

Initializing WProofreader in CKEditor 4

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 CKEditor 4.

1. Initialize CKEditor 4

Before integrating and configuring WProofreader, make sure CKEditor 4 is properly configured in your web application. To initialize CKEditor 4, follow the steps described in [CKEditor 4 Quick Start Guide](#).

2. Initialize WProofreader

Choose one of the options below how to enable spelling and grammar checking functionality of WProofreader in CKEditor 4.

Option A. Using autoSearch

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 CKEditor is in focus. Check a sample below or follow the steps described in the [Get started with WProofreader Server \(autoSearch\)](#).

```
<script>
    window.WEBSPELLCHECKER_CONFIG = {
        autoSearch: true,
            lang: 'auto', //enable language auto-detection option
            theme: 'gray', //adjust color theme of WProofreader to look more native to CKEditor 4
            enableAutoSearchIn: ['.cke_wysiwyg_frame'], //limit WProofreader work only to CKEditor 4. They
hide the original element and create an iframe element with cke_wysiwyg_frame class.
            ...
    };
</script>

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

Option B. Using the init method

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

Initializing WProofreader using init() in CKEditor 4

```

<!-- 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>

<!-- Use this path for the Cloud-based version
<script type="text/javascript" src="https://svc.webspellchecker.net/spellcheck31/wscbundle/wscbundle.js"><
/script>
-->

<div id="ckeditor4-editor">
    <p>These are an examples of a sentences with two misspelled words and grammar problems. Just type text with
    misspelling to see how it works.</p>
</div>

<script>
    CKEDITOR.disableAutoInline = true;

    CKEDITOR.on('instanceReady', function(event) {
        var editor = event.editor;

        WEBSPELLCHECKER.init({
            //catch both iframe and inline modes
            container: editor.window.getFrame() ? editor.window.getFrame().$ : editor.element.$
        });
    });

    CKEDITOR.replace('ckeditor4-editor', {});
</script>

```

3. 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

```

4. Remove old plugins



Remove SCAYT and WSC Dialog plugins for CKEditor 4 which are integrated by default to avoid any possible issues. It can be done by adding the **removePlugins** parameter to CKEditor **config.js** file:

CKEditor config.js:

```
config.removePlugins = 'scayt,wsc';
```

5. 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](#).

You can see the [demo of CKEditor 4 integration](#) on our website.

Still have a question?

Please feel free to contact our [technical team](#) if you are having any difficulties with the configuration.

Can also be useful:

- [API docs](#)
- [WProofreader customization options](#)
- [Get started with WProofreader Server \(autoSearch\)](#)