Auction API documentation  
  
This is the API documentation for connection to SkyCastle Auctions backend.

These APIs are not public.

**Base URL**

For all the endpoints we shall be describing below, they are prefixed by the base url to which our different backend services are hosted.   
Below it the base endpoint.

<https://app.skycastleauctionhub.com>

Account creation and Access  
  
**Register**

POST /auth-service/api/v1/register

Headers  
Content-Type: application/json

All user registration is done through this endpoint.   
  
This endpoint supports a JSON payload i.e. please set your content type to application/json.  
  
The system supports 3 types of users, i.e. ADMIN, SELLER and BUYER.  
This endpoint is currently not protected by any security.  
  
Below is a description of the different request body parameters required to create a new user account.   
Please note, a registered user will fall into a given account type depending on the values of some of the parameter in the request.

Body Parameters

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| username | String | M | This is the account username |
| password | String | M | This is the access password |
| enable2Fa | Boolean | O | Whether or not access should be protected by 2FA, by default this is set to false |
| firstname | String | M | Represents the user’s first name. |
| lastname | String | M | Represents the user’s last name. |
| email | String | M | This is a mandatory field representing users email. |
| role | Enum (ROLE\_ADMIN, ROLE\_SELLER, ROLE\_BUYER) | C | This represents the role and depends on whether the user should be an ADMIN, SELLER or BUYER. Depending on which account you want the user to fall in, please specify user ROLE\_ADMIN, ROLE\_SELLER, ROLE\_BUYER respectively. |
| accountType | Enum (ADMIN, BUYER,SELLER) | C | Depending on which account type you want to create, please specify this accordingly as seen in “role” above (ADMIN, SELLER, BUYER) |
| nin | String | C | This is required for SELLER accounts creation. |
| identification | String | C | This field represents a BUYER company registration number, or individual identification it can be an individual identification, company reg No, etc.  Only supply this when registering a SELLER. |
| settings | Object | O | This is an optional few settings, we shall look at a sample in the sample request. |
| contact | String | O | Telephone number. |

Sample Request.  
  
The sample request below shows how to create an ADMIN user account.

{

"enable2Fa": **false**,

"username": "admin",

"accountType": "ADMIN",

"firstname": "John",

"lastname": "Doe",

"settings":{

"receive\_mails":**false**,

"device\_no":1

},

"gender": "MALE",

"password": "123456",

"email": "john@oddjobs.tech",

"address": "Munyonyo"

}

Response Structure

A 200 response together with the user id is returned.  
Please handle all other status codes as failures.  
{

"data": {

"id": 12,

"username": "admin",

"email": "john@oddjobs.tech",

"address": "Munyonyo",

"settings": **null**,

"name":**”**John **Doe**,

"accountType": "ADMIN",

"nin": **null**,

"identification": **null**,

"firstname": "John",

"lastname": "Doe",

"role": "ROLE\_ADMIN",

"gender": **null**,

"department": **null**,

"createdAt": "2023-08-09T10:39:53.084+00:00",

"qrCode": **null**,

"token": **null**

},

"success": **true**,

"message": "User admin successfully registered",

"timestamp": "2023-08-09T10:39:53.153+00:00",

"status": 200

}

Please note that if the enable2Fa was set to true in the request, the qrCode parameter in the response will contain a base64 string version of the QR-Code. The integrating application, will be required to display this on the UI for the user to scan using any authenticator app like “Google Authentication” or Microsoft Authenticator. The integrating app will also have the burden to render a six-digit OTP verification screen during 2FA login process.

**Login**

POST /login   
  
Header  
Content-Type: multipart/form-data

This endpoint allows a user to be authenticated.  
It takes the username and password as request body parameters.  
Please note that the http client must set the content type to multipart/form-data.

In javascript (axios, please use the Form API and set the content type header accordingly when calling this API)

Request body Parameter

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| username | String | M | Mandatory username |
| password | String | M | Mandatory password |

Response Structure

A 200 response with an access token and refresh token plus user object will be returned.

Please treat all other response status codes as failures.  
  
Inside the user Object you can tell if a user has 2FA enabled or not and act accordingly.

