

Installing WebSpellChecker v5.14+ on Linux with NGINX

This step-by-step guide outlines the main steps for the WebSpellChecker application installation on a Linux-based server with NGINX web server. In this guide we will run you through the whole installation process from downloading installation packages to embedding spell/grammar checking functionality to your web-based system.



Before you begin:

1. Check if your environment configuration meets [Installation requirements](#).
2. Acknowledge with [Architecture diagram](#).
3. Make sure that you have a valid license:

First of all you need to get a valid License Ticket ID to proceed with the license activation. Depending on the server license type, it may be a commercial (e.g. 1 year) or a 30-day trial license. Except the validity period, there is no difference between ticket types from the technical perspective.

Good to know:

- If you do not have a license yet, you can obtain it using one of the following ways:
 - request a 30-day trial license for evaluation and testing purposes;
 - acquire a commercial license by contacting our sales department.
- Starting from v5.8.1 released on December 30, 2020, the license is no longer validated by or tied to the hardware characteristics of a machine or server. License for older versions of the packages is hardware-dependent.
- If you are migrating WebSpellChecker from one server to another, you will need to deactivate (detach) a license on the current server and then reactivate it on a new one. Check carefully the **Migrating License to New Server** guide under the Licensing section.
- If you have problems with your license, contact us.

1. Download installation package
2. Install WebSpellChecker
3. Configure grammar engine
4. Activate License
5. Finalize installation
6. Verify operability
 - WebSpellChecker Server Status
 - WebSpellChecker Server Version
 - WebSpellChecker Demo Samples
7. Further steps
 - Option A. Integration guides
 - Option B. Demo samples

1. Download installation package

Download WebSpellChecker Server installation package for the corresponding version of your operating system (OS). By default, it is provided on a license purchase or renewal. You can also request it by [contacting us](#).

2. Install WebSpellChecker

2.1. Unpack the WebSpellChecker Server installation package (e.g. wsc_app_x64_5.x.x.xx.tar.gz).

```
tar -xvzf wsc_app_x64_5.x.x.xx.tar.gz
```

2.2. Switch to the directory with the unpacked installation package using `cd ./WSC_5.x.x.x/` and execute `automated_install.pl` script.

```
sudo perl automated_install.pl
```



If you are running the installation without root privileges, make sure all necessary directories have **write** permissions for the correct installation process.

2.3. Before proceeding with installation, read and confirm the terms and conditions of [WebSpellChecker Software License Agreement](#).

Welcome to the WebSpellChecker (WSC) application installation wizard!

Before you go ahead with the installation, make sure that your environment configuration meets the WSC application system requirements.

If you are all set, please follow carefully the wizard instructions.

PLEASE MAKE SURE YOU ARE ACKNOWLEDGED AND ACCEPT THE WEBSPELLCHECKER SOFTWARE LICENSE AGREEMENT TERMS AND CONDITIONS PUBLISHED ON THE WEBSPELLCHECKER OFFICIAL WEBSITE. BY DOWNLOADING OR INSTALLING THIS SOFTWARE, YOU ACCEPT THE FULL TERMS OF THIS AGREEMENT.

IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, YOU MUST NOT INSTALL OR USE THE WEBSPELLCHECKER SOFTWARE.

2.4. Select installation path. The default installation path is **/opt/WSC**. However, you may decide to change it and set a desired one. As soon as you define the installation path, press **Enter** to proceed.

Step 1: Check Configuration Settings.

Install path [/opt/WSC]:

2.5. Select the products you want to install. By default, [WProofreader add-on for RTEs](#) (Option 1) is selected.

Step 2: Select Products.

1. WProofreader(HTML Editable Controls, CKEditor 4/5, Froala Editor, TinyMCE, Quill, etc.)
2. Web API
3. SCAYT Plugin for CKEditor 4
4. WSC Dialog Plugin for CKEditor 4
5. Select all products

List the products you want to install. Use the comma as separator. [1]:

2.6. Confirm whether you want Demo Samples and Documentation to be installed. If so, press **Enter** to proceed.

Step 3: Install Demo Samples and Documentation

Do you want to install Demo Samples and Documentation? Enter 'y' or 'n' [y]:

2.7. During this step, the installer copies files to the selected installation directory.

Step 4: Copy the WSC Files to the Installation Directory.

2.8. Define permissions. Set appropriate permissions for the application. Define the files' ownership, a user and group in the **user:group** format (e.g. 33:33).

Step 5: Define the WSC Application Permissions.

