# Integrate Enhanced Media Embed Server

Websites like Facebook and Instagram expose an oEmbed endpoint for developers to utilise. The Iframely service creates embeds from websites on the public Internet. For content on private networks, such as a corporate CMS, this document outlines how to enrich the content or build an API that the Enhanced Media Embed server can utilise to create rich hyperlinks. We'll also provide some information about the Enhanced Media Embed server's summary cards.

There are three options for enhancing the embeds created for private content by the Enhanced Media Embed server:

- Annotate content with Open Graph or other meta tags
- Develop your own custom endpoint that returns JSON in the oEmbed format
- Develop your own custom endpoint that returns JSON in the Tiny Enhanced Media Embed format

## A note on cookies & authentication

If the content or endpoint is on the same domain as the Enhanced Media Embed server, cookies will be forwarded to that server. This should reuse the existing login of the user in a CMS environment.

As an example:

- The Enhanced Media Embed server is accessible at `http://mydomain.example.com/media`
- Your custom oEmbed endpoint is accessible at `http://mydomain.example.com/endpoint`
- You have already signed into your CMS at `http://mydomain.example.com/cms`
- You embed content from `http://mydomain.example.com/cms/secretcontent` that normally requires authentication to access
- Because your content, oEmbed endpoint and Enhanced Media Embed server are all on the same domain, `http://mydomain.example.com/cms/secretcontent` is embeddable using cookie passthrough

## Annotate content with Open Graph or other meta tags

### Opengraph

If your content is marked up with Open Graph tags and is accessible with a HTTP GET request from the Enhanced Media Embed server, then business card style embeds will be created based on your content.

If Iframely is enabled in the configuration, then the Open Graph look up will be performed by Iframely. Iframely requires that the content be publicly accessible on the Internet.

At a minimum, you'll need to define these Open Graph tags:

- `og:title`
- `og:image`

It's a good idea to also define:

- `og:url`
- `og:description`
- `og:site_name`

Additionally, you can specify a video or audio resource to include in the embed by defining:

- `og:video` & `og:video:type` (only MP4 is supported across all browsers)
- `og:audio` & `og:audio:type` (only MP3 is supported across all browsers)

### SEO tags

As an alternative to Open Graph tags, you can include meta tags using the older style recommended by search engines such as Google.

- `<title>...</title>`
- `<meta name="description">...</meta>`
- `<link rel=image_src href="..." />`

#### Pros

- This is the easiest method for creating embeds without an existing embed API
- No server configuration required
- There are existing CMS plugins that will add these tags to content
- The Enhanced Media Embed server will handle creating the embed HTML and styling

#### Cons

- The page must be accessible to the Enhanced Media Embed server (it must not require authentication)
- The embed HTML and styling created by the Enhanced Media Embed server is not configurable
- Only works for HTML URLs

- Iframely can only be enabled or disabled as a global option. If Iframely is enabled, then all content must be world accessible and cookie passthrough will not happen.

## Custom API

As an alternative to including meta tags in your content, you can write a custom API that returns JSON in either the oEmbed or Tiny Enhanced Media Embed formats.

See the docs on configuring a custom endpoint for details on getting the Enhanced Media Embed server to utilise your custom API.

### OEmbed endpoint

This is a popular choice and many CMSs have existing plugins that support oEmbed. While you can create custom HTML embeds this way, they cannot contain scripts. If the HTML contains a script, then the Enhanced Media Embed server will embed a summary card.

#### Pros

- There are existing CMS plugins that support oEmbed
- You can write your own custom HTML

#### Cons

- No room in this spec for multiple embed representations of the same URL
- Must be a separate API rather than just metadata embedded in the content
- Error messages aren't defined as part of the spec

### Tiny Enhanced Media Embed endpoint

The other option for developing a custom API endpoint is to return JSON in the Tiny Enhanced Media Embed format.

