**Supported Models & Relationships**

1. **User**
2. One to one relationship(HasOne) with Store
3. One to one relationship with(HasOne) Picture
4. One to many relationship(HasMany) with Carts

1. **Store**
2. One to one relationship(BelongsTo) with User
3. One to one relationship(HasOne) with Picture
4. One to many relationship(HasMany) with Post

1. **Post**
2. One to one relationship(BelongsTo) with Store
3. One to one relationship(BelongsTo) with Category
4. One to many relationship(HasMany) with Picture
5. One to many relationship(HasMany) with Order

1. **Category**
2. One to one relationship(BelongsTo) with Category(parent)
3. One to many relationship(HasMany) with Category(subs)
4. One to many relationships(HasMany) with Post

1. **Picture**
2. One to one(BelongsTo) ***morphable*** relationship with User, Store & Post

1. **Order**
2. One to one(BelongsTo) relationship with Post
3. One to one(BelongsTo) relationship with Cart

1. **Cart**
2. One to one(BelongsTo) relationship with User
3. One to many(HasMany) relationship with Orders

**Base Routes** *https://api.iocmart.com/v1*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Route** | **Method** | **Parameter Type** | **Payload** | **What happens**  **behind** | **Results** | **Require Auth** | **Require**  **Admin** | **Possible Errors** |
| /user | GET | - | - | Gets the authenticated user with its currently active cart, store and picture | Returns all details of this user | Yes | No | 401- Unauthenticated request |
| /docs | GET | - | - | - | Returns this docs | No | No | - |
| /password | POST | - | {  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 | 401- Unauthenticated request  422- Failed validation. Invalid payload |
| /login | POST | - | {  email:‘’,  password: ‘’  } | Retrieves the token of the user with the provided email and password | Returns a json object  {  data: token  } if successful | No | No | 422- Failed validation. Invalid payload  400- Invalid login credentials |
| /register | POST | - | {  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 | 422- Failed validation. Invalid payload |

**Users Routes** *https://api.iocmart.com/v1/users*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Route** | **Method** | **Parameter Type** | **Payload** | **What happens**  **behind** | **Results** | **Require Auth** | **Require**  **Admin** | **Possible Errors** |
| / | GET | - | - | Gets a list of all rows in the table by default or those that match the conditions passed in the request.  ***See below on Querying the Api*** | Returns this list | No | No | - |
| / | POST | - | {  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 | 422 - Failed validation. Invalid payload |
| /id | GET | id - int | - | Finds the user with the parameter id, with its store and pictures | Returns all details of the found user | No | No | 404- No user with such id found |
| /id | PUT | 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 | 401- Unauthenticated request   1. No user with such id found   422- Failed validation. Invalid payload  403- Authenticated user is not the owner of the profile or is not an admin |
| /id/profile | PUT | 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 | 401- Unauthenticated request  404- No user with such id found  422- Failed validation. Invalid payload  403- Authenticated user is not the owner of the profile or is not an admin |
| /id | DELETE | 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 | 401- Unauthenticated request  404- No user with such id found  422- Unsuccessful delete  403- Authenticated user is not the owner of the profile or is not an admin |

**Posts Routes** *https://api.iocmart.com/v1/posts*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Route** | **Method** | **Parameter Type** | **Payload** | **What happens**  **behind** | **Results** | **Require Auth** | **Require**  **Admin** | **Possible Errors** |
| / | GET | - | - | Gets a list of all rows in the table by default or those that match the conditions passed in the request.  ***See below on Querying the Api*** | Returns this list | No | No | - |
| / | POST | - | {  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 | 401- Unauthenticated request  422 - Failed validation. Invalid payload   1. Authenticated user doesn’t have an existing store |
| /id | GET | 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 | 404- No post with such id found |
| /id | PUT | 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 | 401- Unauthenticated request  404- No post with such id found  422- Failed validation. Invalid payload  403- Authenticated user doesn’t own the post or is not an admin |
| /id | DELETE | 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 | 401- Unauthenticated request  404- No post with such id found  422- Unsuccessful delete  403- Authenticated user doesn’t own the post or is not an admin |

**Stores Routes** *https://api.iocmart.com/v1/stores*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Route** | **Method** | **Parameter Type** | **Payload** | **What happens**  **behind** | **Results** | **Require Auth** | **Require**  **Admin** | **Possible Errors** |
| / | GET | - | - | Gets a list of all rows in the table by default or those that match the conditions passed in the request.  ***See below on Querying the Api*** | Returns this list | No | No | - |
| / | POST | - | {  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 | 401- Unauthenticated request  422 - Failed validation. Invalid payload  400- Authenticated user has an existing store |
| /id | GET | id - int | - | Finds the store with the parameter id, with its owner and logo | Returns all details of the found user | No | No | 404- No user with such id found |
| /id | PUT | 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 | 401- Unauthenticated request  404- No store with such id found  422- Failed validation. Invalid payload  403 - Authenticated user doesn’t own the store or is not an admin |
| /id/logo | PUT | 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 | 401- Unauthenticated request  404- No store with such id found  422- Failed validation. Invalid payload  403 - Authenticated user doesn’t own the store or is not an admin |
| /id | DELETE | 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 | 401- Unauthenticated request  404- No store with such id found  422- Unsuccessful delete  403 - Authenticated user doesn’t own the store or is not an admin |

