

Supported Models & Relationships

1. User

- i. One to one relationship(HasOne) with Store
- ii. One to one relationship with(HasOne) Picture
- iii. One to many relationship(HasMany) with Carts

2. Store

- i. One to one relationship(BelongsTo) with User
- ii. One to one relationship(HasOne) with Picture
- iii. One to many relationship(HasMany) with Post

3. Post

- i. One to one relationship(BelongsTo) with Store
- ii. One to one relationship(BelongsTo) with Category
- iii. One to many relationship(HasMany) with Picture
- iv. One to many relationship(HasMany) with Order

4. Category

- i. One to one relationship(BelongsTo) with Category(parent)
- ii. One to many relationship(HasMany) with Category(subs)
- iii. One to many relationships(HasMany) with Post

5. Picture

- i. One to one(BelongsTo) *morphable* relationship with User, Store & Post

6. Order

- i. One to one(BelongsTo) relationship with Post
- ii. One to one(BelongsTo) relationship with Cart

7. Cart

- i. One to one(BelongsTo) relationship with User
- ii. One to many(HasMany) relationship with Orders

Base Routes <https://api.iocmart.com>

| | Parameter Type | Payload | What happens behind | Results | Require Auth | Require Admin |
|--|----------------|--|---|---|--------------|---------------|
| | - | - | Gets the authenticated user with its currently active cart, store and picture | Returns all details of this user | Yes | No |
| | - | - | - | Returns this docs | No | No |
| | - | { password: ‘’, password_confirmation: ‘’ } | Updates the password of the authenticated user with the payload passed in | Returns a json object { data: ‘true’ } if successful | Yes | No |

| | | | | | | |
|--|---|---|--|--|----|----|
| | - | { email: ‘’, password: ‘’ } | Retrieves the token of the user with the provided email and password | Returns a json object { data: token } if successful | No | No |
| | - | { name: ‘’, email: ‘’, phone:{ phone: ‘’, phone_country: ‘NG’ }, password: ‘’, password_confirmation: ‘’ } | Creates a new user with the payload passed in. After it creates a token for the newly created user | Returns a json object { data: token } if successful | No | No |

Users Routes <https://api.iocmart.com/users>

| Parameter Type | Payload | What happens behind | Results | Require Auth | Require Admin |
|----------------|---|---|---|--------------|---------------|
| - | - | Gets a paginated list of all users, starting from the most recently created, with their stores and pictures | Returns this paginated list | No | No |
| - | { name: ‘’, email: ‘’, phone:{ phone: ‘’, phone_country: ‘NG’ }, password: ‘’, password_confirmation: ‘’ } | Creates a new user with the payload provided | Returns all details of the new user | No | No |
| id - int | - | Finds the user with the parameter id, with its store and pictures | Returns all details of the found user | No | No |
| id - int | { name: ‘’, email: ‘’, phone:{ phone: ‘’, phone_country: ‘NG’ } } | Finds the user with the parameter id and updates it with payload passed in | Returns all details of the updated user | Yes | No |
| id - int | { image: ‘’ } | Finds the user with the parameter id and updates its profile image to the image passed in the payload | Returns a json object { data: ‘true’ } if successful | Yes | No |

| | | | | | | |
|--|----------|---|--|--|-----|----|
| | | | | | | |
| | id - int | - | Finds the user with the parameter id and deletes it, its store ,its tokens, its carts and its profile image. | Returns a json object { data: 'true' } if delete was successful | Yes | No |

Posts Routes *https://api.iocmart.com/posts*

| Parameter Type | Payload | What happens behind | Results | Require Auth | Require Admin |
|----------------|---|--|--|--------------|---------------|
| - | - | Gets a paginated list of all posts, starting from the most recently created, with their category info, their pictures and their store with the owner of the store. | Returns this paginated list | No | No |
| - | { name: ' ', description: ' ', price: ' ', category_id: ' ', quantity: ' ', images: [] } | Creates a new post with the payload provided under the store of the Authenticated user | Returns all details of the new post | Yes | No |
| id - int | - | Finds the post with the parameter id, together with its category info, its pictures and its store with the owner of the store. | Returns all details of the found post | No | No |
| id - int | { name: ' ', description: ' ', price: ' ', category_id: ' ', quantity: ' ', Images: [] } | Finds the post with the parameter id and updates it with payload passed in | Returns all details of the updated post | Yes | No |
| id - int | - | Finds the post with the parameter id and deletes it and its pictures. | Returns a json object { data: 'true' } if delete was successful | Yes | No |