Sample Response  
{

"access\_token": {

"expiresIn": 1200000,

"createdAt": 1691577415718,

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlbHZpc0BvZGRqb2JzLnRlY2giLCJBVVRIRU5USUNBVEVEIjp0cnVlLCJSRUZSRVNIIjpmYWxzZSwiaXNzIjoiXCJTa3kgQ2FzdGxlIEF1Y3Rpb24gSHViXCIiLCJleHAiOjE2OTE1Nzg2MTV9.j3tnJOgCfFwwV1jzOBffcd-z-mnw\_Whw9V26sGF2mUk"

},

"refresh\_token": {

"expiresIn": 2400000,

"createdAt": 1691577416531,

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlbHZpc0BvZGRqb2JzLnRlY2giLCJBVVRIRU5USUNBVEVEIjp0cnVlLCJSRUZSRVNIIjp0cnVlLCJpc3MiOiJcIlNreSBDYXN0bGUgQXVjdGlvbiBIdWJcIiIsImV4cCI6MTY5MTU3OTgxNn0.MLzThWLTSqAQQ2jMkl5gZVCZ4nq8gS9fbD8T03ec54o"

}

}

Note that the access token expires after some time, and it should be refreshed by the refreshToken periodically. The refreshToken expiry time is twice that of the accessToken. Once the refresh token also expires, the user session will be considered expired and will have to login again.  
It is also important to note that when a user 2FA is enabled, only the access token is returned on login. This token can only be used to verify the OTP token.  
   
The integrating application is advised to periodically refresh the access token before it expires.

We shall look at the refresh tokens API next.

**Refresh Tokens**

GET / auth-service/api/v1/tokens/refresh

Headers:   
Authorization: Bearer {{**refreshToken**}}

The client is required to make a get request to this endpoint with the refreshToken as the Authorization header.

Request body

This endpoint has no request body. The client just has to specify the refreshToken in the Authorization header. Do not forget to prefix it with the term “Bearer “. Leave a space between “Bearer” and the token value.  
  
Response

This endpoint returns a new set of access and refresh tokens. Please update your local store until you need to refresh again.

Sample Response.

{

"access\_token": {

"expiresIn": 1200000,

"createdAt": 1691577415718,

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlbHZpc0BvZGRqb2JzLnRlY2giLCJBVVRIRU5USUNBVEVEIjp0cnVlLCJSRUZSRVNIIjpmYWxzZSwiaXNzIjoiXCJTa3kgQ2FzdGxlIEF1Y3Rpb24gSHViXCIiLCJleHAiOjE2OTE1Nzg2MTV9.j3tnJOgCfFwwV1jzOBffcd-z-mnw\_Whw9V26sGF2mUk"

},

"refresh\_token": {

"expiresIn": 2400000,

"createdAt": 1691577416531,

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlbHZpc0BvZGRqb2JzLnRlY2giLCJBVVRIRU5USUNBVEVEIjp0cnVlLCJSRUZSRVNIIjp0cnVlLCJpc3MiOiJcIlNreSBDYXN0bGUgQXVjdGlvbiBIdWJcIiIsImV4cCI6MTY5MTU3OTgxNn0.MLzThWLTSqAQQ2jMkl5gZVCZ4nq8gS9fbD8T03ec54o"

}

}

**Refresh Tokens**  
This endpoint is used to verify a g digit toke during 2FA second stage.

GET / auth-service/api/v1/tokens/otp-verify

Headers   
Authorization: Bearer {{**accessToken**}}

Content-Type: multipart/form-data  
  
Please note that just like the login endpoint, the content type is multipart/form-data.

Request Parameters.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| code | Number | M | Mandatory expiring 6 digit code |

**Response Structure**

{

"access\_token": {

"expiresIn": 1200000,

"createdAt": 1691577415718,

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlbHZpc0BvZGRqb2JzLnRlY2giLCJBVVRIRU5USUNBVEVEIjp0cnVlLCJSRUZSRVNIIjpmYWxzZSwiaXNzIjoiXCJTa3kgQ2FzdGxlIEF1Y3Rpb24gSHViXCIiLCJleHAiOjE2OTE1Nzg2MTV9.j3tnJOgCfFwwV1jzOBffcd-z-mnw\_Whw9V26sGF2mUk"

