

Service Endpoints Overview

The following RESTful service endpoints will facilitate interaction between the user interface and the MongoDB database. Each endpoint is designed based on the goals of user personas, including farmers, agronomists, and agricultural researchers. Each endpoint follows RESTful principles and uses appropriate HTTP verbs.

Endpoint URL	HTTP Verb	Purpose
/api/fields	GET	Retrieve a list of all fields for the logged-in user.
/api/fields/:fieldId	GET	Retrieve detailed information about a specific field.
/api/fields	POST	Create a new field setup.
/api/fields/:fieldId	PUT	Update an existing field's information.
/api/fields/:fieldId	DELETE	Delete a specific field.
/api/field-scans	GET	Retrieve a list of recent field scans.
/api/field-scans/:scanId	GET	Get detailed data of a specific field scan.
/api/field-scans	POST	Add a new field scan record.
/api/alerts	GET	Retrieve a list of all alerts for the logged-in user.
/api/alerts/:alertId	GET	Get detailed information about a specific alert.
/api/users/:userId/preferences	GET	Retrieve user preferences.
/api/users/:userId/preferences	PUT	Update user alert preferences.
/api/reports	POST	Generate a custom report for a user.
/api/reports/:reportId	GET	Retrieve a saved report by ID.
/api/drones	GET	Retrieve a list of all drones and their status.
/api/drones/:droneId/maintenance	POST	Add a new maintenance log for a drone.

Endpoint Descriptions and Example Requests/Responses

GET /api/fields

- **Purpose:** Retrieve all fields created by the logged-in user.
- **Example Request:**

Example Response:

```
[  
  {  
    "_id": "field123",
```

```
"fieldName": "North Field",
"location": { "latitude": 38.1234, "longitude": -91.5678 },
"cropType": "corn"
},
{
  "_id": "field456",
  "fieldName": "East Field",
  "location": { "latitude": 39.1234, "longitude": -92.5678 },
  "cropType": "wheat"
}
]
```

Error Response:

```
{
  "status": 404,
  "message": "No fields found for the user."
}
```

POST /api/fields

- **Purpose:** Create a new field setup.
- **Example Request:**

Example Request:

```
{
  "fieldName": "South Field",
  "location": { "latitude": 37.9876, "longitude": -90.1234 },
  "boundaryCoordinates": [
    { "lat": 37.988, "lng": -90.124 },
    { "lat": 37.986, "lng": -90.122 }
  ],
  "cropType": "soybeans"
}
```

Example Response:

```
{
  "status": 201,
  "message": "Field created successfully.",
  "fieldId": "field789"
}
```

Error Response:

```
{
  "status": 400,
  "message": "Field creation failed. Missing required data."
}
```

```
}
```

GET */api/field-scans/:scanId*

- **Purpose:** Retrieve detailed data for a specific field scan.

Example Request:

```
GET /api/field-scans/scan123
```

Example Response:

```
{
  "_id": "scan123",
  "fieldId": "field123",
  "scanDate": "2024-11-01T14:00:00Z",
  "cropMetrics": {
    "moistureLevel": 35,
    "chlorophyllContent": 75,
    "plantHeight": 120
  },
  "alerts": [
    {
      "alertType": "low moisture",
      "severity": "medium",
      "recommendation": "Increase irrigation."
    }
  ]
}
```

Error Response:

```
{
  "status": 404,
  "message": "Scan not found."
}
```

PUT */api/users/:userId/preferences*

- **Purpose:** Update user alert preferences.

Example Request:

```
{
  "alertThresholds": {
```

```
"moistureLevel": 30,  
"chlorophyllContent": 60  
}  
}
```

Example Response:

```
{  
  "status": 200,  
  "message": "Preferences updated successfully."  
}
```

Error Response:

```
{  
  "status": 400,  
  "message": "Invalid data format."  
}
```

POST */api/drones/:droneId/maintenance*

- **Purpose:** Add a new maintenance log for a drone.

