Conduit Data Formats

Contents

[Article 2](#_Toc486422834)

[Summary 2](#_Toc486422835)

[Example 2](#_Toc486422836)

[Properties 2](#_Toc486422837)

[Custom Properties 6](#_Toc486422838)

[Attributes 7](#_Toc486422839)

[Summary 7](#_Toc486422840)

[Example 7](#_Toc486422841)

[Properties 7](#_Toc486422842)

[Books 8](#_Toc486422843)

[Summary 8](#_Toc486422844)

[Example 9](#_Toc486422845)

[Properties 9](#_Toc486422846)

[Sources 9](#_Toc486422847)

[Summary 9](#_Toc486422848)

[Example 9](#_Toc486422849)

[Properties 11](#_Toc486422850)

# Article

## Summary

An article is the basis of Conduit. Articles must follow a specific format in order to work and display correctly in Conduit, though additional properties may be added and can be referenced through the configuration of other properties and data items.

Articles can be defined statically in a JSON file, create via custom code into a JSON object, or created dynamically from an RSS feed (this dynamic creation will require some configuration in the Source object). Articles are made available in conduit through articles.service, but are also available in the page.controller as $scope.articles.

## Example

{

    "date":"25 May 2017 09:23:86",

"link":"www.example.com/article8675309.html",

"id":"8675309",

"title":"Article Two",

"text":"The text associated with article two",

"images":["www.imgur.com/abc.png","www.imgur.com/xyz.png"],

"selectedImage":0,

"books":[],

"comments":[],

"wasRead":false,

    "inFeed":true,

    "source":"Example Source"

}

## Properties

|  |  |  |
| --- | --- | --- |
| **active** | **Example:** | **true** |
|  | **Type:** | **boolean** |
|  | **Req:** | **Optional** |
|  | **Default:** | **false** |
| This Boolean is used by Conduit to track which article card is currently selected. If true, the card is selected. If false, the article is not selected. Styling is changed accordingly. | | |

|  |  |  |
| --- | --- | --- |
| **activeInBook** | **Example:** | **true** |
|  | **Type:** | **boolean** |
|  | **Req:** | **Optional** |
|  | **Default:** | **false** |
| This Boolean is used by Conduit to uniquely track if an article was selected in the book versus the feed. If true, the ‘active’ styling is switched from the feed to the book. | | |

|  |  |  |
| --- | --- | --- |
| **books** | **Example:** | **["World News", “Local News”]** |
|  | **Type:** | **String array** |
|  | **Req:** | **Optional** |
|  | **Default:** | **An empty array** |
| This property is an array of Strings to represent the books the article is included in. If the array is empty, Conduit will not place the article in any books. Conduit will use this field to track the books the article is in. | | |

|  |  |  |
| --- | --- | --- |
| **build** | **Example:** | **true** |
|  | **Type:** | **boolean** |
|  | **Req:** | **Optional** |
|  | **Default:** | **false** |
| This Boolean is used by Conduit to help reduce loading time. Conduit will not build the card in DOM until build has been set to true. InfiniteScroll will automatically set display to true on articles as the user scrolls and adjusts the filter. | | |

|  |  |  |
| --- | --- | --- |
| **comments** | **Example:** | **["Hello world!", “Here’s my feedback.”]** |
|  | **Type:** | **String array** |
|  | **Req:** | **Optional** |
|  | **Default:** | **An empty array** |
| This property is an array of Strings to represent the comments included with the article. If the array is empty, Conduit will not include any comments with the article. Conduit will use this field to track the comments on the article. | | |

|  |  |  |
| --- | --- | --- |
| **date** | **Example:** | **“25 May 2017 09:23:86”** |
|  | **Type:** | **String** |
|  | **Req:** | **Required** |
|  | **Default:** | **Today’s date** |
| This property is used to display the date the article was created/published and used to filter out articles that are old. | | |

|  |  |  |
| --- | --- | --- |
| **id** | **Example:** | **“8675309”** |
|  | **Type:** | **String, GUID** |
|  | **Req:** | **Required** |
|  | **Default:** | **A randomly generated UUID** |
| This property uniquely identifies this article throughout Conduit. It must be unique, but does not need to fit a specific format. | | |

|  |  |  |
| --- | --- | --- |
| **images** | **Example:** | **["www.imgur.com/abc.png","www.imgur.com/xyz.png"]** |
|  | **Type:** | **String array** |
|  | **Req:** | **Optional** |
|  | **Default:** | **An empty array** |
| This property is an array of urls to article images. If the array is empty, Conduit will recognize this and note that there are no images associated with the project. | | |

