CSV on the Web

Intro to W3C CSV on the Web Specifications
DDI Metadata Workshop – Dagstuhl 2016

Gregg Kellogg
gregg@greggkellogg.net
https://gkellogg.github.com/ddi-csvw
@gkellogg

CSV data is dumb

- It’s a simple text format, data has no inherent meaning.
  - Cells may be data-typed or have a regular format: what does “09/10/2016” mean?
  - Cells may be related to data in other tables/ columns: Foreign Keys
  - Cells may be associated with different entities: Join results

Web CSV

- 5-star Linked Data
  - CSV URLs
  - CSVs link to other CSVs
  - CSVs link to other Resources
  - RDF and JSON conversion

W3C CSV on the Web

- Working Group chartered to allow applications to provide higher interoperability with working with CSV, or similar formats.
  - Use Cases: http://www.w3.org/TR/csvw-ucr/
  - Model for Tabular Data and Metadata on the Web: http://www.w3.org/TR/tabular-data-model/
  - Metadata Vocabulary for Tabular Data: http://www.w3.org/TR/tabular-metadata/
  - Generating JSON from Tabular Data on the Web: http://www.w3.org/TR/csv2json/
  - Generating RDF from Tabular Data on the Web: http://www.w3.org/TR/csv2rdf/

Model for Tabular Data

Table Group
id
notes
tables
other annotations

Table
id
columns
foreign keys
notes
rows
table direction
transformations
url
other annotations

Column
about URL
cells
datatype
default
lang
name
number
ordered
property URL
required
rows
separator
table
text direction
titles
value URL
virtual
other annotations

Row
cells
number
primary key
table
titles
referenced rows
source number
table

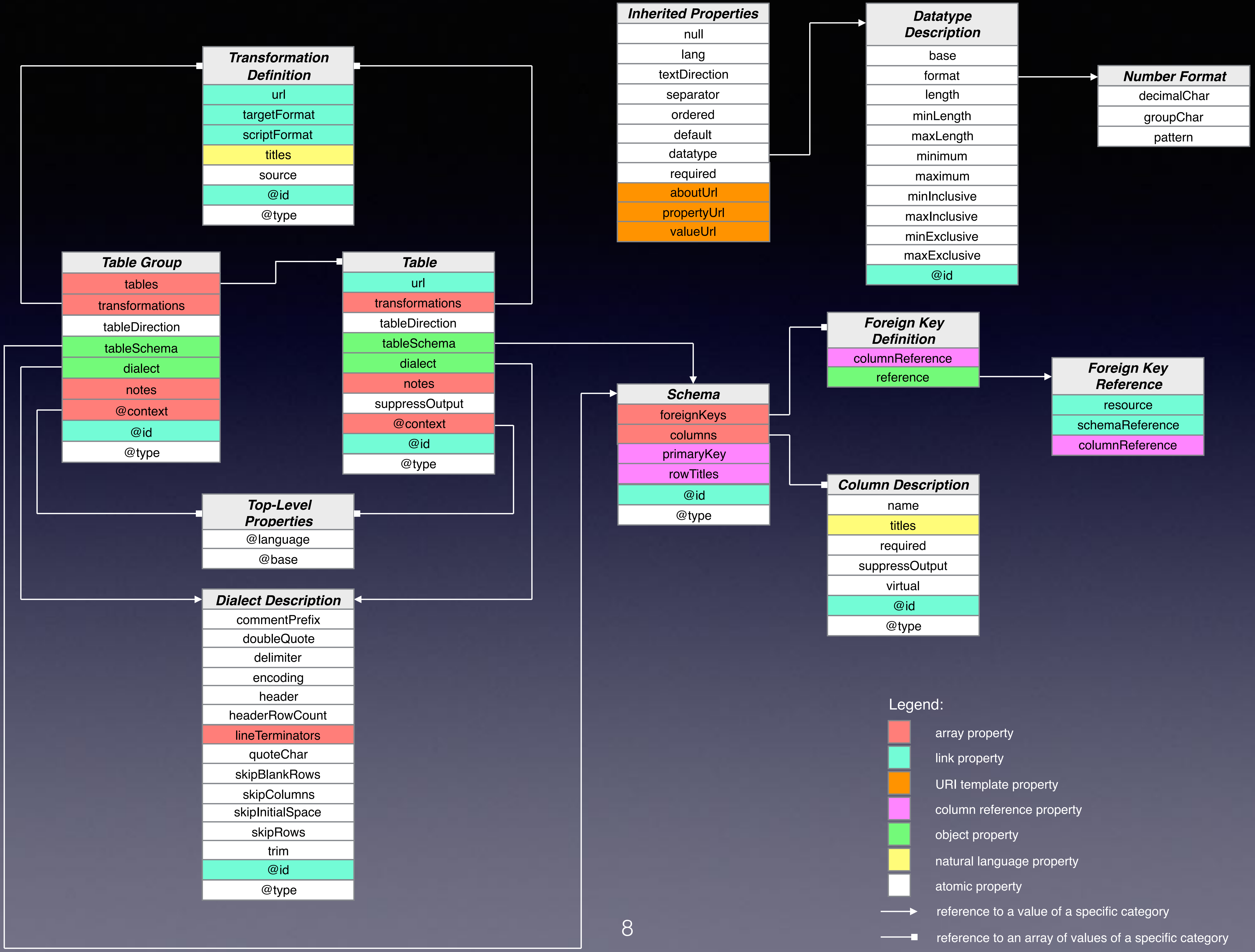
Cell
about URL
column
errors
ordered
property URL
row
string value
table
text direction
value
value URL

Mapping CSV to Model

- Parse CSV: RFC4180 + dialect metadata.
  - delimiter, doubleQuote, headerRowCount, lineTerminators, quoteChar, …
- Dialect Description comes from *Metadata Document*.
- Match Headers to Columns.
- Parse Cells using Column metadata/datatype.
- Abstract data model used for viewing, validation, and conversions.

