TEI Editor

Release 0.10.1

Mark Hall

Oct 08, 2020

CONTENTS:

The TEI Editor is a JavaScript web-application that provides a WYSTYG (What You See, TEI You Get) editor for editing TEI documents. It provides a modern user-interface that supports the creation of complex TEI documents without manual XML editing. However, it also imposes certain restrictions on the TEI, which mean that it may not be applicable for all TEI files that are out there (and that is not the aim, either).

CHAPTER

ONE

FEATURES

The TEI editor is built using Vue.js and can be used both as a stand-alone application and as a component to be loaded within another Vue.js application.

It provides the following features:

- Fully visual editing of text (no XML or TEI in sight)
  - Support for multiple text editors within a single TEI document (text, aparatus, ...)
  - Support for nested documents (footnotes, annotations, ...)
- Editing of the TEI header metadata

CHAPTER

TWO

DESIGN DECISIONS (AND RESTRICTIONS)

In order to create a graphical TEI editor that was usable across a wide range of projects, two big design decisions were made to disallow mixed content and to limit the use of structural elements.

2.1 No mixed content

XML (which TEI is) creates a tree-structure of nested tags to represent its content. In a mixed content document, text and child tags may appear at the same level:

```
<p><place>Camelot</place> is a silly place.</p>
```

This kind of use reduces the number of tags and thus the effort when manually creating the XML. However, the consequence of this is that it becomes impossible to distinguish when white-space is used to format the document for readability or when it has actual semantic content:

```
<p>
  <place>Camelot</place>
  is a silly place.
</p>
```

The first line-break and spaces before the `<place>` tag are probably just formatting. The line-break and spaces before the "is" should be compressed into a single space and the line-break after the full-stop can probably be ignored again. However, none of these assumptions can be guaranteed to be true and may vary between files. This makes parsing and interpreting TEI/XML tricky, making the application code much more complex. To avoid this, the TEI editor does **not** support mixed content. Instead, text content that has meaning **may** only be contained in leaf nodes:

```
<p>
  <place>Camelot</place>
  <seg> is a silly place.</seg>
</p>
```

Obviously this requires using a "this is a text-block" tag (the example used the `<seg>` tag), which has the following consequences:

- Increase the file-size. This is unavoidable, but can be mitigated through compression.
- Explicitly mark out content and formatting white-space.
- Allow for clean formatting of the TEI file.

2.2 Limited structural elements

The editor is built for graphical editing of TEI documents, in a manner similar to Word. It is difficult to create a user-friendly and usable interface for editing non-visual structures (such as TEI's `<div>`) in this context. As a result the TEI editor only supports single-level basic structural elements, such as lists.

2.3 Other Restrictions

Due to the JavaScript nature of the application, there is a performance limit to the size of TEI documents that can be edited. In practice we have found that at about 40k words, performance when typing starts to reduce noticeably.

The TEI editor supports editing nested documents (for example footnotes or annotations). Due to the way this has been implemented it requires a standardised format for the unique identifier for each nested document. The nested document's identifier **must** be in the following format: `nestedDocumentElementName-UniqueNumber`.

CHAPTER THREE

INSTALLATION

The TEI Editor is available as an NPM package and can be installed via:

```
npm install -s tei-editor
```

or:

```
yarn add tei-editor
```

CHAPTER

FOUR

USE

The TEI Editor can be used as a stand-alone application or as a component within a larger Vue.js application.

4.1 Stand-alone

To use the TEI Editor as a stand-alone application, create a new file `tei-editor.html` with the following content:

```
<!DOCTYPE html>
<html>
  <head>
    <title>TEI Editor</title>
    <link rel="stylesheet" href="css/app.css"/>
  </head>
  <body>
    <div id="app"></div>
    <script id="TEIEditorConfig" type="application/json">
    </script>
    <script src="js/chunk-vendors.js"></script>
    <script src="js/app.js"></script>
  </body>
</html>
```

Then copy the `js` and `css` directories from the `tei-editor/dist` directory.

Finally, into the `<script id="TEIEditorConfig">` tag add the configuration as documented here.

If you wish to automatically load a TEI document, add the following before the `<script src="js/chunk-vendors.js">` tag:

```
<script id="TEIEditorData" type="application/xml+tei">
</script>
```

Then add the TEI to load into that element.

4.2 Vue.js Component

Documentation to follow.

CHAPTER

FIVE

CONFIGURATION

The TEI Editor is highly configurable and stylable.

5.1 Editor Sections

The configuration must be made available via a `<script>` tag with the `id` attribute set to "TEIEditorConfig" and the `type` set to "application/json":

```
<script id="TEIEditorConfig" type="application/json">
  {
    "sections": ...
  }
</script>
```

The configuration itself is provided as one large JSON object, which is documented here. The documentation uses a semi-formal format, mainly showing the individual JSON objects that make up the configuration, with some additional markers:

- Optional keys are marked with "?".
- Keys that can be repeated are marked with "+".
- Nested JSON objects are indicated with a camel-case value (e.g. "ParserElement"). When creating an actual configuration object, these are replaced with the nested JSON object.
- Customisable values are indicates by the values "AnyString" (any string content allowed), "AnyHTMLString" (any HTML content is allowed), and "Boolean" (true or false).
- Choices are marked using the pipe "|" symbol.
- Fixed values are indicated using values that start with a lower-case letter.
- Where lists are shown with a single value, this value can be repeated as often as required.

The top-level configuration object is structured as follows:

```
{
  "sections": {
    "+SectionIdentifier": "TextEditorSection | MetadataEditorSection"
  }
}
```

It consists of one or more section identifier keys, each of which has either a `TextEditorSection` or a `MetadataEditorSection` object as its value. The TEI Editor can handle any number of `TextEditorSection`, however there may be at most one `MetadataEditorSection`.

