Tabular Data Resource

A simple format to describe a single tabular data resource such as a CSV file. It includes support both for metadata such as author and title and a schema to describe the data, for example the types of the fields/columns in the data.

Author(s)	Paul Walsh, Rufus Pollock
Created	15 December 2017
Updated	2 May 2017
JSON Schema	tabular-data-resource.json
Version	1

Language

The key words `MUST`, `MUST NOT`, `REQUIRED`, `SHALL`, `SHALL NOT`, `SHOULD`, `SHOULD NOT`, `RECOMMENDED`, `MAY`, and `OPTIONAL` in this document are to be interpreted as described in RFC 2119

Introduction

A **Tabular Data Resource** is a type of Data Resource specialized for describing tabular data like CSV files or spreadsheets.

Tabular Data Resource extends Data Resource in following key ways:

- The `schema` property MUST follow the Table Schema specification, either as a JSON object directly under the property, or a string referencing another JSON document containing the Table Schema
- A new `dialect` property to describe the CSV dialect. This property follows the CSV Dialect specification.

Examples

A minimal Tabular Data Resource, referencing external JSON documents, looks as follows.

```js
// with data and a schema accessible via the local filesystem
{
  "profile": "tabular-data-resource",
  "name": "resource-name",
  "path": [ "resource-path.csv" ],
  "schema": "tableschema.json"
}

// with data accessible via http
{
  "profile": "tabular-data-resource",
  "name": "resource-name",
  "path": [ "http://example.com/resource-path.csv" ],
  "schema": "http://example.com/tableschema.json",
  "dialect": "http://example.com/csvdialect.json"
}
```

A minimal Tabular Data Resource example using the data property to inline data looks as follows.

```js
{
  "profile": "tabular-data-resource",
  "name": "resource-name",
  "data": [
    {
      "id": 1,
      "first_name": "Louise"
    },
    {
      "id": 2,
      "first_name": "Julia"
    }
  ],
  "schema": {
    "fields": [
      },
      {
        "name": "first_name",
        "type": "string"
      }
    ],
    "primaryKey": "id"
  }
}
```

A comprehensive Tabular Data Resource example with all required, recommended and optional properties looks as follows.

```js
{
  "profile": "tabular-data-resource",
  "name": "solar-system",
  "path": "http://example.com/solar-system.csv",
  "title": "The Solar System",
  "description": "My favourite data about the solar system.",
  "format": "csv",
  "mediatype": "text/csv",
  "encoding": "utf-8",
  "bytes": 1,
  "hash": "",
  "schema": {
    "fields": [
      {
        "name": "id",
        "type": "integer"
      },
      {
        "name": "name",
        "type": "string"
      },
      {
        "name": "description",
        "type": "string"
      }
    ],
    "primaryKey": "id"
  },
  },
  "sources": [{
      "title": "The Solar System - 2001",
      "path": "http://example.com/solar-system-2001.json",
      "email": ""
    }],
    "licenses": [{
      "name": "CC-BY-4.0",
      "title": "Creative Commons Attribution 4.0",
      "path": "https://creativecommons.org/licenses/by/4.0/"
    }]
}
```

Specification

A Tabular Data Resource MUST be a Data Resource , that is it MUST conform to the Data Resource specification .

In addition:

- The Data Resource `schema` property MUST follow the Table Schema specification, either as a JSON object directly under the property, or a string referencing another JSON document containing the Table Schema
- There `MUST` be a `profile` property with the value `tabular-data-resource`
- The data the Data Resource describes MUST:
  - If non-inline: Be a CSV file
  - If inline data: be “JSON tabular data” that is array of data rows where each row is an `array` or `object` (see below)

CSV file requirements

CSV files in the wild come in a bewildering array of formats. There is a standard for CSV files described in RFC 4180 , but unfortunately this standard does not reflect reality. In Tabular Data Resource, CSV files `MUST` follow RFC 4180 with the following important exceptions allowed:

File encoding

Files MUST:

- EITHER be encoded as UTF-8 (the default)
- OR the Tabular Data Resource MUST include an `encoding` property and the files `MUST` follow that encoding

NB: the RFC requires 7-bit ASCII encoding.

CSV Dialect

The line terminator character `MUST` be LF or CRLF (the RFC allows CRLF only).

If the CSV differs from this or the RFC in any other way regarding dialect (e.g. line terminators, quote characters, field delimiters), the Tabular Data Resource MUST contain a `dialect` property describing its dialect. The `dialect` property MUST follow the CSV Dialect specification.

The value for the `dialect` property on a `resource` MUST be an `object` representing the dialect OR a `string` that identifies the location of the dialect.

If a `string` it must be a url-or-path, that is a fully qualified http URL or a relative POSIX path. The file at the the location specified by this url-or-path string `MUST` be a JSON document containing the dialect.

JSON Tabular Data

JSON Tabular Data MUST be an `array` where each item in the array MUST be:

- EITHER: an array where each entry in the array is the value for that cell in the table
- OR: an object where each key corresponds to the header for that row and the value corresponds to the cell value for that row for that header

Row Arrays

```js
[
]
```

Row Objects

js

```js
[
  { "A": 1, "B": 2, "C": 3 },
  { "A": 4, "B": 5, "C": 6 }
]
```

Edit this page

Last Updated: 3/31/2021, 4:44:15 PM