|  |  |  |
| --- | --- | --- |
| **inBook** | **Example:** | **true** |
|  | **Type:** | **boolean** |
|  | **Req:** | **Optional** |
|  | **Default:** | **false** |
| This Boolean is used by Conduit to track whether or not an article is being displayed in the book. If true, the article card will appear in the book. If false, the article card will be hidden. | | |

|  |  |  |
| --- | --- | --- |
| **inFeed** | **Example:** | **true** |
|  | **Type:** | **boolean** |
|  | **Req:** | **Optional** |
|  | **Default:** | **true** |
| This Boolean is used by Conduit to track whether or not an article is being displayed in the feed. If true, the article card will appear in the feed. If false, the article card will be hidden. | | |

|  |  |  |
| --- | --- | --- |
| **link** | **Example:** | **“www.example.com/article8675309.html”** |
|  | **Type:** | **String** |
|  | **Req:** | **Optional** |
|  | **Default:** |  |
| This property links back to the original article. | | |

|  |  |  |
| --- | --- | --- |
| **read** | **Example:** | **false** |
|  | **Type:** | **boolean** |
|  | **Req:** | **Optional** |
|  | **Default:** | **false** |
| This Boolean is used by Conduit to track whether or not an article has been read by the user. This will adjust styling. If true, the styling will appear greyed out. If false, the styling will appear normal. | | |

|  |  |  |
| --- | --- | --- |
| **selectedImage** | **Example:** | **0** |
|  | **Type:** | **int** |
|  | **Req:** | **Optional** |
|  | **Default:** | **0** |
| This property represents the image that will display on first load, and then which image has been selected by the user. It should be an int less than the length of the image array. 0 is default and recommended. | | |

|  |  |  |
| --- | --- | --- |
| **source** | **Example:** | **“Example Source”** |
|  | **Type:** | **String** |
|  | **Req:** | **Required** |
|  | **Default:** |  |
| This property identifies the source the article came from. This helps conduit figure out how to treat the article and deal with custom properties. | | |

|  |  |  |
| --- | --- | --- |
| **tags** | **Example:** | **“["Topic 1", "Topic 3"]”** |
|  | **Type:** | **String array** |
|  | **Req:** | **Optional, Recommended** |
|  | **Default:** | **An empty array** |
| This property is an array of tags that will be displayed on article cards. Conduit can be told how to populate this array by configuring sources.json. The article’s source is automatically included as the first tag. | | |

|  |  |  |
| --- | --- | --- |
| **text** | **Example:** | **“Lorem Ipsum”** |
|  | **Type:** | **String** |
|  | **Req:** | **Optional, Recommended** |
|  | **Default:** | **A randomly generated UUID** |
| This property is the article’s body that is displayed in the card and viewer. The article will work without this property, but the user will not see anything in the text fields. | | |

|  |  |  |
| --- | --- | --- |
| **title** | **Example:** | **“Article Two”** |
|  | **Type:** | **String** |
|  | **Req:** | **Optional, Recommended** |
|  | **Default:** | **A randomly generated UUID** |
| This property is the article’s title that is displayed in the card and viewer. The article will work without this property, but the user will not see anything in the title fields. | | |

## Custom Properties

Conduit supports custom properties for articles. How to populate these properties and how Conduit should handle them is defined in the sources.json file. A sample of an article with custom data is shown below.

{

    "date":"25 May 2017 09:23:86",

"link":"www.example.com/article8675309.html",

"id":"8675309",

"title":"Article Two",

"text":"The text associated with article two",

"images":["www.imgur.com/abc.png","www.imgur.com/xyz.png"],

"selectedImage":0,

"tags":["Example", "USA", "EUR", "Topic 1", "Topic 3"],

"topics":["Topic 1", "Topic 3"],

"region":[{"abrv":"USA","name":"United States"}, {"abrv":"EUR","name":"Europe"}],

"books":[],

"comments":[],

"wasRead":false,

    "inFeed":true,

    "source":"Example Source"

}

# Attributes

## Summary

Attributes are used to represent distinct properties, actions, or states of articles in Conduit. In its vanilla implementation, Conduit has attributes to represent articles that have images or comments or that have been removed or commented on.

Attributes are defined in the file attributes.json and made available in Conduit as attributes.service.