5.1.1 TextEditorSection

The `TextEditorSection` represents a section in which the user can edit a TEI text block:

```
{
  "label": "AnyString",
  "type": "TextEditor",
  "parser": "ParserElement",
  "serialiser": "SerialiserElement",
  "schema": [
    "TagElement"
  ],
  "ui": "TextEditorUIElement"
}
```

The `label` can be any string value and is used in the UI to allow the user to navigate to this section. The `type` must be `"TextEditor"`. The `parser` configuration is used to identify the root element in the TEI document that contains the text to be edited in this section. The `serialiser` configuration is used to identify the root element in the TEI document that the edited text is to be written to. The `schema` contains a list of `TagElement` objects that identify the various markup elements that can be used to annotate the text edited in this section. The `ui` contains a `TextEditorUIElement` that configures the sidebar that contains the markup user-interface controls.

There is one limitation in the `schema` and that is that you **must** specify one `TagElement` that has the `name` set to `"text"`. This is the basic text element and is the only required `TagElement`.

TagElement

The `TagElement` represents one markup tag that is used in the TEI text edited in the section it is specified in.

```
{
  "name": "AnyString",
  "type": "block | wrapping | nested | inline | mark",
  "attrs": {
    "+AttributeName": "ElementAttribute"
  },
  "?parser": "ParserElement",
  "?parsers": ["ParserElement"],
  "serialiser": "SerialiserElement",
  "?content": "ElementName",
  "?reference": "NestedReferenceElement"
}
```

The `name` can be any value, but each `name` **must** be unique within the `TextEditorSection` and there **must** be exactly one `TagElement` with the `name` set to `"text"`. The `type` defines the type of markup the `TagElement` represents:

- *block*: A basic block-level element.
- *wrapping*: A block-level element that contains another block-level element. The name of the inner block-level element **must** be specified in the `content` key.
- *nested*: The root element for a nested document. Nested documents **must** have an `"xml:id"` attribute that specifies the unique identifier for each nested document. This must be in the format `nestedDocumentElementName-UniqueNumber`.
- *inline*: An inline element.
- *mark*: A formatting mark that is attached either to text or to an inline element.

The distinction between inline and mark elements is fluid, but in general you should prefer mark elements for formatting and styling markup and inline elements to mark semantic content.

The `attrs` object maps attribute names (which can by any string value) to `ElementAttribute` configurations that specify how the attribute is parsed and serialised.

Each `TagElement` **must** specify either a single `parser` or a list of `parsers` that specify which TEI tags are mapped to this `TagElement`. The `serialiser` entry configures how the `TagElement` is converted back into a TEI tag.

The `content` **must** and **may only** be specified for a `TagElement` that has the `type` `"wrapping"`. In that case it **must** be set to the `name` of the `TagElement` that may be contained by the wrapping `TagElement`.

The `reference` is specified for any `TagElement` that represents the reference to a nested document and specifies how the two are linked together.

ElementAttribute

The `ElementAttribute` specifies the default value for the attribute, how it is parsed and serialised:

```
{
  "default": "AnyString",
  "?parser": "ParserElement",
  "?parsers": ["ParserElement"],
  "serialiser": "SerialiserElement"
}
```

As with the `TagElement`, either a single `parser` or multiple `parsers` **must** be provided to specify how the attribute is parsed from the TEI document.

Likewise the `serialiser` specifies how the attribute is serialised.

The `default` specifies the default value that is set for the attribute if no valid value can be parsed from the TEI document.

NestedReferenceElement

Editing nested documents consists of two steps. First, the user needs to mark up the text that represents the reference to the nested document. Then they need to edit the nested document. The `NestedReferenceElement` specifies the link from the reference element to the nested document.

```
{
  "type": "ElementName",
  "attr": "AttributeName"
}
```

The `type` specifies the `name` of the `TagElement` that represents the nested documents. The `attr` specifies the attribute on the reference element that contains the nested document's unique identifier.

ParserElement

The `ParserElement` specifies how a `TagElement` or `ElementAttribute` is parsed from the TEI document.

```
{
  "selector": "XpathSelector",
  "?type": "static",
  "?value": "AnyString",
  "?text": "xpath-text-selector"
}
```

The `selector` contains an XPath selector. The selector is configured to require the "tei" prefix on all TEI nodes, for example "tei:head[@type="level-1"]".

When used in the `TagElement` for inline or mark elements, the `text` **may** be used and contains a further XPath selector that specfifies how the text content is to be parsed, relative to the TEI element selected via the `selector` XPath.

When used in the `ElementAttribute`, the attribute's value by default is set to the result of the `selector`. However, if the `type` is specified with the value `"static"`, then if the `selector` matches, the attribute's value is set to the value specified in `value`.

SerialiserElement

The `SerialiserElement` specifies how the `TagElement` or `ElementAttribute` are serialised.

```
{
  "?tag": "AnyString",
  "?attrs": {"AttributeName": "AnyString"},
  "?attr": "AnyString",
  "?value": "SubstitutedString"
}
```

When used in the `TagElement`, the `tag` is used to specify the TEI tag to serialise to. This **must** be prefixed with `"tei"`. In the use with the `TagElement`, you can also use the `attrs` object to specify static attributes that are serialised as specified here.

When used in the `ElementAttribute`, the `attr` is used to specify the name of the attribute to serialise to and the `value` is used to specify the serialised value. The `value` supports substitution, by including the special value `{value}`. By setting the `value` to `"{value}"`, the attribute value specified by the user is serialised as is. However, it is possible to also provide additional text that is serialised as static, for example `"#{value}"` prefixes the user-provided value with a #.

TextEditorUIElement

The `TextEditorUIElement` is the root element for configuring the sidebar for the main and any nested documents.

```
{
  "doc | NestedElementName": ["TextEditorUISection"]
}
```

For the main document, the key **must** be `"doc"`. For nested documents, it **must** be the name of the nested document `TagElement`.

TextEditorUISection

The sidebar is configured as a list of `TextEditorUISection` elements that are then displayed vertically below each other in the editor.

```
{
  "label": "AnyString",
  "entities": ["TextEditorUIBlock"],
  "?condition": "TextEditorUICondition"
}
```

The `label` is displayed as the section heading. Each `TextEditorUIBlock` specified in the `entities` is then shown in the specified order below the `label`.

