

WProofreader customization options

Before customizing the WProofreader settings described in this section:

- Initialize it using one of the approaches in [Initialization Options](#) section.
- Acknowledge yourself with the full list of customization options available in [API docs](#).

Being an administrator, you can customize WProofreader default settings such as user interface and application behavior described in this guide.

This guide covers the following topics:

1. [Introduction](#)
 - 1.1. [Server and Cloud version](#)
 - 1.2. [AutoSearch options](#)
 - 1.3. [AutoStartup](#)
 - 1.4. [Deactivating grammar checking](#)
2. [Customizing user interface](#)
 - 2.1. [Change the default style theme](#)
 - 2.2. [Suggestions pop-up view and commands](#)
 - 2.3. [WProofreader badge](#)
 - 2.4. [Settings dialog](#)
 - 2.5. [Actions for proofreading in a dialog](#)
3. [Languages and dictionaries settings](#)
 - 3.1. [Default language options](#)
 - 3.2. [Default language for all users](#)
 - 3.3. [User interface localization language](#)
 - 3.4. [Predefined user dictionary](#)
 - 3.5. [Extending the list of dictionaries](#)
 - 3.6. [Custom dictionaries](#)
4. [Miscellaneous options](#)
 - 4.1. [Minimal word length](#)
 - 4.2. [Excluding elements from checking](#)
 - 4.3. [Disabling proofreading in As You Type mode](#)

1. Introduction

1.1. Server and Cloud version

In general, WProofreader provides the same functionality for both Cloud and Server versions. The difference is mainly about the location of requests processing: it can be either on WebSpellChecker Cloud hosted on [Amazon Web Services](#) (AWS) in data centers in the USA or on your servers such as physical servers, VMs or instances.

With Cloud version, to access the services, you need to use a special activation key in **serviceld** parameter. The path to WebSpellChecker Cloud will be obtained from the path to the **wscbundle.js** file.

For the Server version, you need to instruct WProofreader where the requests will be processed by specifying the following parameters:

- **serviceProtocol**
- **serviceHost**
- **servicePath**
- **servicePort**

Example of setting up the Server version of WProofreader with the **autoSearch** option turned on:

```
<script>
  window.WEBSPELLCHECKER_CONFIG = {
    autoSearch: true,
    serviceProtocol: 'https',
    serviceHost: 'your_host_name',
    servicePort: '443',
    servicePath: 'wscservice/api'
  };
</script>
```

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

For more details, refer to [Get Started with WProofreader Server AutoSearch](#).

Example for setting up the **Cloud** version of WProofreader with the autoSearch option turned on:

```
<script>
  window.WEBSPELLCHECKER_CONFIG = {
    autoSearch: true,
    serviceId: '1:your-service-ID'
  };
</script>
```

```
<script type="text/javascript" src="https://svc.webspellchecker.net/spellcheck31/wscbundle/wscbundle.js"><
/script>
```

For more details, refer to [Get Started with WProofreader Cloud AutoSearch](#).

1.2. AutoSearch options

Once you have enabled autoSearch by adding **autoSearch: true**, in configuration, you can specify where the spelling and grammar check should be enabled on your webpage.

By default, WProofreader with autoSearch function turned on will be enabled only in the area in focus. For the areas which are not focused, WProofreader will not be enabled by default until user places a cursor there and starts typing. For details, refer to the following [AutoSearch Creation options](#):

- **disableAutoSearchIn** lets you disable the autoSearch mechanism by **class, id, data attribute name, and HTML elements**. Please note that if **enableAutoSearchIn** option is specified, this option will be ignored.
- **enableAutoSearchIn** lets you enable the autoSearch mechanism only for elements with provided **class, id, data attribute name or HTML elements type**.



