NinJS User Guide

ninjs User Guide

1. Introduction: ninjs is News in JSON

ninjs standardises the representation of news content in JSON, a lightweight, easy-to-parse data interchange format.

We have identified key properties and structures required to represent news and publishing information, and crafted JSON representations that are designed to result in lightweight instance documents. While keeping it simple, we have been careful to map ninjs concepts to the IPTC News Architecture, so you can rest assured that ninjs concepts can interoperate with other markup languages such as NewsML-G2 (https://iptc.org/standards/newsml-g2/).

We have created some tools to help get users started. These include a simple ninjs generator (https://www.iptc.org/std/ninjs/generator/) that shows how easily a real ninjs document can be created, and a JSON schema to validate ninjs objects.

The latest version of the standard is 3.1. Version 3.1 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_3.1.json) was made available in June 2025.

The latest version in the 2.x branch is 2.2, which was made available in June 2025. See below for more information on the differences between the 1.x and 2.x branches.

The latest version in the 1.x branch is 1.6, which was also made available in June 2025. See below for more information on the differences between the 1.x and 2.x branches.

1.1. Built for both APIs and data at rest

In creating ninjs, we focused on two main use cases:

1. **Data interchange - such as via APIs**

   - ninjs documents can be very concise, allowing for the inclusion of just the most important properties - such as byline, headline, and body text - as determined by a given provider.

   - ninjs allows the provider to indicate whether there are more properties available using the `representationtype: "partial"` construct.

   - Version 3.x of ninjs has been created to work well with both GraphQL and OpenAPI/Swagger, the two main standards for JSON-based APIs.

   - Versions 2.x and 3.x of the ninjs schema have been created in such a way that they can easily be converted into a schema for a binary data serialisation protocol such as Protocol Buffers (https://developers.google.com/protocol-buffers), Avro (https://avro.apache.org/), Thrift (https://thrift.apache.org/) or CBOR (https://cbor.io/) / CDDL (https://datatracker.ietf.org/doc/html/rfc8610). We include a Protocol Buffers schema in the GitHub repository (https://github.com/iptc/newsinjson/tree/main/protobufs) as an example.

2. **Data at rest - such as in search engines or content management systems**

   - ninjs documents can convey a rich set of news publishing properties, such as information that is necessary for the pre-publishing workflow or custom properties that a particular provider wants to express in a standard way.

   - ninjs is designed to be customisable by a particular provider, so that they can express just what they require, without unnecessary overhead. ninjs is suitable for use in JSON-native engines, such as MongoDB (http://www.mongodb.org/) or Elasticsearch (http://www.elasticsearch.org/).

1.2. The ninjs data model in a nutshell

As the below diagram shows, every ninjs object describes an **Item** - either a **News Item**, **Planning Item** or **Event**. All ninjs items must have a unique ID given by a **URI**. They can have administrative metadata (describing who created the news item and when), descriptive metadata (describing **people**, **places**, **genres**, **subjects**, **organisations** and **events** related to the item) and rights-based metadata (**copyright**, **licensing** and **usage terms**).

The content of the item can be a **body**, containing text, HTML or other representation of the text of a news item, **renditions** of the news item as images, videos or multimedia, or both body and renditions.

Associations are a means of linking ninjs items together.
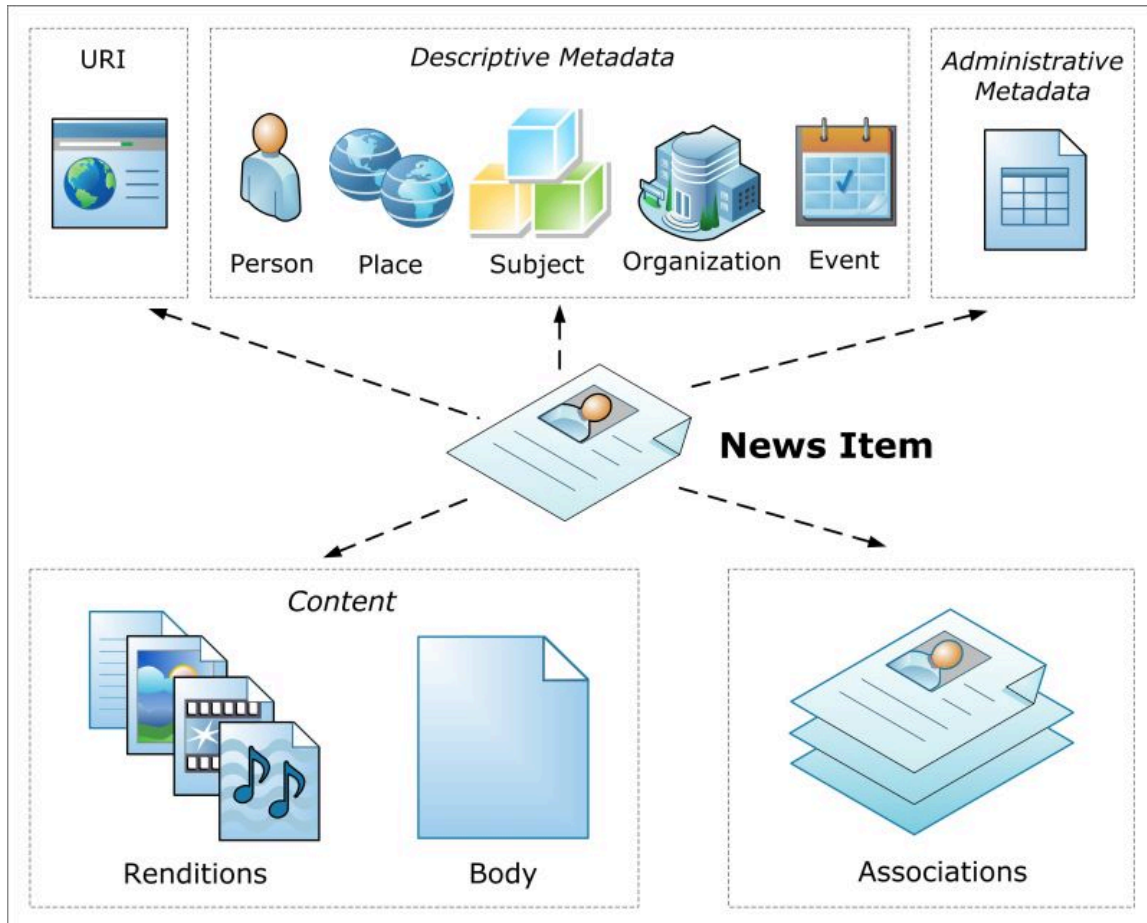
1.3. ninjs object building blocks
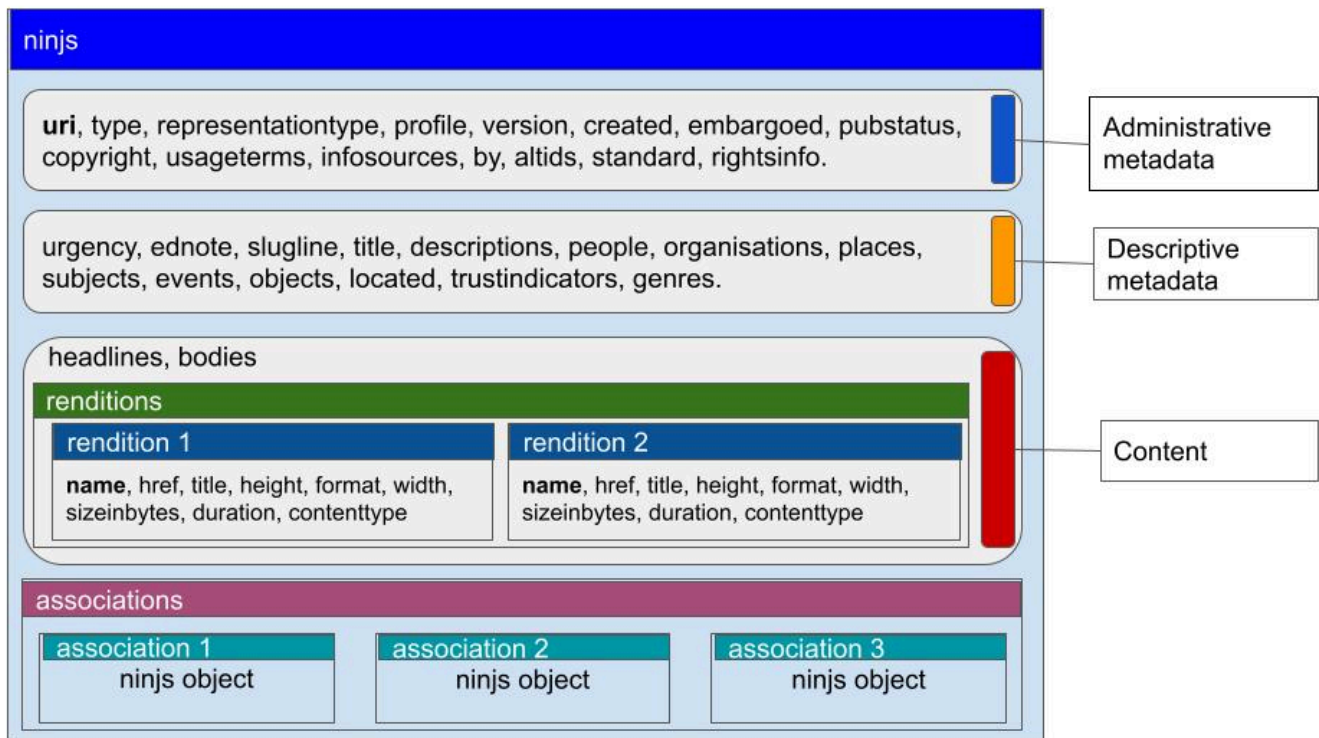
1.4. A simple example

ninjs can be very simple. This is an example of something that might be returned by an API call.