By default a `TextEditorSection` is always shown to the user. However, if the `condition` is specified, then this changes and the default is that the `TextEditorSection` is hidden and only if the `TextEditorUICondition` holds, is the `TextEditorSection` shown.

TextEditorUIBlock

The `TextEditorUIBlock` configures either a vertical list of input elements or a horizontal menubar.

```
{
  "?type": "menubar | list"
  "entities": ["TextEditorUIEntity"]
}
```

The `type` **must** either be `"menubar"` or `"list"`. Generally `"list"` is only used if you need to have a text input element that needs a label.

The individual UI elements are configured via `TextEditorUIEntity` entries in the `entities`.

TextEditorUIEntity

The `TextEditorUIEntity` configures a single element that modifies the document.

```
{
  "type": "setNodeType | toggleMark | selectNodeAttr | setNodeAttrString |␣
↪setNodeAttrNumber | setNodeAttrValue | toggleNodeAttrValue | selectMarkAttr |␣
↪editNestedDoc | linkNestedDoc | closeNested"
  "label": "AnyHTMLString",
  "nodeType": "ElementName",
  "?ariaLabel": "AnyString",
  "?attr": "AttributeName",
  "?value": "AnyString",
  "?values": ["ValueLabelPair"],
  "?targetNodeType": "NestedElementName",
  "min?": "AnyNumber";
  "max?": "AnyNumber";
  "step?": "AnyNumber";
}
```

The `type` configures the type of user-interface element to show and **must** be one of the following:

- *setNodeType*: Set the type of the current text block to the given `nodeType`. If it is a `block TagElement` then this sets the type for the complete block. If it is an `inline TagElement`, then it is changed for the selection.

In this case if the `TagElement` is already set, then it is removed. If it is a `wrapping TagElement`, then the current text block is set to the content `TagElement` and then wrapped in the wrapping `TagElement`.

- *toggleMark*: Toggles the `mark TagElement` on or off.
- *selectNodeAttr*: Allows the user to select the `TagElement` attribute from a drop-down list. The attribute is specified via the `attr` and the potential values to select from via `values`.
- *setNodeAttrString*: Allows the user to enter the `TagElement` attribute's value into a single-line text input. The attribute is specified via `attr`.
- *setNodeAttrNumber*: Allows the user to enter the `TagElement` attribute's value into a single-line number input. The attribute is specified via `attr`.
- *setNodeAttrValue*: Allows the user to set a fixed `TagElement` attribute value by clicking on a button. The attribute is specified via `attr` and the value to set via `value`.
- *toggleNodeAttrValue*: Toggle the value `value` on the `TagElement` attribute. The attribute is specified via `attr`.
- *selectMarkAttr*: Select a `mark TagElement` attribute value from a drop-down list. The attribute is specified via the `attr` and the potential values to select from via `values`.
- *editNestedDoc*: Edit the nested document linked to the current `inline TagElement`. The attribute that contains the unique identifier of the nested document to edit is specified via `attr`, the type of nested document is specified via the `targetNodeType`.
- *linkNestedDoc*: Select the linked nested document for the current `inline TagElement` from a drop-down list. The attribute that the unique identifier will be set in is specified via `attr`. The type of nested document to link is specified via the `targetNodeType`.
- *closeNested*: Closes the nested document editor.

The `label` is the label shown to the user and can by any HTML content. By providing HTML content, images can be used as the label. If using an image for the `label`, then you **must** provide an `ariaLabel` text for accessibility reasons.

ValueLabelPair

The `ValueLabelPair` is used to specify an entry for a drop-down list.

```
{
  "value": "AnyString",
  "label": "AnyString"
}
```

The `value` is what is stored in the attribute, while the `label` is shown to the user.

TextEditorUICondition

The `TextEditorUICondition` is used to specify a condition under which a specific `TextEditorUISection` is displayed.

```
{
  "type": "isActive",
  "activeType": "ElementName"
}
```

The `type` attribute specifies the type of condition to check. Currently only a single type of condition is implemented. `"isActive"` checks whether the `TagElement` set in the `activeType` is currently active.

5.1.2 MetadataEditorSection

The `MetadataEditorSection` configures the metadata editor. Unlike the `TextEditorSection`, of which there can be multiple, there **must** only be one `MetadataEditorSection`.

```
{
  "label": "AnyString",
  "type": "MetadataEditor",
  "schema": ["MetadataEditorElement"],
  "ui": ["MetadataEditorUISection"]
}
```

The `label` can be any string value and is used in the UI to allow the user to navigate to this section. The `type` must be `"MetadataEditor"`. The `schema` specifies how the metadata is parsed from and serialised to the TEI document. The `ui` specifies how the metadata is displayed to the user.

MetadataEditorElement

The `MetadataEditorElement` is used to convert the TEI header into a tree-structure that is then edited via the UI.

```
{
  "tag": "AnyString",
  "?children": ["MetadataEditorElement"],
  "?text": "DottedPath",
  "?multiple": "Boolean",
  "?attrs": {
    "AttributeName": "DottedPath"
  }
}
```

The `tag` specifies the TEI tag that this `MetadataEditorElement` matches. If it matches, then if any `children` are specified, the matching is applied recursively.

If a `text` is specified and if the matched TEI tag has text content, then the text content is placed into the tree structure at the location specified via the dotted path. If any attributes of the matched TEI element are to be set in the tree, then these are specified in the `attrs` and if the attribute with the given `AttributeName` is set on the TEI element, then that value is set in the tree at the position specified via the dotted path.

MetadataEditorUISection

The `MetadataEditorUISection` is used to visually separate sections of the metadata to edit.

```
{
  "label": "AnyString",
  "entries": ["MetadataEditorUIElement"]
}
```

The `label` is used as the heading that is displayed to the user. The `entries` define the editable UI elements.

