EKO Q: How to upload measurement data

This document provides a step-by-step instructions on how to import irradiance measurement data to EKO Q application.

See more support information at https://eko-support.eko-instruments.com/knowledge-base/eko-q

Table of Contents

How to upload data?

Once the sensor has been created, the final step is to upload the corresponding measurement data for analysis. To do so, the user must click either **"Analyze Sensor"** for the specific sensor or select the sensor name from the dropdown list of the created site in the **"Sites"** tab on the right.

Following this action, the user will be redirected to the data upload page, as illustrated in Figure 1 below.

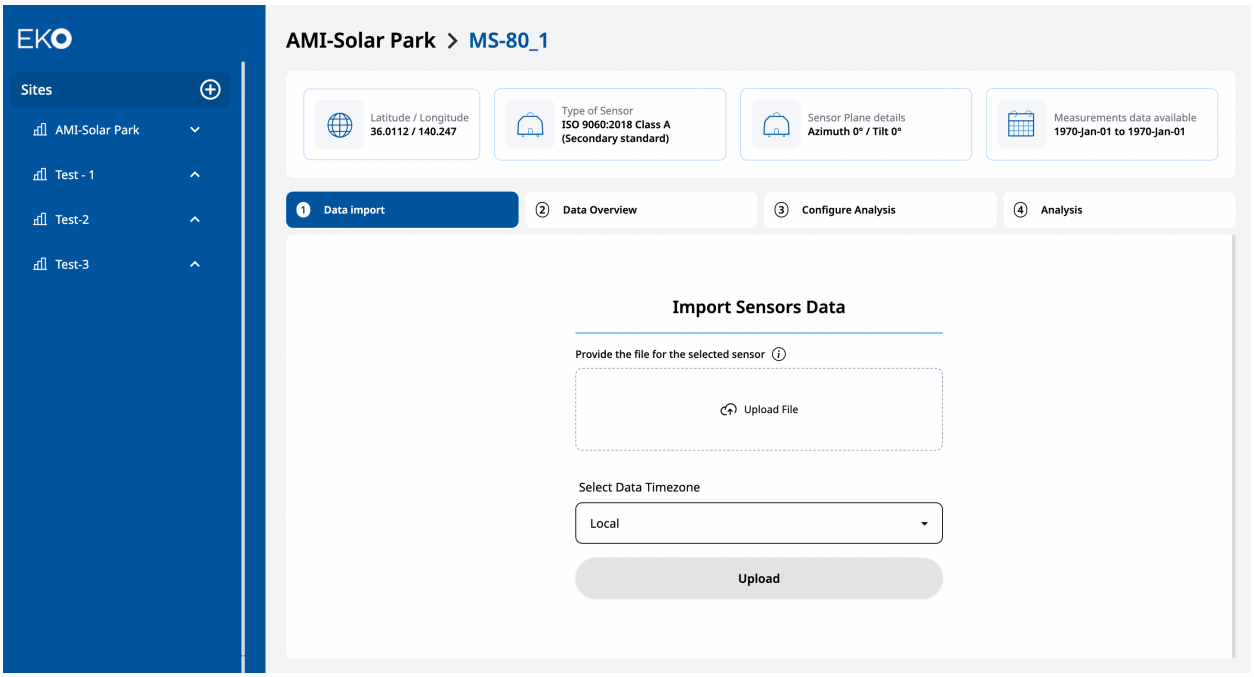


Figure 1: 'Import Sensors Data' page for the selected sensor.

On the data upload page, the user must follow these steps to upload and prepare the sensor data for analysis:

- **Upload File**: Click on **"Upload File"** to select the data file containing the sensor measurements.
- **Select Data Time Zone**: Choose the appropriate time zone for the uploaded data, either **UTC** or **Local**.
- **Click Upload**: Once the file and time zone are selected, click **"Upload"** to proceed with the data import.

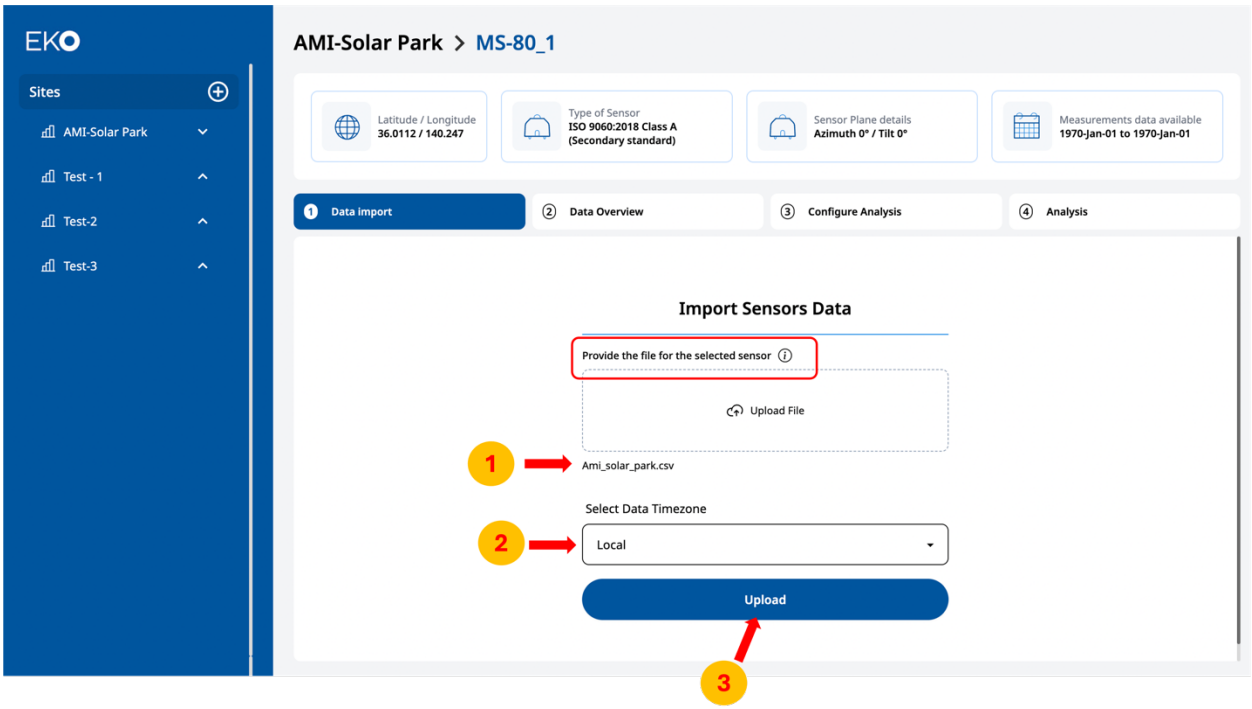


Figure 2: Data uploading process for the selected sensor.

As illustrated in Figure 2 above:

- **Step 1** shows the name of the uploaded file.
- **Step 2** indicates the selected time zone.
- **Step 3** highlights the **"Upload"** button, which is now active and ready to be clicked to finalize the process.

Based on the **format of the time series** (More information on the time series) and the **datetime index** (More information on the datetime index), the user can either proceed directly to the **"Data Overview"** section or perform an intermediate step to assist the tool in reformatting and interpreting the uploaded data. This intermediate step, known as **data parsing**, helps the system correctly format the uploaded file, enabling the app to perform analysis provide insights. If necessary, the tool will prompt the user to define specific parameters, such as date-time format adjustments, column mappings, or missing value handling.

To avoid manual data parsing using the built-in Data Parsing Tool (detailed in the next section), the uploaded file must adhere to one of the accepted timestamp formats. The user can access these formats by clicking on the tool tip labelled "Provide the file for the selected sensor," as marked in Figure 2 above.

The following Figure 3 presents the accepted timestamp formats along with an example.

