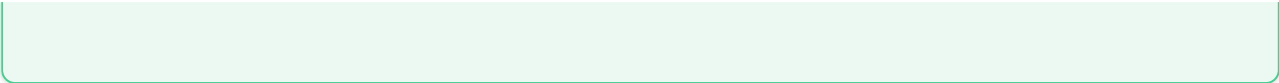
/pet/{petId}/uploadImage uploads an image



POST



/pet Add a new pet to the store









PUT `/pet` Update an existing pet  




GET `/pet/findByStatus` Finds Pets by status  




GET `/pet/findByTags` Finds Pets by tags  



Warning: Deprecated

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



POST

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



DELETE

/pet/{petId} Deletes a pet



store Access to Petstore orders



GET

/store/inventory Returns pet inventories by status



POST

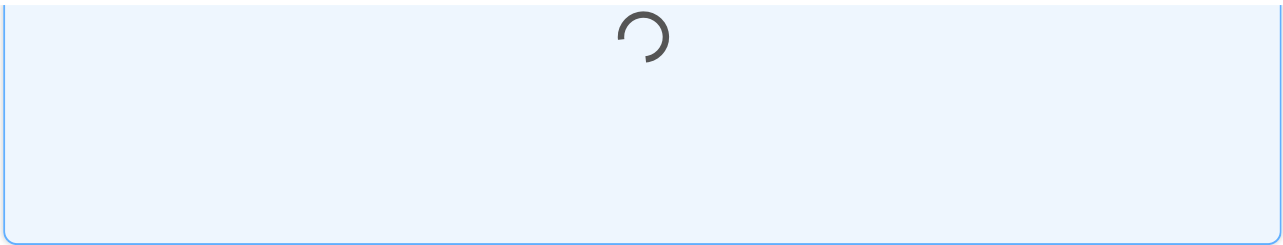
/store/order Place an order for a pet



GET

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





DELETE `/store/order/{orderId}` Delete purchase order by ID ^

user Operations about user

Find out more about our store ^

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

GET `/user/{username}` Get user by user name ^

PUT `/user/{username}` Updated user ^

DELETE

/user/{username} Delete user



GET

/user/login Logs user into the system



GET

/user/logout Logs out current logged in user session



POST

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



POST

/user Create user





Models



ApiResponse >

Category >

Pet >

Tag >

Order >

User >

6. **Untitled**

7. sadsadasd

sadsadsadsa

8. Untitled

9. Untitled

10. sdsadsadas

10.1. Untitled

10.2. **Untitled**

10.3. Untitled

11. Untitled

12. adasdsadsa

dsadsadasdadasdassadasdsada

12.1. s`dasdaasdasdsads

dsadsadasadsadasadsadsadadsadsadsadasdsasdsadasdassadsadas

13. Untitled

14. Test test testerino

sssadsadsadasdas

Un h1 nebun

15. Inline Images

Just a Heading 1 🤔👁️🐱

Lorem Ipsum is simply 🤔👁️🐱 dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, 🤔👁️🐱 but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of 🤔👁️🐱 Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.



Failed to load OpenAPI specification

Unexpected response from server

16. Untitled

17. **Untitled**

Heading

ASDASDSADSADSAD

ASDASDASDASDSA

Heading

asdsadasdsadasd

sadasdasdsad sadsadas dsadas

Heading

asdasdasdasd asdasd asdasdasd

This is my 1 row	hey hey 1	Rapid 1
My second one	hey hey 2	rapid 2
my third one	hey hey 3	rapid 3

This is my 1 row	hey hey 1	Rapid 1
My second one	hey hey 2	rapid 2
my third one	hey hey 3	rapid 3

jkbjkjbjkjk	ljkjknjknkj	kjbnjknbjkjb

kjbnjk jkbnjkbn	jknbjknb jkhbjk	jknbjk bjkb

∨ Expandable heading one

asdasdsadsa

dasdasd

sadasdasasd

Heading 2

sadsadsadsa

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

```
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));
```

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

1

1

1

19. Virtual Account Number API Specification

19.1. 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

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

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

19.2. Get Account Status By Bank Account

GET

https://api.mpay.com.au/receivables/v1/statusByBank... ▾

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

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

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

19.3. 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>`

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

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


19.4. 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](#)

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

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

19.5. 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>

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

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

19.6. Get Account Status By ClientUniqueld

GET

https://api.mpay.com.au/receivables/v1/statusByClien... ▾

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

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

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

19.7. securitySchemes

BASICAUTH

https://api.mpay.com.ausecuritySchemes



Try it

Production URL

Responses

● 200

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

19.8. schemas

AUTOMATCHERCREATEBODY_V1

https://api.mpay.com.auschemas ▾

Try it ▶

Production URL

Responses

● 200

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

19.9. schemas

AUTOMATCHERBATCHDETAILSCLASS

https://api.mpay.com.aus... ▾

Try it ▶

Production URL

Responses

● 200

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

19.10. schemas

AUTOMATCHERBATCHRESPONSE_V1

https://api.mpay.com.ausc... ▾

Try it ▶

Production URL

Responses

● 200

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

19.11. schemas

DEBUGRESPONSE

https://api.mpay.com.auschemas



Try it

Production URL

Responses

● 200

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

19.12. schemas

AUTOMATCHERSTATUSBODY_V1

https://api.mpay.com.auschemas ▾

Try it ▶

Production URL

Responses

● 200

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

19.13. schemas

AUTOMATCHERSTATUSRESPONSE_V1

https://api.mpay.com.aus... ▾

Try it ▶

Production URL

Responses

● 200

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