MetadataEditorUIElement

The `MetadataEditorUIElement` is used to create the actual interface for editing the metadata.

```
{
  "type": "single-text | multi-field | multi-row",
  "label": "AnyString",
  "path": "DottedPath",
  "?entries": ["MetadataEditorUIElement"]
}
```

The `type` specifies the type of editing interface and **must** be one of `"single-text"`, `"multi-field"`, or `"multi-row"`. The `label` is used to label the input element. The `path` is a dotted path that specifies the location in the tree of the metadata to edit. The optional `entries` allow nesting `MetadataEditorUIElement` to enable editing complex metadata structures.

If the `type` is `"single-text"`, then a simple text-input box is shown to the user. If the `type` is `multi-row`, then the `entries` **must** be specified and define the `MetadataEditorUIElements` that make up one row. If the `type` is `multi-field` then the `entries` **must** be specified and define the `MetadataEditorUIElements` that conceptually belong together.

In general the `multi-field MetadataEditorUIElement` are contained within `multi-row MetadataEditorUIElements`.

The full path for accessing the metadata from the tree structure is calculated by concatenating all the `path` values for the nested `MetadataEditorUIElements`.

5.2 Autoload

The TEI editor supports automatically loading text content that is embedded in the current page. To make use of this, ensure that an element with the id `TEIEditorDocument` exists on the page. The content of that element will automatically be loaded when the editor is mounted.

5.3 Styling

The core TEI editor comes with the minimal styling needed to layout the editor.

5.3.1 Text Editor

All `block` and `wrapping TagElements` are rendered as `<div>` elements, with the `class` attribute set to `"node-{TagElementName}"`. `inline TagElements` are rendered as `<span>` elements, with the `class` attribute set to `"node-{TagElementName}"`.

All `mark TagElements` are rendered as `<span>` elements, with the `class` attribute set to `"mark-{TagElementName}"`.

All `TagElements` attributes are added to the respective `<div>` or `<span>` elements as `data-attributeName="attributeValue"`.

5.3.2 Metadata Editor

For each `MetadataEditorUISection` a `<section>` tag is generated. For `MetadataEditorUIElements` that have a `multi-row` or `multi-field` `type` an `<ol>` tag is generated with the `class` attribute set to the `type`.

Additionally for the `multi-row` elements there is an `<ul role="menubar">` that contains the buttons for adding, removing, and re-ordering the multi-row children.

CHAPTER

SIX

DEMO

Below is a demo showing all features provided by the TEI editor:

The editor above uses the following configuration:

```
{
    "sections": {
        "body": {
            "label": "Main Text",
            "type": "TextEditor",
            "parser": {
                "selector": "tei:text/tei:group/tei:body[@type=\"main\"]"
            },
            "serialiser": {
                "tag": "tei:text/tei:group/tei:body",
                "attrs": {
                    "type": "main"
                }
            },
            "schema": [
                {
                    "name": "paragraph",
                    "type": "block",
                    "attrs": {
                        "alignment": {
                            "default": "left",
                            "parsers": [
                                {
                                    "selector": "contains(@style, 'align-left')",
                                    "type": "static",
                                    "value": "left"
                                },
                                {
                                    "selector": "contains(@style, 'align-center')",
                                    "type": "static",
                                    "value": "center"
                                },
                                {
                                    "selector": "contains(@style, 'align-right')",
                                    "type": "static",
                                    "value": "right"
                                },
                                {
                                    "selector": "contains(@style, 'align-justify')",
                                    "type": "static",
```

(continues on next page)

(continued from previous page)

```
                                "value": "justify"
                            }
                        ],
                        "serialiser": {
                            "attr": "style",
                            "value": "align-{value}"
                        }
                    }
                },
                "parser": {
                    "selector": "tei:p"
                },
                "serialiser": {
                    "tag": "tei:p"
                }
            },
            {
                "name": "heading",
                "type": "block",
                "attrs": {
                    "level": {
                        "default": "1",
                        "parser": {
                            "selector": "substring(@type, 7)"
                        },
                        "serialiser": {
                            "attr": "type",
                            "value": "level-{value}"
                        }
                    },
                    "navIdentifier": {
                        "default": "",
                        "parser": {
                            "selector": "@data-navIdentifier"
                        },
                        "serialiser": {
                            "attr": "data-navIdentifier"
                        }
                    }
                },
                "parser": {
                    "selector": "tei:head"
                },
                "serialiser": {
                    "tag": "tei:head"
                }
            },
            {
                "name": "listItem",
                "type": "block",
                "parser": {
                    "selector": "tei:item"
                },
                "serialiser": {
                    "tag": "tei:item"
                }
            },
```

(continues on next page)

(continued from previous page)

```
                {
                    "name": "list",
                    "type": "wrapping",
                    "content": "listItem",
                    "parser": {
                        "selector": "tei:list"
                    },
                    "serialiser": {
                        "tag": "tei:list"
                    }
                },
                {
                    "name": "text",
                    "type": "inline",
                    "parsers": [
                        {
                            "selector": "tei:seg",
                            "text": "text()"
                        },
                        {
                            "selector": "tei:hi",
                            "text": "text()"
                        }
                    ],
                    "serialiser": {
                        "tag": "tei:seg"
                    }
                },
                {
                    "name": "proof",
                    "type": "inline",
                    "parser": {
                        "selector": "tei:metamark[@function=\"proof\"]",
                        "text": "text()"
                    },
                    "serialiser": {
                        "tag": "tei:metamark",
                        "attrs": {
                            "function": "proof"
                        }
                    }
                },
                {
                    "name": "footnoteMarker",
                    "type": "inline",
                    "parser": {
                        "selector": "tei:ref[@type=\"footnote\"]",
                        "text": "text()"
                    },
                    "serialiser": {
                        "tag": "tei:ref",
                        "attrs": {
                            "type": "footnote"
                        }
                    },
                    "attrs": {
                        "target": {
```

(continues on next page)

