



Swiggy Dummy Data API Documentation

This in-house API provides Swiggy-like restaurant and menu data for food delivery applications. The API offers two main endpoints for fetching restaurant listings and detailed menu information.

Base URL: `https://namastedev.com/api/v1`

Important Note: Use the dummy images for menu and restaurant

Endpoints

1. Get Restaurant List

Retrieve a list of all available restaurants with their basic information.

Endpoint

```
GET /listRestaurants
```

Request

URL: `https://namastedev.com/api/v1/listRestaurants`

Method: GET

Query Parameters: None required

Headers: None required

Response

Success Response (200 OK)

```
{
  success: true,
  message: "Restaurant List fetched successfully",
  data: {
    data: {
      cards: [
        {},
        {
          card: {
            card: {
              gridElements: {
                infoWithStyle: {
                  restaurants: [
                    {
                      info: {
                        id: "123456",
                        name: "Pizza Paradise",
                        clouinaryImageId:
                          "rng/md/carousel/production/pizza123",
                        locality: "MG Road",
                        areaName: "Central District",
                        costForTwo: "₹400 for two",
                        cuisines: ["Pizza", "Italian", "Fast Food"],
                        avgRating: 4.3,
                        avgRatingString: "4.3",
                        totalRatingsString: "10K+ ratings",
                        veg: false,
                        sla: {
                          deliveryTime: 30,
                          lastMileTravel: 3.5,
                          slaString: "30 mins",
                        },
                      },
                      aggregatedDiscountInfoV3: {
                        header: "50% OFF",
                        subHeader: "UPTO ₹100",
                      },
                    },
                    // ... more restaurants
                  ],
                },
              },
            },
          },
        },
      ],
    },
  },
}
```

```
    },  
    },  
    },  
    },  
  ],  
},  
},  
};
```

Response Fields

Field	Type	Description
id	String	Unique restaurant identifier
name	String	Restaurant name
cloudinaryImageId	String	Image ID for restaurant logo
locality	String	Street/area location
areaName	String	District/zone name
costForTwo	String	Average cost for two people
cuisines	Array	List of cuisine types
avgRating	Number	Average rating (0-5)
totalRatingsString	String	Total number of ratings
veg	Boolean	Vegetarian restaurant indicator
sla.deliveryTime	Number	Estimated delivery time in minutes
sla.lastMileTravel	Number	Distance in kilometers
aggregatedDiscountInfoV3	Object	Current discount offers

2. Get Restaurant Menu

Retrieve detailed menu information for a specific restaurant including all categories and items.

Endpoint

```
GET /listRestaurantMenu/:id
```

Request

URL: `https://namastedev.com/api/v1/listRestaurantMenu/123456`

Method: `GET`

Path Parameters:

Parameter	Type	Required	Description
<code>id</code>	String	Yes	Restaurant ID (e.g., "123456")

Headers: None required

Response

Success Response (200 OK)

```
{
  status: true,
  message: "Restaurant Menu fetched successfully",
  data: {
    cards: [
      {},
      {},
      {
        card: {
          card: {
            info: {
              id: "123456",
              name: "Pizza Paradise",
              clouidinaryImageId: "rng/md/carousel/production/pizza123",
              locality: "MG Road",
            }
          }
        }
      }
    ]
  }
}
```

```

        areaName: "Central District",
        costForTwo: "₹400 for two",
        costForTwoMessage: "₹400 for two",
        cuisines: ["Pizza", "Italian", "Fast Food"],
        avgRating: 4.3,
        avgRatingString: "4.3",
        totalRatingsString: "10K+ ratings",
        sla: {
            deliveryTime: 30,
            lastMileTravel: 3.5,
            lastMileTravelString: "3.5 km",
            slaString: "30 mins",
        },
    },
},
},
},
{
    groupedCard: {
        cardGroupMap: {
            REGULAR: {
                cards: [
                    {},
                    {
                        card: {
                            card: {
                                "@type":
                                    "type.googleapis.com/swiggy.ItemCategory",
                                title: "Recommended",
                                itemCards: [
                                    {
                                        card: {
                                            info: {
                                                id: "item_001",
                                                name: "Margherita Pizza",
                                                category: "Pizza",
                                                description:
                                                    "Classic delight with 100% real",
                                                imageId: "pizza_margherita_123",
                                                price: 29900,
                                                defaultPrice: 29900,
                                            },
                                        },
                                    },
                                ],
                            },
                        },
                    },
                ],
            },
        },
    },
}

