

Media types

In this article

- Comment body properties
- Git blob properties
- Commits, commit comparison, and pull requests
- Repository contents
- Gists

Learn about media types for specifying the format of the data you want to consume.

Custom media types are used in the API to let consumers choose the format of the data they wish to receive. This is done by adding one or more of the following types to the `Accept` header when you make a request. Media types are specific to resources, allowing them to change independently and support formats that other resources don't.

All GitHub Enterprise Cloud media types look like this:

```
application/vnd.github.param[+json]
```

The most basic media types the API supports are:

```
application/vnd.github+json
application/json
```

Note: In the past, we recommended including `v3` in your `Accept` header. This is no longer required and will have no impact on your API requests.

If you're specifying a property (such as `full/raw/etc` defined below), put it after `github`:

```
application/vnd.github.raw+json
```

Comment body properties

The body of a comment can be written in [GitHub Flavored Markdown](#). The APIs to manage [issues](#), [issue comments](#), [pull request comments](#), and [gist comments](#) all accept these same media types:

Raw media type for comment body properties

```
application/vnd.github.raw+json
```

Return the raw markdown body. Response will include `body`. This is the default if you do not pass any specific media type.

Text media type for comment body properties [↗](#)

```
application/vnd.github.text+json
```

Return a text only representation of the markdown body. Response will include `body_text`.

HTML media type for comment body properties [↗](#)

```
application/vnd.github.html+json
```

Return HTML rendered from the body's markdown. Response will include `body_html`.

Full media type for comment body properties [↗](#)

```
application/vnd.github.full+json
```

Return raw, text and HTML representations. Response will include `body`, `body_text`, and `body_html`:

Git blob properties [↗](#)

The following media types are allowed when [getting a blob](#):

JSON media type for Git blob properties [↗](#)

```
application/vnd.github+json  
application/json
```

Return JSON representation of the blob with `content` as a base64 encoded string. This is the default if nothing is passed.

Raw media type for Git blob properties [↗](#)

```
application/vnd.github.raw
```

Return the raw blob data.

Commits, commit comparison, and pull requests [↗](#)

The REST API to manage [commits](#) and [pull requests](#) support [diff](#) and [patch](#) formats:

diff media type for commits, commit comparison, and pull requests [↗](#)

```
application/vnd.github.diff
```

patch media type for commits, commit comparison, and pull requests [↗](#)

```
application/vnd.github.patch
```

sha media type for commits, commit comparison, and pull requests [↗](#)

```
application/vnd.github.sha
```

Repository contents [↗](#)

Raw media type for repository contents [↗](#)

```
application/vnd.github.raw
```

Return the raw contents of a file. This is the default if you do not pass any specific media type.

HTML media type for repository contents [↗](#)

```
application/vnd.github.html
```

For markup files such as Markdown or AsciiDoc, you can retrieve the rendered HTML using the `.html` media type. Markup languages are rendered to HTML using our open-source [Markup library](#).

Gists [↗](#)

Raw media type for gists [↗](#)

```
application/vnd.github.raw
```

Return the raw contents of a gist. This is the default if you do not pass any specific media type.

base64 media type for gists [↗](#)

```
application/vnd.github.base64
```

The gist contents are base64-encoded before being sent out. This can be useful if your gist contains any invalid UTF-8 sequences.

Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)