# **Global Analysis Endpoints**

## **1. Top Importers**

**Endpoint:** /top\_importers **Method:** POST

**Description:** Retrieves the top 5 importers of strawberry based on the available data.

**Request Body:**

{

"plant\_time": "YYYY-MM-DD",

"code": "product\_code" #081110

}

**Response:**

{

"top\_importers": [

{ year: 2023, Canada: 8828351, France: 3140471, Belgium: 2025136, Netherlands: 1692905, "United States of America": 3444156 }

],

["countries": ["Country1", "Country2", "Country3",...],

[“code” : code],

[ “month” : month ]

}

**Error Response:**

{

"error": "Error message"

}

## **2. Receive Country**

**Endpoint:** POST /receive\_country

**Description:** Receives the selected country code.

**Request Body:**

{

"country": "Egypt"

}

**Response:**

{

"message": "Data received",

"country": "Egypt"

}

## 

## **3. Recommended Month**

**Endpoint:** POST /recommended\_month

**Description:** Calculates the sum of quantities for the selected importer over the last 5 years.

**Request Body:**

{

"month": 3,

"country": "",

"code": "081110"

}

**Response:**

{

"sum\_current\_month": 1000,

"sum\_next\_month": 1200,

"sum\_after\_next\_months": 1400

}

## 

## 

## 

## 

## **4. Country Price**

**Endpoint:** POST /country\_price

**Description:** Fetches the latest price data for the selected country and code.

**Request Body:**

{

"code": "081110",

"country": “China"

}

**Response:**

{

"2024": 50.5,

"2023": 48.0,

"2022": 45.0

}

## 

## 

## 

## **5. Country Quantity**

**Endpoint:** POST /country\_quantity

**Description:** Fetches the latest quantity data for the selected country and code.

**Request Body:**

{

"code": "081110",

"country": "China"

}

**Response:**

{

"2024": 1000,

"2023": 950,

"2022": 900

}

## 

## 

## 

## 

## **6. Country Growth Value**

**Endpoint:** POST /country\_growth\_value

**Description:** Fetches the growth value data for the selected country and code.

**Request Body:**

{

"code": "081110",

"country": ""

}

**Response:**

{

"2024": 10.5,

"2023": 9.8,

"2022": 8.9

}

## 

## 

## 

## 

## **7. Country Growth Quantity**

**Endpoint:** POST /country\_growth\_quantity

**Description:** Fetches the growth quantity data for the selected country and code.

**Request Body:**

{

"code": "081110",

"country": ""

}

**Response:**

{

"2024": 1050,

"2023": 980,

"2022": 890

}

## 

## 

## 

## 

## **8. Competitors (Monafseen)**

**Endpoint:** GET /monafseen

**Description:** Fetches the top competitors in the global supply market.

**Response:**

{

"exporters": [

{

"exporters": "Country Name",

"20014": 104,

"20015": 135,

"20016": 120,

"20017": 115 },

{

"exporters": "Another Country",

"20014": 9,

"20015": 25,

"20016": 38,

"20017": 40

}

]

}

## 

## **9. Zo2 3am**

**Endpoint:** POST /zo2\_3am

**Description:** Fetches the general agricultural conditions for the selected country.

**Request Body:**

{

"country": "China"

}

**Response:**

{

["country": "China",

"preferred\_taste": "Sweet with mild acidity",

"preferred\_size": "Large",

"preferred\_color": "Bright red with gloss",

"packaging\_preferences": "Some value",

"shelf\_life\_importance": "Some value",

"sugar\_content\_brix\_percentage": "Some value",

"strawberry\_variety": "Some variety"],

[ “variety” : variety ]

}

## 

## 

## 

## 

## 

## **10. Fertilization Data (Tasmeed)**

**Endpoint:** GET /tasmeed

**Description:** Fetches fertilization data.

**Response:**

{

"stage": "Vegetative",

"duration\_days": 30,

"npk\_ratio": "10-10-10",

"ec\_ds\_m": 1.5,

"ph": 6.5,

"temperature": 22.5,

"humidity\_percent": 60,

"light\_hours": 12,

"iron\_fe\_ppm": 3.5,

"calcium\_ca\_ppm": 150,

"magnesium\_mg\_ppm": 40,

"boron\_b\_ppm": 0.5,

"zinc\_zn\_ppm": 1.2

}

## 

## **11. Recommend Tasmeed**

**Endpoint:** GET /recommend\_tasmeed

**Description:** Retrieves the recommended fertilizer based on the strawberry variety.

**Request Body:**

{

variety = data.get("variety")

    country = data.get("country")

}

**Response:**

{

return a.jsonify({"variety": variety, "country": country, "recommendation": recommendations[variety][country]})

}

**Error Response:**

{

if not variety or not country:

        return a.jsonify({"error": "Please provide both 'variety' and 'country' as query parameters."}), 400

    # ✅ التحقق من أن `variety` موجود في القاموس

    if variety not in recommendations:

        return a.jsonify({"error": f"Variety '{variety}' not found. Please provide a valid variety."}), 404

    # ✅ التحقق من أن `country` موجود في `variety`

    if country not in recommendations[variety]:

        return a.jsonify({"error": f"No recommendations found for variety '{variety}' in '{country}'."}), 404

}