Specify the files ownership, a user and group in the 'user:group' format.

It can be the same user and group which are used for the Apache HTTP Server, i.e. 'www-data:www-data' or '33:33'.

Specify the files owner as user:group [nobody:nobody]: 33:33

* Setting the WSC application permissions.

You specified 33:33

2.9. Select AI-based languages to be downloaded and installed. By default, AI-based English [1] will be installed. It enables improved proofreading capabilities compared to traditional dictionaries or rules-based engines. If selected, the installer will attempt to download a language model that is required for an AI-based language to work. If you don't have the Internet connection on the server where the installation takes place, you can do it manually later. For details, check [How to download and enable AI-based engine manually](#).

Step 6. Select AI-based languages.

- 0. None
- 1. English
- 2. German
- 3. Select all

Specify your choice using comma as separator: [1]:

2.10. Select the type of protocol (HTTP or HTTPS) that will be used for communication between the client and the WebSpellChecker service. By default, HTTPS [1] is selected. Press **Enter** to confirm the selection.

Step 6: Select Protocol Type.

Select the type of the protocol that will be used for communication between the WebSpellChecker service and a client.

If you plan to use HTTPS, please make sure you have a valid SSL certificate.

- 1. HTTPS
- 2. HTTP

Input the number from 1 to 2 [1]:

2.11. When selecting your web server, choose **Other** [3] to run static files using NGINX.

Step 8: Choose either Web or Application Server Type.

Select a particular Web or Application Server to be used for the WSC Application Server running.

- 1. Apache HTTP Server
- 2. Java Application Server (Apache Tomcat, Oracle WebLogic Server, Oracle GlassFish Server, etc.)
- 3. Other

Input the number from 1 to 3:3



Due to technical limitations, [WSC Dialog plugin for CKEditor 4](#) is not available for setup on Nginx. If it is a requirement for you to use this plugin, please consider setting it up on either [Apache HTTP Server](#) or [Apache Tomcat](#).



If you want to use NGINX as a web server, perform the following steps:

- 1. Set up NGINX.
- 2. Create a **wscservice** configuration file with bindings to WebSpellChecker static files and add the necessary **Access-Control-Allow-Origin** headers.
- 3. Create a symlink for the **wscservice** virtual directory.
- 4. Configure NGINX to act as a reverse proxy to avoid exposing the 2880 port of AppServer. Check in the **config** sample below or visit [NGINX reverse proxy on Ubuntu](#) for more details.

Step 1. Install NGINX.

```
sudo apt-get update
sudo apt-get install nginx
```

Step 2. Customize the firewall.

```
sudo ufw allow 'Nginx Full'
```

Step 3. Check the status of NGINX web server.

```
systemctl status nginx
```

Step 4. Create and customize 'server blocks'.

```
sudo cd /etc/nginx/sites-available/  
sudo cp default wscservice  
sudo nano wscservice
```

Here is an example of **wscservice** configuration file for NGINX.

wscservice

```
# Default server configuration  
#  
server {  
    listen 80 default_server;  
    listen [::]:80 default_server;  
  
    # SSL configuration  
    #  
    # listen 443 ssl default_server;  
    # listen [::]:443 ssl default_server;  
    #  
    # Note: You should disable gzip for SSL traffic.  
    # See: https://bugs.debian.org/773332  
  
    server_name wscservice;  
  
    # bindings of static files  
    location /wscservice/ {  
        alias /opt/WSC/WebComponents/WebInterface/;  
        autoindex on;  
        index index.html index.htm;  
  
    location /wscservice/samples/ {  
        alias /opt/WSC/WebComponents/Samples/;  
        autoindex on;  
        index index.html index.htm;  
    }  
  
    location /wscservice/wscbundle/ {  
        alias /opt/WSC/WebComponents/WebInterface/wscbundle/;  
        autoindex on;  
        add_header Access-Control-Allow-Origin '*';  
    }  
  
    # reverse proxy for proxying requests to AppServer  
    location /wscservice/api {  
        # location /wscservice/api/ with a slash for WebSpellChecker versions 5.15.0.0 only  
        proxy_pass http://your_host:2880;  
    }  
}  
}
```

Step 5. Create symlink for the 'wscservice' file.

```
ln -s /etc/nginx/sites-available/wscservice /etc/nginx/sites-enabled/wscservice
```

Step 6. Restart NGINX.

```
sudo systemctl restart nginx
```

Step 7. NGINX troubleshooting if the restart fails. Probably the default NGINX config already uses the 80/443 port or you added an extra curly brace.

```
sudo nginx -t
```

2.13. Define the settings of the selected web server or Java application server.

- Specify a domain name. The default value is **localhost**. Either enter your domain name or confirm the default value by pressing Enter.