Metadata

- Finding Metadata from a CSV
  - User-specified, Link Header, well-known locations
- Matching Metadata to a CSV
  - CSV must be compatible with metadata (titles/names)
  - Metadata must reference CSV URL



Transformation Definition
- 🟦 url
- 🟦 targetFormat
- 🟦 scriptFormat
- 🟨 titles
- ⬜ source
- 🟦 @id
- ⬜ @type

Table Group
- 🟥 tables
- 🟥 transformations
- ⬜ tableDirection
- 🟩 tableSchema
- 🟩 dialect
- 🟥 notes
- 🟥 @context
- 🟦 @id
- ⬜ @type

Table
- 🟦 url
- 🟥 transformations
- ⬜ tableDirection
- 🟩 tableSchema
- 🟩 dialect
- 🟥 notes
- ⬜ suppressOutput
- 🟥 @context
- 🟦 @id
- ⬜ @type

Top-Level Properties
- ⬜ @language
- ⬜ @base

Dialect Description
- ⬜ commentPrefix
- ⬜ doubleQuote
- ⬜ delimiter
- ⬜ encoding
- ⬜ header
- ⬜ headerRowCount
- 🟥 lineTerminators
- ⬜ quoteChar
- ⬜ skipBlankRows
- ⬜ skipColumns
- ⬜ skipInitialSpace
- ⬜ skipRows
- ⬜ trim
- 🟦 @id
- ⬜ @type

Inherited Properties
- ⬜ null
- ⬜ lang
- ⬜ textDirection
- ⬜ separator
- ⬜ ordered
- ⬜ default
- ⬜ datatype
- ⬜ required
- 🟧 aboutUrl
- 🟧 propertyUrl
- 🟧 valueUrl

Datatype Description
- ⬜ base
- ⬜ format
- ⬜ length
- ⬜ minLength
- ⬜ maxLength
- ⬜ minimum
- ⬜ maximum
- ⬜ minInclusive
- ⬜ maxInclusive
- ⬜ minExclusive
- ⬜ maxExclusive
- 🟦 @id

Number Format
- ⬜ decimalChar
- ⬜ groupChar
- ⬜ pattern

Schema
- 🟥 foreignKeys
- 🟥 columns
- 🟪 primaryKey
- 🟪 rowTitles
- 🟦 @id
- ⬜ @type

Foreign Key Definition
- 🟪 columnReference
- 🟩 reference

Foreign Key Reference
- 🟦 resource
- 🟦 schemaReference
- 🟪 columnReference

Column Description
- ⬜ name
- 🟨 titles
- ⬜ required
- ⬜ suppressOutput
- ⬜ virtual
- 🟦 @id
- ⬜ @type

Legend:
- 🟥 array property
- 🟦 link property
- 🟧 URI template property
- 🟪 column reference property
- 🟩 object property
- 🟨 natural language property
- ⬜ atomic property
- → reference to a value of a specific category
- —■ reference to an array of values of a specific category

Examples

countryCode	latitude	longitude	name
AD	42.5	1.6	Andorra
AE	23.4	53.8	United Arab Emirates
AF	33.9	67.7	Afghanistan

countries.csv

countryRef	year	population
AF	1960	9,616,353
AF	1961	9,799,379
AF	1961	9,989,846

country_slice.csv

Schema

- Column Descriptions
  - Names/Titles
  - Datatype
- Primary Keys
- Foreign Key Relationships

Embedded Metadata

- Generally Column Titles.
- Formats may define CSV conventions for embedded metadata.
- Principally used to determine metadata compatibility.
  - Also serves as default metadata if no file located.

Datatypes

- Basic XSD datatypes
  - maximum/minimum facets
  - minLength/maxLength facets
  - format/pattern
    - RegExp, Boolean, UAX35 date/time picture string, UAX35 number picture string

Other Features

- Split cells into multiple items
- Validate Primary Keys and Foreign Key references (single and multiple columns)
- Define URL properties for columns
- Multiple subjects per column (may be URLs)
- Values as URLs

Conversions: JSON

countryCode	latitude	longitude	name
AD	42.5	1.6	Andorra
AE	23.4	53.8	United Arab Emirates
AF	33.9	67.7	Afghanistan

countries.csv

countries.json

countries-standard.json

```
{
  "tables": [{
    "url": "http://example.org/countries.csv",
    "row": [{
      "url": "http://example.org/countries.csv#row=2",
      "rownum": 1,
      "describes": [{
        "countryCoe": "AD",
        "latitude": "42.5",
        "longitude": "1.6",
        "name": "Andorra"
      }]
    }, {
      "url": "http://example.org/countries.csv#row=3",
      "rownum": 2,
      "describes": [{
        "countryCode": "AE",
        "latitude": "23.4",
        "longitude": "53.8",
        "name": "United Arab Emirates"
      }]
    }, {
      "url": "http://example.org/countries.csv#row=4",
      "rownum": 3,
      "describes": [{
        "countryCode": "AF",
        "latitude": "33.9",
        "longitude": "67.7",
        "name": "Afghanistan"
      }]
    }]
  }]
}
```

Conversions: JSON (min)

countryCode	latitude	longitude	name
AD	42.5	1.6	Andorra
AE	23.4	53.8	United Arab Emirates
AF	33.9	67.7	Afghanistan

countries.csv

countries.json

countries-minimal.json

```
[{
  "countryCode": "AD",
  "latitude": "42.5",
  "longitude": "1.6",
  "name": "Andorra"
}, {
  "countryCode": "AE",
  "latitude": "23.4",
  "longitude": "53.8",
  "name": "United Arab Emirates"
}, {
  "countryCode": "AF",
  "latitude": "33.9",
  "longitude": "67.7",
  "name": "Afghanistan"
}]
```

Conversions: RDF

countryCode	latitude	longitude	name
AD	42.5	1.6	Andorra
AE	23.4	53.8	United Arab Emirates
AF	33.9	67.7	Afghanistan

countries.csv

countries.json

countries-standard.ttl

```
@base <http://example.org/countries.csv> .
@prefix csvw: <http://www.w3.org/ns/csvw#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

_:tg a csvw:TableGroup ;
  csvw:table [ a csvw:Table ;
    csvw:url <http://example.org/countries.csv> ;
    csvw:row [ a csvw:Row ;
      csvw:rownum "1"^^xsd:integer ;
      csvw:url <#row=2> ;
      csvw:describes _:t1r1
    ], [ a csvw:Row ;
      csvw:rownum "2"^^xsd:integer ;
      csvw:url <#row=3> ;
      csvw:describes _:t1r2
    ], [ a csvw:Row ;
      csvw:rownum "3"^^xsd:integer ;
      csvw:url <#row=4> ;
      csvw:describes _:t1r3
    ]
  ] .

_:t1r1
  <#countryCode> "AD" ;
  <#latitude> "42.5" ;
  <#longitude> "1.6" ;
  <#name> "Andorra" .

_:t1r2
  <#countryCode> "AE" ;
  <#latitude> "23.4" ;
  <#longitude> "53.8" ;
  <#name> "United Arab Emirates" .

_:t1r3
  <#countryCode> "AF" ;
  <#latitude> "33.9" ;
  <#longitude> "67.7" ;
  <#name> "Afghanistan" .
```