(continued from previous page)

```
                            "default": "",
                            "parser": {
                                "selector": "substring(@target, 2)"
                            },
                            "serialiser": {
                                "attr": "target",
                                "value": "#{value}"
                            }
                        }
                    },
                    "reference": {
                        "type": "footnote",
                        "attr": "target"
                    }
                },
                {
                    "name": "footnote",
                    "type": "nested",
                    "parser": {
                        "selector": "tei:note[@type=\"footnote\"]"
                    },
                    "serialiser": {
                        "tag": "tei:note",
                        "attrs": {
                            "type": "footnote"
                        }
                    },
                    "attrs": {
                        "id": {
                            "parser": {
                                "selector": "@xml:id"
                            },
                            "serialiser": {
                                "attr": "xml:id"
                            }
                        },
                        "type": {
                            "default": "footnote"
                        }
                    }
                },
                {
                    "name": "annotationMarker",
                    "type": "inline",
                    "parser": {
                        "selector": "tei:ref[@type=\"annotation\"]",
                        "text": "text()"
                    },
                    "serialiser": {
                        "tag": "tei:ref",
                        "attrs": {
                            "type": "annotation"
                        }
                    },
                    "attrs": {
                        "target": {
                            "default": "",
```

(continues on next page)

(continued from previous page)

```
                    "parser": {
                        "selector": "substring(@target, 2)"
                    },
                    "serialiser": {
                        "attr": "target",
                        "value": "#{value}"
                    }
                }
            },
            "reference": {
                "type": "annotation",
                "attr": "target"
            }
        },
        {
            "name": "annotation",
            "type": "nested",
            "parser": {
                "selector": "tei:interp"
            },
            "serialiser": {
                "tag": "tei:interp"
            },
            "attrs": {
                "id": {
                    "parser": {
                        "selector": "@xml:id"
                    },
                    "serialiser": {
                        "attr": "xml:id"
                    }
                }
            }
        },
        {
            "name": "bold",
            "type": "mark",
            "parser": {
                "selector": "contains(@style, 'bold')"
            },
            "serialiser": {
                "tag": "tei:hi",
                "attrs": {
                    "style": {
                        "value": "bold"
                    }
                }
            }
        },
        {
            "name": "italic",
            "type": "mark",
            "parser": {
                "selector": "contains(@style, 'italic')"
            },
            "serialiser": {
                "tag": "tei:hi",
```

(continues on next page)

(continued from previous page)

```
                        "attrs": {
                            "style": {
                                "value": "italic"
                            }
                        }
                    }
                },
                {
                    "name": "fontSize",
                    "type": "mark",
                    "parsers": [
                        {
                            "selector": "contains(@style, 'font-size-large')"
                        },
                        {
                            "selector": "contains(@style, 'font-size-small')"
                        }
                    ],
                    "serialiser": {
                        "tag": "tei:hi"
                    },
                    "attrs": {
                        "size": {
                            "default": "",
                            "parsers": [
                                {
                                    "selector": "contains(@style, 'font-size-large')",
                                    "type": "static",
                                    "value": "large"
                                },
                                {
                                    "selector": "contains(@style, 'font-size-small')",
                                    "type": "static",
                                    "value": "small"
                                }
                            ],
                            "serialiser": {
                                "attr": "style",
                                "value": "font-size-{value}"
                            }
                        }
                    }
                }
            ],
            "ui": {
                "doc": [
                    {
                        "label": "Blocks",
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "setNodeType",
                                        "label": "Heading",
                                        "nodeType": "heading"
                                    },
```

(continues on next page)

(continued from previous page)

```
                                {
                                    "type": "setNodeType",
                                    "label": "Paragraph",
                                    "nodeType": "paragraph"
                                },
                                {
                                    "type": "setNodeType",
                                    "label": "List",
                                    "nodeType": "list"
                                }
                            ]
                        }
                    ]
                },
                {
                    "label": "Annotations",
                    "entities": [
                        {
                            "type": "menubar",
                            "entities": [
                                {
                                    "type": "setNodeType",
                                    "label": "Proof",
                                    "nodeType": "proof"
                                },
                                {
                                    "type": "setNodeType",
                                    "label": "Footnote",
                                    "nodeType": "footnoteMarker"
                                },
                                {
                                    "type": "setNodeType",
                                    "label": "Annotation",
                                    "nodeType": "annotationMarker"
                                }
                            ]
                        }
                    ]
                },
                {
                    "label": "Markup",
                    "entities": [
                        {
                            "type": "menubar",
                            "entities": [
                                {
                                    "type": "toggleMark",
                                    "label": "<svg viewBox=\"0 0 24 24\" class=\
↪"mdi-icon\"><path d=\"M13.5,15.5H10V12.5H13.5A1.5,1.5 0 0,1 15,14A1.5,1.5 0 0,1 13.
↪5,15.5M10,6.5H13A1.5,1.5 0 0,1 14.5,8A1.5,1.5 0 0,1 13,9.5H10M15.6,10.79C16.57,10.
↪11 17.25,9 17.25,8C17.25,5.74 15.5,4 13.25,4H7V18H14.04C16.14,18 17.75,16.3 17.75,
↪14.21C17.75,12.69 16.89,11.39 15.6,10.79Z\" /></svg>",
                                    "markType": "bold",
                                    "ariaLabel": "Bold"
                                },
                                {
                                    "type": "toggleMark",
```

(continues on next page)

(continued from previous page)

```
                                        "label": "<svg viewBox=\"0 0 24 24\" class=\
↪"mdi-icon\"><path d=\"M10,4V7H12.21L8.79,15H6V18H14V15H11.79L15.21,7H18V4H10Z\" /></
↪svg>",
                                        "markType": "italic",
                                        "ariaLabel": "Italic"
                                    },
                                    {
                                      "type": "selectMarkAttr",
                                      "markType": "fontSize",
                                      "attr": "size",
                                      "values": [{"value": "", "label": "Normal"}, {
↪"value": "small", "label": "Small"}, {"value": "large", "label": "Large"}]
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "label": "Heading",
                        "condition": {
                            "type": "isActive",
                            "activeType": "heading"
                        },
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "selectNodeAttr",
                                        "nodeType": "heading",
                                        "attr": "level",
                                        "values": [{"label": "Level 1", "value": "1"},
↪ {"label": "Level 2", "value": "2"}]
                                    }
                                ]
                            },
                            {
                                "type": "list",
                                "entities": [
                                    {
                                        "type": "setNodeAttrString",
                                        "nodeType": "heading",
                                        "attr": "navIdentifier",
                                        "label": "Navigation Identifier"
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "label": "Paragraph",
                        "condition": {
                            "type": "isActive",
                            "activeType": "paragraph"
                        },
                        "entities": [
                            {
```