**Categories Routes** *https://api.iocmart.com/v1/categories*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Route** | **Method** | **Parameter Type** | **Payload** | **What happens**  **behind** | **Results** | **Require Auth** | **Require**  **Admin** | **Possible Errors** |
| / | GET | - | - | Gets a list of all rows in the table by default or those that match the conditions passed in the request.  ***See below on Querying the Api*** | Returns this list | No | No | - |
| / | POST | - | {  name: ‘’,  parent\_id: ‘’,  } | Creates a new category with the payload provided | Returns all details of the new category | Yes | Yes | 401 - Unauthenticated request  422 - Failed validation. Invalid payload  403 - Authenticated user is not an admin |
| /id | GET | id - int | - | Finds the category with the parameter id, with its parent and its subcategories | Returns all details of the found category | No | No | 404- No category with such id found |
| /id | PUT | id - int | {  name: ‘’,  parent\_id: ‘’,  } | Finds the category with the parameter id and updates it with payload passed in | Returns all details of the updated category | Yes | Yes | 401- Unauthenticated request  404- No category with such id found  422- Failed validation. Invalid payload  403 - Authenticated user is not an admin |
| /id | DELETE | id - int | - | Finds the category with the parameter id and deletes it, all its posts and its subcategories. | Returns a json object  {  data: ‘true’  } if delete was successful | Yes | Yes | 401-Unauthenticated request  422- Unsuccessful delete  404- No category with such id found  403 - Authenticated user is not an admin |

**Pictures Routes** *https://api.iocmart.com/v1/pictures*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Route** | **Method** | **Parameter Type** | **Payload** | **What happens**  **behind** | **Results** | **Require Auth** | **Require**  **Admin** | **Possible Errors** |
| / | GET | - | - | Gets a list of all rows in the table by default or those that match the conditions passed in the request.  ***See below on Querying the Api*** | Returns this list | No | No | - |
| / | POST | - | {  filename: ‘’,  imageable\_id: ‘’,  imageable\_type: ‘’  } | Creates a new picture with the payload provided | Returns all details of the new picture | Yes | Yes | 401 - Unauthenticated request  422 - Failed validation. Invalid payload  403 - Authenticated user is not an admin |
| /id | GET | id - int | - | Finds the picture with the parameter id, with what model it belongs to | Returns all details of the found picture | No | No | 401- Unauthenticated request  404- No picture with such id found  403 - Authenticated user is not an admin  d |
| /id | PUT | 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 | 401- Unauthenticated request  404- No picture with such id found  422- Failed validation. Invalid payload  403 - Authenticated user is not associated with the picture or is not an admin |
| /id | DELETE | id - int | - | Finds the picture with the parameter id and deletes it. | Returns a json object  {  data: ‘true’  } if delete was successful | Yes | No | 401-Unauthenticated request  422- Unsuccessful delete  404- No picture with such id found  403 - Authenticated user is not associated with the picture or is not an admin |

**Orders Routes** *https://api.iocmart.com/v1/orders*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Route** | **Method** | **Parameter Type** | **Payload** | **What happens**  **behind** | **Results** | **Require Auth** | **Require**  **Admin** | **Possible Errors** |
| / | GET | - | - | Gets a list of all rows in the table by default or those that match the conditions passed in the request.  ***See below on Querying the Api*** | Returns this list | No | No | - |
| / | POST | - | {  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 | 401 - Unauthenticated request  422 - Failed validation. Invalid payload |
| /id | GET | id - int | - | Finds the order with the parameter id, with the cart and post it belongs to | Returns all details of the found order | No | No | 401- Unauthenticated request  404- No order with such id found  403 - Authenticated user is not the owner of the order or is not an admin  d |
| /id | PUT | 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 | 401- Unauthenticated request  404- No order with such id found  422- Failed validation. Invalid payload  403 - Authenticated user is not the owner of the order or is not an admin |
| /id | DELETE | id - int | - | Finds the order with the parameter id and deletes it. | Returns a json object  {  data: ‘true’  } if delete was successful | Yes | No | 401-Unauthenticated request  422- Unsuccessful delete  404- No order with such id found  403 - Authenticated user is not the owner of the order or is not an admin |