Conversions: RDF (min)

countryCode	latitude	longitude	name
AD	42.5	1.6	Andorra
AE	23.4	53.8	United Arab Emirates
AF	33.9	67.7	Afghanistan

countries.csv

countries.json

countries-minimal.ttl

```
@base <http://example.org/countries.csv> .

_:t1r1
  <#countryCode> "AD" ;
  <#latitude> "42.5" ;
  <#longitude> "1.6" ;
  <#name> "Andorra" .

_:t1r2
  <#countryCode> "AE" ;
  <#latitude> "23.4" ;
  <#longitude> "53.8" ;
  <#name> "United Arab Emirates" .

_:t1r3
  <#countryCode> "AF" ;
  <#latitude> "33.9" ;
  <#longitude> "67.7" ;
  <#name> "Afghanistan" .
```

Tools

- CSVLint
- CKAN – open source data portal platform
- Socrata – cloud-based open data
- Google Fusion Tables – data visualization
- Ruby rdf-tabular – CSVW reference implementation
- RDF Distiller
- Structured Data Linter

More Information

w3c

GitHub

Primer

distiller

linter

Gregg Kellogg

gregg@greggkellogg.net
http://greggkellogg.net/

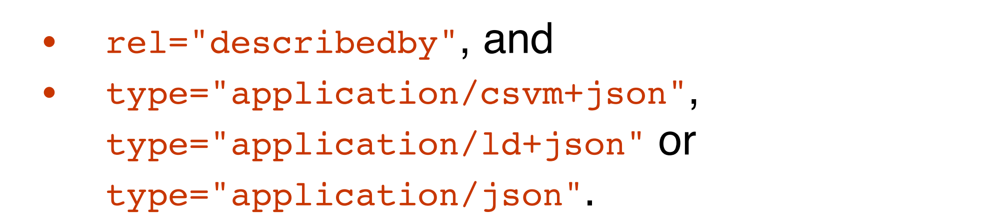
@gkellogg

https://gkellogg.github.com/ddi-csvw/

Deep Dive

Locating Metadata

- Start with Metadata
- HTTP Link header rel="*describedby*"



- `rel="describedby"`, and
- `type="application/csvm+json"`, `type="application/ld+json"` or `type="application/json"`.

- Default locations
  - {+url}-metadata.json
  - csv-metadata.json
  - /.well-known/csvm

```
{+url}-metadata.json
csv-metadata.json
```

- Embedded Metadata

Top-Level Properties

- Constrained JSON-LD Context
  - **MUST** include csvw namespace *http://www.w3.org/ns/csvw*
  - **MAY** include *@base* and/or *@language*

```
{
  "@context": "http://www.w3.org/ns/csvw",

}

{
  "@context": [
    "http://www.w3.org/ns/csvw",
    {
      "@base":      "http://example.org/",
      "@language":  "en-AU"
    }
  ],

}
```

Table Group

- **MUST** include *tables*
- **MAY** include any of the following:
  - *dialect* – how to parse CSV
  - *notes* – Arbitrary JSON-LD
  - *tableDirection*
  - *tableSchema* – defaults for tables not having a *tableSchema*
  - *transformations* – undefined. For transformations to other formats
  - *@id*
  - *@type* – if present **MUST** be “TableGroup”
  - common and inherited properties

```
{
  "@context": "http://www.w3.org/ns/csvw",
  "@type": "TableGroup",
  "dialect": {
    "delimiter": "\t",
    "headerRowCount": 3
  },
  "notes": {
    "type": "Annotation",
    "target": "countries.csv#cell=2,6-*,7",
    "body": "…representative points.",
    "motivation": "commenting"
  },
  "tables": [{
    "url": "countries.csv"
  }, {
    "url": "country-groups.csv"
  }],
  "tableDirection": "ltr",
  “tableSchema": {},
  "transformations": [{   }]
}
```