(continues on next page)

(continued from previous page)

```
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "setNodeAttrValue",
                                        "nodeType": "paragraph",
                                        "attr": "alignment",
                                        "value": "left",
                                        "label": "Left"
                                    },
                                    {
                                        "type": "setNodeAttrValue",
                                        "nodeType": "paragraph",
                                        "attr": "alignment",
                                        "value": "center",
                                        "label": "Center"
                                    },
                                    {
                                        "type": "setNodeAttrValue",
                                        "nodeType": "paragraph",
                                        "attr": "alignment",
                                        "value": "right",
                                        "label": "Right"
                                    },
                                    {
                                        "type": "setNodeAttrValue",
                                        "nodeType": "paragraph",
                                        "attr": "alignment",
                                        "value": "justify",
                                        "label": "Justify"
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "label": "Footnote",
                        "condition": {
                            "type": "isActive",
                            "activeType": "footnoteMarker"
                        },
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "editNestedDoc",
                                        "nodeType": "footnoteMarker",
                                        "attr": "target",
                                        "label": "Edit",
                                        "targetNodeType": "footnote"
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "label": "Annotation",
```

(continues on next page)

(continued from previous page)

```
                    "condition": {
                        "type": "isActive",
                        "activeType": "annotationMarker"
                    },
                    "entities": [
                        {
                            "type": "menubar",
                            "entities": [
                                {
                                    "type": "linkNestedDoc",
                                    "nodeType": "annotationMarker",
                                    "attr": "target",
                                    "targetNodeType": "annotation"
                                },
                                {
                                    "type": "editNestedDoc",
                                    "nodeType": "annotationMarker",
                                    "attr": "target",
                                    "label": "Edit",
                                    "targetNodeType": "annotation"
                                }
                            ]
                        }
                    ]
                }
            ],
            "footnote": [
                {
                    "label": "Footnote",
                    "entities": [
                        {
                            "type": "menubar",
                            "entities": [
                                {
                                    "type": "closeNested",
                                    "label": "Close"
                                }
                            ]
                        }
                    ]
                },
                {
                    "label": "Markup",
                    "entities": [
                        {
                            "type": "menubar",
                            "entities": [
                                {
                                    "type": "toggleMark",
                                    "label": "Bold",
                                    "markType": "bold"
                                },
                                {
                                    "type": "toggleMark",
                                    "label": "Italic",
                                    "markType": "italic"
                                },
```

(continues on next page)

(continued from previous page)

```
                                    {
                                      "type": "selectMarkAttr",
                                      "markType": "fontSize",
                                      "attr": "size",
                                      "values": [{"value": "", "label": "Normal"}, {
↪"value": "small", "label": "Small"}, {"value": "large", "label": "Large"}]
                                    }
                                ]
                            }
                        ]
                    }
                ],
                "annotation": [
                    {
                        "label": "Annotation",
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "closeNested",
                                        "label": "Close"
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "label": "Markup",
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "toggleMark",
                                        "label": "Bold",
                                        "markType": "bold"
                                    },
                                    {
                                        "type": "toggleMark",
                                        "label": "Italic",
                                        "markType": "italic"
                                    },
                                    {
                                      "type": "selectMarkAttr",
                                      "markType": "fontSize",
                                      "attr": "size",
                                      "values": [{"value": "", "label": "Normal"}, {
↪"value": "small", "label": "Small"}, {"value": "large", "label": "Large"}]
                                    }
                                ]
                            }
                        ]
                    }
                ]
            }
        },
```

(continues on next page)

(continued from previous page)

```
        "apparatus": {
            "label": "Apparatus",
            "type": "TextEditor",
            "parser": {
                "selector": "tei:text/tei:group/tei:body[@type=\"apparatus\"]"
            },
            "serialiser": {
                "tag": "tei:text/tei:group/tei:body",
                "attrs": {
                    "type": "apparatus"
                }
            },
            "schema": [
                {
                    "name": "paragraph",
                    "label": "Paragraph",
                    "type": "block",
                    "attrs": {
                        "alignment": {
                            "default": "left",
                            "parsers": [
                                {
                                    "selector": "contains(@style, 'align-left')",
                                    "type": "static",
                                    "value": "left"
                                },
                                {
                                    "selector": "contains(@style, 'align-center')",
                                    "type": "static",
                                    "value": "center"
                                },
                                {
                                    "selector": "contains(@style, 'align-right')",
                                    "type": "static",
                                    "value": "right"
                                },
                                {
                                    "selector": "contains(@style, 'align-justify')",
                                    "type": "static",
                                    "value": "justify"
                                }
                            ],
                            "serialiser": {
                                "attr": "style",
                                "value": "align-{value}"
                            }
                        }
                    },
                    "parser": {
                        "selector": "tei:p"
                    },
                    "serialiser": {
                        "tag": "tei:p"
                    }
                },
                {
                    "name": "heading",
```

(continues on next page)

(continued from previous page)

```
                "label": "Heading",
                "type": "block",
                "attrs": {
                    "level": {
                        "default": "1",
                        "parser": {
                            "selector": "substring(@type, 7)"
                        },
                        "serialiser": {
                            "attr": "type",
                            "value": "level-{value}"
                        }
                    },
                    "navIdentifier": {
                        "default": "",
                        "parser": {
                            "selector": "@data-navIdentifier"
                        },
                        "serialiser": {
                            "attr": "data-navIdentifier"
                        }
                    }
                },
                "parser": {
                    "selector": "tei:head"
                },
                "serialiser": {
                    "tag": "tei:head"
                }
            },
            {
                "name": "text",
                "type": "inline",
                "parsers": [
                    {
                        "selector": "tei:seg",
                        "text": "text()"
                    },
                    {
                        "selector": "tei:hi",
                        "text": "text()"
                    }
                ],
                "serialiser": {
                    "tag": "tei:seg"
                }
            },
            {
                "name": "proof",
                "type": "inline",
                "parser": {
                    "selector": "tei:metamark[@function=\"proof\"]",
                    "text": "text()"
                },
                "serialiser": {
                    "tag": "tei:metamark",
                    "attrs": {
```