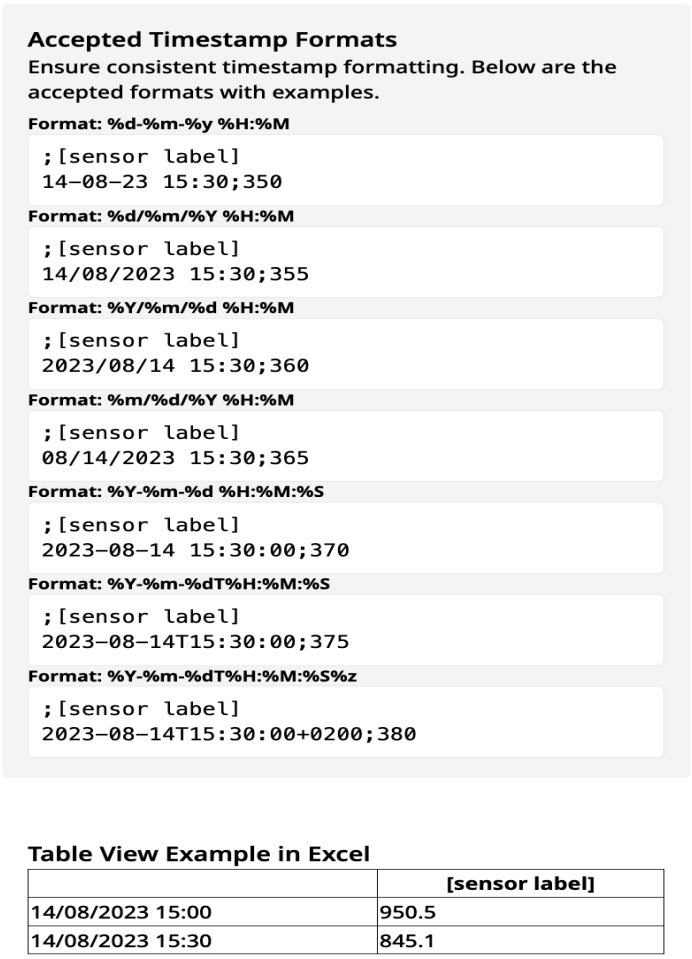


Figure 3: Accepted timestamp formats for the uploaded csv containing sensor data.

What, why and when Data parsing?

In simple terms, data parsing is the process by which the tool reads and interprets the uploaded information. For the tool to correctly process the data, it must “understand the language” of the information provided. Here, the information refers to the uploaded sensor data, and the language refers to its format and structure.

As mentioned earlier, the user can bypass the data-parsing step if the tool automatically detects the correct datetime index and time series format. However, if the tool encounters inconsistencies or formatting issues, the user will be redirected to the data parsing section after clicking “Upload” (as shown in Figure 3 above). In this case, an error message or prompt will guide the user through the necessary steps to correct the format and ensure successful data processing.

Navigating through the built-in Data parsing tool

To simplify the instructions and improve clarity, the data parsing process can be broken down into three main steps:

- **Validating the Sensor Label** : After uploading the data, the tool first checks for the Sensor Label; the same label name assigned in Figure 3 above, in the topmost row of the uploaded csv file, specifically for columns excluding the first one, which is automatically recognized as the datetime index. This requirement highlights the importance of keeping the Sensor Label identical to the header name of the desired column of data to be analyzed, as mentioned in the previous steps. If the Sensor Label is missing, an error will be raised, preventing further processing. This validation is crucial because measurement files often contain additional metadata, such as time zones, coordinates, data conventions, and other non-measurement-related details. Ensuring that the Sensor Label is correctly positioned allows the tool to accurately identify and analyze the relevant data.
- **Inspecting the Datetime Index**: By default, the tool assumes that the first column represents the datetime index. It will inspect the format and attempt to read it. If the format is incorrect or unrecognized, the system will generate error prompts to guide the user in correcting it.
- **Identifying Sensor Measurement Data**: Excluding the first column (datetime index), if the file contains only one additional column, the tool will automatically assume that it represents the sensor measurement (e.g., irradiance data). However, if multiple columns are present, an error prompt will instruct the user to specify which column contains the relevant data for analysis.

Each of these steps is detailed further below with reference to the example dataset used previously.

The following image illustrates the page the user will encounter after clicking the “Upload” button (as shown in Figure 4) if the uploaded csv file does not meet the requirements to bypass the built-in Data Parsing Tool.

The specific requirements for bypassing the parsing tool will be explained progressively throughout this section.

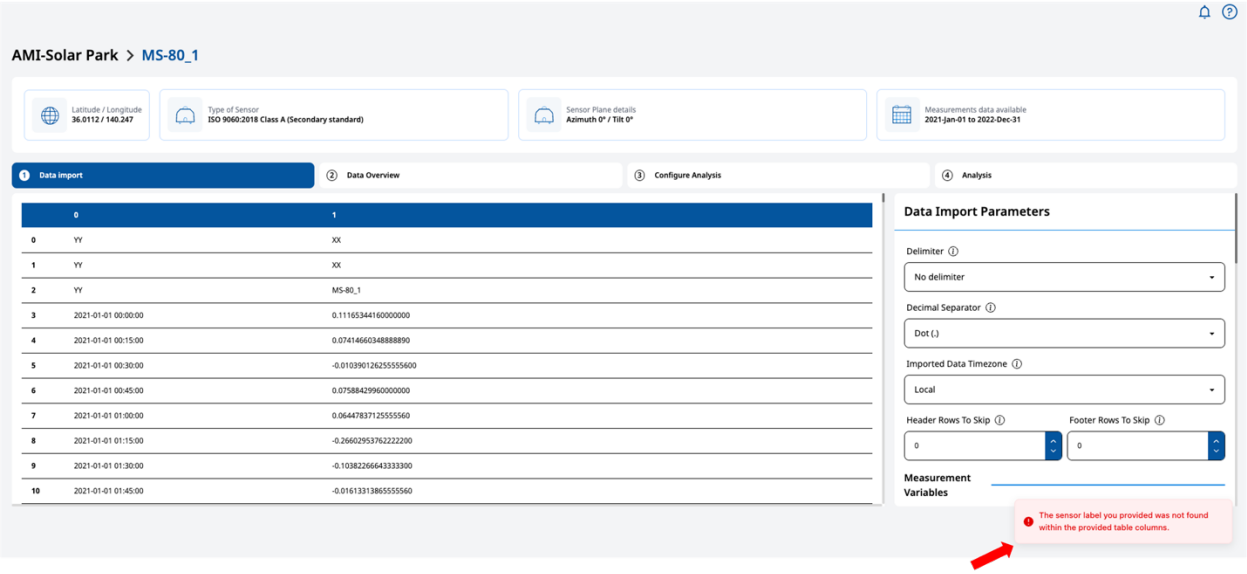


Figure 4: Error in locating the 'Sensor label' for the available column to be analysed.

As seen in Figure 4 above, an error message is immediately displayed, as indicated by the red arrow. The message states:

“The sensor label you provided was not found within the provided table columns.”

This error occurs because the tool’s first validation step is checking for the ‘Sensor label’ that was originally assigned when creating the sensor. In the figure above, the uploaded file does not contain a matching sensor label, leading to this error.

The missing or incorrect sensor label is highlighted and marked in the following Figure 5 to illustrate why this issue occurred.