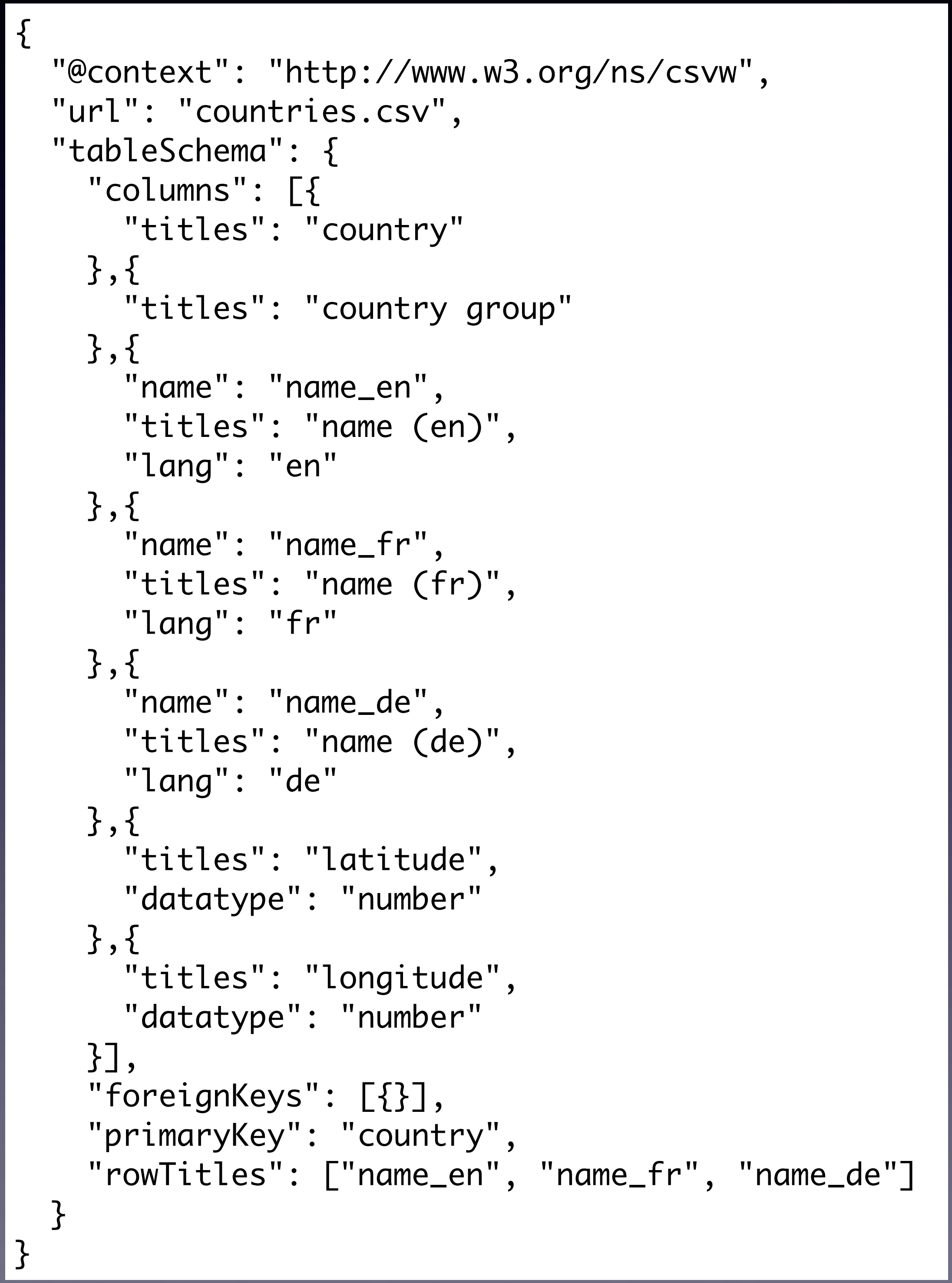
Table

- **MUST** include *url* – reference to CSV
- **MAY** include any of the following:
  - *notes* – Arbitrary JSON-LD
  - *suppressOutput*
  - *tableDirection*
  - *tableSchema* – must be defined someplace, to describe that format of referenced tables
  - *transformations*
  - *@id*
  - *@type* – If present **MUST** be “Table”
  - common and inherited properties

```
{
  "@context": "http://www.w3.org/ns/csvw",
  "@type": "Table",
  "url": "countries.csv",
  "dialect": {   },
  "notes": {   },
  "tableDirection": "..",
  "tableSchema": {
    "columns": [{
      "titles": "country"
    },{
      "titles": "country group"
    },{
      "titles": "name (en)"
    },{
      "titles": "name (fr)"
    },{
      "titles": "name (de)"
    },{
      "titles": "latitude"
    },{
      "titles": "longitude"
    }]
  },
  "transformations": {   }
}
```

Schema

- *columns* – for every column in the CSV. **MAY** also include *virtual columns.*
- *foreignKeys* – to validate against entries in another table.
- *primaryKey* – to determine uniqueness
- *rowTitles* – Reference to column who’s content defines the title for the row.
- *@id*
- *@type* – If present **MUST** be “Schema”
- common and inherited properties



```
{
  "@context": "http://www.w3.org/ns/csvw",
  "url": "countries.csv",
  "tableSchema": {
    "columns": [{
      "titles": "country"
    },{
      "titles": "country group"
    },{
      "name": "name_en",
      "titles": "name (en)",
      "lang": "en"
    },{
      "name": "name_fr",
      "titles": "name (fr)",
      "lang": "fr"
    },{
      "name": "name_de",
      "titles": "name (de)",
      "lang": "de"
    },{
      "titles": "latitude",
      "datatype": "number"
    },{
      "titles": "longitude",
      "datatype": "number"
    }],
    "foreignKeys": [{}],
    "primaryKey": "country",
    "rowTitles": ["name_en", "name_fr", "name_de"]
  }
}
```

Column

- *name* – Used for key referencing and in *URI templates*.

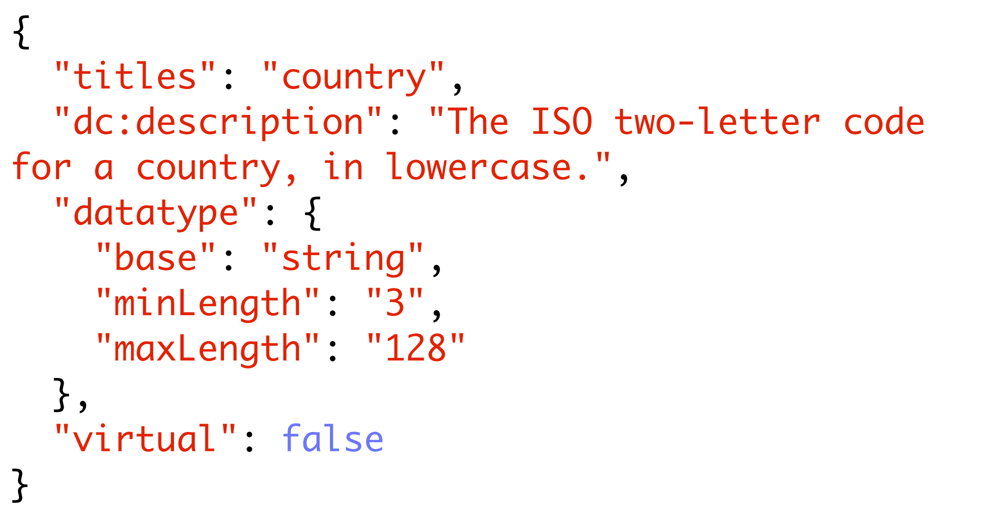
- *titles* – Titles of this column. Some title **MUST** match the header from the CSV. Allows different forms for internationalization.

- *virtual* – For columns not actually in the CSV. If present, comes after other columns. May be used as “glue”.