#### Pros

- You can write your own custom HTML
- The format has the ability to house multiple embed representations of the same URL
- Better defined ability to communicate errors to the media server

#### Cons

- Must be a separate API rather than just metadata embedded in the content
- No support from existing plugins
- The Textbox.io editor does not fully take advantage of this format yet

### Tiny Enhanced Media Embed format

#### HTTP response status codes

- HTTP 200 (OK): `EphoxEmbedObj`
- HTTP 400 (User Error): `ErrorObj`
- HTTP 503 (Upstream Error): `ErrorObj`
- HTTP 500 (Unexpected Error): `ErrorObj`

#### JSON response objects

#### `EphoxEmbedObj`

`rel`, `media` and `html` combine to form the default representation of the embeddable resource that your server has chosen. Clients of the Enhanced Media Embed server (such as the Textbox.io editor) can look for alternative representations in `links`.

- `title` (optional)
  - String containing the document title.
- `author_name` (optional)
  - String containing the author's name.
- `author_iri` (optional)
  - String containing an IRI for the author.
- `provider_iri` (optional)
  - String containing an IRI for the resource provider.
- `provider_name` (optional)
  - String containing the name of the resource provider.
- `short_iri` (optional)
  - String containing a shortened IRI for the resource.
- `canonical_iri` (required)
  - String containing the IRI of the resource.
- `description` (optional)
  - String containing a description of the document.
- `cache_age` (optional)

    - Integer containing the suggested cache lifetime for this resource, in seconds.
- `date` (optional)
    - String containing the date of the document in the format **YYYY-MM-DD**.
- `links` (required)
    - LinksObj
- `rel` (optional)
    - RelObj
- `media` (optional)
    - MediaObj
- `html` (optional)
    - String containing the HTML snippet to be embedded by Textbox.io.

## `RelObj`

An array of tags describing the primary type of an embed, where it came from and whether there are any technical attributes that you may want to know about (autoplay, ssl, file format (flash, html5, etc)).

- `primary` (required)
    - Array of PrimaryRels
- `technical` (required)
    - Array of TechnicalRels
- `source` (required)
    - Array of SourceRel

## `PrimaryRel`

A string describing the primary type of an embed containing one of the following values:

- `player` : A video or audio player
- `thumbnail` : A thumbnail representation of the resource
- `image` : A full sized image for the resource
- `reader`
- `file` : No HTML provided. Should just be a hyperlink to a downloadable file.
- `survey`
- `app` : An embed that will switch over to a mobile app if played on a mobile (e.g. soundcloud)
- `summary` : Summary card (scriptless embed)
- `icon`
- `logo`
- `promo`

## `TechnicalRel`

A string describing technical attributes of an embed containing one of the following values:

- `flash`
- `html5`
- `gifv`
- `inline`
- `ssl`
- `autoplay`

## `SourceRel`

A string describing the source of an embed containing one of the following values:

- `iframely` : From Iframely
- `opengraph` : Generated from Open Graph tags in a resource
- `twitter` : Retrieved from a Twitter Card
- `oembed` : Retreived from an oEmbed API
- `sm4`
- `fallback` : Tiny fallback embeds that look at SEO tags and Open Graph tags.
- `script_censor` : The original embed (from Iframely or oEmbed) had a script in it and has been converted to a summary card.
- `smartframe_censor` : The original embed had an Iframely smart frame and has been censored into a summary card to avoid a content dependency on Iframely.

## `LinksObj`

Represents all of the possible representations of this resource.

- `players` (required)
    - Array of LinkObjs
- `thumbnails` (required)
    - Array of LinkObjs
- `apps` (required)
    - Array of LinkObjs
- `readers` (required)
    - Array of LinkObjs

- `surveys` (required)
    - Array of LinkObjs
- `summary_cards` (required)
    - Array of LinkObjs
- `icons` (required)
    - Array of LinkObjs
- `logos` (required)
    - Array of LinkObjs
- `promos` (required)
    - Array of LinkObjs
- `images` (required)
    - Array of LinkObjs
- `files` (required)
    - Array of LinkObjs

## `LinkObj`

This represents a representation that you could link to / embed.

- `media` (optional)
    - MediaObj
- `rels` (required)
    - RelObj
- `href` (optional)
    - String containing the URL of the resource.
- `mime_type` (required)
    - String containing the mime-type of the resource.
- `html` (required)
    - String containing the embeddable HTML snippet.