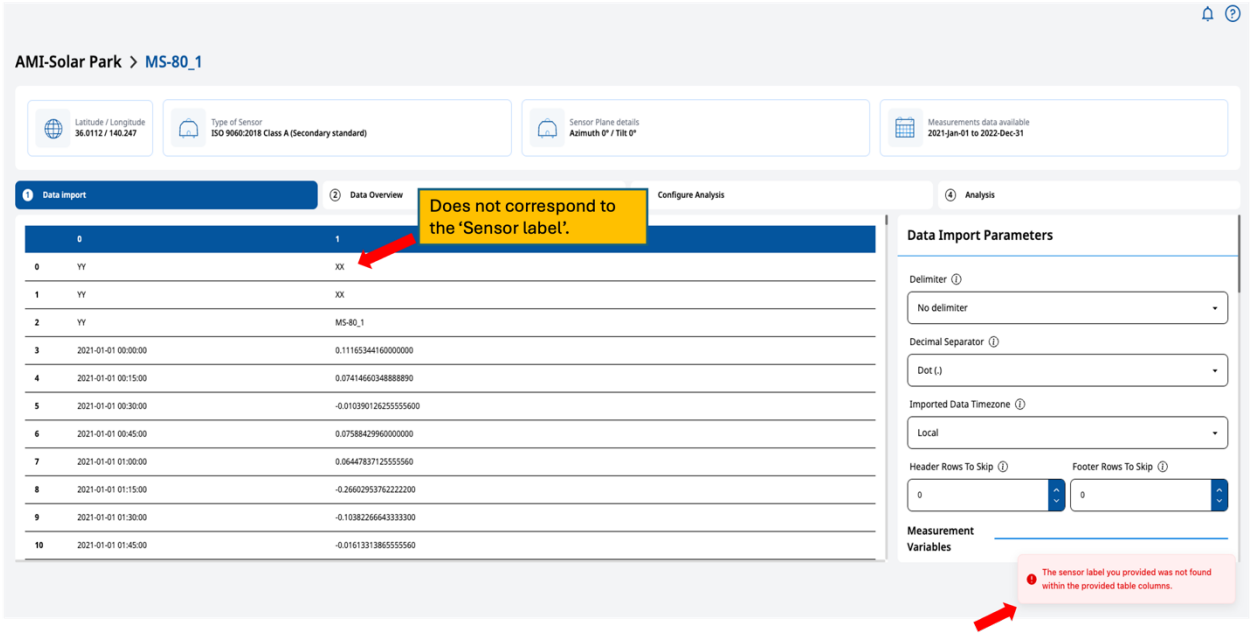


Figure 5: Cause indicated for the error regarding 'Sensor label'.

Although this error can be resolved using the in-built Parsing Tool, it is important to first understand other potential errors and their root causes before proceeding. Assuming the sensor label is correctly placed in the top row of the first column (excluding the datetime index), the tool successfully validates the sensor label as the header and proceeds to the second automatic check.

At this stage, the tool automatically inspects the datetime index format. If the datetime format does not match any of the accepted timestamp formats, an error will be raised, requiring further user intervention. The following Figure 6 illustrates this scenario, where the Sensor Label validation check is passed, allowing the tool to proceed with analyzing the datetime index.

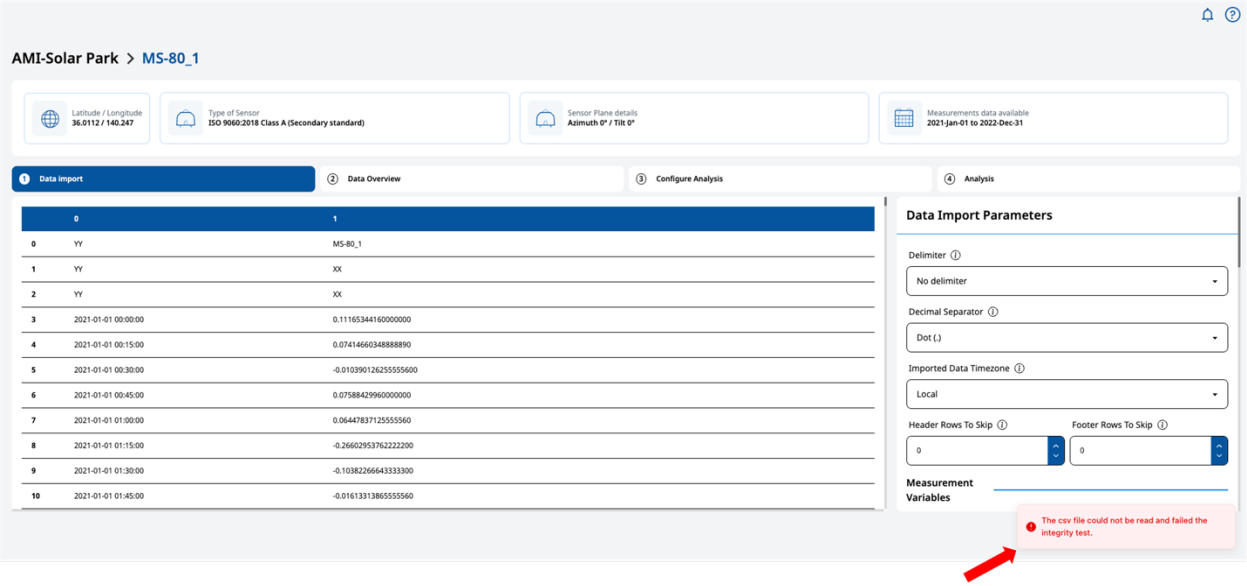


Figure 6: Error in the formatting and placement of the Datetime index.

In the scenario presented in the Figure 6 above, sensor label is correctly placed in the top row of the first column (excluding the datetime index), the tool successfully validates it and proceeds to inspect the datetime index format.

However, as shown in Figure 6 above, an error message appears, marked by the red arrow, stating:

"The csv file could not be read and failed the integrity test."

This error occurs because, following the first check, the tool assumes that the datetime index should begin from the second row of the dataset. It then attempts to validate whether the datetime index follows one of the accepted timestamp formats (as outlined in Figure 6 above). In this case, the datetime format does not meet the expected criteria, leading to the failure of the integrity test. The following Figure 7 highlights the issue in the dataset, illustrating why the datetime index validation check failed.

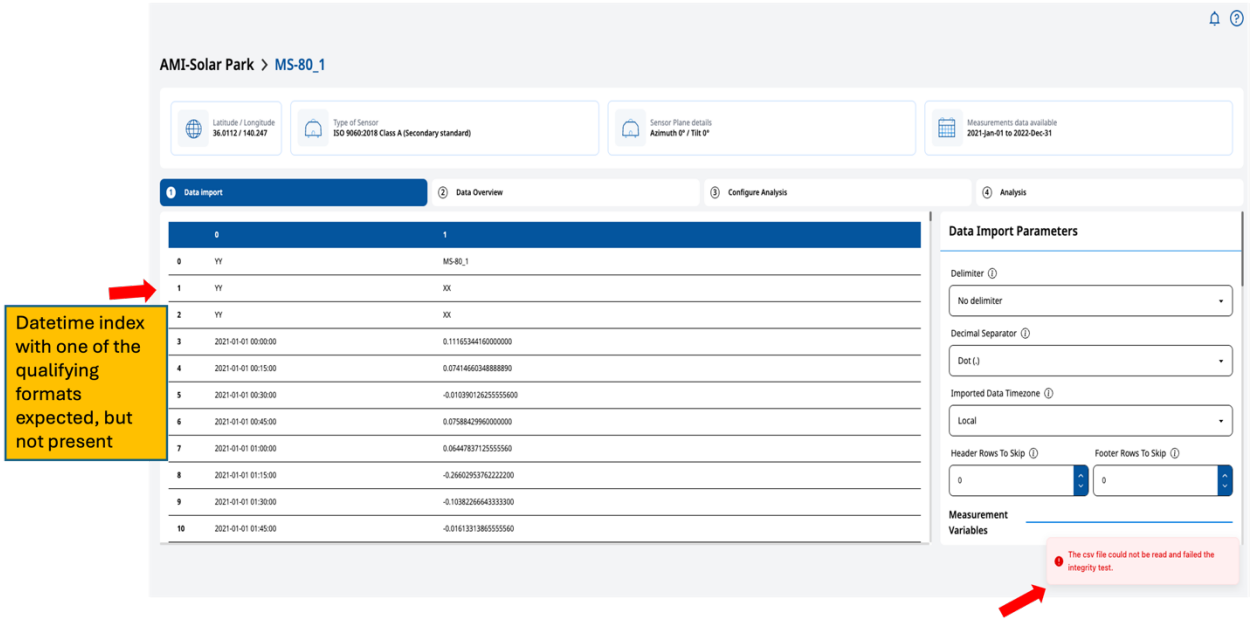


Figure 7: Cause indicated regarding the error in Datetime index placement and formatting.

As shown in Figure 7 above, the tool now checks whether the Datetime Index follows one of the predefined accepted timestamp formats. However, as indicated in the Figure 7, the row identified as the Datetime Index does not contain a valid datetime value and does not match any of the available presets. As a result, an error is raised, prompting the user to correct the issue before proceeding.

Now, consider a scenario shown in the Figure 8 below, where all previous sources of error are bypassed: the first row of the first column correctly contains the sensor label, and the first row of the Datetime index column contains a valid Datetime value that matches one of the accepted formats. In this case, the tool will automatically recognize the datetime format, and no additional steps will be required from the Parsing Tool for datetime validation.

