CRM

Db diagram URL : <https://dbdiagram.io/d/crm-65e9f7f8b1f3d4062c64191d>

Contact info : [Alifareeq@aswar-g.com](mailto:Alifareeq@aswar-g.com)

Last edited : 2024/05/15

# 1- Models

In this app there are 8 models:

Id be carful if i were you  
Ctrl A + del

Hmmmm your worst enemy !! who are you ?/ then you are in danger

1. Opportunity:

* Represents a potential sales opportunity within the CRM system.
* Attributes include name, sales person, customer, expected revenue, expected closing date, stage, internal notes, probability, priority, state, address, country, medium, tags, and associated activity.

- Each opportunity is associated with a sales person, a customer, and various other attributes defining its details.

- Has a dependency on other models such as User, Contact, Stage, SalesTeam, Note, Priority, State, Country, Medium, Tag.

2. LostOpportunity:

- Represents an opportunity that was lost within the CRM system.

- Contains the lost opportunity itself and a description of why it was lost.

- Linked to the Opportunity model.

3. Medium:

- Represents different mediums used for marketing or communication purposes.

- Attributes include name and description.

- Mediums can be channels like advertising platforms, email, etc.

4. Note:

- Represents a note within the CRM system.

- Stores information about notes, including title and content.

- Notes can be associated with customers or other entities.

5. Priority:

- Represents priority levels within the CRM system.

- Attributes include scale and value to define the priority level.

-Priorities help in categorizing opportunities or tasks based on their importance.

6. SalesTeam:

- Represents a sales team associated with opportunities.

- Contains information about sales teams, including name, leader, and members.

- Sales teams are responsible for handling opportunities and collaborating on sales efforts.

7. Stage:

- Represents different stages in the sales process within the CRM system.

- Attributes include name, sequence number, total expected revenue, next stage order, and flags indicating if it's a winning or losing stage.

- Stages help in tracking the progress of opportunities through the sales pipeline.

8. Tag:

- Represents tags within the CRM system.

- Tags are used for categorizing or labeling various entities such as opportunities, contacts, etc.

- Attributes include name and color for visual differentiation.

- Tags provide a flexible way to organize and filter data within the CRM.

Each of these models has its own api for crud operation that i will discuss in the next page

# 2- Api Endpoints

For each model that has its own URL route and extra action that do necessary action for the required model.

The structure for the api as following :

*{{baseUrl}}/<model>/?extra\_parameters=”my extra url parameter”*

Where the <model> is the required model for making crud operation.

For example if you want to create a new note do the following:

Method : POST

URL: {{baseUrl}}/note/

Body:

{

“Title”: ”MY AWESOME TITLE”,

“DESCRIPTION”: ”GIVE IT YOUR BEST ”

}

Here is a list of all possible endpoints :

## SalesTeam

1. List Sales Teams:

* HTTP Method: GET
* URL: {{baseUrl}}/sales-team/
* Description: Retrieve a list of all Sales Team instances.

1. Retrieve Sales Team:

* HTTP Method: GET
* URL: {{baseUrl}}/sales-team/{id}/
* Description: Retrieve details of a specific Sales Team instance by its ID.

1. Create sales Team:

* HTTP Method: POST
* URL: {{baseUrl}}/sales-team/
* Description: Create a new Sales Team instance.

1. Update Sales Team:

* HTTP Method: PUT or PATCH
* URL: {{baseUrl}}/sales-team/{id}/
* Description: Update an existing Sales Team instance by its ID.

1. Delete Sales Team:

* HTTP Method: DELETE
* URL: {{baseUrl}}/sales-team/{id}/
* Description: Delete a Sales Team instance by its ID.

## Note:

SalesTeam

1. List Sales Teams:
   * HTTP Method: GET
   * URL: {{baseUrl}}/sales-team/
   * Description: Retrieve a list of all Sales Team instances.
2. Retrieve Sales Team:
   * HTTP Method: GET
   * URL: {{baseUrl}}/sales-team/{id}/
   * Description: Retrieve details of a specific Sales Team instance by its ID.
3. Create Sales Team:
   * HTTP Method: POST
   * URL: {{baseUrl}}/sales-team/
   * Description: Create a new Sales Team instance.
4. Update Sales Team:
   * HTTP Method: PUT or PATCH
   * URL: {{baseUrl}}/sales-team/{id}/
   * Description: Update an existing Sales Team instance by its ID.
5. Delete Sales Team:
   * HTTP Method: DELETE
   * URL: {{baseUrl}}/sales-team/{id}/
   * Description: Delete a Sales Team instance by its ID.

## 

## 

## Lost opportunity:

Routes and Operations for LostOpportunity Model

1. List Lost Opportunities:
   * HTTP Method: GET
   * URL: {{baseUrl}}/lost-opportunity/
   * Description: Retrieve a list of all Lost Opportunity instances.
2. Retrieve Lost Opportunity:
   * HTTP Method: GET
   * URL: {{baseUrl}}/lost-opportunity/{id}/
   * Description: Retrieve details of a specific Lost Opportunity instance by its ID.
3. Create Lost Opportunity:
   * HTTP Method: POST
   * URL: {{baseUrl}}/lost-opportunity/
   * Description: Create a new Lost Opportunity instance.
4. Update Lost Opportunity:
   * HTTP Method: PUT or PATCH
   * URL: {{baseUrl}}/lost-opportunity/{id}/
   * Description: Update an existing Lost Opportunity instance by its ID.
5. Delete Lost Opportunity:
   * HTTP Method: DELETE
   * URL: {{baseUrl}}/lost-opportunity/{id}/
   * Description: Delete a Lost Opportunity instance by its ID

## 

## Priority:

1. List Priorities:
   * HTTP Method: GET
   * URL: {{baseUrl}}/priority/
   * Description: Retrieve a list of all Priority instances.
2. Retrieve Priority:
   * HTTP Method: GET
   * URL: {{baseUrl}}/priority/{id}/
   * Description: Retrieve details of a specific Priority instance by its ID.
3. Create Priority:
   * HTTP Method: POST
   * URL: {{baseUrl}}/priority/
   * Description: Create a new Priority instance.
4. Update Priority:
   * HTTP Method: PUT or PATCH
   * URL: {{baseUrl}}/priority/{id}/
   * Description: Update an existing Priority instance by its ID.
5. Delete Priority:
   * HTTP Method: DELETE
   * URL: {{baseUrl}}/priority/{id}/
   * Description: Delete a Priority instance by its ID.

