

`_include_wproofreader_plugin_ckeditor5`

WProofreader plugin for CKEditor 5 inherits all the functionality of the WProofreader component with slight adaptation to the view and features of the editor. Right now the plugin is just a “wrapper” for the classic WProofreader. Thus, WProofreader will be more native to CKEditor 5 build procedure.

The main differences between the plugin and the classic WProofreader are:

- part of the build and configuration of CKEditor 5;
- tailored color theme for CKEditor 5 based on previously discussed styles;
- implemented sync in the case of real-time collaboration features and multi-root plugin;
- better work of WProofreader as you type mode in the case of multi-root plugin (languages and settings synchronization between instances).

For more details, please refer to the [official repo](#) or [NPM package](#) of the WProofreader plugin for CKEditor 5.

If your preference is to use the plugin instead of the classic WProofreader initialization, please follow the steps below.

1. Clone [CKEditor 5 Classic editor](#):

```
git clone -b stable git@github.com:ckeditor/ckeditor5-build-classic.git
```

2. Go to the editor folder:

```
cd ckeditor5-build-classic
```

3. Install all the required dependencies:

```
npm install
```

4. Install the WProofreader plugin:

```
npm install @webspellchecker/wproofreader-ckeditor5 --save-dev
```

5. Open `/src/ckeditor.js` file.

6. Import the WProofreader plugin:

```
import WProofreader from '@webspellchecker/wproofreader-ckeditor5/src/wproofreader';
```



To add WProofreader in a bundle with CKEditor 5 with **TypeScript** you need to use the following:

```
//@ts-ignore  
  
import WProofreader from '@webspellchecker/wproofreader-ckeditor5/src/wproofreader';
```

7. Add WProofreader plugin to **ClassicEditor.builtinPlugins**:

```
ClassicEditor.builtinPlugins = [  
    ...,  
    WProofreader  
];
```

8. Add WProofreader toolbar button to **ClassicEditor.defaultConfig**:



The toolbar button is available starting from [version 2.0.0 of the plugin](#) and the general [WebSpellChecker package version 5.8.0](#).

```
ClassicEditor.defaultConfig = {
  ...,
  toolbar: {
    items: [
      ...,
      'wproofreader'
    ]
  }
}
```

9. Add the **wproofreader** field and its options to **ClassicEditor.defaultConfig**:

In the case of the Cloud version, **serviceId** and **srcUrl** are mandatory options.

```
ClassicEditor.defaultConfig = {
  ...,
  wproofreader: {
    lang: 'en_US', // set the default language
    serviceId: 'your-service-ID', // the activation key for the Cloud-based version only
    srcUrl: 'https://svc.webspellchecker.net/spellcheck31/wscbundle/wscbundle.js'
  }
}
```

In the case of the server-based version, you need to clearly define a path to the WebSpellChecker/WProofreader server:

```
ClassicEditor.defaultConfig = {
  ...,
  wproofreader: {
    lang: 'en_US', // set the default language
    serviceProtocol: 'https',
    serviceHost: 'your_host_name',
    servicePort: '443',
    servicePath: 'virtual_directory/api',
    srcUrl: 'https://your_host_name/virtual_directory/wscbundle/wscbundle.js'
  }
}
```



The full list of configuration options for WProofreader is available in [API docs](#).

10. Rebuild the bundle:

```
npm run build
```

That's it. Now you should be able to use the spelling and grammar checking functionality of WProofreader with CKEditor 5.

Try it out on the default sample page: [/sample/index.html](#).