| ninjs 3.1 | ninjs 2.2 | ninjs 1.6 |

```json
{
  "uri" : "http://ninjs.example.com/newsitems/20130709simp123",
  "type" : "text",
  "versionCreated" : "2013-07-09T10:37:00Z",
  "by" : "Paulo Santalucia and Frances d'Emilio",
  "headlines" : [
    {
      "role": "main",
      "value": "Captain of wrecked cruise ship on trial in Italy"
    }
  ],
  "bodies" : [
    {
      "role": "main",
      "value": "GROSSETO, Italy (AP) -- The trial of the captain of the shipwrecked Costa Concordia cruise liner
has begun in a theater converted into a courtroom in Tuscany to accommodate all the survivors and relatives of the
32 victims who want to see justice carried out in the 2012 tragedy. The sole defendant, Francesco Schettino, his
eyes shaded by sunglasses and slipping into a back door, made no comment to reporters as he arrived for his trial
Tuesday on charges of multiple manslaughter, abandoning ship and causing the shipwreck near the island of Giglio.
His lawyer, Domenico Pepe, told reporters that, as expected, the judge was postponing the hearing due to an eight-
day nationwide lawyers' strike. Pepe said some 1,000 witnesses are expected to eventually testify. Many of them are
expected to be from among the 4,200 passengers and crew aboard the ship that struck a jagged reef off Giglio, took
on water and capsized. Schettino has denied wrongdoing."
    }
  ]
}
```

You can make your own ninjs documents for demonstration or testing purposes using the ninjs generator tool on iptc.org (https://www.iptc.org/std/ninjs/generator/).

1.5. Versions of ninjs: ninjs 3.1, 2.2 and 1.6

IPTC manages three versions of ninjs: the 3.x branch (where the latest version is 3.1), the 2.x branch (where the latest version is 2.2) and the 1.x branch (where the latest version is 1.6).

The 1.x series continues the version of ninjs that started with ninjs 1.0 in 2013, adding new features in a way that is fully backwards-compatible. The latest release in this series is 1.6, which adds `digitalsourcetype`.

The 2.x series fixes some issues that made the 1.x series tricky for some developers to work with. The main change is that ninjs 2.x avoids the use of patternProperties, so all properties now have well-defined names. This change is intended to make ninjs easier for projects based on binary protocols such as Protocol Buffers and Avro. We also took the opportunity to make property names more consistent with NewsML-G2 and to add some fields to handle rights.

The 3.x series includes support for news planning and events, including live events. It also adds support for live streamed renditions. In a major change, we also change the case of all ninjs properties to "camelCase" format, in keeping with JSON best practices, especially regarding GraphQL usage. ninjs 3.x was created alongside the LPX (Live Production Exchange) project from the DPP (https://www.thedpp.com/lpx), with the help of experts from CNN, Reuters, Arqiva and more.

All new users are strongly recommended to use version 3.1.

ninjs 3.1, 2.2 and 1.6 were approved by the IPTC Standards Committee on May 16th, 2025.

This document focuses on the 3.1 version, but includes notes where the 2.x and 1.x versions differ. Those who must use version 2.x or 1.x for backwards compatibility reasons can find the JSON Schema file for ninjs 2.2 at https://iptc.org/std/ninjs/ninjs-schema_2.2.json and the schema file for ninjs 1.6 at https://iptc.org/std/ninjs/ninjs-schema_1.6.json.

1.6. Differences between ninjs 1.x, 2.x and 3.x

We maintain the 1.x and 2.x branches for those who can't move to the new 3.x version.

1.6.1. patternProperties vs arrays

In the ninjs 1.x schema, some properties are implemented as "patternProperties", which means that the name must start with a certain text string (such as `headline_`) but can end in any string (such as `subhead`). This was used to distinguish objects from each other.

This can cause problems with binary serialisations and queries based on the schema, including Protocol Buffers, Avro, CBOR and GraphQL which all require well-known, pre- defined property names.

Therefore, in the 2.x and 3.x versions of the schema, we instead use simple arrays with well-defined names, and a `role` property that defines the type of the value.

For example, a subheadline is defined in ninjs 1.x as

```
                                                                    NINJS 1.X JSON
  {
    "headline_subhead": "My subheadline"
  }
```

The same subheadline would be defined in ninjs 2.x or 3.x as

```
                                                               NINJS 2.X OR 3.X JSON
  {
    "headlines": [
      {
        "role": "subhead",
        "value": "My subheadline"
      }
    ]
  }
```

This applies to the following properties:

- `description_XXX` (1.x) / `descriptions` (2.x, 3.x)

- `body_XXX` (1.x) / `bodies` (2.x, 3.x)

- `headline_XXX` (1.x) / `headlines` (2.x, 3.x)

1.6.2. Renditions and Associations converted to arrays

The `renditions` and `associations` properties are objects in ninjs 1.x, using the object name as the distinguishing element. This caused similar problems, so we use arrays in ninjs 2.x and 3.x.

For example, in ninjs 1.x a rendition may look like the following:

```
                                                                  NINJS 1.X JSON
  {
    "renditions": {
      "highres": {
        ...
      }
    }
  }
```

The same rendition in ninjs 2.x or 3.x would look like:

```
                                                              NINJS 2.X OR 3.X JSON
  {
    "renditions": [
      {
        "name": "highres",
        ...
      }
    ]
  }
```

Associations work the same way in all ninjs versions.

1.6.3. Property names converted to plural form

In addition, the following property names were converted to plural form for ninjs 2.x. They were not changed in any other way.

In ninjs 3.x, property names were converted to camelCase.

ninjs 1.x	ninjs 2.x	ninjs 3.x
altid	altids	altIds
event	events	events
genre	genres	genres
infosource	infosources	infoSources
object	objects	objects
organisation	organisations	organisations
person	people	people
place	places	places
subject	subjects	subjects
trustindicator	trustindicators	trustIndicators

1.7. Which version of ninjs should I choose for my project?

If you are starting a green-field project, we recommend using the latest version in the ninjs 3.x branch, currently ninjs 3.1. This version should be easiest for developers to work with.

If you are already using a 1.x or 2.x version of ninjs, we recommend at least upgrading to the latest version in that branch, currently 1.6 and 2.2. This should be an easy change, because all new minor versions in a branch are backwards-compatible with earlier minor versions in the same branch.

1.8. ninjs and other standards

JSON is a popular alternative to (http://blog.programmableweb.com/2011/05/25/1-in-5-apis-say-bye-xml/) (or replacement for (http://blog.appfog.com/why-json-will-continue-to-push-xml-out-of-the-picture/)) XML. ninjs is designed in such a way that it can be used either standalone or alongside other IPTC news markup formats (https://iptc.org/standards/) such as NewsML-G2 (https://iptc.org/standards/newsml-g2/), NITF (https://iptc.org/standards/nitf/), IPTC7901 (https://iptc.org/standards/iptc-7901/) and rNews (https://iptc.org/standards/rnews/).

2. About the IPTC ninjs User Guide

2.1. Copyright

Copyright © 2025 IPTC, International Press Telecommunications Council.

The IPTC ninjs User Guide document is published under the Creative Commons Attribution 4.0 license (http://creativecommons.org/licenses/by/4.0/) - see the full license agreement at http://creativecommons.org/licenses/by/4.0/.

By obtaining, using and/or copying this document, you (the licensee) agree that you have read, understood, and will comply with the terms and conditions of the license.

Materials used in this guide are either in the public domain or are available with the permission of their respective copyright holders. All materials of this IPTC standard covered by copyright shall be licensable at no charge.

2.2. Acknowledgements

This document is the result of a team effort by members of the News in JSON Working Group of the International Press Telecommunications Council (IPTC), with input and assistance from other contributors.

Contributors to this document include: Paul Harman (Bloomberg), Johan Lindgren (TT), Stuart Myles (previously with AP), Brendan Quinn (IPTC), Michael Steidl (previously with IPTC), Evan Sandhaus (previously with NYT), and Ian Young (PA Media / Alamy).

2.3. How to contact IPTC

All users of ninjs are encouraged to join the public IPTC ninjs discussion group: https://groups.io/g/iptc-ninjs/

Staff from organisations who are members of the IPTC are welcome to join the members-only discussion list for the ninjs development team (https://groups.io/g/iptc-news-in-json-dev).

For questions, comments or suggestions, you can raise an issue on our public GitHub repository: https://github.com/iptc/newsinjson/issues

You can also submit a message on our website: https://iptc.org/about-iptc/contact-us/

Visit the ninjs section on IPTC's website: https://iptc.org/standards/ninjs/

To stay up to date on developments with ninjs and other areas, follow IPTC on Twitter: @IPTC (http://www.twitter.com/IPTC/)

2.4. About IPTC

The IPTC, based in London, brings together the world's leading news agencies, publishers and industry vendors. It develops and promotes efficient technical standards to improve the management and exchange of information between content providers, intermediaries and consumers. IPTC standards enable easy, cost-effective and rapid innovation and include the IPTC Photo Metadata standard, the Video Metadata Hub, the news exchange formats NewsML-G2, ninjs, SportsML-G2 and NITF, rNews for marking up online news, the rights expression language RightsML, and NewsCodes taxonomies for categorising news.

IPTC is a not-for-profit membership organisation registered in England. Find more about joining IPTC (https://iptc.org/participate/).

Business address:

```
IPTC International Press Telecommunications Council
25 Southampton Buildings
London WC2A 1AL
United Kingdom
```

3. ninjs in Detail

Here we document each property that can be used in a ninjs object.

The "uri" property is **required** at the top level. In some array objects, other properties may be required; for example, if `rightsInfo` is used, then either `linkedRights` or `encodedRights` is required.