- *@id*

- *@type* – If present, **MUST** be “Column”

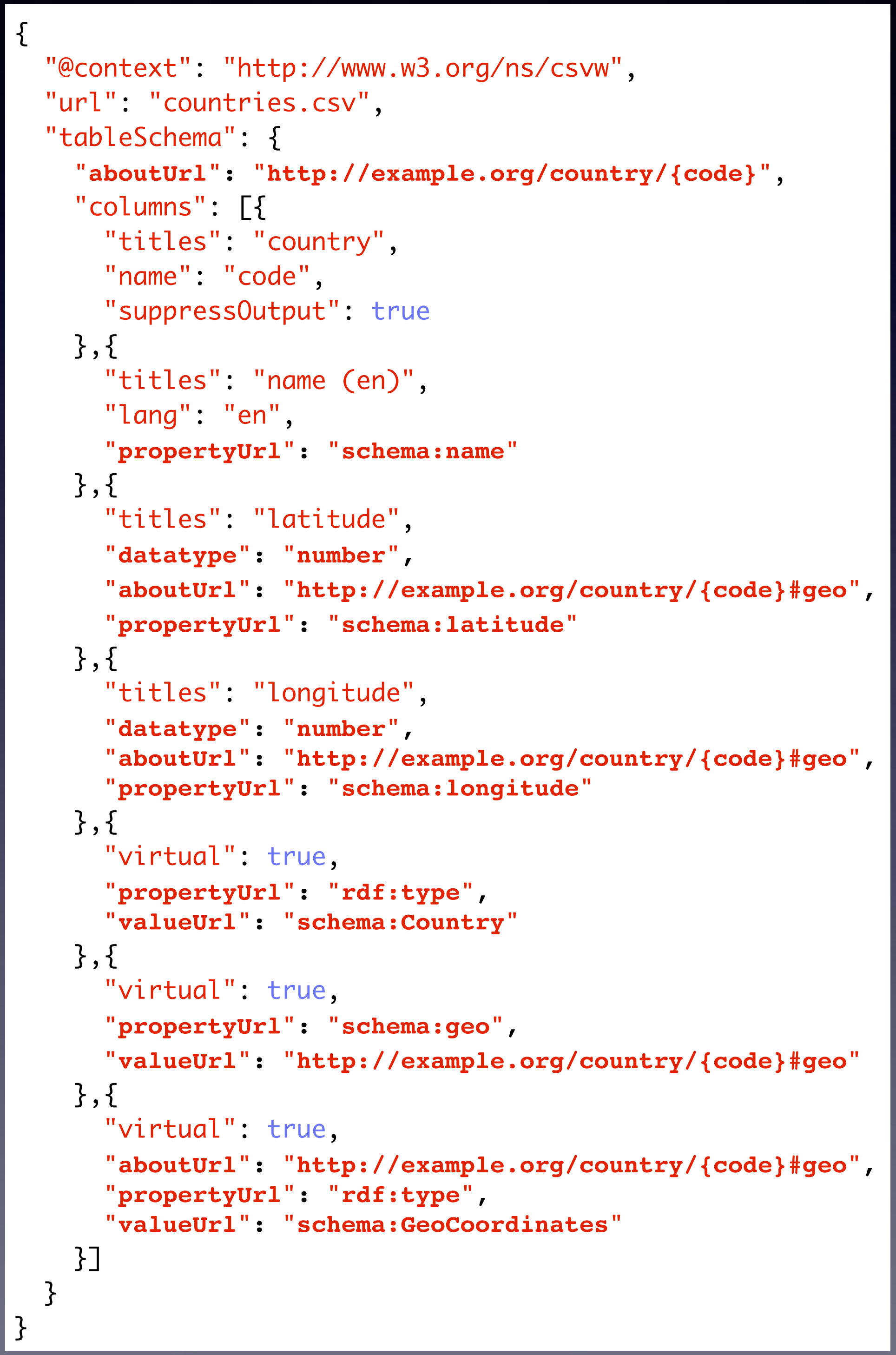
- common and inherited properties



```
{
  "titles": "country",
  "dc:description": "The ISO two-letter code
for a country, in lowercase.",
  "datatype": {
    "base": "string",
    "minLength": "3",
    "maxLength": "128"
  },
  "virtual": false
}
```

Inherited Properties

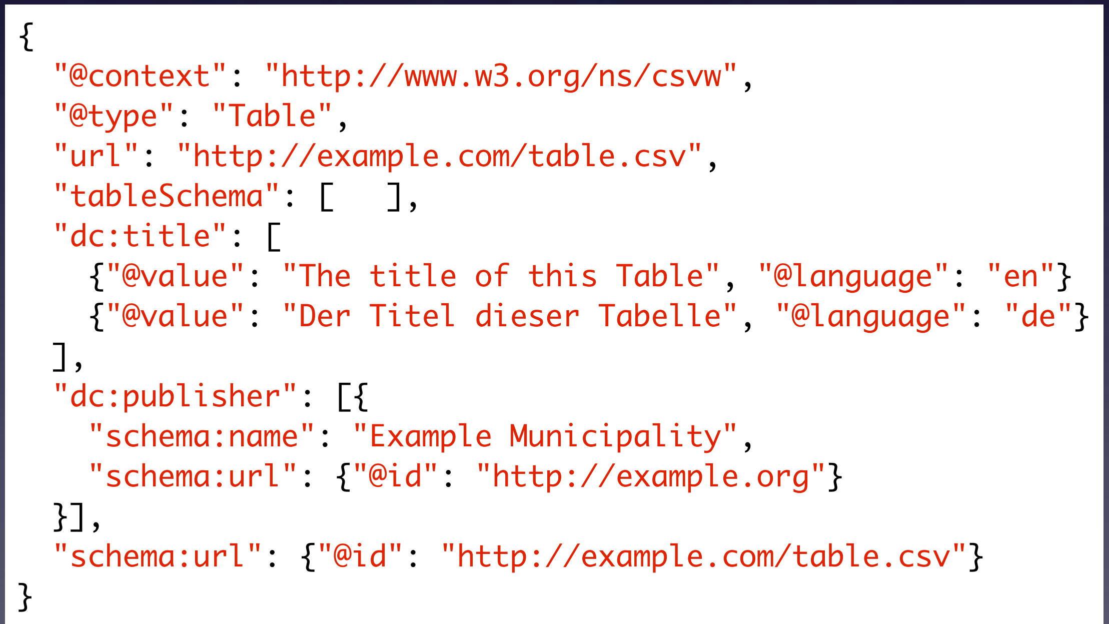
- *aboutUrl* – RDF subject (URI Template)
- *datatype* – See *Built-in Datatypes* and Derived Datatypes
- *default* – when value is null/missing
- *lang* – language for string values
- *null* – values to be considered the same as null
- *ordered* – Multiple values retain order (RDF)
- *propertyUrl* – RDF predicate (URI Template)
- *required* – requires column data to be present
- *separator* – how to split multiple values from a cell
- *textDirection* – “ltr”, “rtl”, “auto”, “inherit”
- *valueUrl* – RDF object (URI Template)



```
{
  "@context": "http://www.w3.org/ns/csvw",
  "url": "countries.csv",
  "tableSchema": {
    "aboutUrl": "http://example.org/country/{code}",
    "columns": [{
      "titles": "country",
      "name": "code",
      "suppressOutput": true
    },{
      "titles": "name (en)",
      "lang": "en",
      "propertyUrl": "schema:name"
    },{
      "titles": "latitude",
      "datatype": "number",
      "aboutUrl": "http://example.org/country/{code}#geo",
      "propertyUrl": "schema:latitude"
    },{
      "titles": "longitude",
      "datatype": "number",
      "aboutUrl": "http://example.org/country/{code}#geo",
      "propertyUrl": "schema:longitude"
    },{
      "virtual": true,
      "propertyUrl": "rdf:type",
      "valueUrl": "schema:Country"
    },{
      "virtual": true,
      "propertyUrl": "schema:geo",
      "valueUrl": "http://example.org/country/{code}#geo"
    },{
      "virtual": true,
      "aboutUrl": "http://example.org/country/{code}#geo",
      "propertyUrl": "rdf:type",
      "valueUrl": "schema:GeoCoordinates"
    }]
  }
}
```