```

```

        card: {
          info: {
            id: "item_002",
            name: "Pepperoni Pizza",
            category: "Pizza",
            description:
              "American classic with",
            imageId: "pizza_pepperoni_456",
            price: 39900,
          },
        },
      },
    },
    {
      card: {
        info: {
          id: "item_003",
          name: "Veggie Supreme",
          category: "Pizza",
          description:
            "Loaded with fresh vegetables...",
          imageId: "pizza_veggie_789",
          price: 34900,
          defaultPrice: 34900,
        },
      },
    },
  ],
},
{
  card: {
    card: {
      "@type":
        "type.googleapis.com/swiggy.presentation",
      title: "Specialty Pizzas",
      itemCards: [
        {
          card: {
            info: {
              id: "item_004",
              name: "BBQ Chicken Pizza",
              category: "Pizza",
              description:
                "Grilled chicken with onions, and cheese",
              imageId: "pizza_bbq_chicken_101",
            },
          },
        },
      ],
    },
  },
},

```

```

        price: 44900,
      },
    },
  },
  {
    card: {
      info: {
        id: "item_005",
        name: "Hawaiian Pizza",
        category: "Pizza",
        description:
          "Tropical delight with ham",
        imageId: "pizza_hawaiian_202",
        price: 42900,
        defaultPrice: 42900,
      },
    },
  },
],
},
{
  card: {
    card: {
      "@type":
        "type.googleapis.com/swiggy.presentation",
      title: "Beverages",
      itemCards: [
        {
          card: {
            info: {
              id: "item_006",
              name: "Coca Cola",
              category: "Beverages",
              description: "Chilled 500ml Coca Cola",
              imageId: "beverage_coke_707",
              price: 5900,
              defaultPrice: 5900,
            },
          },
        },
      ],
    },
  },
},
},

```

```
    ],  
    },  
  },  
},  
],  
},  
},  
};
```

Error Response (404 Not Found)

```
{  
  "success": false,  
  "message": "Restaurant menu not found"  
}
```

Error Response (400 Bad Request)

```
{  
  "success": false,  
  "message": "Restaurant ID is required"  
}
```

Menu Response Fields

Restaurant Info:

Field	Type	Description
id	String	Restaurant ID
name	String	Restaurant name
costForTwoMessage	String	Cost for two people
cuisines	Array	Cuisine types
avgRatingString	String	Average rating
totalRatingsString	String	Total ratings count

Menu Items:

Field	Type	Description
@type	String	Category type identifier
title	String	Category name (e.g., "Recommended")
itemCards	Array	List of menu items in category
id	String	Unique item identifier
name	String	Dish name
category	String	Item category
description	String	Item description
imageId	String	Image identifier
price	Number	Price in paise (divide by 100 for rupees)
defaultPrice	Number	Default/base price in paise

Available Restaurants

Restaurant ID	Name	Cuisine	Cost for Two
123456	Pizza Paradise	Pizza, Italian	₹400
234567	Burger Hub	Burgers, American	₹300
345678	Green Bites	Healthy, Salads	₹250
456789	Spice Kingdom	Indian, Biryani	₹500
567890	Chinese Dragon	Chinese, Asian	₹350
678901	Dessert Delight	Desserts, Ice Cream	₹200
789012	Sushi Station	Japanese, Sushi	₹800
890123	South Spice	South Indian, Dosa	₹300

Restaurant ID	Name	Cuisine	Cost for Two
901234	Pasta Palace	Italian, Pasta	₹450
012345	BBQ Nation	BBQ, Grilled	₹600

Usage Examples

JavaScript (Fetch API)

