**Ovine App - API Design Document**

|  |  |  |  |
| --- | --- | --- | --- |
| Document Revision # | Action Taken, Notes | Modified Date | Editor |
| 0.1 | Creation | 03/02/2023 | Xiaolong Xia |
| 0.2 | Edited further,  incorporated feedback from Patrick Niland | 10/02/2023 | Xiaolong Xia |
|  |  |  |  |

# Overview

This document defines the data protocol interfaces of Ovine Farm to Farm Movement Application for front-end or external access parties, mainly including: login interface, interface for obtaining keeper's Herd ID list, image/document detection and upload interface, ovine movement creation interface and notification related interface etc.

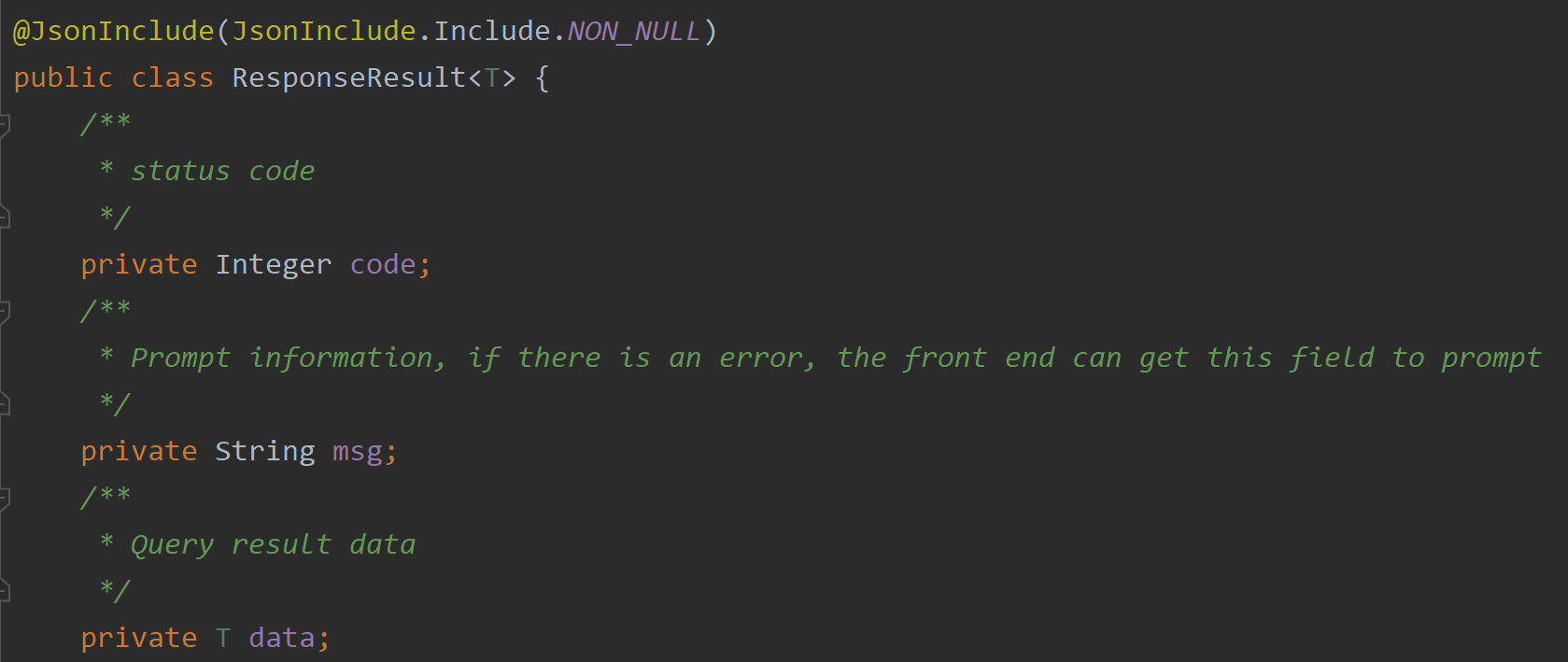
This document is suitable for developers, testers, project leaders or third-party collaborators. More importantly, this document requires continuous updating in order to enhance the readability, reliability and reference of this document.

References: Ovine Farm to Farm Movement Application Requirements

# URI structure

https://[**host**]:[**port**]/{**service name**}/{**resource url**}

# Custom Unified Response Body



# Http response status codes

The server should always return the right HTTP status code to the client.