Common Properties

- Properties which are *prefixed names*.
    - Generally arbitrary JSON-LD to associated with the associated model object.
    - Note that JSON-LD dialect is constrained.

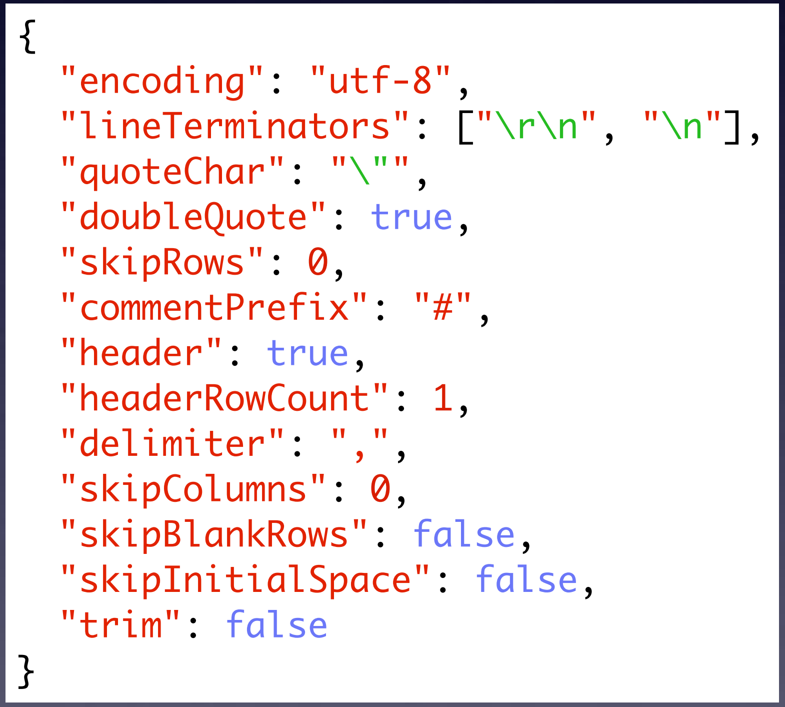


```
{
  "@context": "http://www.w3.org/ns/csvw",
  "@type": "Table",
  "url": "http://example.com/table.csv",
  "tableSchema": [   ],
  "dc:title": [
    {"@value": "The title of this Table", "@language": "en"}
    {"@value": "Der Titel dieser Tabelle", "@language": "de"}
  ],
  "dc:publisher": [{
    "schema:name": "Example Municipality",
    "schema:url": {"@id": "http://example.org"}
  }],
  "schema:url": {"@id": "http://example.com/table.csv"}
}
```

Dialect Description

- commentPrefix
- delimiter
- doubleQuote
- encoding
- header
- headerRowCount
- lineTerminators
- quoteChar

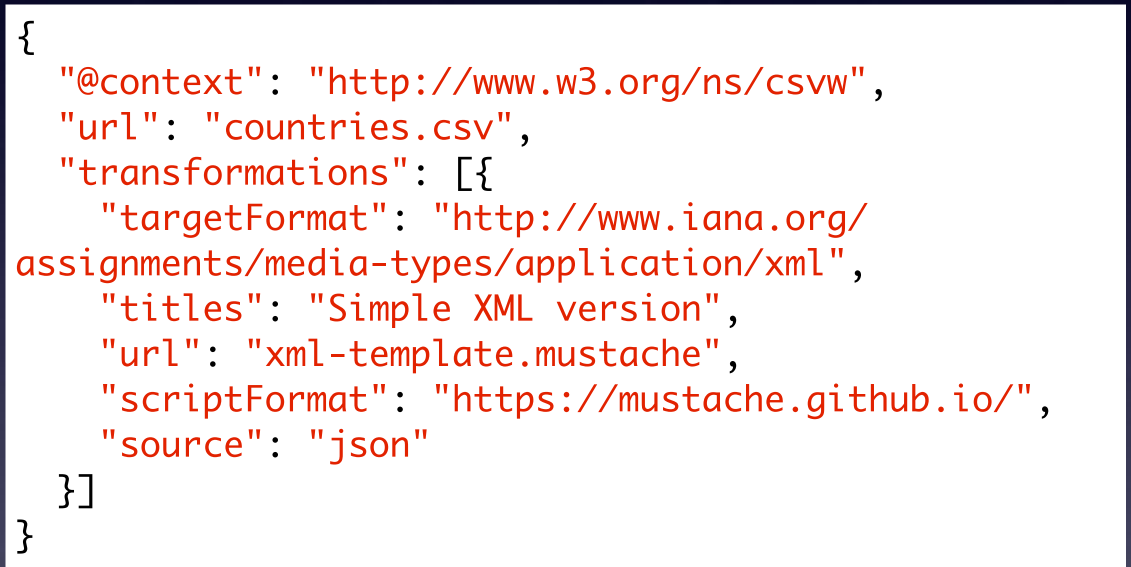
- skipBlankRows
- skipColumns
- skipInitialSpace
- skipRows
- trim
- @id
- @type



```
{
  "encoding": "utf-8",
  "lineTerminators": ["\r\n", "\n"],
  "quoteChar": "\"",
  "doubleQuote": true,
  "skipRows": 0,
  "commentPrefix": "#",
  "header": true,
  "headerRowCount": 1,
  "delimiter": ",",
  "skipColumns": 0,
  "skipBlankRows": false,
  "skipInitialSpace": false,
  "trim": false
}
```