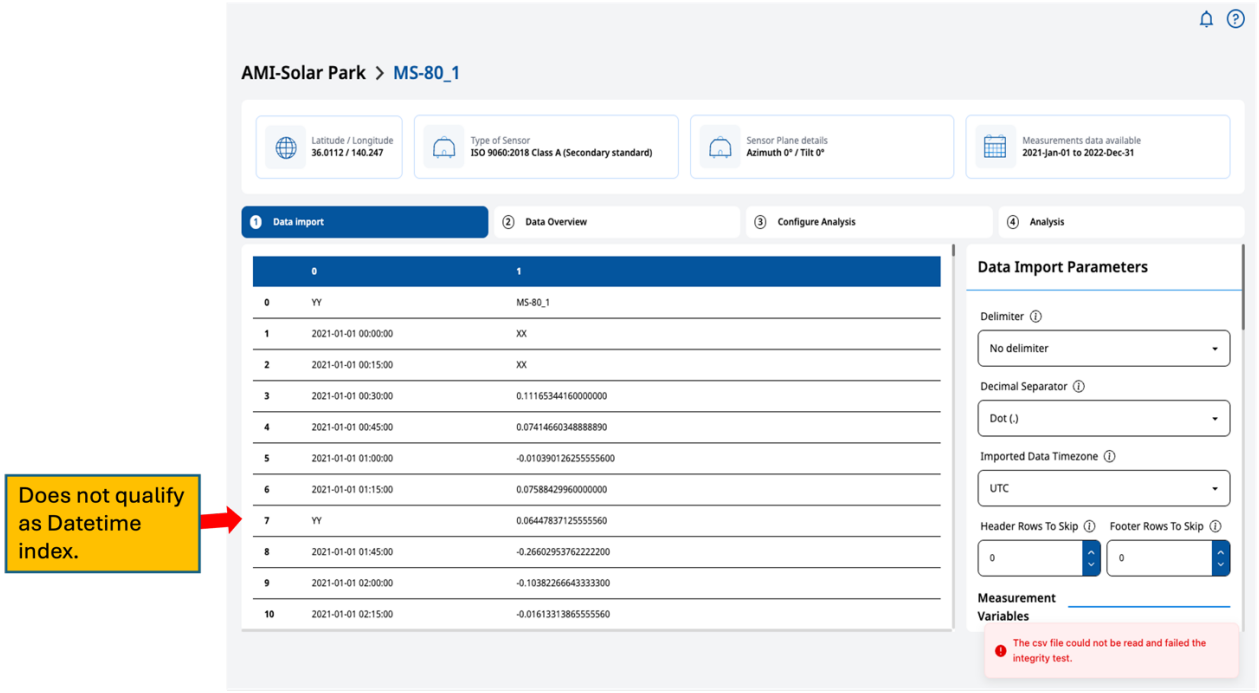


Figure 8: Mistake in the Datetime index detected causing the failure in the integrity test of the uploaded csv file.

Though Figure 8 satisfies the previously mentioned conditions, a similar error can still be observed. The cause is indicated in the same figure; while the tool has successfully validated the sensor label and the datetime index, it now inspects the entire datetime index column. If any row within this column does not conform to an accepted datetime format, an error is raised, prompting the user to correct the issue before proceeding.

Once the sensor label and datetime index are verified, the tool moves on to locating the irradiance column to be analysed. By default, if there is only one additional column (besides the datetime index), the tool assumes that this column contains the irradiance measurements. However, this assumption does not always hold, as csv files often include multiple columns with various types of data. If the file contains multiple additional columns, the tool cannot automatically determine the correct column of interest and raises an error, as shown in Figure 9 below.

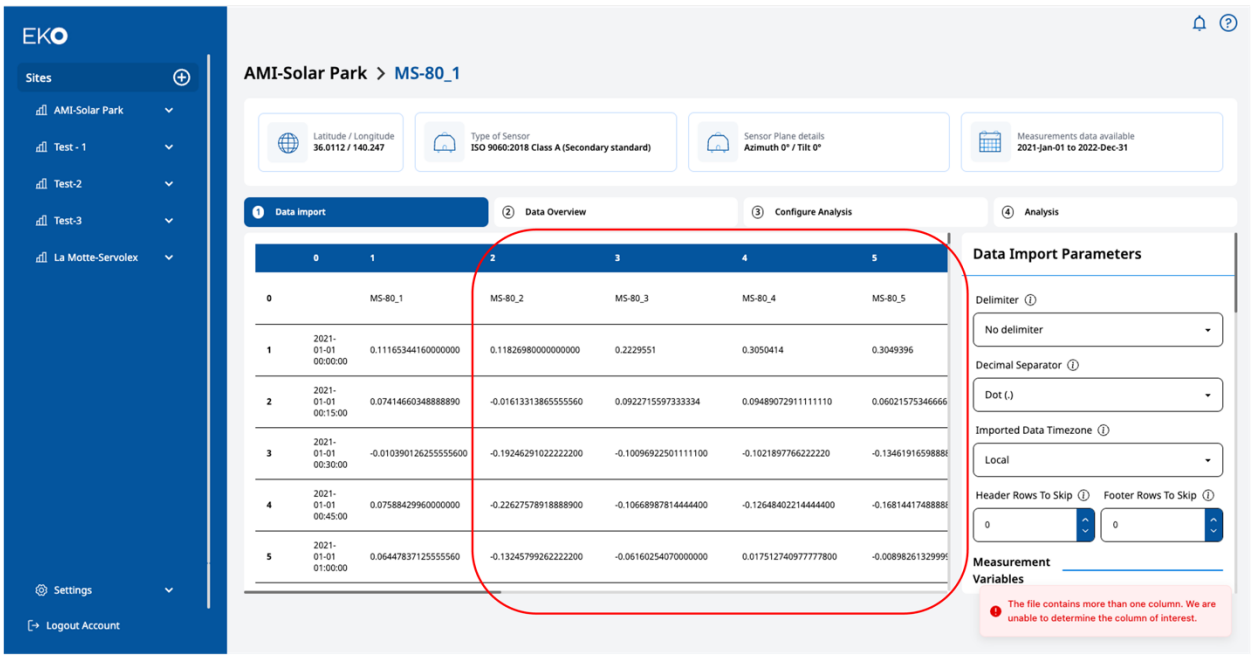


Figure 9: Error prompting that the file contains more than one column, thus disabling the tool to automatically choose the column of interest.

Having successfully verified the sensor label and datetime index, the tool now inspects the data column to perform the analysis. Since there is more than one column in the uploaded file (as marked in Figure 9 above), the tool cannot automatically determine which column contains the relevant measurement data. As a result, a specific error is raised, stating:

"The file contains more than one column. We are unable to determine the column of interest."

This issue can be resolved using the in-built Parsing Tool, without requiring the user to modify the csv file before uploading. The tool will guide the user in selecting the appropriate column of interest, ensuring that the correct data is analysed.

All the above causes of errors can be completely bypassed, and the user will not be directed to the Parsing Tool, if the uploaded file follows the correct structure, as described in the next section.

Ami_solar_park

YY	MS-80_1
2021-01-01 00:00:00	0.11165344160000000
2021-01-01 00:15:00	0.07414660348888890
2021-01-01 00:30:00	-0.010390126255555600
2021-01-01 00:45:00	0.07588429960000000
2021-01-01 01:00:00	0.06447837125555560
2021-01-01 01:15:00	-0.26602953762222200
2021-01-01 01:30:00	-0.10382266643333300
2021-01-01 01:45:00	-0.01613313865555560

Figure 10: File format to bypass the in-built Parsing tool.

The Figure 10 above meets the basic criteria required for seamless data processing. Specifically:

- The first row of the first column contains the Sensor Label.
- The Datetime Index follows one of the accepted formats and starts from the second row, with the first row correctly identified as the header.
- Only one additional column is present, which the tool automatically assumes to be the data column for inspection in the next steps.

Since all conditions are satisfied, no errors will be raised, and the uploaded file will be accepted. As a result, the user receives the following confirmation shown in the Figure 11 below, allowing them to proceed with the next step of data analysis.

