

Configuring WProofreader Server in WYSIWYG editors

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](#).

Supported integrations

The autoSearch 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 control. A single configuration applies to almost all editors and controls.

HTML integration	WYSIWYG editors
HTML editable controls: <input> disabled by default <textarea> Elements with contenteditable attribute set to 'true': <div> <iframe>	• CKEditor 4 • CKEditor 5* • TinyMCE 4,5 • Froala Editor • Kendo UI • Quill • Redactor • Summernote • RadEditor -Telerik ASP.NET Editor • ProseMirror • Trix • Slate.js** • Draft**  * The dialog mode of WProofreader is not supported with real-time collaboration features or multi-root editor modes of CKEditor 5. ** Integration with Slate.js and Draft won't work properly with IE11.

WProofreader also has a plugin for WordPress 4.x and WordPress 5.x. For more details, check the official repository of [WProofreader plugin for WordPress](#).

1. Initialize WProofreader

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

Option A. Using autoSearch



The fastest and easiest approach for WProofreader initialization in a rich text editor.

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

Option . Using the init method

When using this method, WProofreader will be started on the editor load. For details, refer to one of the guides explaining how to use the init() method with the most popular rich text editors:

- [Initializing WProofreader in CKEditor 4](#)
- [Initializing WProofreader in CKEditor 5 v12.0+](#)
- [Initializing WProofreader in Froala Editor 3](#)
- [Initializing WProofreader in TinyMCE 5](#)
- [Getting started with WProofreader plugin for CKEditor 5](#)
- [Integrating WProofreader into Angular, React, Vue.js-based apps](#)

2. WProofreader in action

Wonder how it works? Check demos [here](#).