What is the Feedly JSON representation of an article?

The purpose of this guide is to help you become familiar with the enriched JSON structure Feedly creates to represent articles in the Feedly REST API.

> 🌀 **Pro tip**
> When an article is open in the Feedly Web Application, you can press SHIFT+D to see the JSON presentation appear.

The Core

Feedly aggregates content from millions of websites using RSS. We normalize the content of the articles into a JSON structure. We also use machine learning models (aka Leo) to analyze and enrich the content.

Here is an overview of the core content we aggregate via RSS and normalize into a JSON object:

```
{
  // The article id is unique and immutable
  // (it won't change over time) for a given article.
  "id": "PSNTZO8gXFUe+cpCZyApw0vEKWPT4b14D6teBEocIAE=_174faa24b8c:18d07f2:2694a93d",
  "originId": "https://www.theverge.com/2020/10/5/21503082/slack-down-issues-slow-loading-threads-messages",
  // When was the last time the article was recrawled?
  "recrawled": 1601945947549,
  // What is the language associated with this article
  "language": "en",
  // content.content or summary.content capture the html content (full or snippet)
  // shared by the publisher in the RSS/ATOM feed
  "content": { "content": "html representation of the article", "direction": "ltr" },
  // What is the title of this article
  "title": "Slack experienced performance issues for more than six hours",
  // When was this article crawled/fetched by feedly for the first time?
  "crawled": 1601932774284,
```

```
  // Who is the author of the article?
  "author": "Bijan Stephen",
  // information about the source/origin publishing the RSS feed
  "origin": {
    "streamId": "feed/http://www.theverge.com/rss/full.xml",
    "title": "The Verge",
    "htmlUrl": "https://www.theverge.com/"
  },
  // alternate.href is the URL of the article shared by the publisher
  "alternate": [{
    "href": "https://www.theverge.com/2020/10/5/21503082/slack-down-issues-slow-loading-threads-messages",
    "type": "text/html"
  }],
  // Updated date provided by the publisher
  "updated": 1601932198000,
  // Published date provided by the publisher
  "published": 1601932198000,
  // If available, the canonicalUrl of the article. The alternate might be a non canonical URL
  "canonicalUrl": "https://www.theverge.com/2020/10/5/21503082/slack-down-issues-slow-loading-threads-messages"
}
```

- All timestamps are EPOCH in milliseconds (number of milliseconds since January 1st, 1970 at 00:00 UTC)
- updated and published are provided by the content creator and have proven to be sometimes unreliable. Feedly sorts article by the crawled time.

HTML Content

- The publisher will include some content in the RSS feed. That content is available in `summary.content` or `content.content`
- If Feedly detects that the content included in the RSS feed is just a snippet and the content is not behind a paywall, Feedly will fetch the content from the website, remove some of the chrome and save the cleaned-up content in `fullContent`.

```
"summary": {
    "content": "Autonomous Response recently stopped a Trickbot....",
    "direction": "ltr"
},
"content": {
    "content": "Autonomous Response recently stopped a Trickbot....",
    "direction": "ltr"
},
"fullContent": "<div><h1>Autonomous Response stops a runaway Trickbot intrusion</h1><p>Tony Jarvis..."
```

The content in summary.content, content.content, and fullContent can include HTML markup. Please see the regular expression below to strip the HTML tags:

```
export const RE_STRIP = /<\/?[^>]+>/gi;

function stripTags(htmlContent) {
  return aString
    .replace(/&lt;/gi, '<')
    .replace(/&gt;/, '>')
    .replace(RE_STRIP, ' ')
    .replace(/(\s+)/gi, ' ');
}
```

Article URL

Look in the following order:

- canonicalUrl
- alternate.href

```
"canonicalUrl": "http://www.darktrace.com/en/blog/autonomous-response-stops-a-runaway-trickbot-intrusion/",

"alternate": [
    {
      "type": "text/html",
      "href": "https://www.darktrace.com/en/blog/autonomous-response-stops-a-runaway-trickbot-intrusion"
    }
]
```

Featured visual

For every article, Feedly tries to analyze the content and see if there is a featured visual we can use to present the overview of the article in the Feedly app before the user clicks and opens to read the article.