},

"refresh\_token": {

"expiresIn": 2400000,

"createdAt": 1691577416531,

"token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJlbHZpc0BvZGRqb2JzLnRlY2giLCJBVVRIRU5USUNBVEVEIjp0cnVlLCJSRUZSRVNIIjp0cnVlLCJpc3MiOiJcIlNreSBDYXN0bGUgQXVjdGlvbiBIdWJcIiIsImV4cCI6MTY5MTU3OTgxNn0.MLzThWLTSqAQQ2jMkl5gZVCZ4nq8gS9fbD8T03ec54o"

}

}

**Disable/Enable accounts**  
This endpoint is used by the administrators to disable or enable accounts

GET / auth-service/api/v1/changeStatus/{username}?enable={true/false}

Headers   
Authorization: Bearer {{**accessToken**}}

Request Parameters.

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| username | String (Path Variable) | M | Mandatory account username to enable/disable |
| enable | Boolean (url param) | M | Mandatory Boolean, if set to true, the target account will be enabled if found disabled else, it will be disabled if enabled. |

**Response Structure**

A 200-status code with a descriptive message of what happened.  
  
Please treat all other status codes as failures accordingly.

**Other User related endpoints**

**Get current user**  
This endpoint is used to access details of the logged in user, call it to return the currently logged in user object.  
By currently logged in, we mean self. You only call it to get details of yourself when logged in.

GET / auth-service/api/v1/current-user

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

No request body required. Just make sure to call it with the active access token.

Response

A user object will be returned similar to this one below.

{

"id": 12,

"username": "admin",

"email": "john@oddjobs.tech",

"address": "Munyonyo",

"settings": **null**,

"name":**”**John **Doe**,

"accountType": "ADMIN",

"nin": **null**,

"identification": **null**,

"firstname": "John",

"lastname": "Doe",

"role": "ROLE\_ADMIN",

"gender": **null**,

"department": **null**,

"createdAt": "2023-08-09T10:39:53.084+00:00",

"qrCode": **null**,

"token": **null**

}

**Get Users**

This endpoint is used by Admin accounts to return a paginated list of all users in the system.

GET / auth-service/api/v1/users?username={sample}&page=0&size=10

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| username | String | O | Filter by username |
| Size | Integer | O | Defaults to 10 |
| Page | Integer | O | Defaults to page 0 (meaning db offset 0) |

**Get Seller account details**  
This endpoint is used to retrieve a seller’s account details

GET / auth-service/api/v1/seller/{id}

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| id | Integer (path variable) | M | Mandatory seller db Id. Specify this as a path variable as shown in the enpoint url above. |

**Response structure**

This returns the seller account object. This is similar to the user account response objects seen above.

{

"id": 3,

"accountType": "SELLER",

"username": "embassy@oddjobs.tech",

"role": "ROLE\_SELLER",

"email": "embassy@oddjobs.tech",

"address": **null**,

"settings": {},

"isExpired": **false**,

"status": "ACTIVE",

"name": "American Embassy",

"nin": **null**,

"identification": **null**,

"isApproved": **false**,

"contact": **null**

}

**Approve Seller account**  
This endpoint is used by administrator account to approve or reject seller accounts.  
Once a seller is registered, they cannot transact until their account is approved. This endpoint achieves exactly that.

GET / auth-service/api/v1/seller/{id}/approve?approved=true

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| id | Integer (path variable) | M | Mandatory seller db Id. Specify this as a path variable as shown in the enpoint url above. |
| approved | Boolean | M | Mandatory Boolean flag to indicate whether the account should be approved or note. |

**Response structure**

This returns the seller account object. This is similar to the user account response objects seen above.

**Get Buyer account details**  
This endpoint is used to retrieve a seller’s account details

GET / auth-service/api/v1/buyer/{id}

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| id | Integer (path variable) | M | Mandatory buyer db Id. Specify this as a path variable as shown in the enpoint url above. |

**Response structure**

This returns the seller account object. This is similar to the user account response objects seen above.