Stores Routes *https://api.iocmart.com/stores*

| Parameter | Payload | What happens | Results | Require | Require |
|-----------|---------|--------------|---------|---------|---------|
|-----------|---------|--------------|---------|---------|---------|

| Type | | behind | | Auth | Admin |
|----------|--|---|--|------|-------|
| - | - | Gets a paginated list of all stores, starting from the most recently created, with their owners and their logos | Returns this paginated list | No | No |
| - | { name: ‘ ’ , description: ‘ ’ , email: ‘ ’ , phone:{ phone: ‘ ’ , phone_country: ‘NG’ }, link: ‘ ’ , } | Creates a new store with the payload provided for the Authenticated user | Returns all details of the new store | Yes | No |
| id - int | - | Finds the store with the parameter id, with its owner and logo | Returns all details of the found user | No | No |
| id - int | { name: ‘ ’ , description: ‘ ’ , email: ‘ ’ , phone:{ phone: ‘ ’ , phone_country: ‘NG’ }, link: ‘ ’ , } | Finds the store with the parameter id and updates it with payload passed in | Returns all details of the updated store | Yes | No |
| id - int | { image: ‘ ’ } | Finds the store with the parameter id and updates its logo to the image passed in the payload | Returns a json object { data: ‘true’ } if successful | Yes | No |
| id - int | - | Finds the store with the parameter id and deletes it, all its posts and its logo. | Returns a json object { data: ‘true’ } if delete was successful | Yes | No |

Categories Routes <https://api.iocmart.com/categories>

| Parameter Type | Payload | What happens behind | Results | Require Auth | Require Admin |
|----------------|---------|---|-----------------------------|--------------|---------------|
| - | - | Gets a paginated list of all categories, starting from the most recently created, with their parent and their subcategories | Returns this paginated list | No | No |
| - | { | Creates a new category with the | Returns all details of | Yes | Yes |

| | | | | | |
|----------|---|---|---|-----|-----|
| | <pre>name: '', parent_id: '', }</pre> | payload provided | the new category | | |
| id - int | - | Finds the category with the parameter id, with its parent and its subcategories | Returns all details of the found category | No | No |
| id - int | <pre>{ name: '', parent_id: '', }</pre> | Finds the category with the parameter id and updates it with payload passed in | Returns all details of the updated category | Yes | Yes |
| id - int | - | Finds the category with the parameter id and deletes it, all its posts and its subcategories. | Returns a json object <pre>{ data: 'true' }</pre> if delete was successful | Yes | Yes |

Pictures Routes <https://api.iocmart.com/pictures>

| Parameter Type | Payload | What happens behind | Results | Require Auth | Require Admin |
|----------------|---|---|--|--------------|---------------|
| - | - | Gets a paginated list of all pictures, starting from the most recently created, with what model it belongs to | Returns this paginated list | Yes | Yes |
| - | <pre>{ filename: '', imageable_id: '', imageable_type: '' }</pre> | Creates a new picture with the payload provided | Returns all details of the new picture | Yes | Yes |
| id - int | - | Finds the picture with the parameter id, with what model it belongs to | Returns all details of the found picture | Yes | Yes |

| | | | | | |
|----------|--|---|---|-----|----|
| id - int | { filename: ‘’, imageable_id: ‘’, imageable_type: ‘’ } | Finds the picture with the parameter id and updates it with payload passed in | Returns all details of the updated picture | Yes | No |
| id - int | - | Finds the picture with the parameter id and deletes it. | Returns a json object { data: ‘true’ } if delete was successful | Yes | No |

Orders Routes <https://api.iocmart.com/orders>

| Parameter Type | Payload | What happens behind | Results | Require Auth | Require Admin |
|----------------|---|--|---|--------------|---------------|
| - | - | Gets a paginated list of all orders, starting from the most recently created, with the post and cart it belongs to | Returns this paginated list | Yes | Yes |
| - | { post_id: ‘’, quantity: ‘’, } | Creates a new order with the payload provided into the unchecked-out cart of the authorized user | Returns all details of the new order | Yes | No |
| id - int | - | Finds the order with the parameter id, with the cart and post it belongs to | Returns all details of the found order | Yes | No |
| id - int | { post_id: ‘’, quantity: ‘’, } | Finds the order with the parameter id and updates it with payload passed in | Returns all details of the updated order | Yes | No |
| id - int | - | Finds the order with the parameter id and deletes it. | Returns a json object { data: ‘true’ } if delete was successful | Yes | No |