(continues on next page)

(continued from previous page)

```
                    "function": "proof"
                }
            },
            "attrs": {
                "function": {
                    "default": "proof"
                }
            }
        },
        {
            "name": "bold",
            "type": "mark",
            "parser": {
                "selector": "contains(@style, 'bold')"
            },
            "serialiser": {
                "tag": "tei:hi",
                "attrs": {
                    "style": {
                        "value": "bold"
                    }
                }
            }
        },
        {
            "name": "italic",
            "type": "mark",
            "parser": {
                "selector": "contains(@style, 'italic')"
            },
            "serialiser": {
                "tag": "tei:hi",
                "attrs": {
                    "style": {
                        "value": "italic"
                    }
                }
            }
        },
        {
            "name": "fontSize",
            "type": "mark",
            "parsers": [
                {
                    "selector": "contains(@style, 'font-size-large')"
                },
                {
                    "selector": "contains(@style, 'font-size-small')"
                }
            ],
            "serialiser": {
                "tag": "tei:hi"
            },
            "attrs": {
                "size": {
                    "default": "",
                    "parsers": [
```

(continues on next page)

(continued from previous page)

```
                                {
                                    "selector": "contains(@style, 'font-size-large')",
                                    "type": "static",
                                    "value": "large"
                                },
                                {
                                    "selector": "contains(@style, 'font-size-small')",
                                    "type": "static",
                                    "value": "small"
                                }
                            ],
                            "serialiser": {
                                "attr": "style",
                                "value": "font-size-{value}"
                            }
                        }
                    }
                }
            ],
            "ui": {
                "doc": [
                    {
                        "label": "Blocks",
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "setNodeType",
                                        "label": "Heading",
                                        "nodeType": "heading"
                                    },
                                    {
                                        "type": "setNodeType",
                                        "label": "Paragraph",
                                        "nodeType": "paragraph"
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "label": "Annotations",
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                       "type": "setNodeType",
                                       "label": "Proof",
                                       "nodeType": "proof"
                                    }
                                ]
                            }
                        ]
                    },
                    {
```

(continues on next page)

(continued from previous page)

```
                        "label": "Markup",
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "toggleMark",
                                        "label": "Bold",
                                        "markType": "bold"
                                    },
                                    {
                                        "type": "toggleMark",
                                        "label": "Italic",
                                        "markType": "italic"
                                    },
                                    {
                                      "type": "selectMarkAttr",
                                      "markType": "fontSize",
                                      "attr": "size",
                                      "values": [{"value": "", "label": "Normal"}, {"value": "small", "label": "Small"}, {"value": "large", "label": "Large"}]
                                    }
                                ]
                            }
                        ]
                    },
                    {
                        "label": "Heading",
                        "condition": {
                            "type": "isActive",
                            "activeType": "heading"
                        },
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "selectNodeAttr",
                                        "nodeType": "heading",
                                        "attr": "level",
                                        "values": [{"label": "Level 1", "value": "1"}, {"label": "Level 2", "value": "2"}]
                                    }
                                ]
                            },
                            {
                                "type": "list",
                                "entities": [
                                    {
                                        "type": "setNodeAttrString",
                                        "nodeType": "heading",
                                        "attr": "navIdentifier",
                                        "label": "Navigation Identifier"
                                    }
                                ]
                            }
                        ]
```

(continues on next page)

(continued from previous page)

```
                    },
                    {
                        "label": "Paragraph",
                        "condition": {
                            "type": "isActive",
                            "activeType": "paragraph"
                        },
                        "entities": [
                            {
                                "type": "menubar",
                                "entities": [
                                    {
                                        "type": "setNodeAttrValue",
                                        "nodeType": "paragraph",
                                        "attr": "alignment",
                                        "value": "left",
                                        "label": "Left"
                                    },
                                    {
                                        "type": "setNodeAttrValue",
                                        "nodeType": "paragraph",
                                        "attr": "alignment",
                                        "value": "center",
                                        "label": "Center"
                                    },
                                    {
                                        "type": "setNodeAttrValue",
                                        "nodeType": "paragraph",
                                        "attr": "alignment",
                                        "value": "right",
                                        "label": "Right"
                                    },
                                    {
                                        "type": "setNodeAttrValue",
                                        "nodeType": "paragraph",
                                        "attr": "alignment",
                                        "value": "justify",
                                        "label": "Justify"
                                    }
                                ]
                            }
                        ]
                    }
                ]
            }
        },
        "metadata": {
            "label": "Metadata",
            "type": "MetadataEditor",
            "schema": [
                {
                    "tag": "tei:fileDesc",
                    "children": [
                        {
                            "tag": "tei:titleStmt",
                            "children": [
                                {
```

(continues on next page)

(continued from previous page)

```
                                    "tag": "tei:title",
                                    "text": "fileDesc.titleStmt.title._text"
                                },
                                {
                                    "tag": "tei:author",
                                    "text": "fileDesc.titleStmt.author._text"
                                },
                                {
                                    "tag": "tei:respStmt",
                                    "multiple": true,
                                    "attrs": {
                                        "xml:id": "_attrs.xml:id"
                                    },
                                    "children": [
                                        {
                                            "tag": "tei:resp",
                                            "text": "resp._text",
                                            "multiple": true
                                        },
                                        {
                                            "tag": "tei:name",
                                            "text": "name._text",
                                            "multiple": true
                                        }
                                    ]
                                }
                            ]
                        },
                        {
                            "tag": "tei:publicationStmt",
                            "children": [
                                {
                                    "tag": "tei:distributor",
                                    "text": "fileDesc.publicationStmt.distributor._
↪text"
                                }
                            ]
                        }
                    ]
                }
            ],
            "ui": [
                {
                    "label": "Bibliography",
                    "entries": [
                        {
                            "type": "single-text",
                            "label": "Title",
                            "path": "fileDesc.titleStmt.title._text"
                        },
                        {
                            "type": "single-text",
                            "label": "Author",
                            "path": "fileDesc.titleStmt.author._text"
                        }
                    ]
                },
```