{

"id": 5,

"deleted": **false**,

"enabled": **true**,

"accountType": "BUYER",

"username": "ekeeya@oddjobs.tech",

"role": "ROLE\_BUYER",

"email": "ekeeya@oddjobs.tech",

"address": **null**,

"settings": {},

"using2FA": **true**,

"secret": "EAJ5RJVIJMYDRN3G4OLWGK5XBCKS56KL",

"isExpired": **false**,

"status": "ACTIVE",

"firstname": **null**,

"lastname": "Keeya",

"gender": **MALE**,

"accountNonExpired": **true**,

"accountNonLocked": **true**,

"credentialsNonExpired": **true**

}

We have looked at all required accounts and access related API endpoints, Let’s proceed to the inventory service endpoints.

Inventory service endpoints

**Create Categories**  
Currently the system only supports a single category which is Vehicles and this is set by default. In an event we start to support other categories of products, we can use this endpoint to set them.  
  
Only Admin accounts can create new categories

POST /inventory-service/api/v1/categories

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| categoryNames | List of Strings | M | Mandatory categoryNames. (See sample request body) |

**Sample request body**

{

"categoryNames":[

"Furniture"

]

}

**Response structure**

A list of configured categories is returned.

{

"success": **true**,

"message": "",

"statusCode": 200,

"data": **null**,

"entries": [

{

"success": **true**,

"message": "Successfully added product category(s) FURNITURE",

"statusCode": 200,

"data": {

"id": 2,

"categoryName": "Furniture"

}

}

],

"totalCount": 0,

"offset": 0,

"totalPages": 0,

"size": 1,

"page": 0

}

**GET Categories**  
This is used to fetch all available categories, you will need the categories when uploading products like cars.

GET /inventory-service/api/v1/categories

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

No request parameters required.

**Response structure**

A list of configured categories is returned.

{

"success": **true**,

"message": "",

"statusCode": 200,

"data": **null**,

"entries": [

{

"id": 1,

"categoryName": "Vehicles"

},

{

"id": 2,

"categoryName": "Furniture"

}

],

"totalCount": 2,

"offset": 0,

"totalPages": 1,

"size": 2,

"page": 0

}

**Create Vehicle Models**  
Models can be created by any account user. The integrating app is advised to add a quick way of creating a non-existent vehicle model during vehicle uploading.

POST /inventory-service/api/v1/ vehicle-models

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| model | String | M | Mandatory model name e.g. CAMRY HYBRID) |
| make | String | M | Mandatory vehicke make, e.g TOYOTA |
| year | Integer | M | Vehicle make year e.g 2022 |

**Sample request body**

{

"model":"CAMRY HYBRID",

"make":"TOYOTA",

"year":2016

}

**Response structure**

Below is the sample response

{

"success": **true**,

"message": "Vehicle model CAMRY HYBRID added",

"statusCode": 200,

"data": {

"id": 1,

"model": "CAMRY HYBRID",

"make": "TOYOTA",

"year": 2016

}

}

"totalCount": 0,

"offset": 0,

"totalPages": 0,

"size": 1,

"page": 0

}

**GET Models**  
This is used to fetch all available vehicle models, you will need the these models when uploading products like cars.

GET /inventory-service/api/v1/ vehicle-models  
  
Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

No request parameters required.

**Response structure**

A list of configured vehicle models is returned.

{

"success": **true**,

"message": "",

"statusCode": 200,

"data": **null**,

"entries": [

{

"id": 1,

"model": "CAMRY HYBRID",

"year": 2016,

"make": "TOYOTA"

}

],

"totalCount": 1,

"offset": 0,

"totalPages": 1,

"size": 1,

"page": 0

}

**Create Vehicle Type**  
Use this endpoint to register vehicle types such as Automobiles, ATVs, Motocycles,WaterMobiles etc.

POST /inventory-service/api/v1/ vehicle-types

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| entries | Object | M | Mandatory list of vehicle types, refer to sample payload below. |

**Sample request body**

{

"entries":[

{"vehicleType":"Automobiles"},

{"vehicleType":"ATVs"},

{"vehicleType":"Motorcycles"},

{"vehicleType":"Watermobiles"}

]

}

**Response structure**

Below is the sample response

{

"success": **true**,

"message": "",

"statusCode": 200,

"data": **null**,

"entries": [

{

"success": **false**,

"message": "Vehicle type Automobiles already exists",

"statusCode": 500,

"data": **null**

},

{

"success": **true**,

"message": "Vehicle type ATVs successfully saved",

"statusCode": 200,

"data": {

"id": 2,

"vehicleType": "ATVs",

"vehicleCount": 0,

"createdAt": "2023-08-09 11:03:49"