These tables primarily describe the ninjs 3.1 schema. Where possible, we note differences in the 1.x and 2.x schemas, but the ultimate reference for each version of ninjs is the relevant JSON Schema file.

3.1. Administrative properties

These properties concern the structure, type and creation details of the content.

ninjs 3.1 property	ninjs 1.x / 2.x property	Description	Data type
`uri`	`uri`	The identifier for this news object	URI
`type`	`type`	The generic news type of this news object	One of: `text` `audio` `video` `picture` `graphic` `composite` `component` `event` (3.x only) `planning` (3.x only)
`title`	`title`	A short natural-language name for the item.	string

ninjs 3.1 property	ninjs 1.x / 2.x property	Description	Data type
standard	standard (2.x) $standard (1.x)	An object with information about standard, version and schema this instance is valid against. This property is an object with the following child properties: | Property name	Description	Data type | |---|---|---| | name | The name of the standard, such as ninjs. | string | | version | The version of the standard used, such as 3.1. | string | | schema | The uri of the JSON schema to use for validation, such as https://www.iptc.org/std/ninjs/ninjs-schema_3.1.json | URI |	object
altIds	altids (2.x) altid (1.x)	Alternative identifiers for the item. It is up to the individual provider to name and set type on the alternative identifiers they like to use. In ninjs 2.x and 3.x, this property is an object with the following child properties: | Property name	Description	Data type | |---|---|---| | role | The role of the alternative id (such as "internal") | string | | value | The alternative id value | string |	array of objects (2.x, 3.x) JSON object (1.x)
representationType	representationtype	Indicates the completeness of this representation of a news item. For a full document representation, the value should be full. For a partial representation, for example a short extract provided as the results of an API call, the value partial should be used. NOTE: in ninjs 1.x, the allowed values are complete and incomplete.	One of: full partial
profile	profile	An identifier for the structure of the news object. This can be any string but we suggest something identifying the structure of the content such as 'text-only' or 'text-photo'. Profiles are typically provider-specific.	string
version	version	The version of the news object which is identified by the uri property.	string
firstCreated	firstcreated	Indicates when the first version of the item was created.	date-time

ninjs 3.1 property	ninjs 1.x / 2.x property	Description	Data type
versionCreated	versioncreated	The date and time when this version of the news object was created.	date-time
contentCreated	contentcreated	The date and time when the content of this ninjs object was originally created.	date-time
embargoed	embargoed	The date and time before which all versions of the news object are embargoed. If absent, this object is not embargoed.	date-time
expires	expires	The date and time after which the Item is no longer considered editorially relevant by its provider. nar:expires (Added in ninjs 2.1)	date-time
pubStatus	pubstatus	The publishing status of the news object, its value is `usable` by default.	One of: `usable` `withheld` `canceled`
urgency	urgency	The editorial urgency of the content from 1 to 9. 1 represents the highest urgency, 9 the lowest.	integer (1-9)
edNote	ednote	A note that is intended to be read by internal staff at the receiving organisation, but not published to the end-user.	string
language	language	The human language used by the content. The value should follow IETF BCP47 (https://tools.ietf.org/html/bcp47) format.	string

3.2. Rights properties

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
copyrightHolder	copyrightholder	The person or organisation claiming the intellectual property for the content.	string
copyrightNotice	copyrightnotice	Any necessary copyright notice for claiming the intellectual property for the content.	string
usageTerms	usageterms	A natural-language statement about the usage terms pertaining to the content.	string
digitalSourceType	digitalsourcetype	A natural-language statement about the usage terms pertaining to the content.	string [

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
`rightsInfo`	`rightsinfo`	An expression of rights to be applied to the content. This property is an object with the following child properties: | Property name	Description	Data type
---	---	---
`langId` (`langid` in 1.x and 2.x)	Identifier for the used Rights Expression language.	uri
`linkedRights` (`linkedrights` in 1.x and 2.x)	A link from the current Item to Web resource with rights related information.	uri
`encodedRights` (`encodedrights` in 1.x and 2.x)	Contains a rights expression as defined by a Rights Expression Language.	string | For each array element under rightsinfo, one of either `linkedrights` or `encodedrights` is **required**.	Array of objects

3.3. Editorial properties

Properties representing the actual story that will be shown to the audience.

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
`headlines`	`headlines` (2.x) `headline_<type>` (1.x)	(in 2.x and 3.x) An array of objects containing various types of headlines. This property is an object with the following child properties: | Property name	Description	Data type
---	---	---
`role`	The role of this headline	string
`contentType` (contenttype in 2.x)	The IANA (Internet Assigned Numbers Authority) Media Type (https://www.iana.org/assignments/media-types/media-types.xhtml) of the content of this headline. (Previously known as MIME Type.)	string
`value`	The headline identified with the above role and contenttype. Required in each object in the array.	string		Array of objects
`slugline`	`slugline`	A human-readable identifier for the item.	string
`by`	`by` (2.x) `byline` (1.x)	The name(s) of the creator(s) of the content	string

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
`bodies`	`bodies` (2.x) `body_<type>` (1.x)	An array of body objects with the content as text or with markup. This property is an object with the following child properties: {{BODIES_SUBTABLE}}	Array of objects
`descriptions`	`descriptions` (2.x) `description_<type>` (1.x)	An array of one or more descriptions of the ninjs object. See also ednote for information from provider to receiver. Descriptions are seen as metadata. This property is an object with the following child properties: {{DESCRIPTIONS_SUBTABLE}}	Array of objects

Bodies subtable:

Property name	Description	Data type
`role`	The role of this body	string
`contentType` (`contenttype` in 2.x)	The IANA (Internet Assigned Numbers Authority) Media Type (https://www.iana.org/assignments/media-types/media-types.xhtml) of the content of this body. (Previously known as MIME Type.)	string
`value`	The headline identified with the above role and contentType. **Required** to exist in each array item.	string
`charCount` (charcount in 2.x)	The total character count in the article excluding figure captions.	number
`wordCount` (wordCount in 2.x)	The total number of words in the article excluding figure captions.	number

Descriptions subtable:

Property name	Description	Data type
`role`	The role of this description	string
`contentType` (`contenttype` in 2.x)	The IANA (Internet Assigned Numbers Authority) Media Type (https://www.iana.org/assignments/media-types/media-types.xhtml) of the content of this description. (Previously known as MIME Type.)	string
`value`	The descriptive text identified with the above role and contenttype. Required in each object in the array.	string

3.4. Metadata properties

ninjs allows content to be marked up with one simple string-based metadata property `located`, plus several rich metadata properties that can link to pre-defined controlled vocabularies describing people, organisations, places, subjects, objects and events.

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
`located`	`located`	The name of the location from which the content originates. See also `places` for locations which are mentioned in the content.	string
`genres`	`genres` (2.x) `genre` (1.x)	The nature, intellectual or journalistic form of the content. This property is an array of objects with the following properties: | Property name	Description	Data type
---	---	---
`name`	The human-readable name of the genre, such as `Press Release`.	string
`uri`	The identifier of the genre as a complete uri	URI
`literal`	The code for the genre as a string literal	string		array

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
people	people (2.x) person (1.x)	An array of objects describing individual human beings. This property is an array of objects with the following properties: | Property name	Description	Data type
---	---	---
name	The name of a person	string
rel	The relationship of the content of the news object to the person	string
uri	The identifier for the person as a complete uri with the code.	URI
literal	The code for the person as a literal value.	string
contactInfo (contactinfo in 2.x)	Contact and/or location information for the person as a ContactInfo structure.	Contact Info structure		array
organisations	organisations (2.x) organisation (1.x)	An array of objects describing administrative and functional structures which may act as a business, as a political party or not-for-profit party. This property is an array of organisation objects.	array

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
`places`	`places` (2.x) `place` (1.x)	An array of named locations. This property is an array of objects with the following properties: 	array

Property name	Description	Data type
`name`	The name of the place	string
`rel`	The relationship of the content of the news object to the place	string
`uri`	The identifier for the place as a complete uri	URI
`literal`	The code for the place as a literal	string
`contactInfo` (`contactinfo` in 2.x)	Contact and/or location information for the place as a ContactInfo structure.	Contact Info structure
`geoJSON` (`geojson` in 2.x, `geometry_<type>` in 1.x)	An object holding geographic data of this place. From ninjs 2.0 onwards, the contents of this object must conform to the GeoJSON format (https://geojson.org/schema/GeoJSON.json) defined in RFC 7946 (https://datatracker.ietf.org/doc/html/rfc7946).	object

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
subjects	subjects (2.x) subject (1.x)	An array of objects holding concepts with a relationship to the content. This property is an array of objects with the following properties: | Property name	Description	Data type | |---|---|---| | name | The name of the subject | string | | rel | The relationship of the content of the news object to the subject | string | | uri | The identifier of the subject as a complete uri | URI | | literal | The code for the subject as a string literal | string | | creator | Specifies which entity (person, organisation or system) that has created or last edited the property. | string | | relevance | The relevance of the metadata to the news content to which it is attached. | integer (range 0-100) | | confidence | The confidence with which the metadata has been assigned. | integer (range 0-100) |	array

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
events	events (2.x) event (1.x)	An array of objects describing something which happens in a planned or unplanned manner. This property is an array of objects with the following properties: | Property name	Description	Data type
---	---	---
name	The name of the event	string
rel	The relationship of the content of the news object to the event	string
uri	The identifier for the event as a complete uri	URI
literal	The code for the event as a string literal	string		array
objects	objects (2.x) object (1.x)	An array of objects describing something material, excluding persons. This property is an array of objects with the following properties: | Property name	Description	Data type
---	---	---
name	The name of the object	string
rel	The relationship of the content of the news object to the object	string
uri	The identifier for the object as a complete uri	URI
literal	The code for the object as a string literal	string		array

