WProofreader for HTML editable controls

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 to initialize WProofreader in HTML editable elements such as **<input>** and **<textarea>**, and HTML contenteditable elements such as ** **, **<iframe>** and others. All the steps described in this guide are applicable for the Server version of WProofreader.

1. Add WProofreader script

Add the **wscbundle.js** script on your web page.

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

2. Initialize WProofreader

Initialize the WProofreader functionality for the selected elements and define the required parameters. Below you will find examples of initialization in content-editable elements such ** ** and **<iframe>** and editable elements such as **<input>** and**<textarea>.**

2.1. Contenteditable DIV Element

Initialize WProofreader in **contenteditable element** using either an imperative or declarative notation as it is shown in the examples below.

WProofreader (Imperative Notation):

```
<div contenteditable id="container1">
This sampl text is aimed at demonstrating the work of WProofreader in a contenteditable div element.
WProofreader will find all your speling and grammar mistakes as you type and suggest how to correct them.
Type your text here or use this one too see an few of of the problems that WProofreader will recognized.
Click the underlined words and phras to see the suggestion options for correction.
</div>

<script>
    var instance1 = WEBSPELLCHECKER.init({
        container: document.getElementById("container1"),
        autoStartup: true,
                enableGrammar: true,
        lang: 'en_US',
        serviceProtocol: 'https',
        serviceHost: 'your_host_name',
             servicePort: '443',
        servicePath: 'virtual_directory/api'
    });
</script>
```

WProofreader (Declarative Notation):

```
<div contenteditable id="container2" data-wsc-autocreate="true" data-wsc-lang="en_US">
This sampl text is aimed at demonstrating the work of WProofreader in a plain textarea element.
WProofreader will find all your speling and grammar mistakes as you type and suggest how to correct them.
Type your text here or use this one too see an few of of the problems that WProofreader will recognized.
Click the underlined words and phras to see the suggestion options for correction.
</div>

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

2.2. Contenteditable IFRAME Element

Initialize the WProofreader in **content-editable <iframe> element** using either imperative or declarative notation as it is shown in the examples below.

WProofreader (Imperative Notation):

```
<iframe id="container3" src="editable_doc.html"></iframe>

<script>
    var instance3 = WEBSPELLCHECKER.init({
        container: document.getElementById("container3"),
        autoStartup: true,
                enableGrammar: true,
        lang: 'en_US',
        serviceProtocol: 'https',
        serviceHost: 'your_host_name',
        servicePort: '443',
        servicePath: 'virtual_directory/api'
    });
</script>
```

WProofreader (Declarative Notation):

```
<iframe id="container4" data-wsc-autocreate="true" data-wsc-lang="en_US" src="editable_doc.html"></iframe>

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

Editable_doc.html example:

```
<html>
    <head>
        <title>WProofreader for HTML Editable Controls</title>
    </head>
    <body contenteditable>This sampl text is aimed at demonstrating the work of WProofreader in an iframe
element.
WProofreader will find all your speling and grammar mistakes as you type and suggest how to correct them.
Type your text here or use this one too see an few of of the problems that SCAYT will recognized.
Click the underlined words and phras to see the suggestion options for correction.
    </body>
</html>
```

2.3. INPUT element

Initialize WProofreader in **<input> element** using either imperative or declarative notation as it is shown in the examples below.

WProofreader (Imperative Notation):

```
<input id="container5" type="text" value="This sampl text is aimed at demonstrating the work of WProofreader in
a input textform element.
WProofreader will find all your speling and grammar mistakes as you type and suggest how to correct them.
Type your text here or use this one too see an few of of the problems that WProofreader will recognized.
Click the underlined words and phras to see the suggestion options for correction.">

<script>
        var instance5 = WEBSPELLCHECKER.init({
        container: document.getElementById("container5"),
        autoStartup: true,
        enableGrammar: true,
        lang: 'en_US',
        serviceProtocol: 'https',
        serviceHost: 'your_host_name',
        servicePort: '443',
        servicePath: 'virtual_directory/api'
        });
</script>
```

WProofreader (Declarative Notation):

```
<input id="container6" data-wsc-autocreate="true" data-wsc-lang="en_US" type="text" value="This sampl text is
aimed at demonstrating the work of WProofreader
in a input textform element. WProofreader will find all your speling and grammar mistakes as you type and
suggest how to correct them.
Type your text here or use this one too see an few of of the problems that WProofreader will recognized.
Click the underlined words and phras to see the suggestion options for correction.">

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

2.4. TEXTAREA element

Initialize WProofreader in **<textarea> element** using either imperative or declarative notation as it is shown in the examples below.

WProofreader (Imperative Notation):

```
<textarea id="container7" type="text">This sampl text is aimed at demonstrating the work of WProofreader in a
textarea textform element.
WProofreader will find all your speling and grammar mistakes as you type and suggest how to correct them.
Type your text here or use this one too see an few of of the problems that WProofreader will recognized.
Click the underlined words and phras to see the suggestion options for correction.
</textarea>

<script>
        var instance7 = WEBSPELLCHECKER.init({
        container: document.getElementById("container7"),
        autoStartup: true,
        enableGrammar: true,
        lang: 'en_US',
    serviceProtocol: 'https',
    serviceHost: 'your_host_name',
    servicePort: '443',
    servicePath: 'virtual_directory/api'
        });
</script>
```

WProofreader (Declarative Notation):

```
<textarea id="container8" data-wsc-autocreate="true" data-wsc-lang="en_US" type="text">This sampl text is aimed
at demonstrating the work of WProofreader
 in a textarea textform element. WProofreader will find all your speling and grammar mistakes as you type and
suggest how to correct them.
Type your text here or use this one too see an few of of the problems that WProofreader will recognized.
Click the underlined words and phras to see the suggestion options for correction.
</textarea>

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

3. Adjust default options

You can adjust the default options to suit your needs. Here is a full list of WProofreader API options and their possible values.



To see the result of initializing WProofreader in HTML editable elements, see a demo page with the WProofreader initialization in element on our website.

Can be also useful:

- WProofreader API