}

},

{

"success": **true**,

"message": "Vehicle type Motorcycles successfully saved",

"statusCode": 200,

"data": {

"id": 3,

"vehicleType": "Motorcycles",

"vehicleCount": 0,

"createdAt": "2023-08-09 11:03:50"

}

},

{

"success": **true**,

"message": "Vehicle type Snowmobiles successfully saved",

"statusCode": 200,

"data": {

"id": 4,

"vehicleType": "Watermobiles",

"vehicleCount": 0,

"createdAt": "2023-08-09 11:03:50"

}

}

],

"totalCount": 0,

"offset": 0,

"totalPages": 0,

"size": 4,

"page": 0

}

**GET Vehicle Types**  
This is used to fetch all available vehicle types, you will need the vehicle types when uploading products like cars.

GET /inventory-service/api/v1/ vehicle-models  
  
Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

No request parameters required.

**Response structure**

Below is the sample successful 200 response.

{

"success": **true**,

"message": "",

"statusCode": 200,

"data": **null**,

"entries": [

{

"id": 2,

"vehicleType": "ATVs",

"vehicleCount": 0,

"createdAt": "2023-08-09 11:03:49"

},

{

"id": 1,

"vehicleType": "Automobiles",

"vehicleCount": 0,

"createdAt": "2023-05-29 09:08:13"

},

{

"id": 3,

"vehicleType": "Motorcycles",

"vehicleCount": 0,

"createdAt": "2023-08-09 11:03:50"

},

{

"id": 4,

"vehicleType": "WaterMobiles",

"vehicleCount": 0,

"createdAt": "2023-08-09 11:03:50"

}

],

"totalCount": 4,

"offset": 0,

"totalPages": 1,

"size": 4,

"page": 0

}

**Create Vehicle body styles**  
To configure vehicle body styles use this endpoint

POST /inventory-service/api/v1/vehicle-body-styles

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| bodyTypes | List of body styles | M | Mandatory list of vehicle body styles (see sample request) |

**Sample request body**

{

"bodyTypes":["4dr Spor", "Converti", "Coupe", "Hatchbac", "Sedan 4d", "Station"]

}

**Response structure**

Below is the sample response

{

"success": **true**,

"message": "",

"statusCode": 200,

"data": **null**,

"entries": [

{

"id": 2,

"style": "4dr Spor"

},

{

"id": 1,

"style": "Converti",

}

],

"totalCount": 4,

"offset": 0,

"totalPages": 1,

"size": 4,

"page": 0

}

**GET Vehicle Body Styles**  
This is used to fetch all available vehicle body styles, you will need these when uploading products like cars.

GET /inventory-service/api/v1/vehicle-body-styles  
  
Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

No request parameters required.

**Response structure**

Below is the sample successful 200 response.

{

"success": **true**,

"message": "",

"statusCode": 200,

"data": **null**,

"entries": [

{

"id": 2,

"style": "4dr Spor"

},

{

"id": 1,

"style": "Converti",

}

],

"totalCount": 2,

"offset": 0,

"totalPages": 1,

"size": 2,

"page": 0

}

**Upload Vehicle**  
This endpoint is used by sellers and admins to register vehicles, please note that once a vehicle is uploaded, it will not be available for display to the buyers until an inspection report is uploaded by the admins.