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
infoSources	infosources (2.x) infosource (1.x)	An array of parties (person or organisation) which originated, modified, enhanced, distributed, aggregated or supplied the content or provided some information used to create or enhance the content. This property is an array of objects with the following properties:	array

Property name	Description	Data type
name	The name of the infosource	string
role	The role the infosource in relationship to the content as a uri.	string
uri	The identifier of the infosource as a complete uri	URI
literal	The code for the infosource as a string literal	string
contactInfo (contactinfo in 1.x and 2.x)	Contact and/or location information for the info source as a ContactInfo structure.	Contact Info structure

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
`digitalSourceType`	`digitalsourcetype`	Indicates the type of the source of this content. Some common uses for this property include: • AI-generated content can be indicated using the value http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia • Content captured using a digital camera can be indicated using http://cv.iptc.org/newscodes/digitalsourcetype/digitalCapture • Content captured using a digital camera with AI enhancement (eg a smart phone) can be indicated using http://cv.iptc.org/newscodes/digitalsourcetype/computationalCapture, • Content created by a human using digital tools (eg a digital artwork or a text article written on a computer) can be indicated using http://cv.iptc.org/newscodes/digitalsourcetype/digitalCreation This property is a single instance of an object with the following properties: <table><tr><th>Property name</th><th>Description</th><th>Data type</th></tr><tr><td>name</td><td>The name of the digital source type.</td><td>string</td></tr><tr><td>uri</td><td>The identifier of the digital source type as a complete URI. Suggested CV: http://cv.iptc.org/newscodes/digitalsourcetype/</td><td>URI</td></tr><tr><td>literal</td><td>An identifier for the digital source type as a free-text string.</td><td>string</td></tr></table>	object
`trustIndicators`	`trustindicator` (1.x) `trustindicators` (2.x)	An array of objects to allow links to documents about trust indicators, intended to show the trustworthiness of the news provider. This property is an array of objects with the following properties: <table><tr><th>Property name</th><th>Description</th><th>Data type</th></tr><tr><td>role</td><td>The role of the trust indicator as a complete uri, such as `http://cv.iptc.org/newscodes/ trustindicator/editorialPolicy`</td><td>URI</td></tr><tr><td>href</td><td>The URL for accessing the trust indicator resource, such as `https://www.bbc.co.uk/editorialguidelines/`</td><td>string</td></tr><tr><td>title</td><td>The name of the resource being referenced, such as `BBC Editorial Guidelines`</td><td>string</td></tr></table>	array

3.4.1. Contact Info structure

The Contact Info structure is used for the contents of the `contactinfo` property in items in the `people`, `organisations`, `places` and `infosources` arrays.

Property name	Description	Data type
type	Type would be a method of communication like phone, mobile, address etc.	string
role	Role refers to type and could be private, office etc	string
lang	If this contactinfo object needs to be qualified with what language it is in. The value should follow IETF BCP47.	string
name	Human readable name of the contact method, like name for a web page, name of persons twitter account etc	string
value	Actual phone number, email address, web url etc.	string
address	The address of a person, place or organisation. This property is an object with the following child properties: <table>	object

The address property contains:

Property name	Description	Data type
lines	An array of lines to construct an address. The order is important to construct a correct address.	array
locality	A city/town/village etc. part of the address.	string
area	A subdivision of a country part of the address.	string
postalCode (postalcode in 1.x / 2.x)	A postal code part of the address.	string
country	A country part of the address.	string

3.4.2. Organisation Structure

The Contact Info structure is used for the contents of the `contactinfo` property in items in the `people`, `organisations`, `places` and `infosources` arrays.

Property name	Description	Data type
name	The name of the organisation	string
rel	The relationship of the content of the news object to the organisation	string

Property name	Description	Data type
`uri`	The identifier of the organisation as a complete uri	URI
`literal`	The code for the organisation as a literal	string
`symbols`	Symbols used for a financial instrument linked to the organisation at a specific market place. Array of objects with the properties: • `exchange` : Identifier for the marketplace which uses the ticker symbols of the ticker property (string) • `symboltype` : URI describing the type of the symbol. The CV http://cv.iptc.org/newscodes/financialinstrumentsymboltype is recommended. • `symbol` : Symbol identifier, including ticker symbols. We recommend using this sub-property. • `ticker` : Ticker symbol used for the financial instrument (string). This sub-property is allowed but deprecated.	array
`contactInfo` (`contactinfo` in 2.x)	Contact and/or location information for the organisation as a ContactInfo structure.	Contact Info structure

4. Events and Planned Coverage

This section applies to ninjs 3.x only.

A news organisation may wish to distribute information about forthcoming events and planned news coverage in advance of the event. In this way, customers of the news organisation (for example, news outlets and broadcasters who are customers of a wire service) can plan to broadcast or republish the coverage, or plan their own coverage around the work of the upstream news provider.

To handle this in ninjs, we introduce events and planned coverage. The event includes information about a real-world event that is being covered in some way. The planning item includes information about news coverage that the provider intends to release at some point in the future.

4.1. Events

A ninjs event conveys information about a real-world event that may (or may not) be covered by a news organisation: a press conference, a sports event, a webinar, an election.

Information about the event may change over time. For example, we know that there will be a Summer Olympic Games in Brisbane in 2032, but we don't yet know the start time of the Opening Ceremony.

While specifying an event, we can include as much or as little information as we have at the time.

Here's an example of a simple event:

| ninjs 3.1 | ninjs 2.2 | ninjs 1.6 |

```json
                                                              JSON
  {
    "standard": {
      "name": "ninjs",
      "version": "3.1",
      "schema": "http://www.iptc.org/std/ninjs/ninjs-schema_3.1.json"
    },
    "uri": "urn:nato.int:2023-nato-summit-press-conference",
    "type": "event",
    "title": "2023 NATO Summit Press Conference",
    "eventDetails": {
      "dates": {
        "expectedStartDate": "2023-07-11"
      }
    },
    "places": [
      {
        "name": "LITEXPO",
        "contactInfo": [
          {
            "type": "physical",
            "address": {
              "locality": "Vilnius",
              "country": "Lithuania"
            }
          }
        ]
      }
    ]
  }
```

The main new object to cover such events is the eventDetails object, which contains the following properties:

Property name	Description	Data type
eventStatus	Status of the event (as opposed to the status of coverage of the event). Suggested CV: http://cv.iptc.org/newscodes/eventstatus/ (which contains values such as "scheduled" and "canceled")	uri
plannedCoverageStatus	Status of the coverage (as opposed to the status of the event itself). Suggested CV: http://cv.iptc.org/newscodes/newscoveragestatus/ (which contains values such as "coverage intended" and "coverage upon request")	uri
dates	Dates and times for this event.	Dates Structure
organiser	A person or organisation organising the event.	Organisation Structure

Events include a rich date structure which allows for uncertain start and end dates and times and also recurring events. The data model is based on EventsML-G2 (https://iptc.org/std/NewsML-G2/specification/#introduction-to-eventsml-g2), part of the NewsML-G2 family of standards.

> ℹ️ It's worth noting that this is different from the events metadata object used for news items. That object is intended to convey metadata about an event that is related to a published news item.

4.2. Planned Coverage

A ninjs item may contain a `plannedCoverage` property which contains an array of planned coverage items using the following structure:

Property name (3.x only)	Description	Data type
uri	The local identifier of the element which MUST be persistent for all versions of the item, i.e. for its entire lifecycle.",	string
title	A human-readable title for the planned content",	string
pubStatus	The publishing status of the plannedCoverage object, its value is **usable** by default. nar:pubStatus	string with possible values "usable", "withheld"

type	The generic news type of the planned content.	One of:
		`text`
		`audio`
		`video`
		`picture`
		`graphic`
		`composite`
		`component`
		`event`
		`planning`

commissioned	Information about the commissioning of this content.	Commissioning Structure
dates	Dates and times for this planned content.	Dates Structure
audiences	Intended audiences for the planned content. Array of objects with the following structure: 	Property name	Description	Data type
audience	Intended audience(s) for the content.	string
significance	A qualifier which indicates the expected significance of the content for this specific audience.	integer
exclAudience	Excluded audience(s) for the planned content.	Array of strings		Array of objects
edNote	Additional natural language information about the planned content addressed to the editorial people receiving and processing the item.	string
urgency	The editorial urgency of the planned content, as scoped by the parent element.	number (1-9)

Property name (3.x only)	Description	Data type
language	The human language used by the planned content. The value should follow IETF BCP47.	string
itemCount`	Number of planned items of this kind expressed by a range. This is an object with the following structure: 	Property name	Description	Data type
rangeFrom	Lower bound of the range.	integer
rangeTo	Upper bound of the range.	integer		object
wordCount	The count of words of textual content.	integer
renditions	An array of objects showing the planned renditions of the news object.	Array of rendition objects

4.2.1. Commissioning Structure

Property name (3.x only)	Description	Data type
by	The person or party that commissioned the content.	string
on	The date at which the content was commissioned.	date-time
references	Identifiers that help with the commissioning process such as purchase orders and work order numbers. Array of objects with the following structure: 	Property name	Description	Data type
name	Name of the reference	string
value	Value of the reference	string		Array of objects

4.3. Dates for Events and Planning

ℹ This section applies to dates in events and news coverage planning items in ninjs 3.x only.

A news organisation may wish to distribute information about forthcoming events and planned news coverage that has not yet been finalised. Therefore dates and times may not yet be known.

Also, events may recur at regular intervals.