* Standard HTTP status codes

200 – OK – Everything is working

201 – OK – New resource has been created

204 – OK – The resource was successfully deleted

304 – Not Modified – The client can use cached data

400 – Bad Request – The request was invalid or cannot be served. The exact error should be explained in the error payload. E.g. “The JSON is not valid“

401 – Unauthorized – The request requires a user authentication

403 – Forbidden – The server understood the request but is refusing it or the access is not allowed.

404 – Not found – There is no resource behind the URI.

422 – Unprocessable Entity – Should be used if the server cannot process the entity, e.g. if an image cannot be formatted or mandatory fields are missing in the payload.

500 – Internal Server Error – API developers should avoid this error. If an error occurs in the global catch blog, the stracktrace should be logged and not returned as in the response.

* Use error payloads

All exceptions should be mapped in an error payload. Here is an example how a JSON payload should look like.



# Security

Token Based Authentication:

Ideally, microservices should be stateless so the service instances can be scaled out easily and the client requests can be routed to multiple independent service providers. A token based authentication mechanism should be used instead of session based authentication.

And Cooperate with SpringSecurity to authenticate interface access, which means the Header in each request must carry token.

# API Documentation(Swagger)

* All the RESTful API must follow Swagger Specification for API documentation:

[http://swagger.io/specif ication](http://swagger.io/specif%20ication)

The ONAP-Components must provide swagger files for its RESTful API definitions, there are two possible approaches.

* The developers write the Swagger file, then use CI system to generate the stub codes for implementation.
* **The developers write the Swagger annotation in the source code, then use CI system to generate the Swagger file.**

The Swagger file should be placed under the base url of the service so the API definition could be discovered by clients or management tools at runtime. For example, if the base url of the service is /ovinecensus/, the URL of swagger file should be /ovinecensus/Swagger.json

# Interfaces

The API will be RESTful and will include the following endpoints.