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
|--|--|--|--|--|--|

Carts Routes *https://api.iocmart.com/carts*

| Parameter Type | Payload | What happens behind | Results | Require Auth | Require Admin |
|----------------|--|---|--|--------------|---------------|
| - | - | Gets a paginated list of all the authenticated user's carts, starting from the most recently created, with the orders they contain | Returns this paginated list | Yes | No |
| - | { checked_out: '', posts: [], } | Creates a new cart for the authenticated user with the payload provided and creates new orders for all post ids in the posts array in the payload | Returns all details of the new cart | Yes | No |
| id - int | - | Finds the cart with the parameter id, with the order it contains | Returns all details of the found cart | Yes | No |
| id - int | { checked_out: '', posts: [], } | Finds the cart with the parameter id and updates it with payload passed in | Returns all details of the updated cart | Yes | No |
| id - int | - | Finds the cart with the parameter id and deletes it and all orders in it. | Returns a json object { data: 'true' } if delete was successful | Yes | No |

Errors Explained

401 - Unauthenticated

This is thrown when you attempt to visit a route that requires authentication without an access token. Access tokens are retrieved when you register or when you login successfully.

This token is attached to the authorization header of the requests as so: *Authorization: Bearer {{token}}*.

404 - Not Found

This is thrown when you are attempt to visit a route that doesn't exist or attempt to retrieve a resource that doesn't exist with its id.

403 - Authorized Access

This is thrown when you attempt to perform an action without sufficient permissions. Such actions include: updating or deleting a resource that you(the authenticated user) is not associated with. This means updating or deleting a user that is not you, or a post, store, cart etc that is not yours. Exceptions are given to admins however,

who have the right to perform any of the above listed actions, whether theirs or not

422 - Validation Error

This is thrown on routes where payloads are sent along with. This indicates that the payload contains invalid or missing fields and therefore, cannot be processed by the server.

Attached in the body of this error is usually the further information on which fields failed validation and why.

Included in this docs is a full list of all validations on each resource' s payload routes

400 - Bad Request

This is a generic error thrown, when none of the above errors best describes the situation. Example includes when you try logging in with invalid credentials or when you

try to create a store when you have one already . Attached in the body is usually further information describing the particular situation.

Payloads

| | | | |
|-------------|-----------------|-----------------------|---|
| Base Routes | Login | email | Is required. Must be a string and must be a valid email |
| | | password | Is required. Must be a string |
| | Register | name | Is required. Must be a string and must have a minimum of 3 characters |
| | | email | Is required. Must be a string and must be a valid email. Value also has to be unique in the users table |
| | | password | Is required. Must be a string and have a minimum of 8 characters |
| | | password_confirmation | Is required. Must be a string and must match the password field |
| | | phone | Is required. Must be a JSON object containing the phone and phone_country keys |
| | | phone.phone | Is required. Must be a valid international phone number |
| | | phone.phone_country | Is required. A 2 character representation of the phone number eg NG, US, UK etc |
| | Password Update | password | Is required. Must be a string and have a minimum of 8 characters |
| | | password_confirmation | Is required. Must be a string and must match the password field |
| User Routes | Create | name | Is required. Must be a string and must have a minimum of 3 characters |
| | | email | Is required. Must be a string and must be a valid email. Value also has to be unique in the users table |
| | | password | Is required. Must be a string and have a minimum of 8 characters |
| | | password_confirmation | Is required. Must be a string and must match the password field |
| | | phone | Is required. Must be a JSON object containing the phone and phone_country keys |
| | | phone.phone | Is required. Must be a valid international phone number |
| | | phone.phone_country | Is required. A 2 character representation of the phone number eg NG, US, UK etc |
| | Update | name | Is required. Must be a string and must have a minimum of 3 characters |
| | | email | Is required. Must be a string and must be a valid email. Value also has to be unique in the users table |

| | | | |
|---------------------|---------|---------------------|--|
| | | phone | Is required. Must be a JSON object containing the phone and phone_country keys |
| | | phone.phone | Is required. Must be a valid international phone number |
| | | phone.phone_country | Is required. A 2 character representation of the phone number eg NG, US, UK etc |
| | | | |
| | Profile | image | Is required. Must be a valid Base64 image |
| | | | |
| | | | |
| Post Routes | Create | name | Is required. Must be a string and must have a minimum of 3 characters |
| | | price | Is required. Must be an integer |
| | | description | Optional. Must be a string |
| | | quantity | Is required. Must be an integer |
| | | category_id | Is required. Must be an integer, referring to the id of the category the post will belong to |
| | | images | Optional. Must be a array of base64 images |
| | | | |
| | Update | name | Is required. Must be a string and must have a minimum of 3 characters |
| | | price | Is required. Must be an integer |
| | | description | Optional. Must be a string |
| | | quantity | Is required. Must be an integer |
| | | category_id | Is required. Must be an integer, referring to the id of the category the post will belong to |
| | | images | Optional. Must be a array of base64 images |
| | | | |
| | | | |
| Store Routes | Create | name | Is required. Must be a string and must have a minimum of 3 characters |
| | | description | Optional. Must be a string |
| | | email | Is required. Must be a string and must be a valid email. Value also has to be unique in the stores table |
| | | phone | Is required. Must be a JSON object containing the phone and phone_country keys |
| | | phone.phone | Is required. Must be a valid international phone number |
| | | phone.phone_country | Is required. A 2 character representation of the phone number eg NG, US, UK etc |
| | | link | Optional. Must be a valid url |
| | | | |
| | Update | name | Is required. Must be a string and must have a minimum of 3 characters |
| | | description | Optional. Must be a string |
| | | email | Is required. Must be a string and must be a valid email. Value also has to be unique in the stores table |
| | | phone | Is required. Must be a JSON object containing the phone and |

| | | | |
|------------------------|--------|---------------------|--|
| | | | phone.country keys |
| | | phone.phone | Is required. Must be a valid international phone number |
| | | phone.phone_country | Is required. A 2 character representation of the phone number eg NG, US, UK etc |
| | | link | Optional. Must be a valid url |
| | | | |
| | Logo | image | Is required. Must be a valid Base64 image |
| | | | |
| Picture Routes | Create | filename | Is required. Must be a string |
| | | imageable_type | Is required. Must be a string. Only valid values are ‘App\User’ , ‘App\Store’ and ‘App\Post’ |
| | | imageable_id | Is required. Must be a integer indicating the id of the model the picture is associated with |
| | | | |
| | Update | filename | Is required. Must be a string |
| | | imageable_type | Is required. Must be a string. Only valid values are ‘App\User’ , ‘App\Store’ and ‘App\Post’ |
| | | imageable_id | Is required. Must be a integer indicating the id of the model the picture is associated with |
| | | | |
| Category Routes | Create | name | Is required. Must be a string and must have a minimum of 3 characters. Value must also be unique in the categories table |
| | | parent_id | Is required. Must be an integer indicating the id of the category to which this is a sub of. Passing in ‘0’ means it has no parent |
| | | | |
| | Update | name | Is required. Must be a string and must have a minimum of 3 characters. Value must also be unique in the categories table |
| | | parent_id | Is required. Must be an integer indicating the id of the category to which this is a sub of. Passing in ‘0’ means it has no parent |
| | | | |
| | | | |
| Order Routes | Create | quantity | Is required. Must be an integer indicating how many of the post wants to be ordered. |
| | | post_id | Is required. Must be an integer indicating the id of the post to be ordered. |
| | | | |
| | Update | quantity | Is required. Must be an integer indicating how many of the post wants to be ordered. |
| | | post_id | Is required. Must be an integer indicating the id of the post to be ordered. |
| | | | |
| | | | |

| | | | |
|--------------------|--------|-------------|---|
| Cart Routes | Create | checked_out | Is required.Must be an integer between <i>0</i> & <i>1</i> . Indicates whether the cart is to be checked_out immediately or the user is still shopping |
| | | posts | Must be an array of JSON objects formatted as such <i>{ post_id: 1 }</i> The contents of this array is used to create orders for each post_id passed. |
| | | | |
| | Update | checked_out | Is required.Must be an integer between <i>0</i> & <i>1</i> . Indicates whether the cart is to be checked_out immediately or the user is still shopping |
| | | posts | Must be an array of JSON objects formatted as such <i>{ post_id: 1 }</i> The contents of this array is used to create orders for each post_id passed. |