Get Restaurant List

```
fetch("https://namastudev.com/api/v1/listRestaurants")
  .then((response) => response.json())
  .then((data) => {
    const restaurants =
      data.data.cards[1].card.card.gridElements.infoWithStyle.restaurants;

    console.log(restaurants);
  })
  .catch((error) => console.error("Error:", error));
```

Get Restaurant Menu

```
const restaurantId = "123456";

fetch(`https://namastudev.com/api/v1/listRestaurantMenu/${restaurantId}`)
  .then((response) => response.json())
  .then((data) => {
    const menuCategories =
      data.data.cards[4].groupedCard.cardGroupMap.REGULAR.cards;

    console.log(menuCategories);
  })
  .catch((error) => console.error("Error:", error));
```

JavaScript (Async/Await)

```
// Get all restaurants
async function getRestaurants() {
  try {
    const response = await fetch(
      "https://namastudev.com/api/v1/listRestaurants"
    );
    const data = await response.json();
    return
    data.data.cards[1].card.card.gridElements.infoWithStyle.restaurants;
  } catch (error) {
    console.error("Error fetching restaurants:", error);
  }
}

// Get specific restaurant menu
async function getRestaurantMenu(restaurantId) {
  try {
    const response = await fetch(
      `https://namastudev.com/api/v1/listRestaurantMenu/${restaurantId}`
    );

    const data = await response.json();
    return data.data;
  } catch (error) {
    console.error("Error fetching menu:", error);
  }
}

// Usage
(async () => {
  const restaurants = await getRestaurants();
  console.log("Restaurants:", restaurants);

  const menu = await getRestaurantMenu("123456");
  console.log("Menu:", menu);
})();
```

React Example

```
import { useState, useEffect } from "react";

function RestaurantList() {
  const [restaurants, setRestaurants] = useState([]);
  const [loading, setLoading] = useState(true);
```

```

useEffect(() => {
  fetchRestaurants();
}, []);

const fetchRestaurants = async () => {
  try {
    const response = await fetch(
      "https://namastedev.com/api/v1/listRestaurants"
    );

    const json = await response.json();

    const restaurantData =
      json.data.cards[1]?.card?.card?.gridElements?.infoWithStyle
        ?.restaurants;

    setRestaurants(restaurantData || []);
  } catch (error) {
    console.error("Error:", error);
  } finally {
    setLoading(false);
  }
};

if (loading) return <div>Loading...</div>;

return (
  <div className="restaurant-list">
    {restaurants.map((restaurant) => (
      <div key={restaurant.info.id}>
        <h3>{restaurant.info.name}</h3>

        <p>{restaurant.info.cuisines.join(", ")}</p>

        <p>Rating: {restaurant.info.avgRatingString}</p>
      </div>
    ))}
  </div>
);
}

```

Axios Example

```
import axios from "axios";

const API_BASE_URL = "https://namastedev.com/api/v1";

// Get restaurant list
export const getRestaurants = async () => {
  try {
    const { data } = await axios.get(`${API_BASE_URL}/listRestaurants`);

    return
    data.data.cards[1].card.card.gridElements.infoWithStyle.restaurants;
  } catch (error) {
    throw error;
  }
};

// Get restaurant menu
export const getRestaurantMenu = async (restaurantId) => {
  try {
    const { data } = await axios.get(
      `${API_BASE_URL}/listRestaurantMenu/${restaurantId}`
    );

    return data.data;
  } catch (error) {
    throw error;
  }
};
```

Notes

- **Price Format:** All prices are in paise (Indian currency). Divide by 100 to get rupees.
- Example: 29900 paise = ₹299.00
- **Image URLs:** Use the dummy images for menu and restaurant
- **Data Structure:** The API returns nested data structures similar to Swiggy's actual API for easy integration.
- **Menu Categories:** Filter menu items using the @type field:

```
const MENU_TYPE=  
  "type.googleapis.com/swiggy.presentation.food.v2.ItemCategory";  
const categories=cards.filter(c => c.card?.card?.["@type"] === MENU_TYPE);
```

- **No Authentication Required:** This API is open and doesn't require API keys or authentication tokens.