Transformations

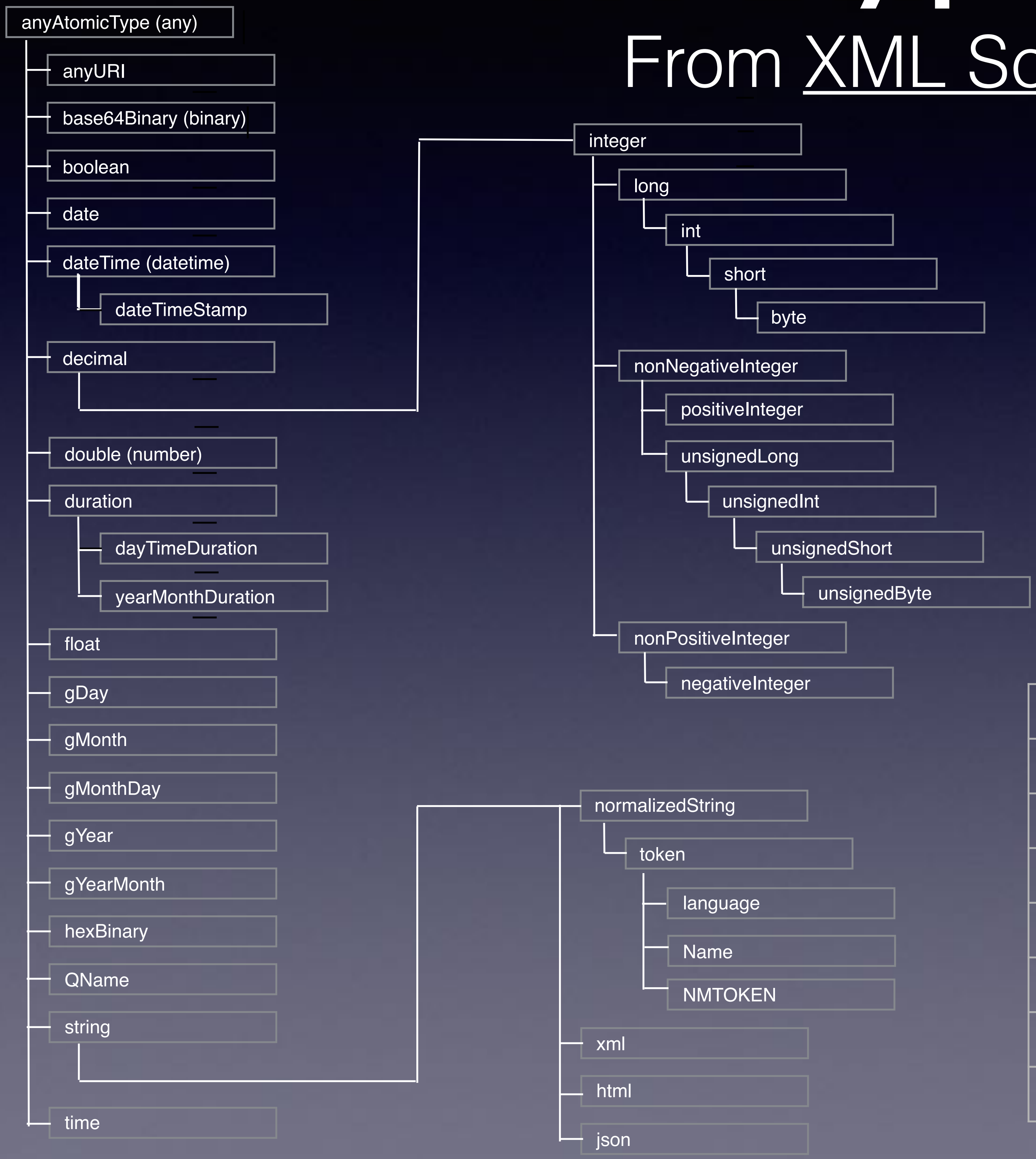
- **MUST** include the following properties:
  - *url* – reference to file containing script/template
  - *scriptFormat* – media type URL describing script format
  - *targetFormat* – media type URL describing target format
- **MAY** include the following:
  - *source* – How to format data before transformation
  - *titles* – for describing format profiles
  - *@id* –
  - *@type* – If present, **MUST** be “Template”



```
{
  "@context": "http://www.w3.org/ns/csvw",
  "url": "countries.csv",
  "transformations": [{
    "targetFormat": "http://www.iana.org/
assignments/media-types/application/xml",
    "titles": "Simple XML version",
    "url": "xml-template.mustache",
    "scriptFormat": "https://mustache.github.io/",
    "source": "json"
  }]
}
```

Built-in Datatypes

From XML Schema Pt. 2



- anyAtomicType (any)
  - anyURI
  - base64Binary (binary)
  - boolean
  - date
  - dateTime (datetime)
    - dateTimeStamp
  - decimal
    - integer
      - long
        - int
          - short
            - byte
      - nonNegativeInteger
        - positiveInteger
        - unsignedLong
          - unsignedInt
            - unsignedShort
              - unsignedByte
      - nonPositiveInteger
        - negativeInteger
  - double (number)
  - duration
    - dayTimeDuration
    - yearMonthDuration
  - float
  - gDay
  - gMonth
  - gMonthDay
  - gYear
  - gYearMonth
  - hexBinary
  - QName
  - string
    - normalizedString
      - token
        - language
        - Name
        - NMTOKEN
    - xml
    - html
    - json
  - time

Datatype Aliases	
number	double
binary	base64Binary
datetime	dateTime
any	anyAtomicType
xml	rdf:XMLLiteral
html	rdf:HTML
json	csvw:JSON

