

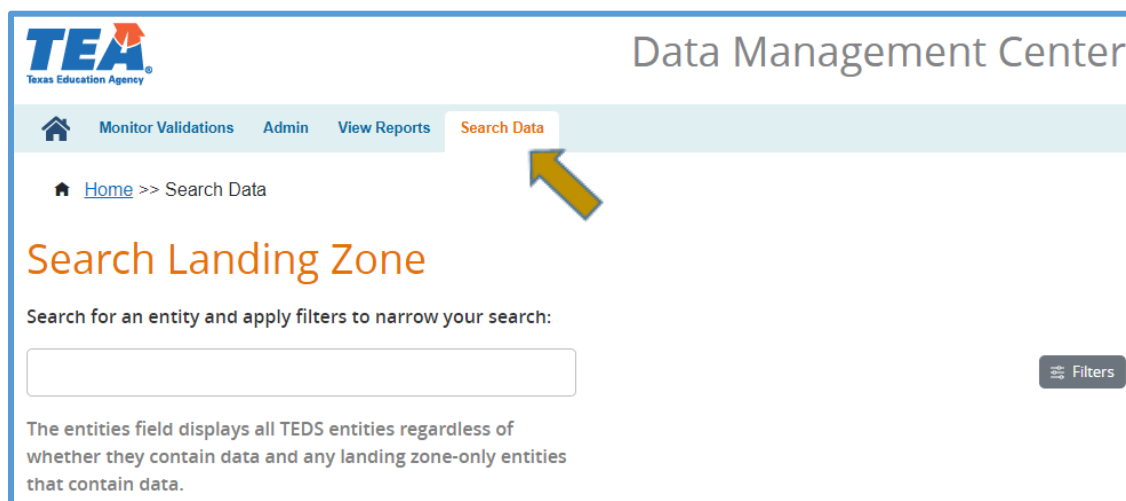
JavaScript Object Notation (JSON) Resource Document

Target Audience: Local education agencies (LEAs) and Education service centers (ESCs).

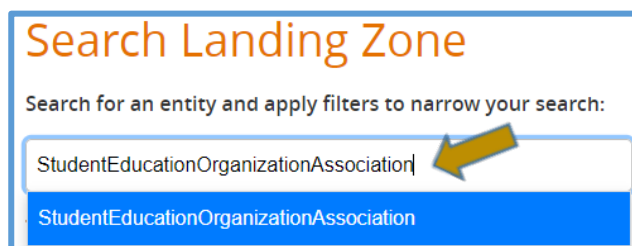
Purpose: To provide guidance on how to read JavaScript Object Notation (JSON) files found within an entity record list in the Individual Operational Data Store (IODS) using Search Data in the Data Management Center (DMC).

Accessing JSON Files:

1. Navigate to Search Data in the DMC.



2. Type the name of the entity that you wish to search (e.g., StudentEducationOrganizationAssociation).



Search Tips HERE!!!

- You must type at least three characters in the search field to retrieve a list of entities consisting of those characters.. The autocomplete feature will display a short list of entities that contain those letters. It is recommended that you enter the first three characters of the entity you are searching for if known.
- Continue typing the name of the entity when the one you are searching for does not populate to the autocomplete display list.

Search Landing Zone

Search for an entity and apply filters to narrow your search:

Stu|

Student

StudentAcademicRecord

StudentApplication (TX)

StudentAssessment

StudentCohortAssociation

StudentCTEProgramAssociation

StudentDisciplineIncidentBehaviorAssociation

StudentEducationOrganizationAssociation

3. Select the name of the entity and apply filters to narrow your search or click **Search** to retrieve a full list of all records published to the IODS.

Search Landing Zone

Search for an entity and apply filters to narrow your search:

×

The entities field displays all TEDS entities regardless of whether they contain data and any landing zone-only entities that contain data.

EducationOrganizationId

StudentUniqueid

Clear All

Search

4. Search Results:

- From the list of search results, click **View** under the JSON column to open the file you wish to check.

Show entries Search Table:

XLSX

Print

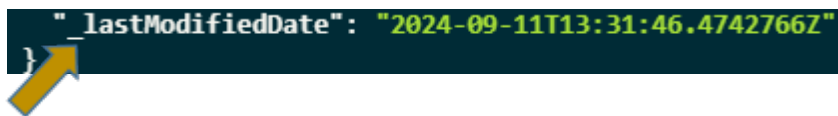
EducationOrganizationId	StudentUniqueid	JSON
701603	1121128866	View
701603	1121335373	View
701603	1121337643	View
701603	1121342361	View
701603	1121352219	View
701603	1121393314	View
701603	1121394418	View
701603	1121423736	View

Key things to note while reading JSON files:

- Data is separated by commas.
- Curly { } brackets hold objects (e.g., data elements).
- Square [] brackets hold arrays (e.g., common types such homelessStatusSets, which may have multiple instances).
 - Technically, if you see a right-facing square bracket ']' the data for that common type ends and a new one begins.
- Some descriptors may be blank which indicates either they are required by Ed-Fi but there is no data to report, no state reportable data, or are optional to report for an entity.

