Technical specification for the feature: Documentation

This is the technical specification for the Documentation feature. The feature was created as part of an EU-funded project called We4Authors Cluster. All features, their technical specification and their video documentation are published on www.accessibilitycluster.com.

If you have any questions or comments, please do not hesitate to reach out to research@funka.com.

The technical specification contains several perspectives:

- the web interface of the feature, based on the experience of the end user or, when it comes to the testing-features, the web author;
- the code of the feature ensuring accessibility by default;
- a description highlighting the key elements of the code;
- reference to a video documentation;
- recommendations for implementation.

Specifications of the feature: Documentation

Supporting the creation of accessible content when more information is needed.

Documentation of accessibility features is important for all authors. This feature provides short contextual information and the possibility to have more granular information provided separately.

The contextual documentation may be followed by a simple icon indicating that more detailed information or help is available. The icon provides access to a popup or modal dialog that allows the web author to learn more about the specific issue.

In addition to this, a section in the /admin/help documentation within the authoring tool explaining accessibility features is recommended.

The documentation must be provided in an easy to read and accessible format, providing good overview and the possibility to search.

Web interface: web author view

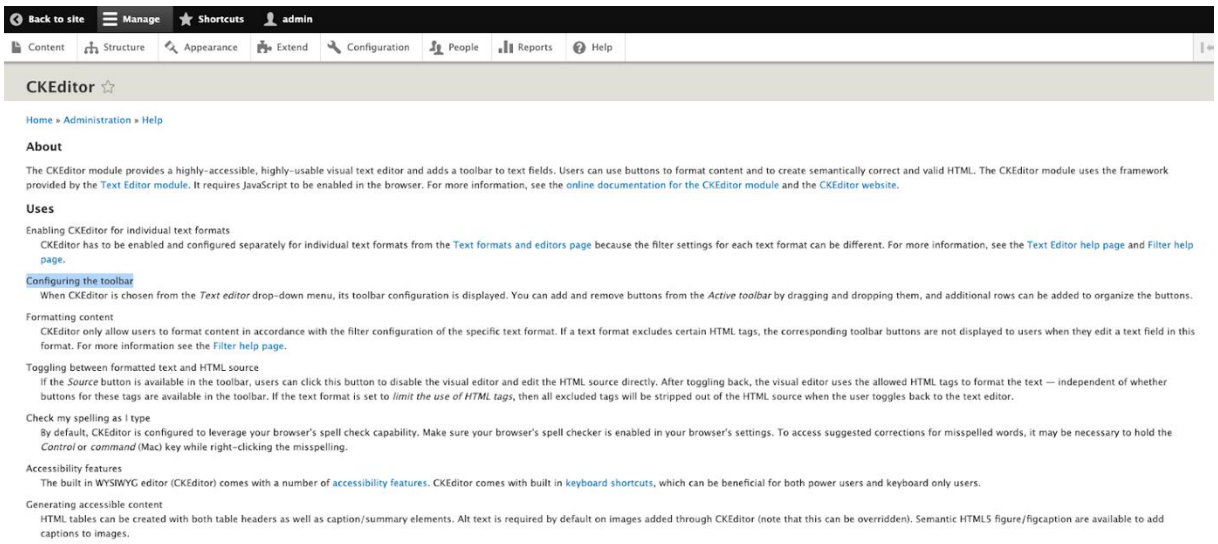

Figure 1: Web author view - example of documentation in CKEditor

Code of web author view

The only common framework for authoring tools is JavaScript. The details will be different per tool, but Scott O'Hara outlined a process for creating Accessible Modal Dialogs with Vanilla JS that can be re-used everywhere. A user could then use a button (that looks like 🙋):

```
<button type="button" class="_your_class(es)_here_"
 data-modal-open="IDREF_of_modal_to_open" disabled>
  <span class="sr-only">Need help?</span>
</button>
```

This launches a dialog that would include:

```
<div id="unique_ID_to_match_data-modal-open" data-modal>
 <h1>
  A descriptive title for the dialog
 </h1>
 <div>
  <!-- primary content of the dialog here -->
 </div>
</div>
```

There are also other models, such as the one posted by Alena Nik titled "Inclusive components: making modals accessible":
https://dev.to/alenanik/inclusive-components-making-modals-accessible-1hn9

Or Ire Aderinokun's post, "Creating An Accessible Modal Dialog":
https://bitsofco.de/accessible-modal-dialog/

Technically, this is how Drupal added help to the CKEditor Help page:
https://www.drupal.org/project/drupal/issues/3150364

Video documentation

This technical specification has been developed in the We4Author Cluster project to reflect recommendations for accessibility features that can be implemented in any authoring tool. The specifications are complemented by a video documentation, covering a live description of:

- web author challenges, and
- feature solutions.

Recommendations for implementation

To make sure the implementation of the features is not causing accessibility problems for web authors with disabilities:

- avoid drag-and-drop for choosing template, and/or always have a keyboard alternative;
- always provide one pointer alternative without specific gestures;
- always support keyboard navigation;
- consider keyboard shortcuts;
- do not rely on sensory characteristics as the sole indicator for understanding and operating content;
- do not indicate important information using colour alone;
- make sure there is enough contrast between text objects and its background colour.