Derived Datatypes

- base – built-in datatype
- format – See Formats
- Length Constraints
  - length – length of cell
  - minLength – minimum length of cell
  - maxLength – maximum length of cell
- Value Constraints
  - minimum/maximum – values of cell
  - minInclusive/maxInclusive
  - minExclusive/maxExclusive
- @id
- @type – “Datatype”

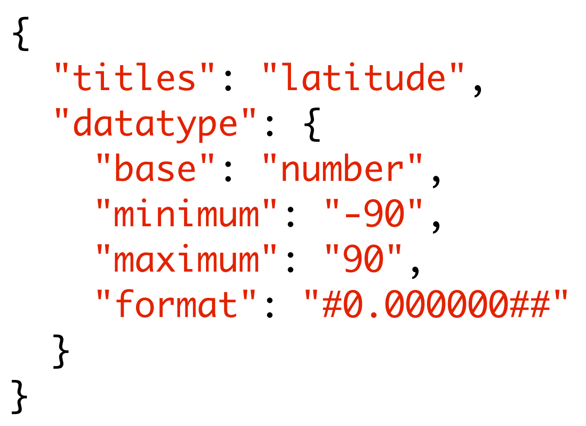
```
{
  "titles": "country",
  "datatype": {
    "dc:title": "Country Code",
    "dc:description": "Country codes as specified in
ISO 3166.",
    "base": "string",
    "format": "[a-z]{2}"
  }
}
```

```
{
  "titles": "name (en)",
  "datatype": {
    "base": "string",
    "minLength": "3",
    "maxLength": "128"
  }
}
```

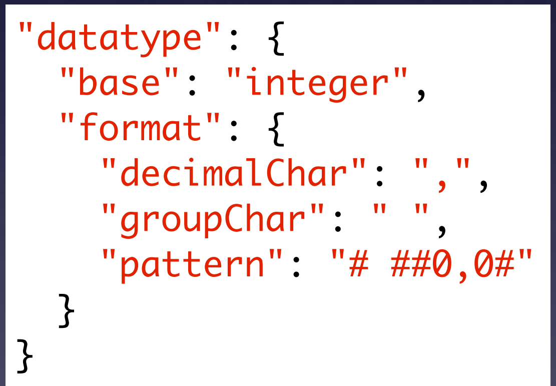
```
{
  "titles": "latitude",
  "datatype": {
    "base": "number",
    "minimum": "-90",
    "maximum": "90"
  }
}
```

Formats for numeric types

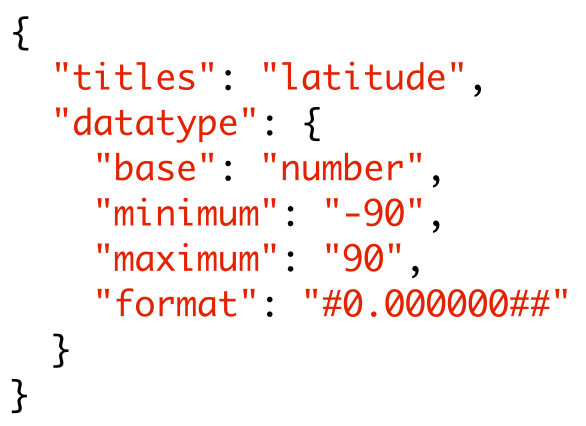
- *pattern* [UAX35]
    - Picture Strings
        - ‘000.0%'
        - ‘###0.#####'
        - ‘#0.0#E+#0'
        - ‘#,00,000'
        - ‘#0.0#,#’
- *decimalChar*
- *groupChar*



```
{
  "titles": "latitude",
  "datatype": {
    "base": "number",
    "minimum": "-90",
    "maximum": "90",
    "format": "#0.000000##"
  }
}
```



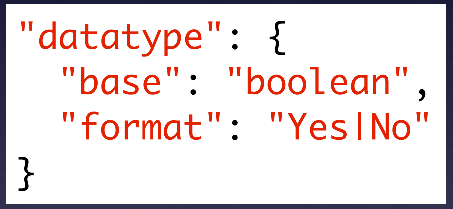
```
"datatype": {
  "base": "integer",
  "format": {
    "decimalChar": ",",
    "groupChar": " ",
    "pattern": "# ##0,0#"
  }
}
```



```
{
  "titles": "latitude",
  "datatype": {
    "base": "number",
    "minimum": "-90",
    "maximum": "90",
    "format": "#0.000000##"
  }
}
```

Formats for booleans

- “Y|N|
- “true|false”
- “1|0”



```
"datatype": {
  "base": "boolean",
  "format": "Yes|No"
}
```

Formats for dates and times

- *pattern* [UAX35]
  - Picture Strings
    - yyyy-MM-dd e.g., 2015-03-22
    - yyyyMMdd e.g., 20150322
    - dd-MM-yyyy e.g., 22-03-2015
    - d-M-yyyy e.g., 22-3-2015
    - MM-dd-yyyy e.g., 03-22-2015
    - M-d-yyyy e.g., 3-22-2015
    - dd/MM/yyyy e.g., 22/03/2015
    - d/M/yyyy e.g., 22/3/2015
    - MM/dd/yyyy e.g., 03/22/2015
    - M/d/yyyy e.g., 3/22/2015
    - dd.MM.yyyy e.g., 22.03.2015
    - d.M.yyyy e.g., 22.3.2015
    - MM.dd.yyyy e.g., 03.22.2015
    - M.d.yyyy e.g., 3.22.2015
    - HH:mm:ss.S – 1+ trailing “S”
    - HH:mm:ss
    - Hummus
    - HH:mm
    - Hmm
    - yyyy-MM-ddTHH:mm:ss.S
    - yyyy-MM-ddTHH:mm:ss
    - yyyy-MM-ddTHH:mm
    - MM/dd/yyyy HH:mm:ss
    - MM/dd/yyyyX – 1+ trailing “X”

```
"datatype": {
  "base": "date",
  "minimum": "2000-01-01",
  "format": "dd/MM/yyyy"
}
```

Serializations

- JSON – not JSON-LD, but uses similar conventions

- RDF – transformation to the RDF data model, with any available serialization

- ~~XML~~ – XML was in the charter, but no champion emerged to define such a serialization.

- All formats encapsulate provenance information from original table; can be excluded using “minimal” mode.