## Example

{

    "name":"Comments",

    "alt":"Article has comments",

    "glyphicon":"comment",

    "compare":"comments",

    "filter":false,

    "action":true,

    "bookAction":true,

    "checked":false

}

## Properties

|  |  |  |
| --- | --- | --- |
| **action** | **Example:** | **true** |
|  | **Type:** | **boolean** |
| This property is used to highlight important attributes, usually actions. If true, actions will appear at the top left of the cards in the primary color. If false, the attributes will appear with the tags. | | |

|  |  |  |
| --- | --- | --- |
| **alt** | **Example:** | **“Article has comments”** |
|  | **Type:** | **String** |
| This property is used to define the alternate/title text for the attribute, which will appear when the user hovers over the attribute icon. | | |

|  |  |  |
| --- | --- | --- |
| **bookAction** | **Example:** | **true** |
|  | **Type:** | **boolean** |
| This property is used to identify whether or not this action attribute will be visible in cards that are in the book. If true, the attribute will appear in the book. If false, the attribute will not appear in the book (but will still appear in the feed). | | |

|  |  |  |
| --- | --- | --- |
| **checked** | **Example:** | **true** |
|  | **Type:** | **boolean** |
| This property is used to determine the default state for the attribute in the filter. If true, the filter for this attribute will be turned on (and only articles with the attribute will be shown). If false, the filter will not be turned on. | | |

|  |  |  |
| --- | --- | --- |
| **compare** | **Example:** | **“comments”** |
|  | **Type:** | **String** |
| This property is used to identify which property of an article will be used to determine whether or not to display the attribute icon. In this example, the comments attributes icon will only be shown if the comment property of an article is not undefined. | | |

|  |  |  |
| --- | --- | --- |
| **filter** | **Example:** | **true** |
|  | **Type:** | **boolean** |
| This property is used to identify whether or not this attribute will appear in the filter. If true, the user will be able to use the filter to show only products that have the attribute. If false, this option will not be available. | | |

|  |  |  |
| --- | --- | --- |
| **glyphicon** | **Example:** | **“comment”** |
|  | **Type:** | **String** |
| This property is used to identify which bootstrap glyphicon will be used. Only the last unique portion of the glyphicon needs to be identified. Conduit will automatically create the span for it. | | |

|  |  |  |
| --- | --- | --- |
| **name** | **Example:** | **“Comments”** |
|  | **Type:** | **String** |
| This property is used to uniquely identify each attribute. | | |

# Books

## Summary

Books are simply used to populate the books dropdown on the right side of the screen.

Books are defined in the file books.json and made available in Conduit as books.service.

## Example

{"name":"Word News"}

## Properties

|  |  |  |
| --- | --- | --- |
| **name** | **Example:** | **“World News”** |
|  | **Type:** | **String** |
| This property is used to create the name that will be shown in the books dropdown. | | |

# Sources

## Summary

Sources is the core location where the developer can tell Conduit how to handle data. It identifies where the data comes from, what data type it is, how the source data should be bound to Conduit, and how the data should display and act within conduit.

Attributes are defined in the file sources.json and made available in Conduit as sources.service.

## Example

{

    "tag":"NWS VA",

    "name":"National Weather Service VA",

    "link":"https://alerts.weather.gov/cap/va.php?x=1",

    "type":"rss",

    "binding":[{

            "local":"date",

            "source":"published"

        },{

            "local":"link",

            "source":"link"

        },{

            "local":"title",

            "source":"title"

        },{

            "local":"text",

            "source":"summary"

        },{

            "local":"cap:event",

            "source":"cap:event"

        },{

            "local":"cap:urgency",

            "source":"cap:urgency"

        },{

            "local":"cap:severity",

            "source":"cap:severity"

        },{

            "local":"cap:certainty",

            "source":"cap:certainty"

        }

    ],

    "tags":["cap:event","cap:urgency","cap:severity","cap:certainty"],

    "checked":true,

    "filter":[{

        "name":"Event",

        "binding": {

            "property":"cap:event"

        },

        "values":[],

        "checked":false

    },{

        "name":"Urgency",

        "binding": {

            "property":"cap:urgency"

        },

        "values":[],

        "checked":false

    },{

        "name":"Severity",

        "binding": {

            "property":"cap:severity"

        },

        "values":[],

        "checked":false

    }]