To handle this situation, in ninjs 3.x we introduce a `dates` object which contains several properties:

Property name (3.x only)	Description	Data type
startDate	The date and time at which the event starts (if known)	date-time
endDate	The date and time at which the event ends (if known)	date-time
expectedStartDate	The approximate date (and optionally time) at which the event or coverage is expected to start.	Truncated DateTime
expectedEndDate	The approximate date (and optionally time) at which the event or coverage is expected to end.	Truncated DateTime
expectedDuration	The time period that the event or coverage is expected to last. Must use the iCalendar duration format (RFC 5545) (https://www.rfc-editor.org/rfc/rfc5545#section-3.3.6). (Actual duration can be calculated from startDate and endDate)	string
recurrence	Specifies recurrence information about the event. This property is an object with the following child properties:

Property name	Description	Data type
recurrenceDates	Date(s) (and optionally times) on which the event occurs.	array of Truncated DateTime
recurrenceRules	Rules that specify the recurrence of this event. This property is an array of Recurrence Rule objects.	Array of Recurrence Rules

| object |

The above date object can be used for [eventDetails/dates] and [plannedCoverage/dates].

4.3.1. Truncated DateTime

Used in expectedStartDate, expectedEndDate and the recurrenceDates property of Recurrence Rules, the "truncated DateTime" is a property that allows dates to be "truncated from the right", i.e. specified with resolution of only the year, year/month, year/month/day or other combinations.

This is based on the equivalent TruncatedDateTime type in NewsML-G2 (https://www.iptc.org/std/NewsML-G2/guidelines/#TruncatedDateTime) and is implemented in ninjs using a regular expression.

The full list of allowed date/time parts is:

- year only (xs:gYear, YYYY)
- year/month (xs:gYearMonth, YYYY-MM)
- year/month/day (xs:date YYYY-MM-DD)

- full datetime (xs:dateTime, YYYY-MM-DDTHH:MM:SS+HH:MM)

All of these can carry an optional timezone suffix.

Note that this does NOT include ISO8601 month of year (xs:gMonth, --MM) or yearly day (xs:gMonthDay, --MM-DD)

Note that other date objects (startDate, endDate, firstCreated, versionCreated etc) require a fully-specified ISO8601 date and time.

4.3.2. Recurrence Rules

Used to specify recurring events within the recurrenceRules property.

This follows the structure of EventsML-G2 / NewsML-G2 recurrence rules, which in turn follows the iCalendar specification.

Property name (3.x only)	Description	Data type
frequency	The FREQ rule part identifies the type of recurrence rule.	string: one of: secondly minutely hourly daily weekly monthly yearly
interval	The INTERVAL rule part contains a positive integer representing how often the recurrence rule repeats.	integer
until	The UNTIL rule part defines a date-time value which bounds the recurrence rule in an inclusive manner.",	date-time
count	The COUNT rule part defines the number of occurrences at which to range-bound the recurrence.	integer

4.4. Renditions

The `renditions` property is a wrapper for different renditions of content of the news object, such as articles, images, videos or PDF documents.

It contains a set of one or more keys, each with a name and an object as the value. The name can be any text or numbers, and should represent a given rendition type such as "thumbnail" or "preview". The object should include the below properties.

4.4.1. Properties of renditions

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
name	name (2.x) *Name of object* (1.x)	The name of this object in the array of renditions. Required.	string
href	href	The URL for accessing the rendition as a resource.	URI
contentType	contenttype	The IANA Media Type which applies to this rendition.	string

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type
`title`	`title`	A title for the link to the rendition resource.	string
`height`	`height`	For still and moving images: the height of the display area measured in pixels.	integer (3.x) number (1.x, 2.x)
`width`	`width`	For still and moving images: the width of the display area measured in pixels.	integer (3.x) number (1.x, 2.x)
`sizeInBytes`	`sizeinbytes`	The size of the rendition resource in bytes.	integer (3.x) number (1.x, 2.x)
`duration`	`duration`	The total time duration of the content in seconds (fractional seconds are allowed)	number
`format`	`format`	A refinement of a generic content type (i.e. IANA media type) by a literal string value.	string
`aspectRatio`	n/a (3.x only)	Aspect ratio of the video file, which is the ratio of the width of video to its height, such as 16:9 and 4:3.	string
`videoCodec`	n/a (3.x only)	The video encoding system used to create the content. Suggested CV: http://cv.iptc.org/newscodes/videocodec/	string
`frameRate`	n/a (3.x only)	The number of video frames per second, which is the rate at which the material should be shown in order to achieve the intended visual effect	number
`poi`	n/a (3.x only)	The point of interest of the rendition. This property is an object with the following child properties: See table below	object
`transportProtocol`	n/a (3.x only)	The transport/flow type of the flow (B2B stream).	string

Child properties of `poi`:

Property name	Description	Data type
x	The x-coordinate of the point of interest of the rendition.	integer
y	The y-coordinate of the point of interest of the rendition.	integer

ninjs 3.x property	ninjs 1.x / 2.x property	Description	Data type

scanType	n/a
	(3.x only)
		The scan type of the video. The only allowed values are 'progressive' and 'interlaced'.	One of:
			`progressive`
			`interlaced`

bitrate	n/a (3.x only)	The bitrate of the video. Should always include units, eg 400 kbps.	string

4.4.2. Renditions example

| ninjs 3.1 | ninjs 2.2 | ninjs 1.6 |

```json
                                                                    JSON
  {
    "uri": "renditions-example",
    "bodies": [
      {
        "value": "..."
      }
    ],
    "renditions": [
      {
        "name": "thumbnail",
        "href":  "http://mms.businesswire.com/media/newsItemId/en/221373/2/dell_blue_rgb.jpg",
        "contentType": "image/jpeg",
        "height": 70,
        "width": 70,
        "sizeInBytes": 4380
      },
      {
        "name": "highres",
        "href":  "http://mms.businesswire.com/media/newsItemId/en/221373/5/dell_blue_rgb.jpg",
        "contentType": "image/jpeg",
        "height": 432,
        "width": 432,
        "sizeInBytes": 33116
      }
    ]
  }
```

4.5. Associations

The `associations` property contains associated valid ninjs objects, each with a **required** `name` explaining the nature of the association.

This can be used to link news items together, or to associate images with a text story, for example.

4.5.1. Associations example

This example shows a press release. The body of the story is in the single object in the `bodies` property. A logo and an image are associated to the story. Both the logo and the image have two renditions, a "thumbnail" and a "highres" version.

| ninjs 3.1 | ninjs 2.2 | ninjs 1.6 |

```json
                                                                              JSON
  {
    "uri": "http://www.businesswire.com/news/home/20130515006361/en",
    "type":"composite",
    "version": "1",
    "versionCreated": "2013-05-16T04:01:00Z",
    "headlines": [
      {
        "role": "simple",
        "contentType": "text/plain",
        "value": "Dell Redefines Workstation Computing Boundaries with Smallest Tower and Most Powerful Rack
  Workstations"
      }
    ],
    "bodies": [
      {
        "role": "richtext",
        "contentType": "application/xhtml+xml",
        "value": "[ ... ]"
      }
    ],
    "associations": [
      {
        "name": "logo",
        "uri":"http://mms.businesswire.com/media/newsItemId/en/221373/3/dell_blue_rgb.jpg",
        "type":"graphic",
        "versionCreated": "2013-05-16T04:01:00Z",
        "renditions": [
          {
            "name": "thumbnail"
          },
          {
            "name": "highres"
          }
        ]
      },
      {
        "name": "photo",
        "uri":"http://mms.businesswire.com/media/newsItemId/en/369394/3/0025lf.jpg",
        "type":"picture",
        "versionCreated": "2013-05-16T04:01:00Z",
        "renditions": [
          {
            "name": "thumbnail"
          },
          {
            "name": "highres"
          }
        ]
      }
    ]
  }
```

5. ninjs Examples

5.1. Text-only examples

5.1.1. A simple text article

Key features:

- A `uri` must be present as an identifier for this content

- `type` indicates that the generic content type is 'text'

- `by` and `headlines` are typical text news metadata

- The text of the article itself is represented by `bodies`, in two format variants: plain text and HTML.

| ninjs 3.1 | ninjs 2.2 | ninjs 1.6 |

```json
                                                                                        JSON
  {
    "uri" : "http://ninjs.example.com/newsitems/20130709simp123",
    "type" : "text",
    "versionCreated" : "2013-07-09T10:37:00Z",
    "by" : "Paulo Santalucia and Frances d'Emilio",
    "headlines" : [
      {
        "role": "main",
        "value": "Captain of wrecked cruise ship on trial in Italy"
      }
    ],
    "bodies" : [
      {
        "role": "main",
        "contentType": "text/plain",
        "value": "GROSSETO, Italy (AP) -- The trial of the captain of the shipwrecked Costa Concordia cruise liner
  has begun in a theater converted into a courtroom in Tuscany to accommodate all the survivors and relatives of the
  32 victims who want to see justice carried out in the 2012 tragedy. The sole defendant, Francesco Schettino, his
  eyes shaded by sunglasses and slipping into a back door, made no comment to reporters as he arrived for his trial
  Tuesday on charges of multiple manslaughter, abandoning ship and causing the shipwreck near the island of Giglio.
  His lawyer, Domenico Pepe, told reporters that, as expected, the judge was postponing the hearing due to an eight-
  day nationwide lawyers' strike. Pepe said some 1,000 witnesses are expected to eventually testify. Many of them are
  expected to be from among the 4,200 passengers and crew aboard the ship that struck a jagged reef off Giglio, took
  on water and capsized. Schettino has denied wrongdoing."
      },
      {
        "role": "richtext",
        "contentType": "body_xhtml",
        "value": "<p>GROSSETO, Italy (AP) -- The trial of the captain of the shipwrecked Costa Concordia cruise liner
  has begun in a theater converted into a courtroom in Tuscany to accommodate all the survivors and relatives of the
  32 victims who want to see justice carried out in the 2012 tragedy.</p> <p>The sole defendant, Francesco Schettino,
  his eyes shaded by sunglasses and slipping into a back door, made no comment to reporters as he arrived for his
  trial Tuesday on charges of multiple manslaughter, abandoning ship and causing the shipwreck near the island of
  Giglio.</p> <p>His lawyer, Domenico Pepe, told reporters that, as expected, the judge was postponing the hearing
  due to an eight-day nationwide lawyers' strike.</p> <p>Pepe said some 1,000 witnesses are expected to eventually
  testify. Many of them are expected to be from among the 4,200 passengers and crew aboard the ship that struck a
  jagged reef off Giglio, took on water and capsized.  Schettino has denied wrongdoing.</p>"
      }
    ]
  }
```

5.1.2. A more complex example

- Metadata about the content are added to the example above.