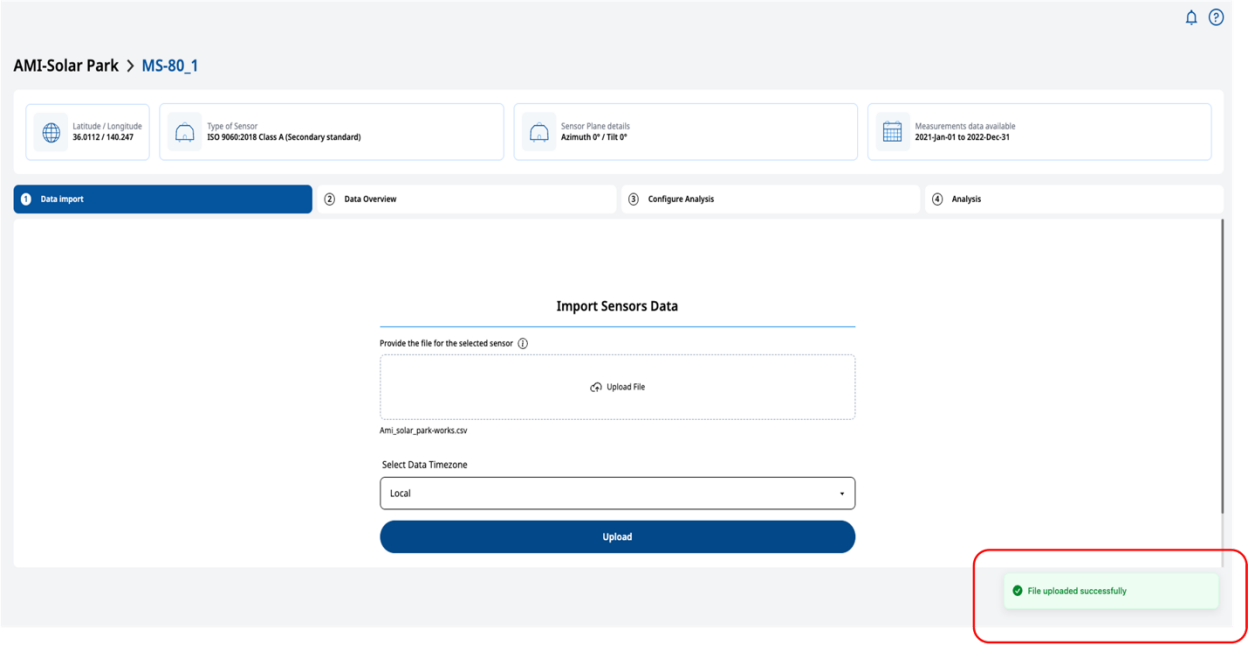


Figure 11: Successful file upload confirmation, highlighted in the red rectangle, indicating that the uploaded file has met all required criteria.

Therefore, based on the preliminary checks outlined above, the user can either adjust the file format before uploading or efficiently utilize the Parsing Tool with a better understanding of the potential errors. By leveraging the functionalities offered in the Parsing Tool, the user can correct formatting issues directly within the interface, ensuring that the uploaded data meets the required structure.

Assuming the user uploads a file that does not pass the automatic quality checks, the system will prompt them with errors. The following Figure 12 illustrates such a case, displaying the content of a non-compliant file that requires adjustments before proceeding.

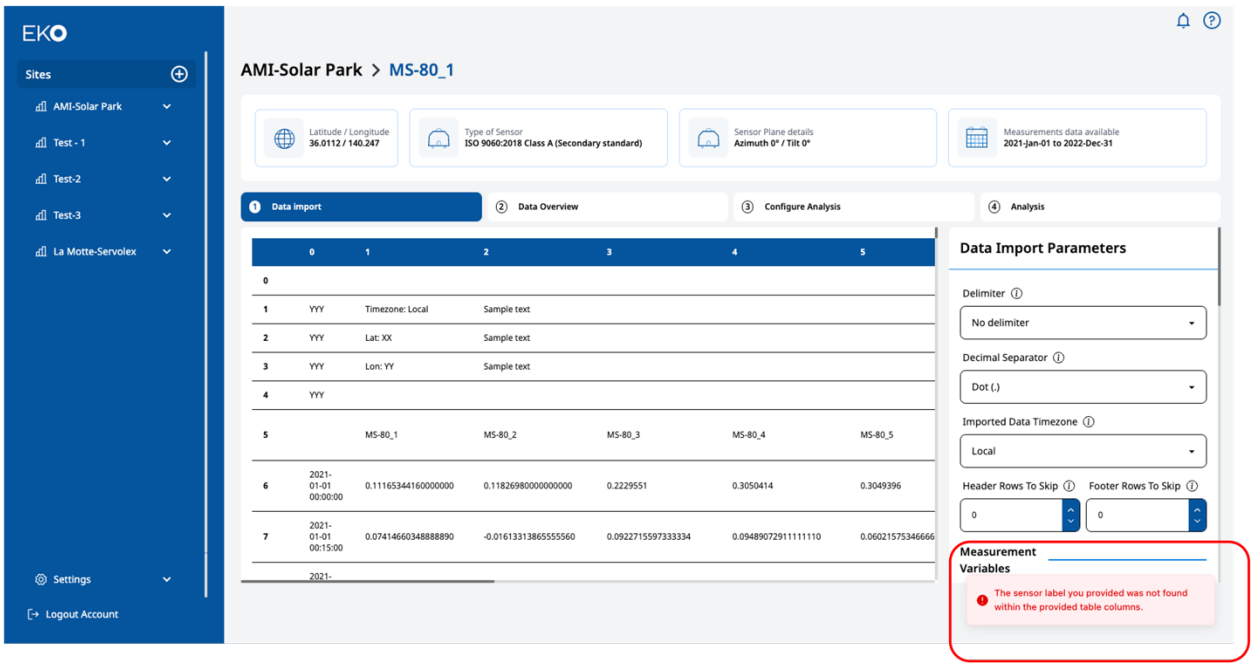


Figure 12: Test file showing the error related to 'Sensor label' location.

With a clear understanding of the sequence of automatic checks performed by the tool and the corresponding errors raised, the first error displayed in Figure 12 is expected, given the structure of the uploaded file. The system identifies formatting issues regarding the location of ‘sensor label’ and prompts the user to make necessary corrections before proceeding.

To resolve these errors, the user can utilize the Parsing Tool by providing the following inputs, which help restructure the file for ‘Sensor label’ location. The next section details these inputs and their role in correcting the format.

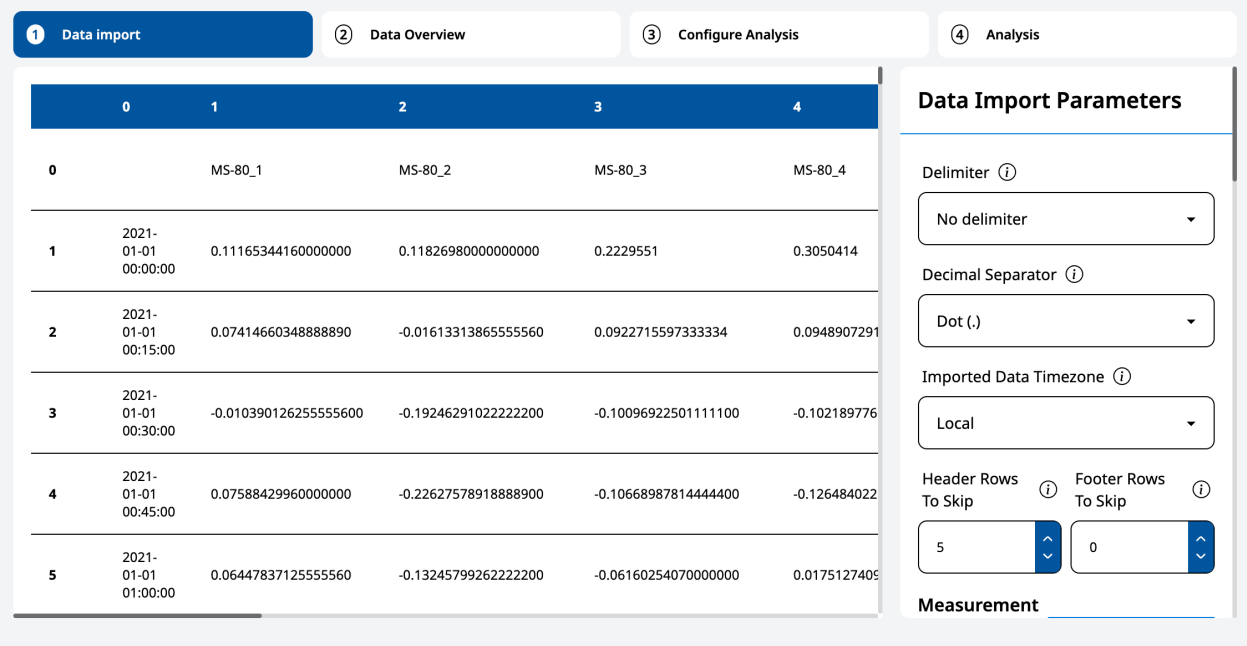


Figure 13: Input parameters to address the first check where 'Sensor label' position is being inspected.