## Login

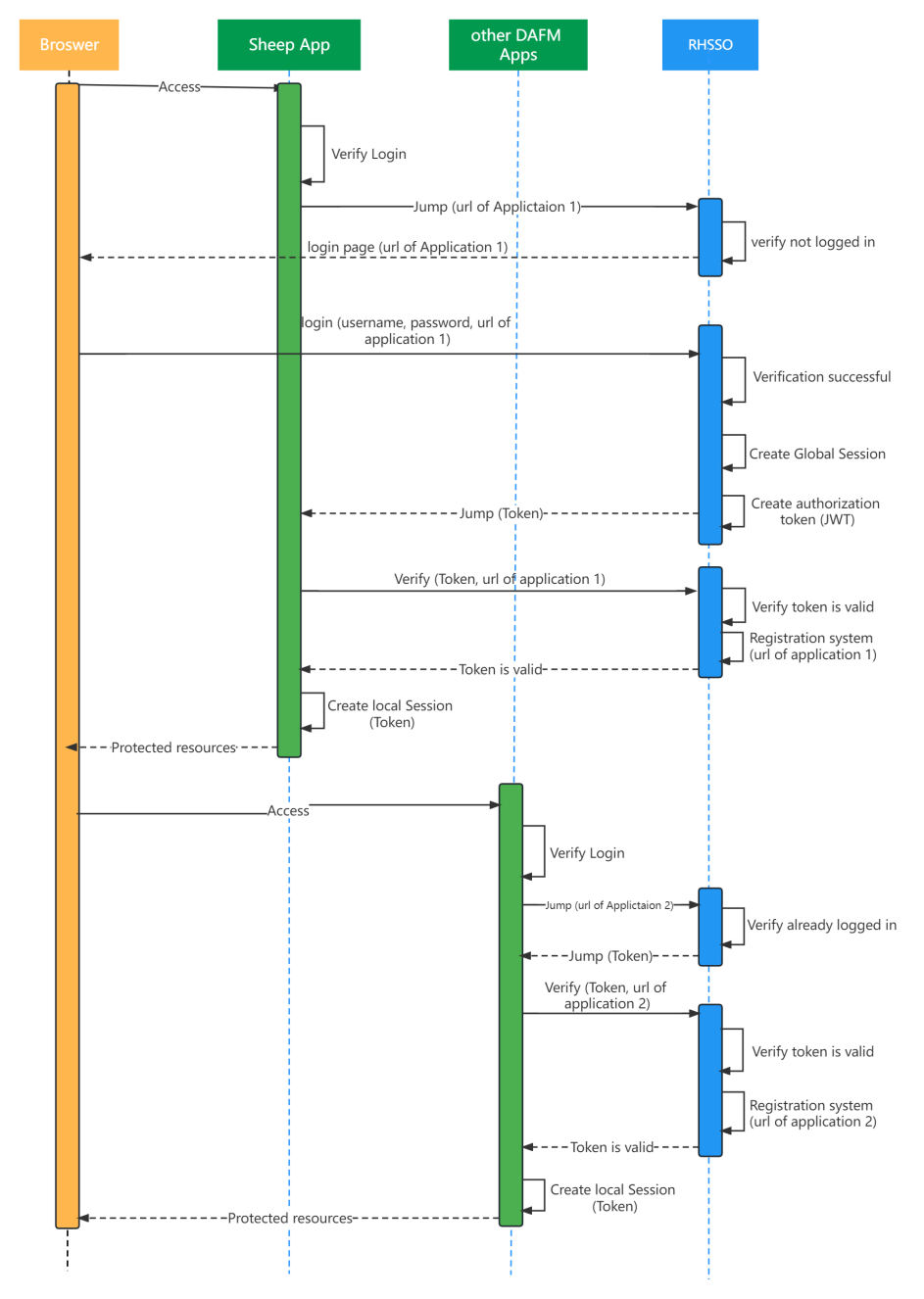


Figure 1. Login - RHSSO process

**Description:** Existing RHSSO login page will be used to access this app and return a valid access token which is required in every request header.

**{**

**Resource URL:** /login

**Type:** Post

**Content Type:** application/json

**Request Example:**

{

“username”: user@agriculture.gov.ie,

“password”: asas123dnjdsd.fdfsadfsdhuhu2341212121adfsdfasdfasdfadfu4412112122

}

**Request Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| username | String | No | The username of existing user in DB. |
| password | String | No | Encrypted password |

**Response Example:**

{

“code”: 200,

“msg”: “Login successfully”,

“data”:{

“token”:dfasdfasdfasdfxasasdfasdfasdfasdfas.asfasfasdfasdf131231.sfdasf3131231

}

}

**Response Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| token | String | No | Access token which is required in every request header. |

} -- redundant, can be deleted

## Users

### Get Herd ID list

**Description:** Query the CCS DB which keeps user data and fetch the **active** Herd ID list of the specified user.

**Resource URL:** /users/herd/ids/{id}

**Type:** Get

**Request Example:** /users/herd/ids/2001

**Request Parameters:** null

**Response Example:**

{

“code”: 200,

“msg”: “Get Herd ID list successfully”,

“data”:{[

{

“desHerdId”: “Q1350634”

},

{

“desHerdId”: “Y1404001”

}

]}

}

**Response Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| desHerdId | String | No | The active herd id of current logged in user |

**Possible errors:**

|  |  |  |
| --- | --- | --- |
| Error Code | Description | Error Message |
| 400 Bad Request | Required fields were invalid, not specified | “” |
| 401 Unauthorized | The access token is invalid or has been revoked | “” |
| 403 Forbidden | The server understood the request but is refusing it or the access is not allowed | “” |

## Documents

### Upload document image

**Description:** The dispatch document image needs to be uploaded when the purchaser or dealer creates a movement. In order to reduce the database space occupied by document, it is necessary to compress the document photo while ensuring the quality of the image, and performing blur detection on the compressed image to ensure that the image remains clear. Then the information in document photo is validated by ‘Validation Service’, for example, verify whether the Herd ID and Document Serial Number are correct. After successful validation, the image file needs to be converted into a pdf file and then saved into the database in order to adapt to the current document storage rules. So this API contains functions of image compression, image blur detection, document format conversion and database storage.

**Resource URL:** /movements/document/upload

**Type:** Post

**Content Type:** application/form-data

**Request Example:**

{

“file”: dispatch\_document.jpg

}

**Request Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| file | Multipartfile | No | Dispatch document photo to upload |

**Response Example:**

{

“code”: 200,

“msg”: “Upload successfully”,

“data”:{

“fileId”: 258153889,

“fileName”: “dispatch\_document.pdf”

}

}

**Response Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| fileId | Long | No | The primary key ID of the document saved in database |
| fileName | String | No | The file name of pdf document. |

**Possible errors:**

|  |  |  |
| --- | --- | --- |
| Error Code | Description | Error Message |
| 400 Bad Request | Required fields were invalid, not specified | “” |
| 401 Unauthorized | The access token is invalid or has been revoked | “” |
| 403 Forbidden | The server understood the request but is refusing it or the access is not allowed | “” |

## Movements

### Get Herd ID by Document Serial No.

**Description:** When creating a permanent movement, the user needs to manually enter the document serial number and query to obtain the corresponding (from) Herd ID based on this number.