POST /inventory-service/api/v1/vehicles

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| name | String | M | Mandatory label you wish to call the vehicle |
| description | String | O | Optional simple notes about the vehicle |
| seller | Integer | M | Mandatory seller db ID to which the vehicle belongs |
| condition | Enum (USED, NEW, SALVAGED, REFURBISHED) | M | Select from either USED, NEW, SALVAGED or REFURBISHED |
| fuelType | Enum (ELECTRIC, DIESEL, FLEXIBLE\_FUEL, PETROLEUM, HYBRID) | M | Mandatory fuelType |
| transmission | Enum (AUTOMATIC, MANUAL) | M | Represents the vehicle transmission nature. |
| bodyStyle | Integer | M | Body style is picked from a list of body styles configured from earlier APIs. It must be the db Id of the desired bodystyle. |
| buyNow | Boolean | M | Specifies whether the vehicle is available for immediate purchase or not. If yes then the sellingPrice in the next param has to be set |
| sellingPrice | Currency | C | This is conditional when the above buyNow is set to true |
| reservePrice | Currency | O | This an optional amount below which the vehicle cannot be sold after an auctioning session. |
| minBidAmount | Currency | O | This is the minimum amount of money a single bid cannot go below. If not set, it is automatically set to 100,000 |
| regNo | String | O | This represents the vehicle’s registration number if any |
| vin | String | M | This represents the unique manufacturer assigned vehicle serial number. |
| odometer | Integer | M | Represents the vehicles mileage, if not set, it defaults to 0 |
| vehicleType | Integer | M | Selected from the previously configured vehicle types as we saw in the preceding APIs. You are most likely going to set the type to AutoMobiles |
| model | Integer | M | This is a mandatory field representing the vehicle model. If the model does not exist in the database, we recommend that the integrating app, find easier ways for users to add it during vehicle upload. |
| images | List of base64 image strings | M | A vehicle must have very clear pictures of it. Please note that the first picture to be selected to the list with be the one displayed for the buyers |
| properties | Object | O | This indicates a whole list of other properties that describe this vehicle, e.g., color, age, previous owner, keys, anything you can come up with. Please see the request structure below on how to structure additional properties. |

**Sample request body**

{

"name": "2013 TOYOTA SIENNA XLE",

"description": "This vehicle is in great condition.",

"properties": {

"Highlights": "Run and Drive",

"Primary Damage": "",

"Secondary Damage": "",

"Color": "WHITE",

"keys": "yes"

},

"seller": 2,

"condition": "USED",

"fuelType": "PETROLEUM",

"transmission": "AUTOMATIC",

"bodyStyle": 5,

"buyNow": false,

"reservePrice": 14000000.0,

"regNo": "UBT 343C",

"images": ["….”,"….”

],

"vin": "51OYK30C70S370514",

"odometer": 1236344,

"model": 3,

"vehicleType": 1

}

**Response structure**

A 200-status code response as one below.

{

"success": true,

"message": "Vehicle 2013 TOYOTA CAMRY HYBRID successfully registered!",

"statusCode": 200,

"data": {

"id": 9,

"name": "2013 TOYOTA CAMRY HYBRID",

"description": "This vehicle is in great condition.",

"properties": {

"Highlights": "Run and Drive",

"Primary Damage": "",

"Secondary Damage": "",

"Color": "WHITE",

"keys": "yes"

},

"seller": 3,

"sellerDetails": null,

"condition": "USED",

"images": [……………….],

"lot": null,

"status": "IN\_EVALUATION",

"categoryName": null,

"vin": "51OYX30C000S370514",

"odometer": 1236344,

"model": "CAMRY HYBRID",

"modelId": 1,

"make": "TOYOTA",

"vehicleType": "Automobiles",

"buyNow": false,

"sellingPrice": 0,

"reservePrice": 14000000.0,

"bodyStyle": "Sedan 4d",

"regNo": "UBT 143C",

"transmission": "AUTOMATIC",

"fuelType": "PETROLEUM",

"inspected": false

}

}

**Get Vehicle**  
This endpoint helps you get details of an individual vehicle.

GET /inventory-service/api/v1/vehicles/{id}

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

Only the vehicle id parsed as a path variable is required.

**Response Structure**{

"success": true,

"message": "Vehicle 2013 TOYOTA CAMRY HYBRID successfully registered!",

"statusCode": 200,

"data": {

"id": 9,

"name": "2013 TOYOTA CAMRY HYBRID",

"description": "This vehicle is in great condition.",

"properties": {

"Highlights": "Run and Drive",

"Color": "WHITE",

},

"seller": 3,

"sellerDetails": null,

"condition": "USED",

"images": [……………….],

"lot": null,

"status": "IN\_EVALUATION",

"categoryName": null,

"vin": "51OYX30C000S370514",

"odometer": 1236344,

"model": "CAMRY HYBRID",

"modelId": 1,

"make": "TOYOTA",

"vehicleType": "Automobiles",

"buyNow": false,

"sellingPrice": 0,

"reservePrice": 14000000.0,

"bodyStyle": "Sedan 4d",

"regNo": "UBT 143C",

"transmission": "AUTOMATIC",

"fuelType": "PETROLEUM",

"inspected": false

}

}

**Get vehicle seller info**  
This endpoint helps you get a vehicle seller details

GET /inventory-service/api/v1/vehicles/{vehicleId}/seller

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

Only the vehicle id parsed as a path variable is required.

**Response Structure**{

"success": **true**,

"message": "",

"statusCode": 200,

"data": {

"id": 3,

"accountType": "SELLER",

"username": "embassy@oddjobs.tech",

"role": "ROLE\_SELLER",

"email": "embassy@oddjobs.tech",

"address": **null**,

"settings": {},

"isExpired": **false**,

"status": "ACTIVE",

"name": "American Embassy",

"nin": **null**,

"identification": **null**,

"isApproved": **false**,

"contact": **0414783822344**

}

}

**Get popular vehicles**  
This endpoint returns a paginated list of popular vehicle products; it is an open endpoint.

GET /inventory-service/api/v1/popular-vehicles

**Request parameters**

These will be specified as url query params

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| model | Integer | O | Filter results by model |
| offset | Integer | O | Represents pages starting from index 0 default is 0 |
| Limit | Integer | O | Represents the number of records to pull, be default it is set to 10 |

**Response structure**

The response is a list of vehicles

{

"success": **true**,

"message": "",

"statusCode": 200,

"data": **null**,

"entries": [

{

"id": 9,

"name": "2013 TOYOTA CAMRY HYBRID",

"description": "This vehicle is in great condition.",

"properties": {

"Highlights": "Run and Drive",

"Color": "WHITE",

},

"seller": 3,

"sellerDetails": null,

"condition": "USED",

"images": [……………….],

"lot": null,

"status": "IN\_EVALUATION",

"categoryName": null,

"vin": "51OYX30C000S370514",

"odometer": 1236344,

"model": "CAMRY HYBRID",

"modelId": 1,

"make": "TOYOTA",

"vehicleType": "Automobiles",

"buyNow": false,

"sellingPrice": 0,

"reservePrice": 14000000.0,

"bodyStyle": "Sedan 4d",

"regNo": "UBT 143C",

"transmission": "AUTOMATIC",

"fuelType": "PETROLEUM",

"inspected": false

}

],

"totalCount": 0,

"offset": 0,

"totalPages": 0,

"size": 0,

"page": 0

}

**Search vehicles**

This endpoint will be used by the vehicle finder to apply different filters, it takes in various search filters to allow you get desired filtered results.

NOTE this functionality is still a work in progress so you might encounter un-desired results.

GET /inventory-service/api/v1/vehicle-search

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

These are all parsed as url query params

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | Type | MOC | Description |
| odometers | min and max mileage values | O | Filters results by supplied odometer values |
| years | Min and max year values | O | Filters results by supplied year ranges |
| make | String | O | Filters results by make name |
| model | Integer | O | Filters results by selected model |
| transmission | String | O |  |
| offset | Integer | O | Defaults to 0 |
| limit | Integer | O | Defaults to 10 |

**Response structure**

The response is a list of vehicles just as one we saw above.

**Get vehicles**

GET /inventory-service/api/v1/get-vehicles

Headers   
Authorization: Bearer {{**accessToken**}}

This endpoint also returns a list of vehicles. The only difference between this and the one above is will be used to search a vehicle by name, sellerId, buyerId as well as using the isPublished flag.  
You can use this endpoint for buyers to search for published vehicles by model name, it can also be used in admin portals by bother admins and sellers to list vehicle.   
Note that when called by a seller, even it you do not specify the sellerId, only vehicles belonging to that seller will be returned.

**Request body parameters**

These are all parsed as url query params

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **MOC** | **Description** |
| queryString | String | O | A car name, this. Can come from the model and name e.g Tesla model X |
| sellerId | Integer | O | Can be used by admins to filter vehicles by seller |
| buyerId | Integer | O | Can be used to filter vehicles by buyer |
| isPublished | Boolean | O | Defaults to true, when set to true, only vehicles whose inspection reports have been uploaded will be returned along other filters will also apply. By default it is set to **true** |

**Response structure.**

A list of vehicles similar to one seen in the above endpoints.

**Upload vehicle inspection report**

This endpoint allows admins only to upload a vehicle inspection report after inspecting a vehicle.

Upon successful inspection report upload, the vehicle will be published for buyers to see and bid on.  
If there is a pending lot belonging to this seller, the vehicle is automatically added to this lot, else a new one is created automatically under this seller and the vehicle is added.  
The lot will inform when the auction begins.

POST /inventory-service/api/v1/vehicles/{id}/upload-inspection

Headers   
Authorization: Bearer {{**accessToken**}}

Content-Type: multipart/form-data

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **MOC** | **Description** |
| id | Integer | M | Vehicle id |
| name | String | M | Inspection title |
| descriptions | Text | M | Inspection descriptions |
| file | MultipartFile(PDF) (20Mbs limit) | O | Due to storage concerns this is optional |

**Response structure**

{

"success": **true**,

"message": "Inspection report Inspection-report9 has been uploaded successfully",

"statusCode": 200,

"data": {

"id": 1,

"name": "Inspection-report9",

"description": "This is the final inspection report for vehicle 9"

}

}

**Get vehicle inspection report**

This endpoint allows one pull a full vehicle inspection report if a file was uploaded.

GET /inventory-service/api/v1/vehicles/{vehicleId}/get-inspection

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

The vehicle Id parsed as a path variable in the url above.

**Response body**

If a file was uploaded during the first stage, this endpoint downloaded it to the local computer. You are advised to call this with a browser or handle the blob properly in axios.

**Create Vehicle Lot**

This endpoint allows admins and sellers create lots manually.  
Please note that lots are automatically created by the system and vehicles are automatically added to them. In an invent this does not happen. This endpoint helps a seller and admin create a lot.

POST /inventory-service/api/v1/create-lot

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **MOC** | **Description** |
| seller | Integer | M | Seller to which the lot should belong to |
| vehicles | List of vehicle Ids | O | Once supplied these will be added to the lot new automatically. They all must belong to the seller above |

**Response Structure**

A 200-response specifying the lot details

{

"id": 1,

"seller": 3,

"startDate": "yyyy-MM-dd HH:mm:ss",

"endDate": "yyyy-MM-dd HH:mm:ss"

"products": []

"status": “PENDING”

…..

}

**Add/Remove vehicles from lots**

This endpoint allows admins and sellers move vehicles to desired lots.

Vehicle lot allocation is automatic but you may wanna move vehicles around lots.

GET /inventory-service/api/v1/lot/add-remove/{lotId}

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **MOC** | **Description** |
| lotId | Integer | M | Parsed as a path variable to the URL and represents the target lot |
| vehicleId | Integer | M | Represents the vehicle db id which you want to add/remove from a given lot |
| add | Boolean | M | When set to true the vehicle will be added to that lot/ else it will be removed. |

**Response Structure**

Success 200 status code

Auctions service endpoints

These endpoints represent all auction services capabilities available via API.

Remember that auction sessions are automatically created when a vehicle is loaded and it’s dates are automatically adjusted when the vehicle is added to a lot.

**Get auctions**

This endpoint allows users to pull up a list of auctions. Sellers calling this endpoint will only see auctions related to their account.

GET /inventory-service/api/v1/auctions

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **MOC** | **Description** |
| lower\_date | Date (yyyy-MM-dd HH:mm:ss) | O |  |
| upper\_date | Date (yyyy-MM-dd HH:mm:ss) | O |  |
| status | Enum PENDING, IN\_PROGRESS, COMPLETED, CANCELLED | O |  |
| seller | Integer | O |  |
| page | Integer | O | Defaults to 0 |
| size | Integer | O | Defaults to 10 |

**Update auction**

This endpoint allows sellers and admins to update a give auction session.

POST /inventory-service/api/v1/ auction/{auctionId}

Headers   
Authorization: Bearer {{**accessToken**}}

**Request body parameters**

This Api only allows you to adjust the vehicle reservePrice, minBidAmount related to this auction. You can also do this by updating the vehicle details.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Type** | **MOC** | **Description** |
| reservePrice | currency | O | Change the car auction reservePrice |
| minBidAmount | Integer | O | Adjust the minimum bid amount during bidding |

**Response Structure**

Success 200 status code with the new updated auction object