When the uploaded file does not meet the expected format to locate the ‘Sensor Label’, the Parsing Tool allows users to configure specific parameters to ensure proper data interpretation. As observed in Figure 13, the Sensor Label is present, but it is not in the topmost header row. To correctly position the sensor label at the top, any irrelevant information in additional rows must be skipped.

The following available parameters can be adjusted to achieve this:

1. Delimiter: The delimiter defines how values are separated in the uploaded csv file. The available options are:

- **No delimiter** (default)
- **Comma (,)**
- **Semicolon (;)**
- **Tab**
- **Space**
- **Pipe (|)**

Selecting the correct delimiter ensures that the file is correctly structured, preventing misalignment of data columns. In the example chosen, the “No delimiter” option works correctly. However, for demonstration purposes, selecting an incorrect delimiter, such as “Semicolon (;)”, could result in incorrect formatting, which the user can immediately visualize.

As shown in Figure 14 below, selecting the wrong delimiter causes the data to be misaligned, allowing the user to experiment with different options until the columns are properly structured.

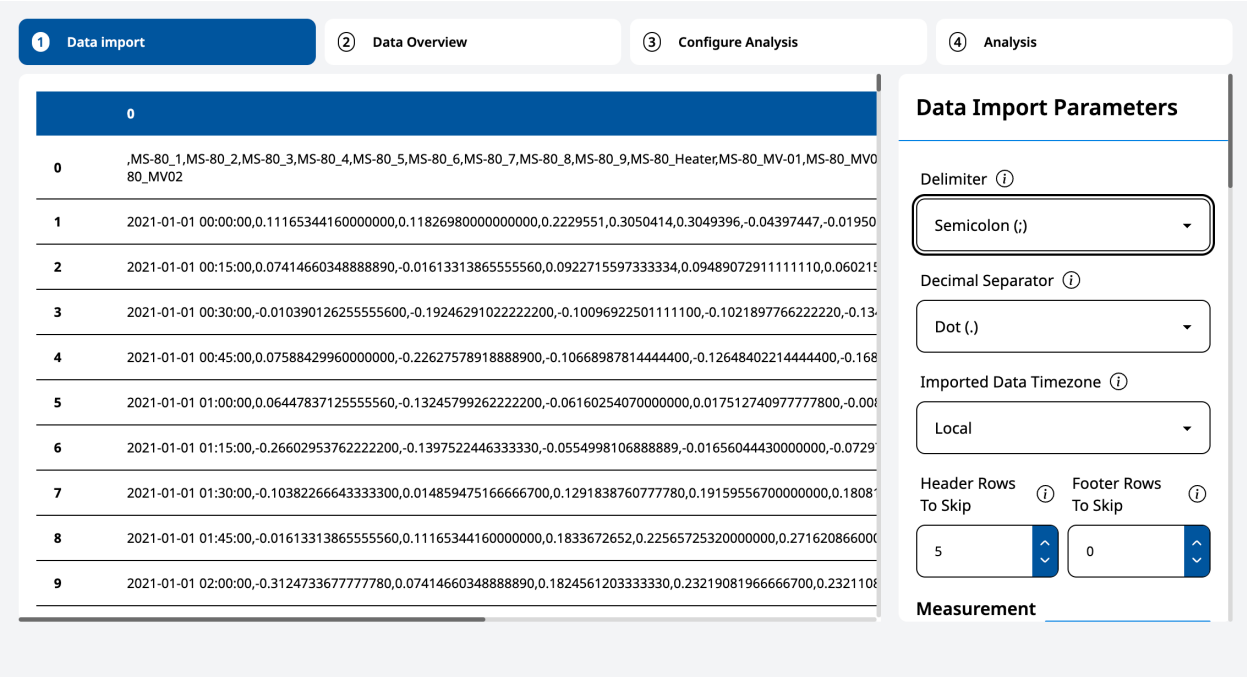


Figure 14: Illustration of misalignment of the data upon choosing the wrong delimiter.

2. Decimal Separator: The decimal separator determines how numerical values are formatted in the dataset. The available options are:

- **Dot (.) (default)**
- **Comma (,)**

Choosing the correct decimal separator ensures that numerical data is interpreted correctly, especially in international datasets where different conventions are used. For this example, as it can be seen from the columns from the Figure 14 above, Dot (.) convention is used and selected accordingly.

3. Imported Data Timezone: This setting allows the user to define whether the uploaded dataset follows a **'UTC'** or **'Local'** timezone. Selecting the correct timezone ensures that timestamps in the dataset are accurately interpreted based on the intended reference time. For this example, the **'Local'** timezone has been selected.

4. Header Rows to Skip : This option lets the user specify the number of header rows to be ignored before processing the dataset. This is useful for datasets that contain

metadata or descriptions above the actual data. In this example, five rows needed to be skipped, so the value was set to five, as seen in Figure 14 above.

5. Footer Rows to Skip: If the dataset includes extra rows at the bottom (e.g., summary statistics or notes), this option allows the user to specify how many rows should be ignored from the bottom of the file. In this example, no footer rows are present in the dataset, so the value remains zero, as shown in Figure 14 above.

Upon correcting the first cause of error, the user can click the “Save” button to proceed. However, upon doing so, the second error is immediately prompted, corresponding to the second automatic check performed by the tool pertaining the Datetime index. As seen in Figure 15 below, this new error arises because the system now moves to the next validation step, identifying any remaining inconsistencies that must be addressed before proceeding with data processing.

