

# MySQL API Server (0.0.1)

Download OpenAPI specification: [Download](#)

Allows users to access a database with access management

## Authentication

### OAuth2PasswordBearer

<b>Security Scheme Type</b>	OAuth2
<b>password OAuth Flow</b>	<b>Token URL:</b> /v1/auth/get_token <b>Scopes:</b>

## Authentication

### Authenticate API user

Authenticate an API user and return a token that will be passed in the authorization header for subsequent requests

REQUEST BODY SCHEMA: application/x-www-form-urlencoded

grant_type	string (Grant Type) password
username required	string (Username)
password required	string (Password)
scope	string (Scope) Default: <input type="text" value=""/>
client_id	string (Client Id)
client_secret	string (Client Secret)

## Responses

> **200** Returns user private access token. Make sure to keep the token secret.

> **400** Bad Request

> **422** Validation Error

POST /v1/auth/get\_token

## Response samples

200

400

422

Content type  
application/json

Copy Expand all Collapse all

```
{  
  "access_token": "string",  
  "token_type": "string"  
}
```

---

# Query

## Create a table in the database

AUTHORIZATIONS: OAuth2PasswordBearer

### QUERY PARAMETERS

---

table\_name  
required

string (Table Name)  
Table name in database, eg `categories`

---

field\_defs  
required

string (Field Defs)  
List of column names followed by their types, eg `Category_ID int(11),  
Category_Name varchar(500)`

---

## Responses

> **200** Returns status of request

> **400** Bad Request

> **401** Unauthorized

> **403** Forbidden

> **422** Validation Error

POST /v1/query/create

Response samples

200

400

401

403

422

Content type  
application/json

Copy Expand all Collapse all

```
{
  "detail": "string"
}
```

# Read record from the database

Read a record from the database and return all rows as a list of tuples

AUTHORIZATIONS: OAuth2PasswordBearer

QUERY PARAMETERS

what_to_select required	string (What To Select) List of columns, or * to indicate all columns, eg <code>Category_Name, ICON_URL</code>
which_table required	string (Which Table) Name of table to retrieve data from, eg <code>categories</code>
conditions_to_satisfy	string (Conditions To Satisfy) If present, specifies one or more conditions that records must satisfy for retrieval, eg <code>Category_ID = 1</code>

## Responses

> 200 Returns the query result as a list of tuples

> 400 Bad Request

> 401 Unauthorized

> 403 Forbidden

> 422 Validation Error

GET /v1/query/read

## Response samples

200

400

401

403

422

Content type  
application/json

Copy Expand all Collapse all

```
{
  - "rows": [
    null
  ]
}
```

## Insert record into database table

AUTHORIZATIONS: OAuth2PasswordBearer

QUERY PARAMETERS

table\_name  
required

string (Table Name)  
Name of table to insert record into, eg categories

values  
required

string (Values)

List of field values for the record with each surrounded by single quotes, eg

```
'1', 'IT', 'google.com'
```

column\_names

string (Column Names)

List of column names that you are specifying values for in the record, eg

`Category_ID, Category_Name, ICON_URL`. If you are adding values for all the columns of the table, you do not need to specify this parameter

## Responses

> **200** Returns status of request

> **400** Bad Request

> **401** Unauthorized

> **403** Forbidden

> **422** Validation Error

POST /v1/query/insert

## Response samples

200

400

401

403

422

Content type

application/json

Copy

Expand all

Collapse all

```
{
  "detail": "string"
}
```

# Update record in the database

AUTHORIZATIONS:

OAuth2PasswordBearer

## QUERY PARAMETERS

---

table\_name  
*required*

string (Table Name)

Name of table in which record will be updated, eg `categories`

---

set\_statements  
*required*

string (Set Statements)

List of columns to modify and their respective values (surrounded by single quotes), eg `Category_Name = 'IT', ICON_URL = 'google.com'`

---

where\_condition

string (Where Condition)

Specifies the conditions that identify which records to update, eg

`Category_ID = 1`. If omitted, all records are updated

---

## Responses

> **200** Returns status of request

> **400** Bad Request

> **401** Unauthorized

> **403** Forbidden

> **422** Validation Error

PUT /v1/query/update

## Response samples

**200**

**400**

**401**

**403**

**422**

Content type

application/json

```
{  
  "detail": "string"  
}
```

## Delete record from the database

AUTHORIZATIONS: OAuth2PasswordBearer

### QUERY PARAMETERS

table_name required	string (Table Name) Name of table in which record will be deleted, eg <code>categories</code>
where_condition	string (Where Condition) Specifies which record or records should be deleted, eg <code>Category_Name = 'IT'</code> . If omitted, all records will be deleted!

## Responses

> **200** Returns status of request

> **400** Bad Request

> **401** Unauthorized

> **403** Forbidden

> **422** Validation Error

DELETE /v1/query/delete

## Response samples



**200****400****401****403****422****Content type**

application/json

Copy

Expand all

Collapse all

```
{  
  "detail": "string"  
}
```

## Info

### Retrieve information regarding user given their token

Retrieve information regarding a user (such as permissions) given their token

AUTHORIZATIONS:[OAuth2PasswordBearer](#)

### Responses

> **200** Returns information about user

> **401** Unauthorized

> **403** Forbidden

GET /v1/info/get\_info

## Response samples

200

401

403

Content type

application/json

Copy

Expand all

Collapse all

```
{
  "username": "string",
  "is_admin": 0,
  "action_permissions": "string",
  "table_permissions": "string"
}
```