

AstroUMA Backend API Documentation

Kilo Code

June 8, 2025

Contents

1 Introduction

This document provides a detailed explanation of the AstroUMA backend API, focusing on the **reportZ** and **bolide** endpoints. It covers the functionality, logic, and middlewares used for each route.

2 Core Middlewares

Several middlewares are used across different routes to handle common tasks like authentication, data validation, and data transformation.

2.1 `validate-jwt.js`

This middleware is responsible for validating the JSON Web Token (JWT) sent in the **x-token** header of the request. It ensures that the user is authenticated before allowing access to protected routes.

2.2 `convertSexagesimalToDecimal.js`

This middleware provides functions to convert astronomical coordinates from sexagesimal format (degrees, minutes, seconds) to decimal degrees, which is necessary for calculations. It includes functions for both individual and bulk conversions.

2.3 `extractJWT.js`

This utility extracts the user ID from the JWT payload, allowing the application to identify the user making the request.

3 ReportZ Endpoints

The **reportZ** endpoints are primarily used for managing and retrieving detailed meteor reports (**Informe_Z**).

3.1 `GET /api/reportz/:id`

- **Controller Function:** `getReportZ` in `reportZController.js`
- **Description:** Retrieves a single, comprehensive **Informe_Z** by its ID. This is one of the most complex endpoints, gathering data from multiple tables.
- **Logic:**
 1. It first checks for a JWT to determine the user's role.
 2. Fetches the main report from the **Informe_Z** table.
 3. Retrieves associated data from other tables:
 - **Observatorio:** Gets details of the observatories involved.
 - **Puntos_ZWO:** Gets ZWO point data.
 - **Elementos_Orbitales:** Gets orbital elements.
 - **Trayectoria_medida** and **Trayectoria_por_regresion:** Gets trajectory data.
 - **Informe_Fotometria:** Checks for a related photometry report.
 4. **Coordinate Conversion:** Uses `convertCoordinates` and `individuaConvertSexagesimalToDecimal` to transform coordinates into a usable format.
 5. **Fuzzy Logic for Showers:**
 - **IMOShowers:** Calculates a membership index for established IMO meteor showers based on orbital parameters (eccentricity, semi-major axis, inclination) and the distance between the radiant and the trajectory (DMRT). It uses a weighted fuzzy logic model.
 - **IAUShowers:** Identifies potential associations with IAU meteor showers by calculating the angular distance between the meteor's radiant points (from ZWO data) and the shower's radiant. It calculates a probability based on this distance.

6. **Error Reports:** Based on the user's role (admin or regular user), it fetches relevant error reports (`Informe_Error`) associated with the main report.
 7. Finally, it assembles all the retrieved and processed data into a single JSON response.
- **Middlewares:** No specific middleware is applied at the route level, but the controller itself handles token extraction and role checking.

3.2 Other reportZ Routes

- **GET /api/reportz:** (`getAllReportZ`) - Retrieves all `Informe_Z` records. Requires JWT validation.
- **POST /api/reportz/advice:** (`saveReportAdvice`) - Saves an error report submitted by a user. Requires JWT validation.
- **DELETE /api/reportz/advice/:id:** (`deleteReportAdvice`) - Deletes an error report. Requires JWT validation.
- **GET /api/reportz/search:** (`getReportzWithCustomSearch`) - Performs a custom search for reports based on query parameters. Requires JWT validation.
- **POST /api/reportz/showerInfo/...:** (`getReportZListFromRain`) - Retrieves reports associated with a specific meteor shower, including moon phase and illumination data calculated using the `suncalc` library.

4 Bolide Endpoints

The `bolide` endpoints are used for searching, retrieving, and exporting data about meteors (`Meteoro` table) and their associated reports.

4.1 GET /api/bolide/search

- **Controller Function:** `getBolideWithCustomSearch` in `bolideController.js`
- **Description:** Provides a powerful and flexible search functionality for meteors based on a variety of criteria.
- **Logic:**
 1. It takes several query parameters: `heightFilter`, `latFilter`, `lonFilter`, `ratioFilter`, `date ranges`, and `reportType`.
 2. **Dynamic Query Building:** The core of this function is a `switch` statement that constructs a SQL query dynamically based on the `reportType` parameter. This allows filtering for meteors that have specific types of reports (`Informe_Z`, `Informe_Radiante`, `Informe_Fotometria`).
 3. It uses `LEFT JOIN` and `INNER JOIN` to link the `Meteoro` table with the different report tables.
 4. **Filtering:**
 - **Date Range:** Filters results based on a start and end date.
 - **Height:** Filters based on the initial height of the trajectory, extracted from the `Inicio_de_la_trayectoria` field in `Informe_Z`.
 - **Location:** After the initial database query, it performs an additional in-memory filtering step if location filters are provided. It uses the `isPointInRadius` middleware to check if the meteor's starting point is within a given radius of a specified latitude and longitude.
 5. **Pagination:** Uses `LIMIT` and `OFFSET` for pagination, fetching 50 results at a time.
 6. It returns a JSON object containing the filtered data and the total number of items.
- **Middlewares:**
 - `isPointInRadius`: Used internally by the controller for location-based filtering.
 - `individuaConvertSexagesimalToDecimal`: Used to convert coordinates for the radius check.

4.2 GET /api/bolide/search/csv

- **Controller Function:** `getBolideWithCustomSearchCSV` in `bolideController.js`
- **Description:** This endpoint has the same filtering logic as `/api/bolide/search`, but instead of returning JSON, it generates and streams an Excel (`.xlsx`) file.
- **Logic:**
 1. It first fetches the filtered list of meteors using the same dynamic query logic as the search endpoint.
 2. It then uses the `exceljs` library to create a workbook with multiple sheets.
 3. It fetches the detailed data for each type of report (`Informe_Z`, `Informe_Radiante`, `Informe_Fotometria`) associated with the filtered meteors.
 4. It populates separate worksheets in the Excel file for each data type.
 5. Finally, it sets the appropriate HTTP headers for a file download and writes the Excel workbook to the response stream.
- **Middlewares:** Same as `/api/bolide/search`.

4.3 Other bolide Routes

- **GET /api/bolide:** (`getAllBolide`) - Retrieves all records from `Informe_Z`. Requires JWT. Note: The function name is misleading as it queries `Informe_Z`, not `Meteoro`.
- **GET /api/bolide/:id:** (`getBolideById`) - Retrieves a single meteor by its ID.
- **GET /api/bolide/search/reports:** (`getReportData`) - Fetches the data for different report types based on a list of IDs provided as query parameters.