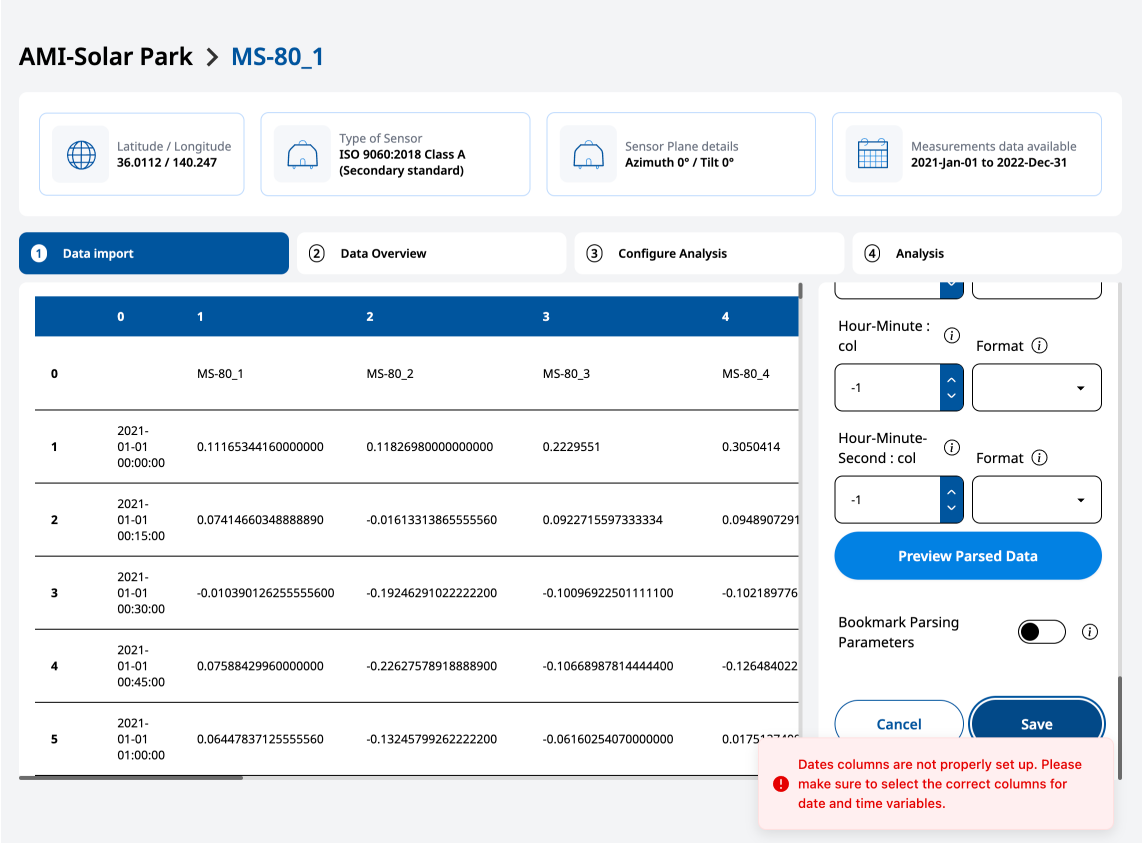
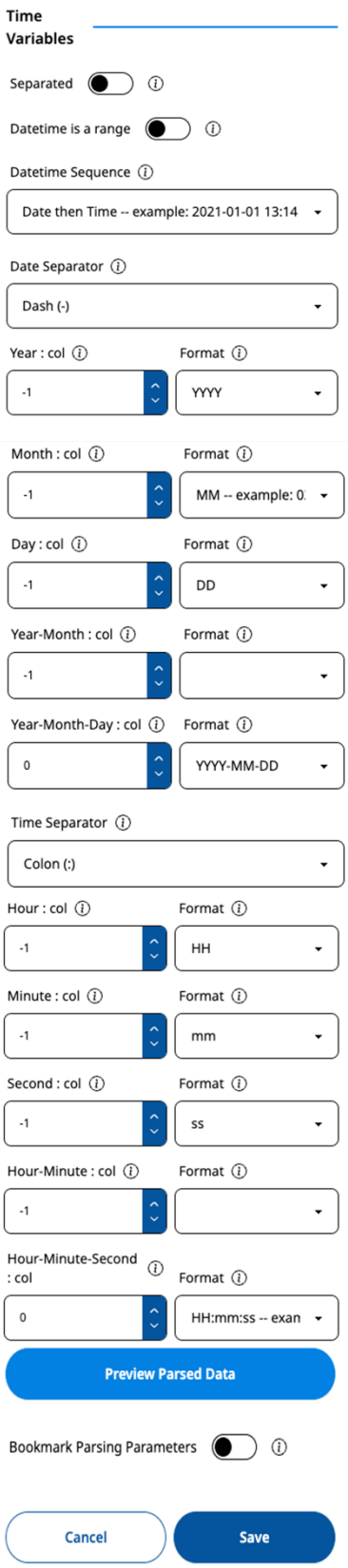


Figure 15: Test file showing the error related to the formatting of the Datetime index.

This error, as seen in Figure 15 above, arises because the datetime index is not configured. Since the uploaded file did not automatically qualify for the checks, each step must now be manually configured using the Parsing Tool.

At this stage, the user must define the format of the datetime index so that the tool can correctly interpret the timestamps in the dataset. This is done by selecting the appropriate format from the available input options, as shown below. The chosen input for the selected example is also highlighted in Figure 16 below to illustrate the correct selection.



1. Separated Toggle

• **Off** : When datetime values are stored in a single column (e.g., 2025-01-01 00:00:00).

• **On:** When datetime values are stored in multiple columns (e.g., separate columns for Year, Month, Day, Hour, Minutes, Seconds).

• For this example, the toggle is set to Off, indicating a single column datetime format.

2. Datetime is a Range Toggle

• **Off:** When each row contains a single timestamp (e.g., 2023-09-30 00:00:00).

• **On:** When each row contains a time range (e.g., 2023-09-30 00:00:00.0/2023-09-30 01:00:00.0).

• For this example, the toggle is set to Off, indicating single timestamp values in each row.

3. Date and Time Format Configuration

• **Datetime Sequence**: Defines whether the dataset follows date-then-time (YYYY-MM-DD HH:MM:SS) or time-then-date (HH:MM:SS YYYY-MM-DD). For this example, date-then-time is chosen.

• **Date Separator:** Defines the separator used between date components (/, -, .). For this example, a dash (-) is used (YYYY-MM-DD).

• **Year, Month, Day Columns:** Since the Separated toggle is Off, the individual Year: col, Month: col, Day: col, and Year-Month: col values are set to -1, meaning these columns do not exist separately.

• **Year-Month-Day Column:** The tool expects the datetime format to be structured as a single column, so the Year-Month-Day column is set to column 0 (YYYY-MM-DD).

4. Time Format Configuration

• **Time Separator:** Defines the separator between hour, minute, and second (:, ., -,). For this example, a colon (:) is used (HH:MM:SS).

• **Hour, Minute, Second Columns** : Since the Separated toggle is Off, individual columns for hours, minutes, and seconds are set to -1.

• **Hour-Minute-Second Column:** The datetime format follows HH:MM:SS, so this is set to column 0 (HH:mm:ss).

Figure 16: Parsing tool inputs to configure Date-time variables.

Once the data structure appears correct, the user can try uploading the data by clicking the **"Save"** button shown in Figure 16 above. After uploading, a confirmation window will appear if the sensor data has been successfully added. If any issues remain, the Parsing Tool will prompt additional error messages, guiding the user to make the necessary corrections before finalizing the upload.

In this case, upon clicking "Save", the tool prompts an error, indicating the final automatic check has failed.

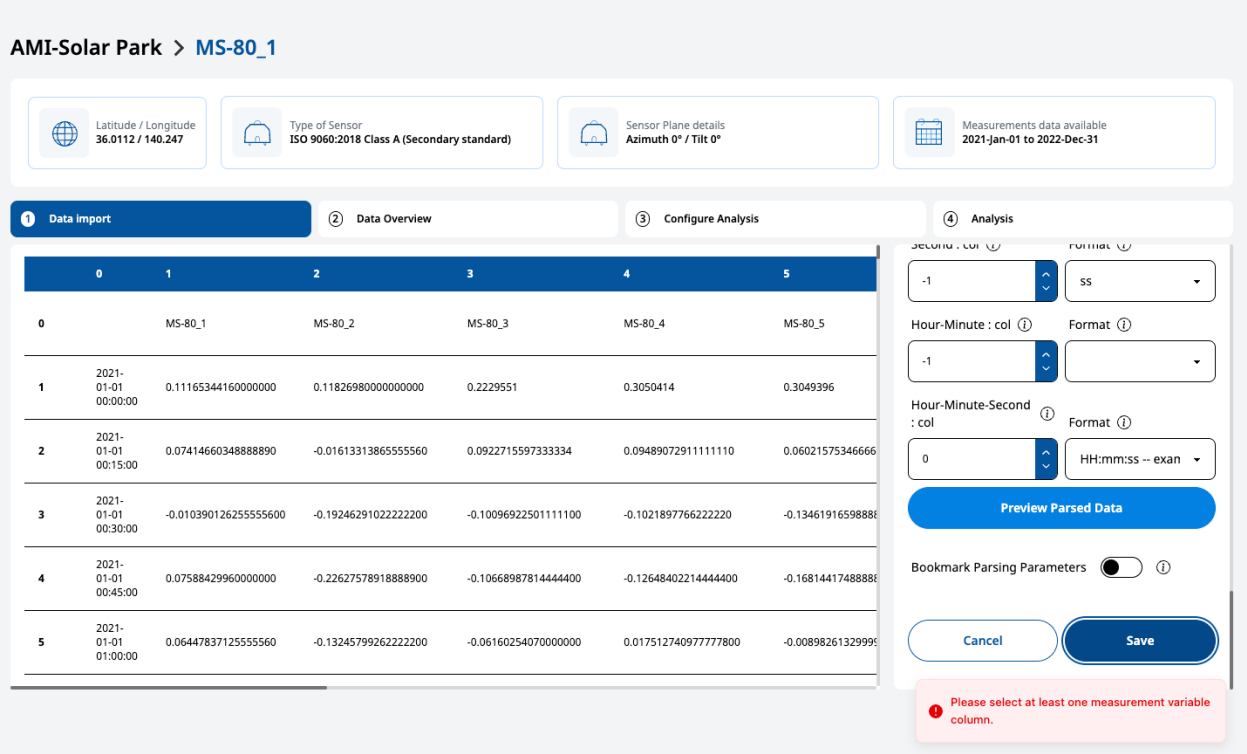


Figure 17: Error prompt for the test dataset corresponding to the presence of more than one column.