- `person`, `place` and `organisation` employ a structure which provides human readable names for the entity but also machine readable identifiers by the `scheme` and `code` properties.

| ninjs 3.1 | ninjs 2.2 | ninjs 1.6 |

```json
{
  "uri" : "http://ninjs.example.com/newsitems/20130709med123",
  "type" : "text",
  "profile" : "text-only",
  "versionCreated" : "2013-07-09T10:37:00Z",
  "copyrightNotice" : "Copyright 2013 The News Agency, www.tna.org - all rights reserved.",
  "language" : "en",
  "people" : [
    {
      "name" : "Francesco Schettino",
      "rel" : "about",
      "uri" : "http://www.wikidata.org/entity/Q4421821"
    }
  ],
  "places" : [
    {
      "name" : "Grossetto",
      "rel" : "mentions"
    },
    {
      "name" : "Tuscany",
      "rel" : "mentions"
    },
    {
      "name" : "Italy",
      "rel" : "mentions",
      "uri" : "http://cvx.iptc.org/iso3166-1a2/it"
    }
  ],
  "organisations" : [
    {
      "name" : "Costa Crociere SpA",
      "rel" : "mentions"
    }
  ],
  "by" : "Paulo Santalucia and Frances d'Emilio",
  "located" : "Grossetto, Italy",
  "headlines" : [
    {
      "role": "main",
      "value": "Captain of wrecked cruise ship on trial in Italy"
    }
  ],
  "bodies": [
    {
      "role": "main",
      "contentType": "text/plain",
      "value" : "GROSSETO, Italy (AP) -- The trial of the captain of the shipwrecked Costa Concordia cruise liner has begun in a theater converted into a courtroom in Tuscany to accommodate all the survivors and relatives of the 32 victims who want to see justice carried out in the 2012 tragedy. The sole defendant, Francesco Schettino, his eyes shaded by sunglasses and slipping into a back door, made no comment to reporters as he arrived for his trial Tuesday on charges of multiple manslaughter, abandoning ship and causing the shipwreck near the island of Giglio. His lawyer, Domenico Pepe, told reporters that, as expected, the judge was postponing the hearing due to an eight-day nationwide lawyers' strike. Pepe said some 1,000 witnesses are expected to eventually testify. Many of them are expected to be from among the 4,200 passengers and crew aboard the ship that struck a jagged reef off Giglio, took on water and capsized. Schettino has denied wrongdoing."
    },
    {
      "role": "richtext",
      "contentType": "application/xhtml+xml",
      "value": "<p>GROSSETO, Italy (AP) -- The trial of the captain of the shipwrecked Costa Concordia cruise liner has begun in a theater converted into a courtroom in Tuscany to accommodate all the survivors and relatives of the 32 victims who want to see justice carried out in the 2012 tragedy.</p> <p>The sole defendant, Francesco Schettino, his eyes shaded by sunglasses and slipping into a back door, made no comment to reporters as he arrived for his trial Tuesday on charges of multiple manslaughter, abandoning ship and causing the shipwreck near the island of Giglio.</p> <p>His lawyer, Domenico Pepe, told reporters that, as expected, the judge was postponing the hearing due to an eight-day nationwide lawyers' strike.</p> <p>Pepe said some 1,000 witnesses are expected to eventually testify. Many of them are expected to be from among the 4,200 passengers and crew aboard the ship that struck a jagged reef off Giglio, took on water and capsized. Schettino has denied wrongdoing.</p>"
    }
  ]
}
```

5.1.3. More Examples on GitHub

You can find more examples of ninjs representations of text-based news releases in the <u>ninjs GitHub repository</u> <u>(https://github.com/iptc/newsinjson/tree/main/examples)</u>.

5.2. Photo-only example

A simple photo-only example.

Key features:

- A "uri" must be present
- "type" indicates that the generic content type is 'a picture'
- "by", "headlines", "descriptions" (in plain text and HTML format) are typical photo metadata
- The image itself is represented by "renditions". Two are available in this example: a main version and a small version.

| ninjs 3.1 | ninjs 2.2 | ninjs 1.6 |

```json
{
  "uri" : "http://ninjs.example.com/newsitems/20130709simpPh123",
  "type" : "picture",
  "versionCreated" : "2013-07-08T08:12:00Z",
  "by" : "Paulo Santalucia",
  "headlines" : [
    {
      "role": "main",
      "value": "Costa Concordia cruise ship"
    }
  ],
  "descriptions": [
    {
      "role": "text",
      "contentType": "text/plain",
      "value": "The Costa Concordia cruise ship lies on its side in the waters of the Tuscan island of Giglio, Italy, Monday, July 8, 2013. The luxury cruise ship ran aground off the coast of Tuscany on Jan 13, 2012, sending water pouring in through a 160-foot (50-meter) gash in the hull and forcing the evacuation of some 4,200 people from the listing vessel early."
    },
    {
      "role": "richtext",
      "contentType": "description_xhtml",
      "value": "<p>The Costa Concordia cruise ship lies on its side in the waters of the Tuscan island of Giglio, Italy,<br /> Monday, July 8, 2013. The luxury cruise ship ran aground off the coast of Tuscany on Jan 13, 2012, sending water pouring in through a 160-foot (50-meter) gash in the hull and forcing the evacuation of some 4,200 people from the listing vessel early.</p>"
    }
  ],
  "renditions" : [
    {
      "name": "main",
      "href" : "http://hosted.ap.org/photos/2/2643c588-dc8d-4923-bebd-e3b904edbb3a-big.jpg",
      "contentType": "image/jpg",
      "title" : "Mid Resolution",
      "height" : 281,
      "width" : 429
    },
    {
      "name": "small",
      "href" : "http://hosted.ap.org/photos/2/2643c588-dc8d-4923-bebd-e3b904edbb3a-small.jpg",
      "contentType": "image/jpg",
      "title" : "Low Resolution",
      "height" : 117,
      "width" : 179
    }
  ]
}
```

5.3. Multimedia example

Example for multimedia content.

The ninjs object includes:

- text news as main object
- a main picture illustrating the text news (in ninjs terms: associated with the text news)
- a portrait of a person the text news is about (also: associated)