Reading JSON Files:

1. At the bottom of the JSON file read the line for '_lastModifiedDate' to determine when the last time this specific JSON file was updated.



```
"_lastModifiedDate": "2024-09-11T13:31:46.4742766Z"
```

- If you made changes to the data but the date does not reflect when the data should have been updated, please contact your source system vendor for assistance.
- Example of how date is formatted: "2024-09-11T13:31:46.4742766Z"
 - Date Format: YYYY-MM-DD
 - Time Format: HH:MM:SS (starts after "T" which stands for timestamp)
 - The numbers and letters that come after the timestamp are part of the String data that you do not need to be concerned with.

2. Go back to the top of the JSON record and scroll down until you see the last hypertext reference (href) line.
 - All data above this is referential data. All the referential data which is based on data dependency must be in the IODS before any of the other data in the JSON file can be published.

```
1 {
2   "id": "72bb96877b824eb98774aea4a22e5ed7",
3   "educationOrganizationReference": {
4     "educationOrganizationId": 701603,
5     "link": {
6       "rel": "LocalEducationAgency",
7       "href": "/ed-fi/localEducationAgencies/9ceb9927b0e04cca9c96dfb244b34407"
8     }
9   },
10  "studentReference": {
11    "studentUniqueId": "1121128866",
12    "link": {
13      "rel": "Student",
14      "href": "/ed-fi/students/a2c263490f1d4ccca3441a93f34b1ff6"
15    }
16  },
17  "hispanicLatinoEthnicity": true,
18  "sexDescriptor": "uri://tea.texas.gov/SexDescriptor#M",
```

- All the data below the last 'href' line is the data that has been published in connection to the referential data previously listed.

3. Reading the data below the last 'href' line:

```
14    "href": "/ed-fi/students/a2c263490f1d4ccca3441a93f34b1ff6"
15  }
16 },
17 "hispanicLatinoEthnicity": true,
18 "sexDescriptor": "uri://tea.texas.gov/SexDescriptor#M",
19 "_ext": {
20   "tx": {
21     "studentAcceleratedEducationPlan": true,
22     "acceleratedInstructionSets": [],
23     "dyslexiaRiskSets": [
24       {
25         "beginDate": "2024-08-01",
26         "dyslexiaRiskDescriptor": "uri://tea.texas.gov/DyslexiaRiskDescriptor#02",
27         "dyslexiaScreeningExceptionReasonDescriptor": "uri://tea.texas.gov/DyslexiaScreeningExceptionReasonDescriptor#01",
28         "endDate": "2024-10-31"
```

- Data standards that are:
 - Left aligned are typically Ed-Fi standards.
 - Indented are typically Texas-specific extensions.
- You can determine if the standards are Texas-specific extensions by the lines that precede them that show 'ext' and 'tx'.
- Typically, the data in the JSON file is listed in alphabetical order, but not always as it is based on the source system vendor's software.

4. Reading data standard sets:

- Example: 'earlyReadingIndicatorDescriptor'

```
"earlyReadingIndicatorSets": [  
  {  
    "beginDate": "2024-08-01",  
    "earlyReadingIndicatorDescriptor": "uri://tea.texas.gov/EarlyReadingIndicatorDescriptor#1",  
    "endDate": "2024-10-31"  
  },  
  {  
    "beginDate": "2024-11-01",  
    "earlyReadingIndicatorDescriptor": "uri://tea.texas.gov/EarlyReadingIndicatorDescriptor#2",  
    "endDate": "2024-12-31"  
  },  
  {  
    "beginDate": "2025-01-01",  
    "earlyReadingIndicatorDescriptor": "uri://tea.texas.gov/EarlyReadingIndicatorDescriptor#3",  
    "endDate": "2025-05-31"  
  }  
]
```

- A set or common type typically begins with 'beginDate'.
 - The common type can consist of data elements with the following data types: Boolean, Number, or Descriptor.
- To the left of the colon is the data element name, and to the right of the colon is the data type value.
- Sets that are 'beginDate' sensitive require an 'endDate' when reporting multiple instances like in the example in the image above.
 - If there is no 'endDate' data published or the 'endDate' for all instances are null or after snapshot, the promotion logic will automatically choose one of the instances at random, and it might not be correct.

5. Reading URLs:

```
"unschooledAsyleeRefugeeDescriptor": "uri://tea.texas.gov/UnschooledAsyleeRefugeeDescriptor#1"
```

- The url that you see to the right of the colon:
 - "uri://tea.texas.gov/ - indicates the namespace that is being used which can be TEA, Ed-Fi, or local.
 - UnschooledAsyleeRefugee - indicates the TEDS descriptor name
 - #1" – indicates the value for the above TEDS descriptor based on the Descriptor Table that can be found in [TWEDS](#).