If the Feedly platform is able to associate a featured visual with the article, it will be available as part of the `visual` property of the JSON structure.

```
"visual": {
  "url": "https://cdn.vox-cdn.com/thumbor/tHeS1L4BtQs00BtD5OC7LYxkqmg=/0x0:2040x1360/1310x873/cdn.vox-cdn.com/uploads/chorus_image/image/67
  "width": 1310,
  "height": 873,
  "contentType": "image/jpeg",
}
```

Folders, Boards, Leo Web Alerts, and Leo Priorities

The `categories` element represents the different folders the feeds this article belongs to have been added to.

The `tags` element represents the different boards an article has been saved to.

The `sources` element represents the list of Leo Web Alerts this article might belong to.

```
"categories": [
  {
    "id": "enterprise/feedly/category/7f0d55eb-e3e6-49cd-8338-c16bd11f6007",
    "label": "Climate Change"
  },
  {
    "id": "enterprise/feedly/category/f74cc032-f22c-4308-b3af-5b06865d6535",
    "label": "Tech"
  }
],
"tags": [
  {
      "id": "enterprise/feedly/tag/cf4d8710-5118-4570-be14-10f1b5f3543f",
      "label": "Design",
      "addedBy": "00e7f395-db29-44be-a688-bba8ddebdcd9",
      "actionTimestamp": 1648056502895,
  },
  {
      "id": "enterprise/feedly/tag/8aedf010-f832-428a-a090-76fd62c8b930",
      "label": "Hiring",
      "addedBy": "00e7f395-db29-44be-a688-bba8ddebdcd9",
      "actionTimestamp": 1648056507155,
  }
],
"sources": [
  {
      "streamId": "feed/https://feedly.com/f/alert/2716ae91-c31e-479f-9adb-fba694026eea",
      "title": "Competitors",
      "feedlyFeedType": "WebAlert",
      "searchTerms": {
        "parts": [
          {
            "text": "Amazon"
          },
          {
            "id": "nlp/f/publicationBucket/byf:industry",
            "label": "Trade Publications"
```

```
          }
        ],
        "isComplexFilter": true
      }
    }
]
```

Read/unread status

When an article is explicitly read by a user, it is tagged with `global.read`, and a `readTime` is assigned. If the Feedly engine is marking an article that is older than 30days as read, it is flagged with `unread` set to false.

```
"unread": false,
"readTime": 99986,
"tags": [
  {
      "id": "user/19c7c186-e129-423e-8fdb-f9e1156d4cf4/tag/global.read",
      "label": ""
  }
]
```

Notes and Highlights

When a user adds notes and highlights to an article, those annotations are captured in the `annotations` element. Highlights are captured using a `highlight` element and notes are captured in a `comment` element.

```
"annotations": [
  {
      "author": "00e7f395-db29-44be-a688-bba8ddebdcd9",
      "authorFirstInitial": "",
      "created": 1648056403659,
      "entryId": "1m7WZiH0rIU9oT5IjVi8urE+Euir4o0PfzR8lqLw7pk=_17f92d356ab:128121c:6d2ffb5e",
      "highlight": {
        "version": 1,
        "start": 12788,
        "end": 12954,
        "text": "The first part of any debrief is spent discussing the candidate, but the remainder is spent talking about how the intervie
      },
      "id": "17fb7d106cb:7b71cc:884eff50",
  },
  {
      "author": "00e7f395-db29-44be-a688-bba8ddebdcd9",
      "authorEmail": "xxxxx@feedly.com",
      "authorFirstInitial": "C",
      "authorName": "Catalina de León Belloc",
      "authorPicture": "https://lh3.googleusercontent.com/a-/AOh14GiHX19pk5_wYpkWqA1v-jkj3uPFsFhYy-qkYboM=s256",
      "comment": "@eduardo here are a couple of plugins worth exploring when we decide to incorporate tokens",
      "created": 1647989499876,
      "emailMentions": [],
      "entryId": "7YhsNHQuRN8Meh5Ie2RK9UcbEyQj51n2NP6hKhpLqKY=_17fb3373196:2906234:59331c86",
      "id": "17fb3d427e4:457b1a:4f769b1b",
      "slackMentions": [
        "@eduardo"
      ],
      "slackMsgIds": {
        "@eduardo": {
          "ts": "1647989500.504149",
          "channelId": "D01E0RW6859"
        }
      },
  }
]
```

Leo Named Entities

Leo includes a set of NER (Named Entity Recognition) models that are responsible for tagging companies, people, location, products, and other named entities defined in Wikipedia (by name or by aliases). When a named entity is recognized, it is added to

the `entities` array. The `mentions` array specifies what fragment of content in the article triggered the association. The `disambiguated` boolean defines if Leo uses a simple model or a more advanced disambiguation model.

Leo also includes a list of industry companies. When an industry company list is associated with an article, the `causes` array defines which entity in the list was recognized and triggered the tagging of the list.

`salienceLevel` is sent to `about` if the matching occurred in the title of the article and `mention` when the matching occurred in the entire content of the article.


```
"entities": [
    {
      "type": "org",
      "disambiguated": true,
      "id": "nlp/f/entity/gz:org:kotak-mahindra-bank",
      "label": "Kotak Mahindra Bank",
      "mentions": [
        {
          "text": "Kotak Mahindra Bank"
        }
      ],
      "salienceLevel": "mention"
    },
    {
      "type": "org",
      "disambiguated": true,
      "id": "nlp/f/entity/gz:org:finance-industry-companies",
      "label": "Finance Companies",
      "causes": [
        {
          "id": "nlp/f/entity/gz:org:kotak-mahindra-bank",
          "label": "Kotak Mahindra Bank"
        }
      ],
      "mentions": [],
      "salienceLevel": "mention"
    }
  ],
```


Leo Topics

Leo includes a set of topic classification models (Industries, Technologies, Trends, etc...). When the content is analyzed and a topic is detected, it is added to the `commonTopics` array. The salience and score capture the level of aboutness.


```
"commonTopics": [
    {
      "type": "industryTopic",
      "id": "nlp/f/topic/4011",
      "label": "Automotive Industry",
      "score": 0.812,
      "salienceLevel": "about"
    },
    {
      "type": "industryTopic",
      "id": "nlp/f/topic/4016",
      "label": "Energy Industry",
      "score": 0.676,
      "salienceLevel": "about"
    },
    {
      "type": "topic",
      "id": "nlp/f/topic/3049",
      "label": "Automotive",
      "score": 0.881,
      "salienceLevel": "about"
    }
  ],
```


Leo Business Events

Leo includes a set of event detection models (product launches, funding event, partnerships, etc...). When the content is analyzed and an event is detected, it is added to the `businessEvents` array. The salience and score capture the level of aboutness.

```
"businessEvents": [
    {
      "id": "nlp/f/businessEvent/productLaunch",
      "label": "Product Launch",
      "score": 0.987,
      "salienceLevel": "about"
    }
],
```

Leo Threat Actors

Leo includes a set of models that integrate with Malpedia and identify threat actor mentions by name or aliases. When a threat actor is identified, it is added to the `entities` array with the type `threatActor`.

Leo also includes a set of Threat Actor lists (for example Threat Actors from North Korea). When a threat actor list is tagged, you can rely on the `causes` array to determine what specific threat actor in the list was identified in the article.

```
"entities": [
  {
      "type": "threatActor",
      "disambiguated": true,
      "mentions": [
        {
          "text": "Lazarus Group"
        },
        {
          "text": "LazarusGroup"
        },
        {
          "text": "Lazarus group"
        }
      ],
      "salienceLevel": "about",
      "id": "nlp/f/entity/gz:ta:68391641-859f-4a9a-9a1e-3e5cf71ec376",
      "label": "Lazarus Group"
  },
  {
      "type": "threatActor",
      "disambiguated": true,
      "causes": [
        {
          "id": "nlp/f/entity/gz:ta:68391641-859f-4a9a-9a1e-3e5cf71ec376",
          "label": "Lazarus Group"
        }
      ],
      "mentions": [],
      "salienceLevel": "about",
      "id": "nlp/f/entity/gz:tac:kp",
      "label": "Threat Actors from North Korea"
},
```

Leo Malware Families

Leo includes a set of models that integrate with Malpedia and identify malware families mentioned by name or aliases. When a malware family is identified, it is added to the `entities` array with the type `malwareFamily`.

```
"entities": [
    {
      "type": "malwareFamily",
      "disambiguated": true,
      "mentions": [
        {
          "text": "Trickbot"
        }
      ],
      "salienceLevel": "about",
      "id": "nlp/f/entity/gz:mal:c824813c-9c79-4917-829a-af72529e8329",
```

```
      "label": "TrickBot"
    },
```

Leo Cyber Attacks

Leo includes a set of models that determine if an article is about a cyber attack (and who the target of the attack is.

When a cyber-attack instance is detected, a common topic of type `cyberEvent` and id `nlp/f/topic/4009` is added to the `commonTopics` array. The `targets` array identifies the organization(s) being targeted by the attack.

```
"commonTopics": [
    {
      "type": "cyberEvent",
      "targets": [
        {
          "confidence": 0.9963997602462769,
          "text": "Microsoft",
          "id": "nlp/f/entity/gz:org:microsoft"
        }
      ],
      "id": "nlp/f/topic/4009",
      "label": "Cyber Attacks",
      "score": 0.961,
      "salienceLevel": "about"
    },
  ],
```

Leo CVE Information

Leo includes a set of models that look for vulnerabilities and aggregate CVE and CVSS information from NVD and vendor advisories. When a CVE is detected, an object is added to the entities array with a `vulnerabilityInfo` object capturing an overview of the information Leo has aggregated about this vulnerability.

If the CVE does not have a CVSS score, Leo uses a machine learning model to predict the CVSS score and captured the output of the model in the `estimatedCVSS` score. `HIGH` means that the predicted CVSS is higher than 8, `MEDIUM` is between 5 and 8 and `LOW` is lower than 5.

```
"entities": [
    {
      "vulnerabilityInfo": {
        "description": "Lodash versions prior to 4.17.21 are vulnerable to Command Injection via the template function.",
        "cvssScore": 7.2,
        "hasExploit": true,
        "hasPatch": true
      },
      "id": "vulnerability/m/entity/CVE-2021-23337",
      "mentions": [
        {
          "text": "CVE-2021-23337"
        }
      ],
      "label": "CVE-2021-23337"
    },
    {
      "vulnerabilityInfo": {
        "description": "Lodash versions prior to 4.17.21 are vulnerable to Regular Expression Denial of Service (ReDoS) via the toNumber, t
        "cvssScore": 5.3,
        "cvssCategoryEstimate": "MEDIUM",
        "hasExploit": true,
        "hasPatch": true
      },
      "id": "vulnerability/m/entity/CVE-2020-28500",
      "mentions": [
        {
          "text": "CVE-2020-28500"
        }
      ],
      "label": "CVE-2020-28500"
```

```
    },
],
"estimatedCVSS": {
    "category": "HIGH"
 },
```

Leo IoC Information

Leo includes a set of models that identify indicators of compromise in articles and threat intelligence reports. All the flagged IoCs are available in the `indicatorsOfCompromise` element.

This element also includes an `exports` section that captures a link to both a `stix2.1` and `markdown` export. The STIX export includes both the IoCs but also references to threat actors, malware families, and vulnerabilities mentioned in the article.

```
"indicatorsOfCompromise": {
    "exports": [
      {
        "type": "markdown",
        "url": "https://exports.feedly.com/ioc/78664a4c09bdd4bfb6696741b04016e1/20220317.090430.all-ioc.md"
      },
      {
        "type": "stix2.1",
        "url": "https://exports.feedly.com/ioc/78664a4c09bdd4bfb6696741b04016e1/20220317.090430.all-ioc.json"
      }
    ],
    "mentions": [
      {
        "text": "modernmeadow[.]co",
        "type": "domain",
        "canonical": "modernmeadow[.]co"
      },

      {
        "text": "23.81.246[.]187:443",
        "type": "ip",
        "canonical": "23[.]81.246.187:443"
      },
      {
        "text": "9fdec91231fe3a709c8d4ec39e25ce8c55282167c561b14917b52701494ac269",
        "type": "hash",
        "canonical": "9fdec91231fe3a709c8d4ec39e25ce8c55282167c561b14917b52701494ac269"
      },
    ]
  },
```

Leo MITRE ATT&CK TTPs

Leo includes a set of models that identify MITRE ATT&CK TTPs in articles and threat intelligence reports. It is a hierarchical model that captures tactics, techniques, and sub-techniques.

TTPs are added to the `entities.`

We also include an `attackNavigator` element that links to a JSON visual mapping of the list of TTPs included in the article into the MITRE ATT&CK Navigator tool.

```
"entities": [
  {
      "type": "mitreAttack",
      "disambiguated": true,
      "label": "Spearphishing Attachment (Enterprise T1566.001)",
      "id": "nlp/f/entity/gz:mi:attack-pattern-2e34237d-8574-43f6-aace-ae2915de8597",
      "mentions": [
        {
          "text": "email accounts attackers would then send spear phishing"
        }
      ],
      "salienceLevel": "mention"
    },{
      "type": "mitreAttack",
      "disambiguated": true,
      "label": "Command and Scripting Interpreter (Enterprise T1059)",
```

```
      "id": "nlp/f/entity/gz:mi:attack-pattern-7385dfaf-6886-4229-9ecd-6fd678040830",
      "mentions": [
        {
          "text": "running executable files"
        }
      ],
      "salienceLevel": "mention"
    },{
      "type": "mitreAttack",
      "disambiguated": true,
      "label": "Execution (Enterprise TA0002)",
      "id": "nlp/f/entity/gz:mi:x-mitre-tactic-4ca45d45-df4d-4613-8980-bac22d278fa5",
      "causes": [
        {
          "label": "Windows Management Instrumentation (Enterprise T1047)",
          "id": "nlp/f/entity/gz:mi:attack-pattern-01a5a209-b94c-450b-b7f9-946497d91055"
        },
        {
          "label": "Command and Scripting Interpreter (Enterprise T1059)",
          "id": "nlp/f/entity/gz:mi:attack-pattern-7385dfaf-6886-4229-9ecd-6fd678040830"
        }
      ],
      "mentions": [],
      "salienceLevel": "mention"
    }
]
"attackNavigator": {
    "ttpCount": 5,
    "url": "https://exports.feedly.com/attacks/3432ed1745df027d416d8c74ba92e13e/leo-wizard-spider-one-article.json"
},
```


Leo Custom Topics

You can use the new custom topic feature to group a set of Leo Concept commonly used into a single, easier to manage and update Leo Concept called a Leo Custom Topic.

When a custom topic is tagged, a new entity if type `customTopic` is added to the `entities` list.

The `causes` element identifies which item in the custom topic was identified/triggered the association.


```
"entities": [
  {
      "type": "customTopic",
      "disambiguated": true,
      "id": "nlp/f/entity/cl:feedly:BrandFamily-rbumwm",
      "causes": [
        {
          "id": "nlp/f/entity/gz:org:microsoft",
          "label": "Microsoft"
        }
      ],
      "mentions": [],
      "salienceLevel": "mention",
      "label": "Brand Family"
  }
]
```


Leo Summary Sentences

Leo includes a model that analyzes the content of the article and picks the two most salient sentences. We call these two sentences the Leo summary. Think of these two sentences as the most interesting sentences to read after the title. Those two sentences are available in the `leoSummary` element.


```
"leoSummary": {
    "sentences": [
      {
        "text": "After the successful launch of Snow+ Electric Scooter in the Indian Markets, Crayon Motors, an Indian manufacturer of e-mo
        "position": 0,
        "score": 0.504
      },
      {
        "text": "The Made in India scooters are designed and manufactured at the Ghaziabad manufacturing unit of Crayon Motors.",
```

```
        "position": 6,
        "score": 0.327
      }
    ]
  },
```

Leo Deduplication

The `duplicates` element represent the list of other articles in your Feedly that have an 85%+ overlap in terms of content to the given article. This happens when you follow different RSS feeds that publish the same content or when you follow sources that publish a press release and modify it slightly.

```
"duplicates": [
    {
      "entryId": "8ujH1fGEaFhXsf/pXdaYjpZo9sm/bPPnVkmadcEorCU=_17fb7aa9ca9:2f10554:d033c07c",
      "feedId": "feed/https://techcrunch.com/feed/",
      "feedTitle": "TechCrunch",
      "unread": true
    }
],
```

Leo Clustering

When important news breaks (Apple launches a new product, major cyber-attack was announced, etc.), lots of different sources will cover that event (from different angles using different content).

Leo includes a machine learning model that continuously reads through millions of articles and tries to cluster articles related to the same news/event.

When an article is part of a cluster, a `clusters` element is added to the article JSON. You can use the list of clusters and their ids to group related articles.

```
"clusters": [
    {
      "id": "topic/fintech/meme/e31d0b42-aacd-11ec-bd05-aa32f38babdb"
    }
]
```

The clustering model has the ability to determine which clusters are trending. When a cluster is trending, it is promoted to a featured meme and exposed via the `featuredMeme` element. A label is programmatically created based on the most salient entities referenced by the story.

```
"featuredMeme": {
  "id": "topic/tech/meme/7e7caa2a-aae7-11ec-bd05-aa32f38babdb",
  "label": "Google Spotify Android",
  "score": 1,
  "featured": true
}
```

Leo Company Lists

Feedly Enterprise customers have access to an add-on feature called Large Company Lists. Customers give us a large list of companies (up to 1,000). We create machine learning models that disambiguate those company names and aliases and we create a custom Leo Concept (entityList entity that represents mentions of any of the companies in the list.

This is a powerful way to monitor attacks on third-party partners or track news about customers or leads.

When an article is tagged by Leo with a Large Company list, you will see an `entityList` entity being added to the list of `entities` associated with the article.

The `causes` element of the entityList will list individual companies/organizations that have been identified by Leo.

```
"entities": [{
  "type": "entityList",
   "disambiguated": true,
    "label": "Feedly FCT Batch 1",
    "id": "nlp/f/entity/cl:feedly:FinTechCcompanies",
     "salienceLevel": "mention",
     "causes": [
      {
         "label": "Aave",
         "id": "nlp/f/entity/gz:org:aave"
      },
      {
         "label": "Uniswap",
         "id": "nlp/f/entity/gz:org:uniswap"
      }
    ],
}
```

Engagement / Social Sharing

Feedly will try to aggregate social engagement information for articles that are popular and generate engagement on social media.

The engagement metadata is captured by two properties: `engagement` and `engagementRate`.

`engagement` is an absolute marker that can be a proxy for the amount of social sharing.

`engagementRate` is a relative marker: a value above 1 indicates a hot/viral article compared to other articles published by the same source but also takes into account the age of the article. The engagement rate allows you to determine if an article is hot/going viral.

```
"engagement": 16,
"engagementRate": 0.53,
```

Linked Entries

Feedly extracts content from links in tweets, Reddit posts, and other sources, and includes the content is the `linked` property:

```
"linked": [
    {
        "id": "xfZjaIdrVYSWquChnLMg+mTmB8EydHoN/EFCSXSwxIM=_17fb88d1171:30c4cea:d033c07c",
        "language": "en",
        "title": "Spotify and Google Announce User Choice Billing — Spotify",
        ...
    },
    ...
]
```

Each item in this list is a full-fledged article, with the same structure as the container. It has its own content, URLs, topics, entities, etc.

Twitter/Reddit Author Details

For Twitter and Reddit sources, articles will contain extra information about the tweet or post author. Here's an example for a tweet:

```
"authorDetails": {
    "fullname": "Daniel Ek",
    "icon": "https://pbs.twimg.com/profile_images/988146044449034241/j3BQbodY_normal.jpg",
    "source": "twitter",
    "url": "https://twitter.com/eldsjal",
    "username": "eldsjal",
```

```
    "picture": "https://pbs.twimg.com/profile_images/988146044449034241/j3BQbodY.jpg"
  },
```

And here's an example for a Reddit post:

```
"authorDetails": {
    "source": "reddit",
    "username": "petr_feedly",
    "url": "https://www.reddit.com/user/petr_feedly"
  },
```

Search Terms

If the article is a response to a search request, the article will include a `previewSearchTerms` property that will give you more information about what parts of the article match your search query. This is useful if you want to highlight or extract the most relevant part of the article.

```
"previewSearchTerms": {
    "parts": [
      {
        "id": "nlp/f/entity/wd:21637",
        "label": "American Civil Liberties Union",
        "type": "org",
        "disambiguated": true,
        "mentions": [
          {
            "text": "american civil liberties union"
          }
        ],
        "salienceLevel": "mention"
      },
      {
        "id": "nlp/f/publicationBucket/byf:industry-bundle",
        "label": "Market Intel"
      }
    ],
    "isComplexFilter": false
  }
```