| ninjs 3.1 | ninjs 2.2 | ninjs 1.6 |

```json
{
  "uri" : "http://ninjs.example.com/newsitems/20130709cplx456",
  "type" : "composite",
  "profile" : "text-photo",
  "versionCreated" : "2013-07-09T10:39:00Z",
  "copyrightNotice" : "Copyright 2013 The News Agency, www.tnag.org - all rights reserved.",
  "language" : "en",
  "people" : [
    {
      "name" : "Francesco Schettino",
      "rel" : "about",
      "uri" : "http://www.wikidata.org/entity/Q4421821"
    }
  ],
  "places" : [
    {
      "name" : "Grossetto",
      "rel" : "mentions"
    },
    {
      "name" : "Tuscany",
      "rel" : "mentions"
    },
    {
      "name" : "Italy",
      "rel" : "mentions",
      "uri" : "http://cvx.iptc.org/iso3166-1a2/it"
    }
  ],
  "organisations" : [
    {
      "name" : "Costa Crociere SpA",
      "rel" : "mentions"
    }
  ],
  "by" : "Paulo Santalucia and Frances d'Emilio",
  "located" : "Grossetto, Italy",
  "headlines" : [
    {
      "value": "Captain of wrecked cruise ship on trial in Italy"
    }
  ],
  "bodies": [
    {
      "role": "text",
      "contentType": "text/plain",
      "value": "GROSSETO, Italy (AP) -- The trial of the captain of the shipwrecked Costa Concordia cruise liner has begun in a theater converted into a courtroom in Tuscany to accommodate all the survivors and relatives of the 32 victims who want to see justice carried out in the 2012 tragedy. The sole defendant, Francesco Schettino, his eyes shaded by sunglasses and slipping into a back door, made no comment to reporters as he arrived for his trial Tuesday on charges of multiple manslaughter, abandoning ship and causing the shipwreck near the island of Giglio. His lawyer, Domenico Pepe, told reporters that, as expected, the judge was postponing the hearing due to an eight-day nationwide lawyers' strike. Pepe said some 1,000 witnesses are expected to eventually testify. Many of them are expected to be from among the 4,200 passengers and crew aboard the ship that struck a jagged reef off Giglio, took on water and capsized. Schettino has denied wrongdoing."
    },
    {
      "role": "text",
      "contentType" : "application/xhtml+xml",
      "value": "<p>GROSSETO, Italy (AP) -- The trial of the captain of the shipwrecked Costa Concordia cruise liner has begun in a theater converted into a courtroom in Tuscany to accommodate all the survivors and relatives of the 32 victims who want to see justice carried out in the 2012 tragedy.</p> <p>The sole defendant, Francesco Schettino, his eyes shaded by sunglasses and slipping into a back door, made no comment to reporters as he arrived for his trial Tuesday on charges of multiple manslaughter, abandoning ship and causing the shipwreck near the island of Giglio.</p> <p>His lawyer, Domenico Pepe, told reporters that, as expected, the judge was postponing the hearing due to an eight-day nationwide lawyers' strike.</p> <p>Pepe said some 1,000 witnesses are expected to eventually testify. Many of them are expected to be from among the 4,200 passengers and crew aboard the ship that struck a jagged reef off Giglio, took on water and capsized. Schettino has denied wrongdoing.</p>"
    }
  ],
  "associations" : [
    {
      "name": "mainpic",
      "uri" : "http://ninjs.example.com/newsitems/20130709simpPh123",
```

```json
      "type" : "picture",
      "versionCreated" : "2013-07-08T08:12:00Z",
      "copyrightNotice" : "Copyright 2013 The News Agency, www.tna.org - all rights reserved.",
      "objects" : [
        {
          "name" : "Costa Concordia",
          "rel" : "about",
          "uri" : "http://www.wikidata.org/entity/Q190542"
        }
      ],
      "places" : [
        {
          "name" : "Giglio",
          "rel" : "mentions"
        },
        {
          "name" : "Tuscany",
          "rel" : "mentions"
        },
        {
          "name" : "Italy",
          "rel" : "mentions",
          "uri" : "http://cvx.iptc.org/iso3166-1a2/it"
        }
      ],
      "organisations" : [
        {
          "name" : "Costa Crociere SpA",
          "rel" : "mentions"
        }
      ],
      "by" : "Paulo Santalucia",
      "headlines" : [
        {
          "value": "Costa Concordia cruise ship"
        }
      ],
      "descriptions": [
        {
          "role": "text",
          "contentType": "text/plain",
          "value": "The Costa Concordia cruise ship lies on its side in the waters of the Tuscan island of Giglio,
Italy, Monday, July 8, 2013.  The luxury cruise ship ran aground off the coast of Tuscany on Jan 13, 2012, sending
water pouring in through a 160-foot (50-meter) gash in the hull and forcing the evacuation of some 4,200 people
from the listing vessel early."
        },
        {
          "role": "richtext",
          "contentType": "application/xhtml+xml",
          "value": "<p>The Costa Concordia cruise ship lies on its side in the waters of the Tuscan island of
Giglio, Italy,</ br> Monday, July 8, 2013. The luxury cruise ship ran aground off the coast of Tuscany on Jan 13,
2012, sending water pouring in through a 160-foot (50-meter) gash in the hull and forcing the evacuation of some
4,200 people from the listing vessel early. </p>"
        }
      ],
      "renditions" : [
        {
          "name": "main",
          "href" : "http://hosted.ap.org/photos/2/2643c588-dc8d-4923-bebd-e3b904edbb3a-big.jpg",
          "contentType": "image/jpg",
          "title" : "Mid Resolution",
          "height" : 281,
          "width" : 429
        },
        {
          "name": "small",
          "href" : "http://hosted.ap.org/photos/2/2643c588-dc8d-4923-bebd-e3b904edbb3a-small.jpg",
          "contentType": "image/jpg",
          "title" : "Low Resolution",
          "height" : 117,
          "width" : 179
        }
      ]
    },
    {
```

```
        "name": "portrait",
        "uri" : "http://ninjs.example.com/newsitems/20130709simpPh456",
        "type" : "picture",
        "versionCreated" : "2013-07-09T10:12:00Z",
        "copyrightNotice" : "Copyright 2013 The News Agency, www.tna.org - all rights reserved.",
        "people" : [
          {
            "name" : "Francesco Schettino",
            "rel" : "about",
            "uri" : "http://www.wikidata.org/entity/Q4421821"
          }
        ],
        "places" : [
          {
            "name" : "Grossetto",
            "rel" : "mentions"
          },
          {
            "name" : "Tuscany",
            "rel" : "mentions"
          },
          {
            "name" : "Italy",
            "rel" : "mentions",
            "uri" : "http://cvx.iptc.org/iso3166-1a2/it"
          }
        ],
        "organisations" : [
          {
            "name" : "Costa Crociere SpA",
            "rel" : "mentions"
          }
        ],
        "by" : "Paulo Santalucia",
        "headlines" : [
          {
            "value": "Francesco Schettino at court"
          }
        ],
        "descriptions": [
          {
            "value": "Francesco Schettino the captain of the Costa Concordia cruise ship at court in Grossetto,
Italy, on 9 July 2013"
          }
        ],
        "renditions" : [
          {
            "name": "main",
            "href" : "http://hosted.ap.org/photos/2/2643c588-dc8d-4923-bebd-e3b904xyz000-big.jpg",
            "contentType": "image/jpg",
            "title" : "Mid Resolution",
            "height" : 430,
            "width" : 280
          },
          {
            "name": "small",
            "href" : "http://hosted.ap.org/photos/2/2643c588-dc8d-4923-bebd-e3b904xyz000-small.jpg",
            "contentType": "image/jpg",
            "title" : "Low Resolution",
            "height" : 180,
            "width" : 120
          }
        ]
      }
    ]
  }
}
```

6. ninjs Best Practices and How Tos

Here we describe some best practices for working with ninjs in your own organisation.

6.1. Best Practice for identifying objects in arrays

The first generation of ninjs used patternproperties to identify and name bodies, headlines etc. With 2.x and 3.x these are now arrays of objects. If, for example, you only have one headline, you only need to use the value property. If there are multiple headlines you can use the role and/or contenttype properties to convey to the receivers of your items what each headline is meant for and what format it has.

6.2. How to escape XML content for inclusion in JSON objects

JSON string values cannot contain characters: `"`, `\`, and various control characters like tabs. Therefore, we need to escape these characters in our content when marshaling data into JSON objects. We have provided an XSL stylesheet to do this using XSL.

The XSL Template at https://raw.githubusercontent.com/iptc/newsinjson/master/tools/xslt/nitf-to-json.xslt extracts the body of an NITF document (NITF samples are available from https://iptc.org/standards/nitf/using-nitf/) and creates a simple JSON object. The resulting object is not yet a valid ninjs object, however it will be valid JSON.

6.3. How to create provider-specific extensions

In some cases, your news releases may have content that does not fit into the objects that are defined in the ninjs standard. In these cases, you can (and should) copy and extend the IPTC ninjs schema.

To do this, do the following:

1. make a copy of the schema file

2. change the URL of the "id" property of IPTC's ninjs 3.1 …

```JSON
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "id" : "http://www.iptc.org/std/ninjs/ninjs-schema_3.1.json#",
  "type" : "object",
  ...
```

   to your own ID, for example `example.com-ninjs-0.1`:

```JSON
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "id" : "http://www.example.com/extended-ninjs-schema_0.1.json#",
  "type" : "object",
  ...
```

3. add your own properties to your copy of the schema. For example, to add a property explaining the reason for an embargo:

```JSON
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "id" : "http://www.example.com/extended-ninjs-schema_0.1.json#",
  "type" : "object",
  ...
  "embargoed": {
    "title": "Embargoed",
    "description": "The date and time before which all versions of the news object are embargoed. If absent, this
object is not embargoed. nar:embargoed",
    "type": "string",
    "format": "date-time"
  },
  "embargoedReason" : {
    "description" : "Textual description of why article is embargoed.",
    "type" : "string"
  },
  ...
```

4. save your new schema to a location where users of the schema can access it, which should be the same as the URL used for the "id" property. For the example above, the file should be accessible at http://www.example.com/e-ninjs-schema_0.1.json

This is an example ninjs document that includes the added property giving the reason for the embargo in an extended ninjs document:

```json
{
  "uri": "http://ninjs.example.com/newsitems/20130709simp123",
  "type": "text",
  "versioncreated": "2013-07-09T10:37:00Z",
  "embargoed": "2013-07-09T14:00:00Z",
  "embargoedreason": "The nomination is made public at this time."
}
```

6.3.1. Using `standard`

With the `standard` field, providers can indicate in each JSON file exactly which schema the file is valid against. So the above example of extending the schema could be shown like this:

```json
{
  "uri": "http://ninjs.example.com/newsitems/20130709simp123",
  "standard": {
      "name": "e-ninjs-schema",
      "schema" : "http://www.example.com/e-ninjs-schema_0.1.json",
      "version" : "0.1"
  },
  "type": "text",
  "versioncreated": "2013-07-09T10:37:00Z",
  "embargoed": "2013-07-09T14:00:00Z",
  "embargoedreason": "The nomination is made public at this time."
  ...
}
```

7. Tools for developing with ninjs

7.1. Validating a ninjs document

JSON Schema is the de-facto standard for validating JSON documents. The IPTC maintains schemas for each version of the ninjs standard. These schemas can be used to validate ninjs documents.

The latest IPTC ninjs schema is available for download at https://iptc.org/std/ninjs/ninjs-schema_3.1.json

7.2. Sample Code

Sample code demonstrating how to validate ninjs documents, along with example ninjs documents, can be found in the IPTC newsinjson github repository (https://github.com/iptc/newsinjson).

7.3. Online "ninjs generator" tool

The IPTC has created a simple form-based tool that can be used to generate sample ninjs documents. It supports ninjs versions 1.6, 2.2 and 3.1, so it is also a handy way to see the differences between each schema version.

The tool can be accessed at https://iptc.org/std/ninjs/generator/.

7.4. TypeScript type definition

Ian Young of PA Media Group has created a TypeScript type for ninjs 2 (https://www.npmjs.com/package/@pa-media-group/iptc-ninjs-2-type) that can be used to ease development with ninjs in TypeScript. We plan to release a type for ninjs 3.x soon.

8. Mapping between ninjs and NewsML-G2

ninjs and NewsML-G2 are not intended to be equivalent to each other, but they both adhere to the IPTC News Architecture (https://www.iptc.org/standards/news-architecture/), so they share many properties.

Here is a summary of the properties shared by ninjs and NewsML-G2:

ninjs 3.x property name	ninjs 2.x property name	NewsML-G2 equivalent
uri	uri	nar:newsItem@guid (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#identification-and-versioning)
type	type	nar:itemClass (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#item-class-itemclass)
title	title`	nar:itemMeta/title (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#item-title)
altIds	altids	nar:altId (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#alternative-identifier-altid)
profile	profile	nar:profile (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#profile-profile)
version	version	nar:newsItem@version (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#identification-and-versioning)
firstCreated	firstcreated	nar:firstCreated (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#date-first-created---firstcreated)
versionCreated	versioncreated	nar:versionCreated (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#date-version-created---versioncreated)
expires	expires	nar:expires (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#expires)
embargoedUntil	embargoed	nar:embargoed (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#date-item-embargo-ends-embargoed)
pubStatus	pubstatus	nar:pubStatus (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#publish-status-pubstatus)
urgency	urgency	nar:urgency (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#urgency-urgency)
edNote	ednote	nar:edNote (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#editorial-note)
language	language	nar:language (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#language-language)
copyrightHolder	copyrightholder	nar:copyrightHolder (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#rights-information)

ninjs 3.x property name	ninjs 2.x property name	NewsML-G2 equivalent
copyrightNotice	copyrightnotice	nar:copyrightnotice (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#rights-information)
usageTerms	usageterms	nar:usageTerms (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#usage-terms-usageterms)
rightsInfo	rightsinfo	nar:rightsInfo (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#rights-information)
headlines	headlines	nar:headline (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#headline-headline) with @role
headlines / contentType	headlines / contenttype	nar:contentType
slugline	slugline	nar:slugline (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#slugline-slugline)
by	by	nar:by (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#by---by)
bodies	bodies	nar:data or nar:inlineXML (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#newsitem-content)
bodies / charCount	bodies / charcount	nar:charcount
bodies / wordCount	bodies / wordcount	nar:wordcount
bodies / contentType	bodies / contenttype	nar:contentType
descriptions	descriptions	nar:description (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#description-description)
descriptions / contentType	descriptions / contenttype	nar:contentType
digitalSourceType	digitalsourcetype	nar:digitalSourceType (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#digitalsourcetype)
located	located	nar:located (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#located-located)
genre	genre	nar:genre (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#genre-genre)
people	people	nar:subject (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#subject-subject)

ninjs 3.x property name	ninjs 2.x property name	NewsML-G2 equivalent
organisations	organisations	nar:subject (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#subject-subject)
places	places	nar:subject (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#subject-subject)
subjects	subjects	nar:subject (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#subject-subject)
objects	objects	nar:subject (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#subject-subject)
infoSources	infosource	nar:infoSource (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#information-source-infosource)
trustIndicators	trustindicators	nar:link (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#item-links) with @rel
renditions	renditions	nar:remoteContent (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#remote-content-remotecontent)
renditions / href	renditions / href	nar:remoteContent@ref
renditions / contentType	renditions / contenttype	nar:remoteContent@contenttype
renditions / height	renditions / height	nar:remoteContent@height
renditions / width	renditions / width	nar:remoteContent@width
renditions / format	renditions / format	nar:remoteContent@format
renditions / duration	renditions / duration	nar:remoteContent@duration
associations	associations	nar:link (https://www.iptc.org/std/NewsML-G2/2.34/specification/NewsML-G2-2.34-specification.html#item-links)

9. ninjs Revision History

9.1. ninjs 1.0

Version 1.0 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_1.0.json) was approved by the IPTC Standards Committee on 23 October 2013.

9.2. ninjs 1.1

Version 1.1 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_1.1.json) was made available in March 2014.

Version 1.1 moved to JSON Schema version 4, and added the `urgency`, `usageterms` and `geometry_*` properties.

An error in the association reference URI in the version 1.1 JSON Schema document was corrected in September 2017.

9.3. ninjs 1.2

Version 1.2 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_1.2.json) was approved by the IPTC Standards Committee and made available in October 2019.

Version 1.2 added the `firstcreated`, `charcount`, `wordcount`, `slugline`, `ednote`, `infosource`, `title` and `slugline` properties. It also added the value `component` under type. It changed the description of the `renditions` property and added sub-properties `duration` and `format`.

9.4. ninjs 1.3

Version 1.3 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_1.3.json) was approved by the IPTC Standards Committee and made available in May 2020.

Version 1.3 added the `headline_`, `altids`, `trustindicator`, `$standard` and `genre` properties.

9.5. ninjs 1.4

Version 1.4 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_1.4.json) was approved by the IPTC Standards Committee in May 2022. It contains new fields *contentcreated, expires* and *rightsinfo*.

9.6. ninjs 1.5

Version 1.5 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_1.5.json) was approved by the IPTC Standards Committee in May 2023. It includes the new `contactInfo` type which can be applied to the `person`, `organisation`, `place` and `infosource` properties. ninjs 1.5 also includes the new `symboltype` and `symbol` properties under `symbols`, and new properties `creator`, `relevance` and `confidence` under `subject`. Use of `ticker` under `symbols` is now deprecated.

In addition, some changes to the schema were made to fix a validation bug in previous versions. In order to accommodate these changes, the ninjs 1.5 schema uses the `http://json-schema.org/draft-07/schema#` version of JSON Schema.

9.7. ninjs 2.0

Version 2.0 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_2.0.json) was approved by the IPTC Standards Committee and made available in October 2021.

Many changes were made for Version 2.0, some of them breaking backwards compatibility:

- `headline_*` changed to `headlines`
- `body_*` changed to `bodies`
- `description_*` changed to `descriptions`
- `altid_*` changed to `altids`
- `byline` changed to `by` to become more consistent with IPTC News Architecture and NewsML-G2
- `person` changed to `people`
- `organisation` changed to `organisations`
- `place` changed to `places`
- `event` changed to `events`
- `object` changed to `objects`
- `infosource` changed to `infosources`
- `trustindicator` changed to `trustindicators`
- `infosource/rel` changed to `infosource/role` to be more consistent with other properties and the IPTC News Architecture
- all metadata properties now use URIs with an optional literal instead of schemes and codes
- Clarified definition of `uri`
- Removed `mimetype` property

- Added `contentcreated` property

- Changed `representationtype` enum values from *complete* and *incomplete* to *full* and *partial*

- `charcount` and `wordcount` moved to properties of `bodies` elements

- `` `$standard" `` changed to "standard"

- Added a new `rightsinfo` property with either `linkedrights` or `embeddedrights` as a child property

An errata release of 2.0 in May 2022 fixed a problem with the GeoJSON property in the `places` object.

9.8. ninjs 2.1

Version 2.1 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_2.1.json) was approved by the IPTC Standards Committee in May 2023 and made available in June 2023.

ninjs 2.2 includes the new `contactInfo` type which can be applied to the `people`, `organisations`, `places` and `infosources` properties. ninjs 2.1 also includes the new `symboltype` and `symbol` properties under `symbols`, and new properties `creator`, `relevance` and `confidence` under `subjects`. The `ticker` sub-property under `symbols` is now deprecated.

In addition, some changes to the schema were made to fix a validation bug in previous versions. In order to accommodate these changes, the ninjs 2.1 schema uses the `https://json-schema.org/draft/2020-12/schema` version of JSON Schema.

9.9. ninjs 3.0

Version 3.0 of the ninjs schema (https://iptc.org/std/ninjs/ninjs-schema_3.0.json) was approved by the IPTC Standards Committee in October 2024 and made available in November 2024.

ninjs 3.0 was a comprehensive update including several major new features:

- Support for news events and planned news coverage, including expected dates (to be used when exact dates and times are not known), event recurrence information and support for commissioning information on planned coverage

- Switch to camelCase for all properties, to better align with JSON and GraphQL best practice

- Additional rendition properties such as `videoCodec`, `frameRate`, `scanType` and `transportProtocol` to better support video content, including live streamed video

- Removal of deprecated properties such as `ticker` and tidying up of descriptions throughout

9.10. ninjs 3.1, 2.2 and 1.6

Version 3.1 (https://iptc.org/std/ninjs/ninjs-schema_3.1.json), Version 2.2 (https://iptc.org/std/ninjs/ninjs-schema_2.2.json) and Version 1.6 (https://iptc.org/std/ninjs/ninjs-schema_1.6.json) of the ninjs schema were approved by the IPTC Standards Committee in May 2025 and made available in June 2025.

All three versions were updated to add a new property, `digitalSourceType` to allow ninjs documents to convey whether the content being delivered was created by a human or by a generative AI system (among other types).

Versions 1.6 and 2.2 also fix an error in previous versions whereby the data type of `urgency` was "number" where it should have been "integer". The specification of `urgency` in 1.6 and 2.2 is now aligned with 3.x.

10. Additional Resources

10.1. On the Web

The IPTC web site www.iptc.org (https://www.iptc.org) has a wealth of resources for implementers. The page www.iptc.org/standards (https://www.iptc.org/standards) summarises all of the IPTC's standards and provides links to further resources for each, including ninjs (https://www.iptc.org/standards/ninjs/), NewsML-G2 (https://www.iptc.org/standards/newsml-g2/), SportsML-G2 (https://www.iptc.org/standards/sportsml-g2/) and RightsML (https://www.iptc.org/standards/rightsml/).

There are also user groups at Groups.io for all those seeking answers to questions about ninjs or other IPTC standards, and for those who would like to participate in development work:

- https://groups.io/g/iptc-ninjs/ is the public group for discussing ninjs and other representations of news content in the JSON format.

- https://groups.io/g/iptc-newsml-g2/ is the public group for NewsML-G2 and Events in NewsML-G2 topics.

- Staff from IPTC Member organisations are welcome to join the <u>members-only discussion list for the ninjs development team</u> (https://groups.io/g/iptc-news-in-json-dev).

10.2. Join the IPTC

The IPTC welcomes new members. Membership is the backbone of the IPTC, and levels vary for both organisations and individuals. Additionally, several "working parties" and "working groups" focus on specific topic areas and standards. IPTC also hosts two face-to-face meetings a year around the globe, where its working parties discuss and develop their ideas to support information exchange and the rapid innovation of media.

To find out more, visit https://www.iptc.org/participate