Initializing WProofreader in TinyMCE 5

WProofreader add-on for rich text editors enables grammar and spell check, combining spelling and grammar suggestions while you type or work with your text in a floating dialog.

Its auto-searching feature enables detecting new editable fields on the page and proofreading the text they contain automatically on hover selection. No additional actions or plugins are required for enabling WProofreader in a specific WYSIWYG editor or HTML editable element. A single configuration applies to almost all editors and controls. Here is the list of WProofreader supported integrations.

This guide outlines the main steps that you need to follow for initializing WProofreader in TinyMCE 5 rich rext editor.

1. Initialize TinyMCE 5

Before integrating and configuring WProofreader, make sure TinyMCE 5 is properly configured in your web application. To initialize TinyMCE 5, follow the steps described in the Quick start.

3. Initialize WProofreader

Choose one of the options below how to enable the spelling and grammar checking functionality of WProofreader in the rich text editor.

Option A. Using autoSearch (the easiest)

When using this method, you need a **CONFIG** and a **wscbundle.js** script. After setting up the **autoSearch**, WProofreader will be enabled as soon as the editable container with TinyMCE is in focus. Follow the steps described in the Get started with WProofreader Server (autoSearch).

To limit WProofreader work only to TinyMCE 5 areas, you can use the **enableAutoSearchIn** option where you need to indicate the class of TinyMCE 5.

```
enableAutoSearchIn: ['..tox-edit-area_ _iframe'],
```

Option B. Using the init method

When using this method, WProofreader will be started on the editor load.

Initializing WProofreader using init() in TinyMCE 5

```
<!-- Include the WEBSPELLCHECKER_CONFIG variable. -->
<script>
    window.WEBSPELLCHECKER_CONFIG = {
                autoSearch: true,
                autoDestroy: true,
                ...
    };
</script>

<script type="text/javascript" src="http(s)://host_name/wscservice/wscbundle/wscbundle.js"></script>


<div id="tinymce5-editor">
    <p>These are an examples of a sentences with two mispelled words and grammar problems. Just type text with
misspelling to see how it works.</p>
</div>

<script>
    tinymce.init({
     selector:'div',
     init_instance_callback : function (tinymce_editor){
     WEBSPELLCHECKER.init({
               container: tinymce_editor.iframeElement
              })
          },
   });
</script>
```

4. Specify additional server parameters

Make sure you have specified the following server-specific options in CONFIG, which are instructing where spelling and grammar checking requests will be processed:

```
serviceProtocol: 'https',
serviceHost: 'your_host_name',
servicePort: '443',
servicePath: 'virtual_directory/api' //by default the virtual directory name is wscservice
```

5. Remove other spellcheckers

⚠ If you used to have other spellcheck tools, please make sure you removed everything to avoid any conflicts or confusion.

6. Further actions

After completion of the initialization steps, you can customize WProofreader options, its user interface, and behavior using WProofreader customization options. More options and methods for advanced customization are available in API docs.

Still have a question?

Please feel free to contact our technical team if you are having any difficulties with the configuration.