Only one option can be used at the same time. The **enableAutoSearch** option has higher priority compared with **disableAutoSearchIn**. If you try using both of them, it can lead to unpredictable behavior and conflicts.

```
<script>
  window.WEBSPELLCHECKER_CONFIG = {
    autoSearch: true,
    disableAutoSearchIn: ['.class', '#id', '[data-attribute]', 'textarea'],
    enableAutoSearchIn: ['.class', '#id', '[data-attribute]', 'textarea'],
    ...
  }
</script>
```

1.2.1 Enable proofreading in inputs

By default, WProofreader is disabled in HTML **<input>** editable elements due to possible compatibility issues. However, you can change this behavior by using the **enableAutoSearchIn** option:

```
enableAutoSearchIn: ['input'],
```

1.3. AutoStartup

The **autoStartup** parameter is set to **true** by default, meaning that WProofreader starts in the enabled state and starts processing available text for grammar, spelling, and style errors. If, for some reason, you do not want WProofreader to start checking text at once and allow your users to choose when to start, you have an option to start WProofreader in the disabled state. To do so, set the **autoStartup** value as **false**, as shown below:

```
autoStartup: false,
```



When WProofreader starts in a disabled state, the badge is grayed out. To activate it, users need to go to the badge and click the turn on icon.

1.4. Deactivating grammar checking

Grammar checking is enabled by default in WProofreader. If you want to deactivate grammar checking, specify the **enableGrammar** parameter and set it to false as shown below:

```
enableGrammar: false,
```



After you have deactivated grammar checking, it becomes disabled for all languages. Only spell check will be performed then.

2. Customizing user interface

With [WProofreader API](#) you can customize:

- Change the default style theme;
- Suggestions balloon view and commands available;
- Badge view and commands available;
- Settings dialog view and additional preferences;
- Actions for Proofread in dialog mode.



Here you can find the [full list of API options and methods](#) available for WProofreader.

2.1. Change the default style theme

You have an option to change the default style theme using **theme** option. At the moment, the following themes are available: **default**, **gray**, **dark**, **custom**. The **gray** theme perfectly matches the color palette of modern rich text editors.

Besides, you can adjust CSS styles to make them look native inside your web app. To do this, simply use the **custom** theme option and write your CSS styles. Find out more in our how-to guide: [How to customize the look and feel of WProofreader?](#)

```
theme: 'gray',
```

2.2. Suggestions pop-up view and commands

You can customize WProofreader options available in the suggestions pop-up. For example:

- Limit the number of suggestions shown to users;
- Prevent users from adding words to the user custom dictionary;
- Add more commands to the list of action items, for example, the possibility to switch to the **Settings** dialog.



The latter is possible only in case of disabled badge. All the action items from the badge are moved to the suggestions pop-up then. For details, see [Changing the Badge View and Options](#) section.

2.2.1. Changing the number of spelling suggestions

By default, WProofreader offers 3 suggestions in its suggestions pop-up. An admin can customize this number using an API option and provide both bigger and smaller number of suggestions.

For Cloud version, up to 8 suggestions are available. For the Server version, this number can be increased. You can use the **suggestionsCount** parameter to increase or decrease the number of spelling suggestions shown to your end users.



Displaying a significant number of suggestions may decrease the speed of the service.

To display only 2 suggestions:

```
suggestionsCount: 2,
```

2.2.2. Changing action items order and availability

Use the **actionItems** parameter to add or hide the menu items in the suggestion pop-up and modify the order of these items. To do so, change the **'addWord'**, **'ignoreAll'**, **'ignore'**, **'settings'**, **'toggle'**, **'proofreadDialog'** array values accordingly.

For example, if you want to prevent your users from adding words to their user custom dictionary, remove the 'addWord' value from the array of parameter values:

```
actionItems: ['ignoreAll', 'ignore', 'settings', 'toggle', 'proofreadDialog'],
```



Removing the **addWord** from the array of parameter values removes the **Add word** command for both suggestion balloon and for proofreading in dialog mode.