Example Request:

```
{  
  "maintenanceDate": "2024-11-03T09:00:00Z",  
  "maintenanceType": "battery replacement",  
  "performedBy": "Technician John",  
  "notes": "Replaced with new high-capacity battery."  
}
```

Example Response:

```
{  
  "status": 201,  
  "message": "Maintenance record added successfully."  
}
```

Error Response:

```
{  
  "status": 404,  
  "message": "Drone not found."  
}
```

GET /api/alerts

- **Purpose:** Retrieve a list of all active and historical alerts for the logged-in user.

Example Request:

GET /api/alerts

Example Response:

```
[
  {
    "_id": "alert001",
    "fieldId": "field123",
    "alertType": "low moisture",
    "severity": "high",
    "alertMessage": "Critical moisture levels detected in North Field.",
    "dateIssued": "2024-11-02T10:30:00Z",
    "status": "unresolved"
  },
  {
    "_id": "alert002",
    "fieldId": "field456",
    "alertType": "pest presence",
    "severity": "medium",
    "alertMessage": "Signs of pest infestation in East Field.",
    "dateIssued": "2024-11-01T15:00:00Z",
    "status": "resolved"
  }
]
```

Error Response:

```
{
  "status": 404,
  "message": "No alerts found for the user."
}
```

GET /api/alerts/:alertId

- **Purpose:** Retrieve detailed information about a specific alert.

Example Request:

GET /api/alerts/alert001

Example Response:

```
{
  "_id": "alert001",
  "fieldId": "field123",
  "alertType": "low moisture",
  "severity": "high",
  "alertMessage": "Critical moisture levels detected in North Field.",
  "dateIssued": "2024-11-02T10:30:00Z",
  "recommendation": "Increase irrigation immediately.",
  "status": "unresolved",
  "actionTaken": null
}
```

Error Response:

```
{
  "status": 404,
  "message": "Alert not found"
}
```

POST /api/reports

- **Purpose:** Generate a custom report based on user-specified parameters.

Example Request:

```
{
  "userId": "user123",
  "fieldsIncluded": ["field123", "field456"],
  "dateRange": {
    "start": "2024-10-01T00:00:00Z",
    "end": "2024-11-01T00:00:00Z"
  },
  "metricsIncluded": ["moistureLevel", "plantHeight"]
}
```

Example Response:

```
{
  "status": 201,
  "message": "Report generated successfully.",
  "reportId": "report789"
}
```

Error Response:

```
{
  "status": 404,
  "message": "Failed to generated report",
}
```

```
}
```

GET /api/reports/:reportId

- **Purpose:** Retrieve a saved report by its ID.

Example Request:

```
GET /api/reports/report789
```

Example Response:

```
{
  "_id": "report789",
  "userId": "user123",
  "fieldsIncluded": ["field123", "field456"],
  "dateRange": {
    "start": "2024-10-01T00:00:00Z",
    "end": "2024-11-01T00:00:00Z"
  },
  "metricsIncluded": ["moistureLevel", "plantHeight"],
  "dataSummary": [
    {
      "fieldId": "field123",
      "averageMoistureLevel": 42,
      "averagePlantHeight": 115
    },
    {
      "fieldId": "field456",
      "averageMoistureLevel": 38,
      "averagePlantHeight": 120
    }
  ],
  "createdOn": "2024-11-02T14:00:00Z"
}
```

Error Response:

```
{
  "status": 404,
  "message": "Report not found."
}
```

GET /api/drones

- **Purpose:** Retrieve a list of all drones and their status for management purposes.

Example Request:

```
GET /api/drones
```

Example Response:

```
[
  {
    "_id": "drone001",
    "droneModel": "DJI Phantom 4",
    "serialNumber": "SN123456789",
    "status": "active",
    "lastMaintenanceDate": "2024-10-15T08:30:00Z"
  },
  {
    "_id": "drone002",
    "droneModel": "Parrot Anafi",
    "serialNumber": "SN987654321",
    "status": "maintenance",
    "lastMaintenanceDate": "2024-11-01T09:45:00Z"
  }
]
```

Error Response:

```
{
  "status": 404,
  "message": "No drones found."
}
```

DIAGRAMS:



