

/rss

The RSS feeder resource is mostly a feed agent that collects data on demand, transforming the content into a RSS feed. The purpose of this resource is actually to provide more visitors to web publishers.

*This may sound a bit contra productive with the fact that there may already exist RSS feeds for things we monitor. However, in **that** case, **this** feed fetcher is instead built for merging several RSS feeds into one big compilation. There are not very many resources available here, at least not publicly.*

The first and primary resource is a json based entry:

/rss

The [/rss](#) resource is used to just give information about the feed, which sites are currently monitored. This is given in json, with a few exceptions. As we were building this resource, it turned out that some sites wants exclusive rights for the feeds. By means, there are feeds not showing up in this state, unless you use the right API url resource. As an example, we could simply use [Earth616](#) as an example. For that site, there is another contrary exception: They want to show only feeds that are explicitly Marvel related. So in that [API resource](#) we only show a few of them. For such cases, they may also be globally available.

Another example is sites that is politically involved, that we ourselves wish to not publish as a supported feed at the ToolsAPI. They are not shown in this list.

/rss/feed

As long as the [/rss/feed](#) resource is not added with an id or a name, it works mostly like [/rss](#)

/rss/feed/id-or-name

This is the big one. The RSS feed itself. This uri is given either an id or the name of a feed. For example: Giving the feed just an id like [1](#) - which is the [Swedish MovieZine](#) feed - the RSS feed will publish that RSS feed as is. If you instead wants to get feed from [marvel.com](#), you can instead enter the resource as [/rss/feed/marvel](#). Depending on how you use the uri, you will now have a merged list of several sites or feeds in RSS.

Bonus feature

If you prefer to have something simpler than RSS/XML, you can simply add an extra variable in the URL. For example [/rss/feed/marvel?as=json](#). The feeder will in that case, instead return content as json.

How does the fetching works

In its simplest way, we use feed agents, that is fetching the content for us. The fetched content will then be collected and merged in a RSS container. The agents are there to secure some kind of uptime. If we fetch data from only one place, and that site loses its internet connectivity for a few hours or days, we suddenly have no feed content to show.

What about RSS-less sites?

It's a complex process, but as long as the site is consequent with the content, we can provide a RSS feed for it.

Eventually upcoming features

While writing this page, an idea of instead of returning content as xml/rss or json, it might probably be an idea to return content as pre-html (card view) that could simply be fetched as complete html blocks linking and featuring the site we monitor instantly.