As seen from the Figure 17 above, this error occurs when more than one column is present, excluding the Datetime Index, preventing the tool from determining the correct column of interest. The next section details how to resolve this issue.

When multiple columns are present in the dataset (excluding the Datetime Index), the tool cannot automatically determine which column contains the sensor measurement data. To resolve this, the user must manually select the appropriate column under the Measurement Variables section.

Figure 18 below displays the available inputs under **"Measurement Variables"**, which allows the user to configure the column containing the desired measurement data. Specifically, the user can:

- **Set “Global Irr: col”**: Select the appropriate column number corresponding to the **irradiance data**.
- **Define the Unit**: Choose the correct unit of measurement (W/m² as default), ensuring accurate interpretation of the values.

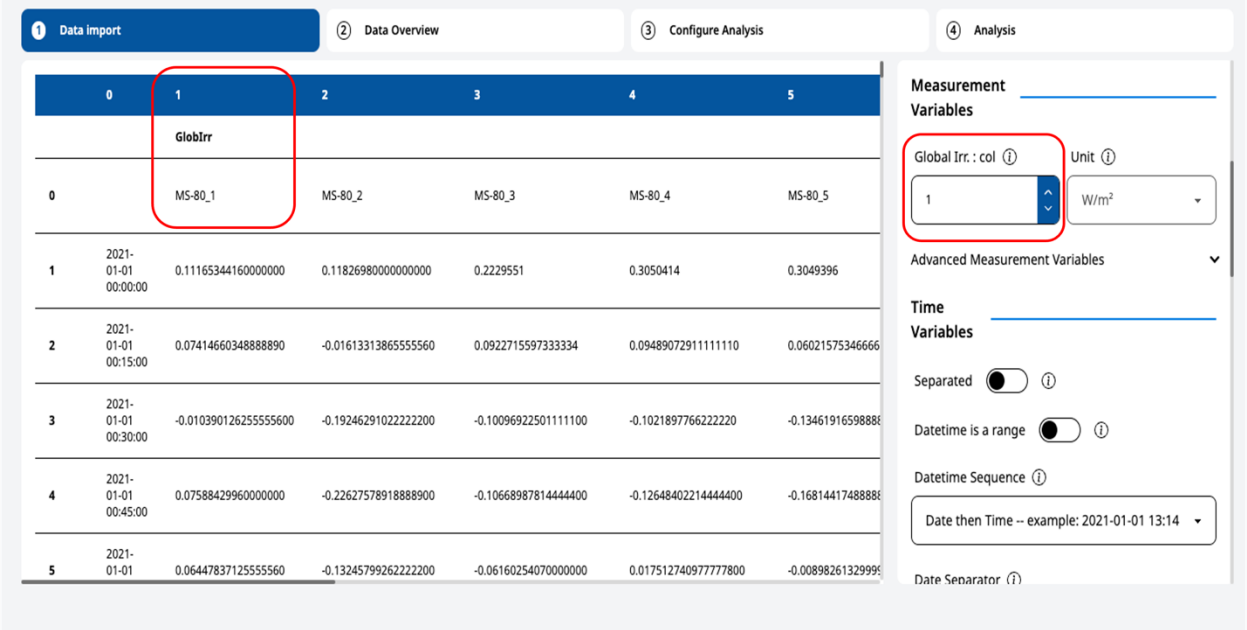


Figure 18: Selecting the desired column to be analysed.

As seen in Figure 19 below, the ‘Global Irr: col’ is set to 1, which corresponds to the column containing the right ‘Sensor label’ . Upon selecting this column, the label **“GlobIrr”** appears at the top of the header for column 1, confirming that the desired column for analysis has been successfully selected.

With this change, the user can now proceed to the next step by clicking the **“Save”** button, as shown in Figure 19 below. This action finalizes the parsing configuration and, if all settings are correct, allows the user to move forward to the **“Data Overview”** section for further analysis.

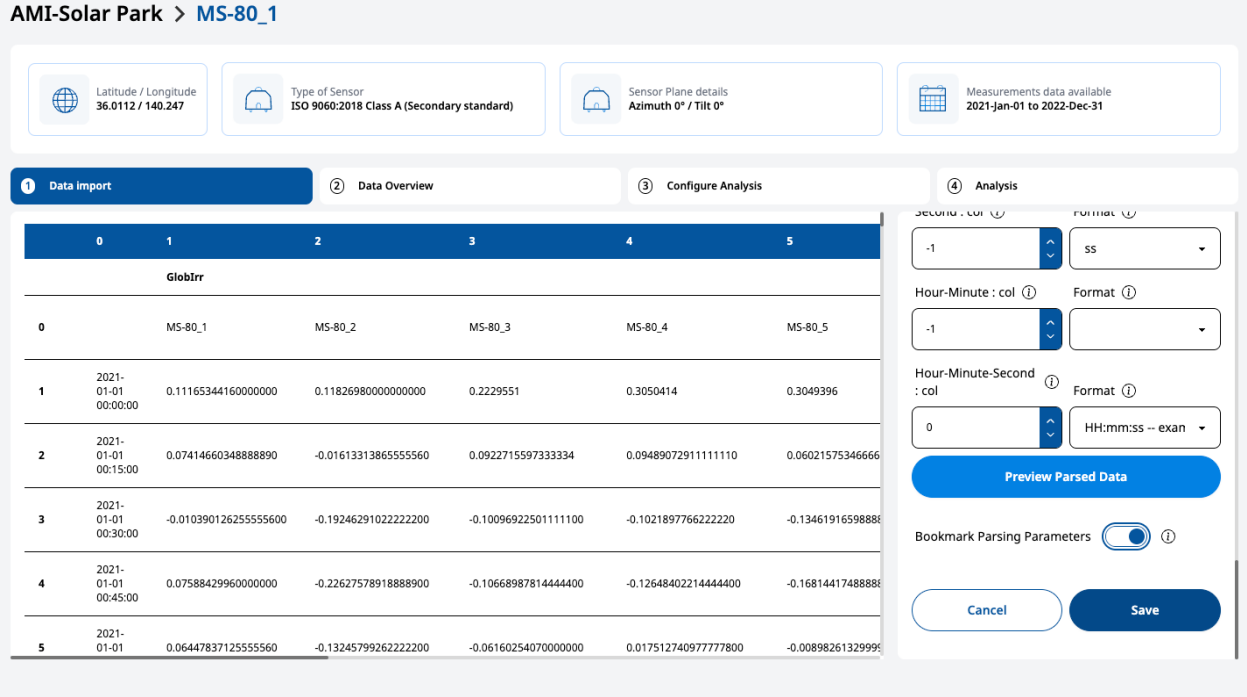


Figure 19: Parsing inputs chosen for a particular sensor can be previewed and bookmarked.

Before proceeding to click “Save”, the user has the option to Preview Parsed Data, which can be done at any stage during the parsing process. This feature allows the user to verify that the data has been correctly interpreted before finalizing the configuration.

Once all inputs are set and validated, the user can toggle “Bookmark Parsing Parameters”, as shown in Figure 19 above to save these parsing settings for future use on the same sensor. By enabling this option, all future uploads for this sensor will automatically apply the same parsing parameters, reducing the need for manual reconfiguration.

Upon clicking “Save”, a successful upload confirmation message is displayed, indicating that the user can now proceed to the next step: “Data Overview.”