To hide **Settings** dialog for your users, remove the 'settings' from the array of parameter values:

```
actionItems: ['addWord', 'ignoreAll', 'toggle', 'proofreadDialog'],
```

2.2.3. Enabling or disabling the more suggestions sub-menu item

By default, the **moreSuggestionsCount** parameter is set to 0, meaning that additional suggestions, if there are any more than specified in the **suggestion sCount** parameter are not shown. To display 3 suggestions on the suggestions balloon view and show 5 more suggestions in the **More Suggestions** section:

```
suggestionsCount:3,  
moreSuggestionsCount:5,
```

2.3. WProofreader badge

The badge in the lower-right part of the editable area shows the status of the checking process (namely, the number of errors, the process itself – spinner, all checked) and allows to access more options:

- Turn on/off the proofreading;
- Switch to the 'Proofread in dialog' mode;
- Switch to the Settings dialog to manage spell checking options, language, and user custom dictionary;
- Promptly select a language for check.

2.3.1. Badge view and options

You can change the following badge options:

- Turn off badge pulsation,
- Completely remove a badge,
- Remove some actions from the badge, for example, **Settings** , **Disable/enable WProofreader** , or **Proofread in dialog** .

2.3.2. Turning off badge pulsation

When WProofreader starts, the badge starts pulsing three times to get the user's attention to the discovered issues. If it turns out to be distracting for the users, you can disable pulsation by setting the **disableBadgePulsation** parameter as **true** as shown in the sample below:

```
disableBadgePulsation:true,
```

2.3.3. Removing a badge

If the **enableBadgeButton** parameter is set **false**, part of action items, specifically **Settings**, **Disable/Enable** and **Proofread in dialog** from the badge area will be moved to the suggestions pop-up:

```
enableBadgeButton:false,
```



Users will not have an indication of the total number of errors detected in the control. The spinner that shows the grammar and spelling checking process is also disabled.

2.3.4. Removing actions from badge

To remove some actions from the badge, for example, **Settings** , **Disable/enable WProofreader** , or **Proofread in dialog** , use the **actionItems** parameter. An example below shows how you can allow the users to access proofreading in dialog only from the badge:

```
actionItems: [ 'proofreadDialog'],
```

2.3.5. Enabling language selector in badge

There is an option available that allows adding a **language selector** that will be present on the badge. This helps a user switch quickly among languages without going to the settings.

```
enableLanguagesInBadgeButton: true,
```

2.4. Settings dialog

The **Settings** dialog contains the following sections: **Options**, **Languages**, **Dictionaries**, and **About**.

Name	Actions available	Description
Ignore options	Predefine some or all available options for application users.	Specifying the most common cases to ignore spelling checks: <ul style="list-style-type: none">• words with all letters capitalized (for example, "SCAYT"),• domain names (for example, "support@webspellchecker.net"),• words with mixed case (for example, "WebSpellChecker"),• words with numbers (for example, "2nd").
Languages	Hide the section from your users so that they use only predefined options and set predefined language for all of them.	Selecting the default language for grammar and spelling checking.
Dictionaries	Hide the section from your users so that they use only predefined options and set up a predefined user dictionary for all of them.	Adding and deleting words to/from personal user dictionaries. Connecting and disconnecting selected dictionaries.
About	Hide the branding information if you have a fully-fledged version of WebSpellChecker.	Displaying information on WProofreader copyright info and version number.

Depending on your web app usage preferences or requirements, you may want to customize the **Settings** dialog, for example:

- Hide one of the sections;
- Change the order of the mostly used tabs first;
- Disable commands for users so that they have the admin-defined settings and presets only.

With the **settingsSections** parameter, you can change the **Settings** dialog sections visibility and order. Example below places **Options** tab after **Languages** and **Dictionaries** tabs:

```
settingsSections: ['languages', 'dictionaries', 'options'],
```

2.4.1. Ignore options

Settings on the **Options** section let you define the following spelling exceptions:

- All caps words like 'EXAMPL'.
- Domain names like '<http://example.com>'.
- Words with mixed case like 'eXaMpl'.
- Words with numbers like 'examp17'.

You can change the default behavior using the following parameters:

```
ignoreAllCapsWords: true,  
ignoreDomainNames: true,  
ignoreWordsWithMixedCases: true,  
ignoreWordsWithNumbers: true,
```

To disable ignoring, set the value of the desired option as **false**.