}

## Properties

|  |  |  |
| --- | --- | --- |
| **binding** | **Example:** | ***(see below)*** |
|  | **Type:** | **Object array** |
|  | **Req:** | **Required** |
|  | **Default:** | **A randomly generated UUID** |
| This property contains an array of JSON objects. Each object will correlate a “local” property value to a “source” property value. In essence, this tells Conduit how to translate the source data into a format that it knows. | | |
| **Properties:**   |  |  |  | | --- | --- | --- | | **local** | **Example:** | **“date”** | |  | **Type:** | **String** | |  | **Req:** | **Required** | |  | **Default:** |  | | This identifies the Conduit-known article property (see Articles for more information). Custom properties can be configured here simply be defining them. Conduit will add any properties that it does not recognize to the article object so that they can be referenced by other custom configuration or code. | | | | **source** | **Example:** | **“published”** | |  | **Type:** | **String** | |  | **Req:** | **Required** | |  | **Default:** |  | | This identifies the source’s property that should be bound to the Conduit-known article property. | | | | | |

|  |  |  |
| --- | --- | --- |
| **checked** | **Example:** | **true** |
|  | **Type:** | **boolean** |
|  | **Req:** | **Optional** |
|  | **Default:** | **true** |
| This property is used to determine the default state for the source in the filter. If true, the filter for this attribute will be source on (and the articles in that source will be shown). If false, the filter will not be turned on (and the articles will be hidden). | | |

|  |  |  |
| --- | --- | --- |
| **filter** | **Example:** | ***(see below)*** |
|  | **Type:** | **Complex object array** |
|  | **Req:** | **Required** |
|  | **Default:** |  |
| This property contains an array of JSON objects. Each object will define a filter for this source and identify how it will be bound to the article. | | |
| **Properties:**   |  |  |  | | --- | --- | --- | | **name** | **Example:** | **“Event”** | |  | **Type:** | **String** | |  | **Req:** | **Required** | |  | **Default:** |  | | This defines the name of the filter, as it will appear in the filter above the drop down. | | | | **binding** | **Example:** | ***(see below)*** | |  | **Type:** | **Object** | |  | **Req:** | **Required** | |  | **Default:** |  | | This tells the filter how where it should get the properties to populate the filter options, and to actually filter out articles. Be default, only the “property” property (why did I do this to myself?) is required. However, Conduit supports a special format that allows the data to be displayed to the user differently than it is tracked in Conduit. The following properties will follow the example below of a state lookup.  //not all code shown  "name":"State"  "binding": {      "property":"state",      "data":"abrv", //=> VA      "name":"fullName" //=> Virginia   }   |  |  |  | | --- | --- | --- | | **data** | **Example:** | **“abrv”** | |  | **Type:** | **String** | |  | **Req:** | **Optional** | |  | **Default:** |  | | The data that is known to Conduit, placed in tags, actually searched in the filter, etc. | | | | **name** | **Example:** | **“fullName”** | |  | **Type:** | **String** | |  | **Req:** | **Optional** | |  | **Default:** |  | | Additional data that Conduit can use to display the data to the user in a different way. | | | | **property** | **Example:** | **“state”** | |  | **Type:** | **String** | |  | **Req:** | **Required** | |  | **Default:** |  | | This property is required for all filter bindings. It references which property in the article object Conduit will use when filter. It can be a custom object as defined in source.binding. | | | | | | | | |

|  |  |  |
| --- | --- | --- |
| **link** | **Example:** | **“https://alerts.weather.gov/cap/va.php?x=1”** |
|  | **Type:** | **String** |
|  | **Req:** | **Required** |
|  | **Default:** |  |
| This property defines the url at which the articles reside. This will usually point to an API call, but can point directly to a file. | | |

|  |  |  |
| --- | --- | --- |
| **name** | **Example:** | **“National Weather Service VA”** |
|  | **Type:** | **String** |
|  | **Req:** | **Required** |
|  | **Default:** |  |
| This property defines the name of the source as it will be shown in the filter. | | |

|  |  |  |
| --- | --- | --- |
| **tag** | **Example:** | **“NWS VA”** |
|  | **Type:** | **String** |
|  | **Req:** | **Required** |
|  | **Default:** |  |
| This property defines a shortened name of the source, as will be shown in tags. | | |

|  |  |  |
| --- | --- | --- |
| **type** | **Example:** | **“rss”** |
|  | **Type:** | **String** |
|  | **Req:** | **Required** |
|  | **Default:** |  |
| This property lets Conduit know how to deal with the data at the source link (rss would be handled differently than JSON).  Currently only “rss” is supported automatically. To work with another source type, set this to “custom” and build out custom code in articles.service.  Automatic support for JSON will be coming soon. | | |