## `MediaObj`

The media object describes the bounds of the embed. It can either be responsive or fixed.

- `type` (required)
    - String with the value `fixed` or `responsive`

Fields when `type` is `fixed`:

- `width` (required)
    - Integer containing width in pixels.
- `height` (required)
    - Integer containing height in pixels.
- `paddingBottom` (optional)
    - Integer

Fields when `type` is `responsive`:

- `aspectRatio` (optional)
    - Double
- `paddingBottom` (optional)
    - Integer
- `width` (required)
    - DimensionBoundObj
- `height` (required)
    - DimensionBoundObj

## `DimensionBoundObj`

The dimension bounds define the height or width of a responsive embed.

- `type` (required)
    - String with the value of `fixed`, `constrained` or `unbounded`

Fields when `type` is `fixed`:

- `pixels` (required)
    - Integer

Fields when `type` is `constrained`:

- `min_pixels` (optional)
    - Integer
- `max_pixels` (optional)
    - Integer

No additional fields when `type` is `unbounded`.

### `ErrorObj`

- `code` (required)
  - Integer with the value of `400` (User Input Error) or `503` (Upstream Failure)
- subcode (required)
  - Integer with one of the following values:
    - When `code` is **503**:
      - `1` - Upstream connection issue
      - `2` - Upstream returned not OK
      - `3` - Upstream returned a response that didn’t make sense to the server
    - When `code` is **501**:
      - `1` - Support for URI not implemented
    - When `code` is **400**:
      - `1` - URI Failed to parse
      - `2` - URI was relative
      - `3` - URI was empty
      - `4` - URI was not http or https
      - `5` - Max width was not a positive integer
- `msg` (required)
  - A string message for developers / support people.

## Summary cards

When the Enhanced Media Embed server generates a summary card of a URL (using the title, thumbnail, description and website), it returns a HTML snippet like the following. You should apply styles to the document style to suit these to your preference.

```
<div class="ephox-summary-card">
  <a class="ephox-summary-card-url-thumbnail" href="http://www.imdb.com/title/tt0117500/">
    <img class="ephox-summary-card-thumbnail" src="https://images-na.ssl-images-amazon.com/images/M
/MV5BZDJjOTE0N2EtMmRlZS00NzU0LWE0ZWQtM2Q3MWMxNjcwZjBhXkEyXkFqcGdeQXVyNDk3NzU2MTQ@._V1_UY1200_CR90,0,630,1200
_AL_.jpg">
  </a>
  <a class="ephox-summary-card-link" href="http://www.imdb.com/title/tt0117500/">
    <span class="ephox-summary-card-title">The Rock (1996)</span><br><br>

 <span class="ephox-summary-card-description">Directed by Michael
Bay.  With Sean Connery, Nicolas Cage, Ed Harris, John Spencer. A
mild-mannered chemist and an ex-con must lead the counterstrike when a
rogue group of military men, led by a renegade general, threaten a nerve
 gas attack from Alcatraz against San
Francisco.</span><br><br>
    <span class="ephox-summary-card-website">IMDb</span>
  </a>
</div>
```

### Recommended CSS

```
.ephox-summary-card {
    border: 1px solid #AAA;
    box-shadow: 0 2px 2px 0 rgba(0,0,0,.14), 0 3px 1px -2px rgba(0,0,0,.2), 0 1px 5px 0 rgba(0,0,0,.12);
    padding: 10px;
    overflow: hidden;
    margin-bottom: 1em;
}

.ephox-summary-card a {
    text-decoration: none;
    color: inherit;
}

.ephox-summary-card a:visited {
    color: inherit;
}

.ephox-summary-card-title {
    font-size: 1.2em;
    display: block;
}

.ephox-summary-card-author {
    color: #999;
    display: block;
    margin-top: 0.5em;
}

.ephox-summary-card-website {
    color: #999;
    display: block;
    margin-top: 0.5em;
}

.ephox-summary-card-thumbnail {
    max-width: 180px;
    max-height: 180px;
    margin-left: 2em;
    float: right;
}

.ephox-summary-card-description {
    margin-top: 0.5em;
    display: block;
}
```