(continues on next page)

(continued from previous page)

```
                {
                    "label": "Digital Version",
                    "entries": [
                        {
                            "type": "single-text",
                            "label": "Distributor",
                            "path": "fileDesc.publicationStmt.distributor._text"
                        }
                    ]
                },
                {
                    "label": "Responsibilities",
                    "entries": [
                        {
                            "type": "multi-row",
                            "path": "fileDesc.titleStmt.respStmt",
                            "entries": [
                                {
                                    "type": "multi-field",
                                    "path": "",
                                    "entries": [
                                        {
                                            "type": "single-text",
                                            "label": "Identifier",
                                            "path": "._attrs.xml:id"
                                        },
                                        {
                                            "type": "multi-row",
                                            "path": ".resp",
                                            "entries": [
                                                {
                                                    "type": "single-text",
                                                    "label": "Responsible for",
                                                    "path": "._text"
                                                }
                                            ]
                                        },
                                        {
                                            "type": "multi-row",
                                            "path": ".name",
                                            "entries": [
                                                {
                                                    "type": "single-text",
                                                    "label": "Name",
                                                    "path": "._text"
                                                }
                                            ]
                                        }
                                    ]
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    }
```

(continues on next page)

(continued from previous page)

```
}
```

The editor also uses the following default document:

```
<script id=TEIEditorDocument type=application/xml+tei>
  <tei:TEI xmlns:tei="http://www.tei-c.org/ns/1.0">
    <tei:teiHeader>
      <tei:fileDesc>
        <tei:titleStmt>
          <tei:title>Test Document</tei:title>
          <tei:author>Mark Hall</tei:author>
          <tei:respStmt xml:id="markhall">
            <tei:resp>Most things</tei:resp>
            <tei:resp>Some other things</tei:resp>
            <tei:name>Mark Hall</tei:name>
            <tei:name>Mark M Hall</tei:name>
          </tei:respStmt>
        </tei:titleStmt>
        <tei:publicationStmt>
          <tei:distributor>Mark Hall</tei:distributor>
        </tei:publicationStmt>
      </tei:fileDesc>
    </tei:teiHeader>
    <tei:text>
      <tei:group>
        <tei:body type="main">
          <tei:head type="level-1" data-navIdentifier="heading-1">
            <tei:seg>TEI Editor Test</tei:seg>
          </tei:head>
          <tei:p style="align-left">
            <tei:seg>This text is </tei:seg>
            <tei:hi style="bold">bold</tei:hi>
            <tei:seg> and </tei:seg>
            <tei:hi style="bold italic">bold-italic</tei:hi>
            <tei:seg> and left aligned.</tei:seg>
          </tei:p>
          <tei:p style="align-right">
            <tei:seg>This text is right aligned.</tei:seg>
          </tei:p>
          <tei:p>
            <tei:seg>Here is some </tei:seg>
            <tei:hi style="font-size-large">large</tei:hi>
            <tei:seg> and </tei:seg>
            <tei:hi style="font-size-small">small</tei:hi>
            <tei:seg> text.</tei:seg>
          </tei:p>
          <tei:list>
            <tei:item>
              <tei:seg>A bullet point</tei:seg>
            </tei:item>
            <tei:item>
              <tei:seg>Another bullet point</tei:seg>
            </tei:item>
          </tei:list>
          <tei:p>
            <tei:metamark function="proof">This is proven.</tei:metamark>
          </tei:p>
```

(continues on next page)

(continued from previous page)

```
          <tei:p>
            <tei:seg>This line has a </tei:seg>
            <tei:ref type="footnote" target="#footnote-1">footnote</tei:ref>
            <tei:seg>.</tei:seg>
          </tei:p>
          <tei:p>
            <tei:seg>This is </tei:seg>
            <tei:ref type="annotation" target="#annotation-1">annotated</tei:ref>
            <tei:seg>.</tei:seg>
          </tei:p>
          <tei:p>
            <tei:seg>Unlike the footnote, the </tei:seg>
            <tei:ref type="annotation" target="#annotation-1">annotation</tei:ref>
            <tei:seg> can be referenced multiple times.</tei:seg>
          </tei:p>
          <tei:note xml:id="footnote-1" type="footnote">
            <tei:p>
              <tei:seg>This is </tei:seg>
              <tei:hi style="bold">just</tei:hi>
              <tei:seg> a simple footnote.</tei:seg>
            </tei:p>
          </tei:note>
          <tei:interp xml:id="annotation-1">
            <tei:p>
              <tei:hi style="bold">1, annotation</tei:hi>
              <tei:seg>]</tei:seg>
            </tei:p>
            <tei:p>
              <tei:seg>An annotation is a piece of commentary on the text.</tei:seg>
            </tei:p>
          </tei:interp>
        </tei:body>
        <tei:body type="apparatus">
          <tei:head type="level-1">
            <tei:seg>Apparatus</tei:seg>
          </tei:head>
          <tei:p>
            <tei:seg>This section contains comments that the annotator thought were </
↪tei:seg>
            <tei:hi style="bold">important</tei:hi>
            <tei:seg>.</tei:seg>
          </tei:p>
        </tei:body>
      </tei:group>
    </tei:text>
  </tei:TEI>
</script>
```

The TEI Editor is MIT licensed.

Documentation https://tei-editor.readthedocs.io/en/latest/

Source Code https://github.com/scmmmh/tei-editor

NPM Package https://www.npmjs.com/package/tei-editor

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search