

1 SVG Preparation

To ensure that the SVG is properly handled by the Expivi Configurator, a few rules apply.

- Not all SVG's are equally supported by the Expivi Configurator, try to keep the SVG as simple as possible.
- All elements that are changed by the Attributes and their supporting elements must have an ID.
 - An example of the supporting element would be the `<image data-container="true">` element required by the SVG Text Attribute.

1.1 Placeholder support

The Expivi Configurator supports the removal of elements that should not be visualized. These elements include placeholders, explanation texts within the SVG texture themselves, and backgrounds that should not be included in the print export.

All elements within the SVG texture that are marked with the `data-placeholder="true"` XML attribute are excluded when requesting the print export. These elements can be made visible however while the visitor is configuring the 3D model.

1.2 Printmark support

Similar to the support for placeholders, the Expivi Configurator also supports print marks. These are essential lines and markings for printers that should not be visible to the visitor.

All elements within the SVG texture that are marked with the `data-printmark="true"` XML attribute are included when requesting the print export. These elements are not visible while the visitor is configuring the 3D model.

2 Exporting a Print Ready File

In order to obtain a print ready file for a configuration, the configuration bundle must first be obtained from the Expivi Configurator. This can typically be done by calling:

```
let configuration_bundle = await expivi.saveConfiguration()
```

This configuration must be sent towards the expivi backend in order to make the resources available for conversion. This is done by POSTing the configuration in the following manner:

```
curl -X POST \
  -H "Content-type: application/json"
  -H "Authorization: Bearer <token>"
  -d @body.json \
  https://expivi.net/api/configured_products/bundle
```

This returns a mapping of the configured product uuids to a number, which is the ID of the configured product in the backend. For example:

```
[
  "cfc55944-c3cc-4d6b-9e08-4d7b72d5a122": 1234567
]
```

 An API token with write access (such as the Team Owner role) is required to access this endpoint. This API token must never be exposed publicly.

 The configuration will only remain available for a period of 3 months. Any operations that work on this resource will fail after this period.

Internally, each print ready file within a product is associated with a media ID. These are referenced within the configured product in the configuration bundle obtained in the first step.

For each media ID in a configured product, a conversion request must be sent to the back end in the following manner:

```
curl -X POST \
  -H "Content-type: application/json"
  -H "Authorization: Bearer <token>"
  -d @body.json \
  https://expivi.net/api/svg/conversion/request/{format}
```

The output format of the request is determined by the format parameter, which currently supports:

- PNG
- PDF
- SVG

The body of the the request should be as follows:

```
{
  "configured_product_id": <the id returned in step 2>,
  "media_id": <a media id found in the configured product>,
  "resolution": <desired resolution for height and width, max=6000>,
  "dpi": <optional dpi, default=300>,
  "hide_placeholders": true,
  "hide_printmarks": true
}
```

This request returns a response of the following format:

```
{
  "hash": <string>,
  "status": <in_progress|complete|not_processing>,
  "message": <optional string>
  "files": [
    <url to the converted file>
  ]
}
```

The hash can be used to poll the status of the conversion, which returns the same response format as above:

```
curl -X GET\
-H "Content-type: application/json"
-H "Authorization: Bearer <token>"
https://expivi.net/api/svg/conversion/check/<hash>
```

Once the status in the message is “complete”, the response will also contain a URL in the “files” array, at which the converted file can be downloaded.

 No authorization is needed to download the resulting file.