**Carts Routes** *https://api.iocmart.com/v1/carts*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Route** | **Method** | **Parameter Type** | **Payload** | **What happens**  **behind** | **Results** | **Require Auth** | **Require**  **Admin** | **Possible Errors** |
| / | GET | - | - | Gets a list of all rows in the table by default or those that match the conditions passed in the request.  ***See below on Querying the Api*** | Returns this list | No | No | - |
| / | POST | - | {  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 | 401 - Unauthenticated request  422 - Failed validation. Invalid payload |
| /id | GET | id - int | - | Finds the cart with the parameter id, with the order it contains | Returns all details of the found cart | No | No | 401- Unauthenticated request  404- No order with such id found  403 - Authenticated user is not the owner of the cart or is not an admin |
| /id | PUT | 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 | 401- Unauthenticated request  404- No cart with such id found  422- Failed validation. Invalid payload  403 - Authenticated user is not the owner of the cart or is not an admin |
| /id | DELETE | 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 | 401-Unauthenticated request  422- Unsuccessful delete  404- No cart with such id found  403 - Authenticated user is not the owner of the cart or is not an admin |

**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 payloaded 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 *0* & *1*. 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  *{ post\_id: 1 }*  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 *0* & *1*. 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  *{ post\_id: 1 }*  The contents of this array is used to create orders for each post\_id passed. |

**Querying the API**

All model routes come provided with a GET route to *https://api.iocmart.com/v1/{{model}}/query which is important for*

*Performing complex queries on the model’s table. These queries are attached as parameters to the request. All available parameters are listed*

*below and examples are provided.*

***NB:*** *All queries return a paginated list of 50 items of the requested query, however if the Limit*query is present in the queries list,

there is an exception, a normal list is returned as Laravel doesn’t work well with *Limit* & *Paginate* together.

1. *Distinct*

*This returns all distinct values in the column provided*

*Eg: [https://api.iocmart.com/v1/users?distinct=name](https://api.iocmart.com/v1/posts/query?distinct=name) returns all distinct names in the users table*

1. *Group\_By*

*This returns the generic SQL result of a group\_by action. Results are very identical to the distinct query*

*Eg: [https://api.iocmart.com/v1/users?group\_by=name](https://api.iocmart.com/v1/posts/query?distinct=name) returns a grouped collection of all rows in the users table*

1. *Limit*

*This is the only query that doesn’t return a paginated list. When present in any request, it returns a typical list.*

*It is used to limit the returned rows of a query to the amount passed in. The max limit proceeded is however 50.*

*Any value passed above 50 will simply return 50 items.*

*Eg: [https://api.iocmart.com/v1/users?limit=10](https://api.iocmart.com/v1/posts/query?distinct=name) returns a list of the first 10 rows in the users table*

1. *Offset*

*This is typically used in conjunction with the Limit query. It is used to offset some rows when trying to perform a limit*

*Eg: [https://api.iocmart.com/v1/users?offset=10&limit=10](https://api.iocmart.com/v1/posts/query?distinct=name) returns a list of the eleventh to twentieth rows in the users table.*

*The offset is used to skip the first 10 rows.*

1. *Order\_By*

*This is used to sort the results along the column provided with the order provided.*

*Eg: [https://api.iocmart.com/v1/users?order\_by=name](https://api.iocmart.com/v1/users/query?order_by=name) returns a list of all users in ascending order of the name column*

1. *Order*

*This is only used when the Order\_By is passed in. It is used to pass the order with which the sorting occurs. Recognized values are asc* and *desc.*

Default value is *asc, so result is sorted in ascending order if Order is not passed in.*

*Eg: [https://api.iocmart.com/v1/users?order\_by=name&order=desc](https://api.iocmart.com/v1/users/query?order_by=name&order=desc) returns a list of all users in descending order of the name column*

1. *Where*

*This is used to compare values with rows in the database. It is usually accompanied with two more queries Ops*  and *Value.*

*Eg: [https://api.iocmart.com/v1/users?where=id&op=<&value=5](https://api.iocmart.com/v1/users/query?where=id&op=<&value=5) returns all users with id less than 5*

1. *Op*

*This is used to provide the operation for the Where query. Default value is = and supported values include*

*=, !=, <, <=, >, >=, like.*

*Eg: [https://api.iocmart.com/v1/users?where=id&op=<&value=5](https://api.iocmart.com/v1/users/query?where=id&op=<&value=5) returns all users with id less than 5*

1. *Value*

*This is used to provide the value for comparison for the Where query.*

*Eg: [https://api.iocmart.com/v1/users?where=id&op=<&value=5](https://api.iocmart.com/v1/users/query?where=id&op=<&value=5) returns all users with id less than 5*

1. *With*

*This is used to attach the required relationships to the result.*

*Eg: [https://api.iocmart.com/v1/users?with=store,picture,carts](https://api.iocmart.com/v1/users/query?with=store,picture,cart) returns all users joined with their store, picture and carts.*

***NB:*** *All queries that accepts database column or table names such as* where, with, order\_by, distinct etc expects only such names, and are filtered

through a safeguard lists of names to avoid SQL injections.