**Resource URL:** /movements/herdid

**Type:** Post

**Content Type:** application/json

**Request Example:**

{

“serialNumber”: “41983-0449”

}

**Request Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| serialNumber | String | No | The serial number of dispatch document |

**Response Example:**

{

“code”: 200,

“msg”: “Get successfully”,

“data”:{

“herdId”: “U1761281”

}

}

**Response Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| herdId | String | No | The herd ID corresponding to the dispatch document serial number |

**Possible errors:**

|  |  |  |
| --- | --- | --- |
| Error Code | Description | Error Message |
| 400 Bad Request | Required fields were invalid, not specified | “” |
| 401 Unauthorized | The access token is invalid or has been revoked | “” |
| 403 Forbidden | The server understood the request but is refusing it or the access is not allowed | “” |

### Create a movement

**Description:** Permanent movement, temporary movement moving away from Owner farm and temporary movement moving back to Owner farm can be created through this interface, and the movement type is distinguished by the 'movementType' parameter. All types of movements are recorded in a same database table.

**Resource URL:** /movements

**Type:** Post

**Content Type:** application/json

**Request Example 1:**

{

“dispDocSerialNum”: “41983-0449”,

“movedFromId”: “Q1350634”,

“movedToId”: “Q1761292”,

“fileId”: 258153889,

“movementDate”: 20/10/2022,

“numAnimals”: 60,

“movementType”: “XXX”

}

**Request Example 2:**

{

“dispDocSerialNum”: “41983-0449”,

“movedFromId”: “Q1350634”,

“movedToId”: “Q1761292”,

“fileId”: 258153889,

“movementDate”: 20/10/2022,

“numAnimals”: 60,

“movementType”: “ZZZ”,

“deadAnimalNum”: 3,

“missingAnimalNum”: 2,

“stolenAnimalNum”: 0

}

**Request Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| dispDocSerialNum | String | No | The serial number of dispatch document |
| movedFromId | String | No | The Herd Id of movement from |
| movedToId | String | No | The Herd Id of movement to |
| fileId | long | No | The uploaded Image document Id in DB |
| movementDate | Date | No | Movement Date |
| numAnimals | int | No | Number of Sheep |
| movementType | String | No | The movement type.  The type of permanent movement is XXX;  The type of temporary movement moving away from Owner farm is YYY;  The type of temporary movement moving back to Owner farm is ZZZ |
| deadAnimalNum | Integer | Yes | The number of dead sheep.  When ‘movementType’ is equal to “ZZZ”, it cannot be null |
| missingAnimalNum | Integer | Yes | The number of missing sheep.  When ‘movementType’ is equal to “ZZZ”, it cannot be null |
| stolenAnimalNum | Integer | Yes | The number of stolen sheep.  When ‘movementType’ is equal to “ZZZ”, it cannot be null |

**Response Example:**

{

“code”: 200,

“msg”: “Create successfully”

}

**Possible errors:**

|  |  |  |
| --- | --- | --- |
| Error Code | Description | Error Message |
| 400 Bad Request | Required fields were invalid, not specified | “” |
| 401 Unauthorized | The access token is invalid or has been revoked | “” |
| 403 Forbidden | The server understood the request but is refusing it or the access is not allowed | “” |

### Get Temporary Movement Records

**Description:** Get the records of temporary movements moving away from owner farm when owner moves the sheep back to own holding.

**Resource URL:** /movements/records

**Type:** Post

**Content Type:** application/json

**Request Example:**

{

“movedFromId”: “Q1350634”

}

**Request Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Request Parameter | type | Nullable | Description |
| movedFromId | String | No | The Herd Id of temporary movement from |

**Response Example:**

{

“code”: 200,

“msg”: “Get successfully”,

“data”:{[

{

“dispDocSerialNum”: “41983-0449”,

“tempMovedToId”: “Q1761292”,

“numAnimals”: 60,

“movementDate”: 20/10/2022,

“fileId”: 258153889

},

{

“dispDocSerialNum”: “41983-1101”,

“tempMovedToId”: “Y1761232”,

“numAnimals”: 40,

“movementDate”: 05/09/2021,

“fileId”: 258153800

}

]}

}

**Response Parameters:**

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | type | Nullable | Description |
| dispDocSerialNum | String | No | The serial number of dispatch document |
| tempMovedToId | String | No | The Herd Id of temporary movement to |
| numAnimals | Integer | No | The number of sheep |
| movementDate | Date | No | The date of movement |
| fileId | long | No | The ID of the dispatch document uploaded when creating a temporary movement |

**Possible errors:**

|  |  |  |
| --- | --- | --- |
| Error Code | Description | Error Message |
| 400 Bad Request | Required fields were invalid, not specified | “” |
| 401 Unauthorized | The access token is invalid or has been revoked | “” |
| 403 Forbidden | The server understood the request but is refusing it or the access is not allowed | “” |

## Notifications

(To be continued...)