## 

## 

## 

## Stage:

1. List Stages:
   * HTTP Method: GET
   * URL: {{baseUrl}}/stage/
   * Description: Retrieve a list of all Stage instances.
2. Retrieve Stage:
   * HTTP Method: GET
   * URL: {{baseUrl}}/stage/{id}/
   * Description: Retrieve details of a specific Stage instance by its ID.
3. Create Stage:
   * HTTP Method: POST
   * URL: {{baseUrl}}/stage/
   * Description: Create a new Stage instance.
4. Update Stage:
   * HTTP Method: PUT or PATCH
   * URL: {{baseUrl}}/stage/{id}/
   * Description: Update an existing Stage instance by its ID.
5. Delete Stage:
   * HTTP Method: DELETE
   * URL: {{baseUrl}}/stage/{id}/
   * Description: Delete a Stage instance by its ID.

## 

## 

## Tag:

1. List Tags:
   * HTTP Method: GET
   * URL: {{baseUrl}}/tag/
   * Description: Retrieve a list of all Tag instances.
2. Retrieve Tag:
   * HTTP Method: GET
   * URL: {{baseUrl}}/tag/{id}/
   * Description: Retrieve details of a specific Tag instance by its ID.
3. Create Tag:
   * HTTP Method: POST
   * URL: {{baseUrl}}/tag/
   * Description: Create a new Tag instance.
4. Update Tag:
   * HTTP Method: PUT or PATCH
   * URL: {{baseUrl}}/tag/{id}/
   * Description: Update an existing Tag instance by its ID.
5. Delete Tag:
   * HTTP Method: DELETE
   * URL: {{baseUrl}}/tag/{id}/
   * Description: Delete a Tag instance by its ID.

## 

## 

## 

## Medium:

1. List Mediums:
   * HTTP Method: GET
   * URL: {{baseUrl}}/medium/
   * Description: Retrieve a list of all Medium instances.
2. Retrieve Medium:
   * HTTP Method: GET
   * URL: {{baseUrl}}/medium/{id}/
   * Description: Retrieve details of a specific Medium instance by its ID.
3. Create Medium:
   * HTTP Method: POST
   * URL: {{baseUrl}}/medium/
   * Description: Create a new Medium instance.
4. Update Medium:
   * HTTP Method: PUT or PATCH
   * URL: {{baseUrl}}/medium/{id}/
   * Description: Update an existing Medium instance by its ID.
5. Delete Medium:
   * HTTP Method: DELETE
   * URL: {{baseUrl}}/medium/{id}/
   * Description: Delete a Medium instance by its ID

## 

## 

## Opportunity :

1. List Opportunities
   * URL: {{baseUrl}}/opportunity/
   * HTTP Method: GET
   * Description: Retrieve a list of all Opportunity instances.
2. Retrieve Opportunity
   * URL: {{baseUrl}}/opportunity/{id}/
   * HTTP Method: GET
   * Description: Retrieve details of a specific Opportunity instance by its ID.
3. Create Opportunity
   * URL: {{baseUrl}}/opportunity/
   * HTTP Method: POST
   * Description: Create a new Opportunity instance.
4. Update Opportunity
   * URL: {{baseUrl}}/opportunity/{id}/
   * HTTP Method: PUT or PATCH
   * Description: Update an existing Opportunity instance by its ID.
5. Delete Opportunity
   * URL: {{baseUrl}}/opportunity/{id}/
   * HTTP Method: DELETE
   * Description: Delete an Opportunity instance by its ID.
6. Move to Win Stage
   * URL: {{baseUrl}}/opportunity/{id}/move\_to\_win\_stage/
   * HTTP Method: PUT or PATCH
   * Description: Moves the specified Opportunity instance to a winning stage.
7. Move to Lost Stage
   * URL: {{baseUrl}}/opportunity/{id}/move\_to\_lost\_stage/
   * HTTP Method: PUT or PATCH
   * Description: Moves the specified Opportunity instance to a losing stage.
8. Create Quotation
   * URL: {{baseUrl}}/opportunity/{id}/create\_quotation/
   * HTTP Method: POST
   * Description: Creates a quotation based on the specified Opportunity instance.

## Response Examples

<https://devserv2.aswar.solutions/swagger/>

4-

# 3- designed flow

* Make sure you have created the following objects:

1. Contact, having associated currency (from base app) with him
2. User

* Create 3 stages one is winning stage, the other is lost , and finally a stage that isn’t won or lost
* Create sales team
* Create sales person or choose exist one (you should have their ID)
* Now you should be good to go for testing basic opportunity operations
* (optional) create Tag, Medium, Note, Tag and Priority as referred in [This section](#_py8p5dycihhf)
* After creating an opportunity you can create a quotation from this specific opportunity using Create quotation referred in the endpoints
* Also you move specific opportunity to a winning stage or lost stage using Move to winning stage or lost stage endpoint referred above.