2.4.2. Language

The **Language** section of the **Settings** dialog contains the list of available languages for spelling and grammar checking and enables users to set the default checking language.

To set the predefined language for all application users, use the **lang** parameter and specify the desired language short code as shown in the [Default Language for all Users](#) section.

To hide the **Language** section from your users so that they cannot modify the selected language, use the **settingsSections** parameter as shown below:

```
settingsSections: ['options', 'dictionaries', 'about'],
```

2.4.3. Dictionaries

Customizing Dictionaries user interface

These are some options for altering **Dictionaries** user interface elements and features:

1. Hiding **Dictionary** view at all.
2. Hiding or removing the Preferences section on the **Dictionary** view to prevent users from adding/changing/removing the dictionary.
3. Setting up a predefined dictionary for all users.

Hiding preferences section on the Dictionary tab

The **Preferences** section of the **Dictionaries** tab contains action items and buttons for managing user dictionaries, specifically: create a new dictionary, connect/disconnect existing dictionary, rename and delete a dictionary.

You may need to hide or remove the **Preferences** section when users have a general user dictionary and you do not want your users to make any changes to it except adding or removing words.

To disable the dictionary preferences UI for your users, specify the **disableDictionariesPreferences** option as **true**:

```
disableDictionariesPreferences: true,
```

Removing Dictionaries tab

You may want to remove the **Dictionaries** tab from user interface in the following cases:

- Users can only add new words using the **Add word** command from the suggestion balloon or when proofreading in a dialog mode. The dictionary itself cannot be deleted or modified.
- There is no predefined dictionary and you want to prevent storing the dictionaries on a server. Instead all the words are saved in users browsers local storage.



Instead of completely removing the **Dictionaries** tab, consider disabling only the user interface elements responsible for creating, deleting, renaming, and connecting a dictionary as in the examples above and when allowing your users to see and manage words added to the dictionary.



If the users have not saved the words added to the dictionary on the server and use local storage instead, these words become inaccessible when switching to a different browser or a computer.

Use the sample below:

```
settingsSections: ['languages', 'options'],
```



For details refer to [Configuring Server Personal Dictionaries](#) and [Configuring Cloud Personal User Dictionaries](#), Personal User Dictionary Interface in WProofreader sections and [FAQ](#) section.



If users try to use a different browser, they won't be able to access the dictionary. If there is no **Dictionary** tab in the **Settings** dialog, they cannot remove some words they might have added by mistake.

2.4.4. About tab



You can remove the **About** tab of WProofreader dialog and branding information if you have purchased a **Custom** or a Server version of WProofreader package.

To remove the branding of WProofreader, namely the title on the UI of the product and **About** tab with copyrights and logo, set the `removeBranding` parameter value `true` as shown in the code sample below:

```
removeBranding:true,
```

2.5. Actions for proofreading in a dialog

Actions available for configuring WProofreader dialog mode are the same as for the suggestions balloon, namely:

- Hiding or turning off descriptions of discovered grammar problems;
- [Changing the number of suggestions](#);
- Hiding some actions, for example, **Add to dictionary** or **Ignore All**;
- Preventing users from switching to the **Settings** dialog mode.

You can turn off descriptions for discovered grammar issues both in suggestions balloon and WProofreader dialog. To do so, set the value `disableProblemDescription` as `true`.

```
disableProblemDescription:true,
```

To hide the **Add to dictionary** command, specify the `actionItems` parameter values as shown in the sample below:

```
actionItems: ['ignoreAll', 'settings', 'toggle', 'proofreadDialog'],
```

To hide or remove the **Settings** icon so that the users cannot navigate and change WProofreader settings, exclude the `settings` value from the array of parameter values:

```
actionItems: ['ignoreAll', 'toggle', 'proofreadDialog'],
```

3. Languages and dictionaries settings

3.1. Default language options

The default language setting for WProofreader users is **American English** (`en_US`). But being the administrator, you can change this default language for all users.

Besides the default languages list, you can have more dictionaries to extend the linguistics coverage. Here are available options:

- **Auto Detect** language. This is a feature that auto-detects the input language and does the check using the proper language. Its support covers around 80 languages. This feature is available in all paid plans except the Basic one. To enable it by default for all users, set **'auto'** as a default language (see details below).
- [Additional languages list](#) including 150+ languages available with WebSpellChecker paid version. No additional configuration is required.
- [Specialized languages](#) such as legal and medical are not included into a default bundle and can be configured on demand for your daily used professional terms.

For details, contact WebSpellChecker [support team](#).

3.2. Default language for all users

You can specify the default language for all WProofreader users using the `lang` option. For example, if you set the **lang: 'de_DE'**, German locale is specified for all users.

```
lang: 'de_DE',
```

To enable the language locale, be sure to use the proper language code. For details, refer to the supported [language short codes list](#).



Users can modify the default language locale using the **Languages** tab of the **Settings** dialog but the locale they have selected will be used until they reload the browser. After reloading the browser a language specified by the administrator is used again.

3.3. User interface localization language

You can change WProofreader interface localization to match the locale preferences. The default interface localization language is American English. All the available short codes for localization are available in the description of localization parameter in [API options](#).

For example, to set German language as the localization option in WProofreader, use the localization parameter as in sample below:

```
localization: 'de',
```

3.4. Predefined user dictionary

To activate one dictionary common for all users, specify the `userDictionaryName` parameter by providing the dictionary name. After that, all words added by users via the **Add word** command are added to this particular dictionary.

```
userDictionaryName: 'user_dictionary_name',
```



If you have authentication enabled in your web app, you can predefine certain dictionaries for specific users using this approach.



Users can add words to a dictionary for spelling suggestions only.

3.5. Extending the list of dictionaries

If you want to use some dictionaries different from the default ones, refer to the list of [additional language dictionaries](#) available on our web site and contact [our sales team](#) for purchasing details.

3.6. Custom dictionaries

Custom dictionaries allow creating company-wide dictionaries and extending the vocabulary of the standard dictionary with words specific to industry or domain. To configure a global custom dictionary in WProofreader, use the `customDictionaryIds` parameter and add a string with required dictionaries IDs which should be separated by commas:

```
customDictionaryIds: '100694, 100695',
```

In the example above, ID is the unique dictionary ID will be assigned to your custom dictionary. For details on setting global custom dictionaries for Server version, refer to [Configuring Server Global Custom Dictionaries](#) section, namely:

- [Create Custom Dictionary for Default Language](#);
- [Create Custom Dictionary for Additional Language](#).

For details on setting custom dictionaries for Cloud version, refer to [Configuring Cloud Custom Dictionaries guide](#).

4. Miscellaneous options

4.1. Minimal word length

By default, WProofreader enables spell check starting with 3 and more letters per word. You can customize this option using the `minWordLength` parameter:

```
minWordLength: 2,
```

After you have used this option, the words which have two (2) and more letters in them will be spell checked.

4.2. Excluding elements from checking

There are a list of options such as `ignoreElements`, `ignoreClasses` and `ignoreAttributes` that help to manage where the text checking should not be done.

Use the `ignoreElements` parameter to instruct WProofreader not to check the HTML elements in the text such as, typeface types, links, tables and others. By default, text within the `<a>` tag are ignored as well as `<style>` and `<script>` tags.

You can additionally choose to ignore spelling and grammar checking in certain web page markup elements by classes or attributes.

In example below, text inside table elements, elements containing specified class or attribute will be skipped during checking:

```
ignoreElements: 'table',  
ignoreClasses: ['ignore-checking'],  
ignoreAttributes: ['data-wsc-ignore-checking']
```

4.3. Disabling proofreading in As You Type mode

Proofreading during typing in the so called 'as you type mode' is enabled by default. You can disable this mode using the **proofreadAsYouType** parameter thus proofreading becomes possible only in dialog mode. To do so, change the value of **proofreadAsYouType** parameter to **false**:

```
proofreadAsYouType: false,
```

After disabling proofread as you type mode, users can access proofreading by initiating WProofreader dialog mode from the badge by clicking **Proofread in dialog**  icon.