Step 9: Specify Web Server or Application Server Settings.

Specify a domain name, the service virtual directory and a port number that will be used by a web server or application server.

Domain name [localhost]:

- Specify the port number of a web server. The installer will suggest the port number based on the specified protocol and selected web server.

Web or application server port [443]:

- Provide the name of the virtual directory for WebSpellChecker service. The default value is **wscservice**.

Service virtual directory [wscservice]:

On this step the installer summarizes the information that you have entered on the previous steps.

```
* PROTOCOL = https
* DOMAIN NAME = localhost
* INTERNAL APPLICATION SERVER = 2880
* WEB / JAVA APPLICATION SERVER PORT = 443
* VIRTUAL DIRECTORY = wscservice
* SERVICE URL = https://localhost:443/wscservice
```

3. Configure grammar engine

The Grammar engine requires Java 8 and higher. The installation script will try to auto-detect a path to an existing Java Virtual Machine (JVM) and record the path in the AppServerX.xml configuration files. If the path is not detected, the script will ask you to specify the correct path to JVM again.

You have an option to **skip** this step during the setup and set a path to JVM manually. Refer to [Enabling grammar engine](#) guide to find out more.

Step 10: Specify Path to Java Virtual Machine.

At least Java 8 is required to use the Grammar engine. Please specify the path to Java Virtual Machine (JVM). You can skip this step and add the path to JVM manually in the '/opt/WSC/AppServer/AppServerX.xml' file (the PathToJavaVirtualMachine tag).

Do you want to specify a path to JVM? Enter 'y' or 'n' [y]:

4. Activate License

On this step you need to choose whether to run the license activation process. You can skip the license activation step if you already have a license activated or you prefer to proceed with the license activation step later. If you want to go ahead with the license activation, press **Enter** and then proceed with the steps described in the license activation guide below.

Refer to the [License activation on Linux](#) guide to find out how to activate a license on a Linux server.

```
Step 11: Activate License.
```

```
Do you want to proceed with the license activation? Enter 'y' or 'n' [y]:
```

Upon successful license activation, the installer script will start AppServer.

```
Congratulations! Your license is confirmed and activated successfully.  
* Starting AppServer
```

5. Finalize installation

If no errors happened during the setup, and everything is configured properly, you will receive a confirmation that the application has been installed.

```
Congratulations! The WSC application has been successfully installed to /opt/WSC directory.
```

6. Verify operability

Now you are ready to check if the installation process of the application went well, and all the components are functioning properly.

```
Step 12: Verify the WSC Application Operability.
```

```
Check the application version, status, and access to the demo samples to verify if the application works properly after the installation.
```

```
a. Version:
```

```
https://localhost:443/wscservice/api/?cmd=ver
```

```
b. Status:
```

```
https://localhost:443/wscservice/api/?cmd=status
```

```
c. Default demo samples:
```

```
https://localhost:443/wscservice/samples/
```

Find the correct responses for the application version and status below.

WebSpellChecker Server Status

Here is a correct status response of WebSpellChecker Server:

```
{  
  "SpellCheckEngine": {  
    "active": true  
  },  
  "GrammarCheckEngine": {  
    "active": true,  
  },  
  "EnglishAIModel": {  
    "active": true  
  },  
  "GermanAIModel": {  
    "active": true  
  },  
  "SpanishAIModel": {  
    "active": true,  
  },  
  "EnglishAutocomplete": {  
    "active": true,  
  }  
}
```


WebSpellChecker Server Version

Here is a correct version response for WebSpellChecker Server 5.x.x.x on a Linux server:

```
{
  "Copyright": "(c) 2000-2023 WebSpellChecker LLC",
  "ProductWebsite": "webspellchecker.com",
  "ProgramVersion": "5.x.x.x x64 master:xxxxxxx (xxxx) #xx",
  "PackageVersion": "5.x.x.x master:xxxxxxx (xxx) #xx"
}
```

WebSpellChecker Demo Samples

Here is the correct view of the WebSpellChecker demo samples:

 WebSpellChecker

WProofreader

SpellCheckAsYouType

WebSpellChecker Dialog

OVERVIEW

HTML EDITABLE CONTROLS

WYSIWYG EDITORS

[Multiple Controls](#)

Multiple Editors

WProofreader in HTML Editable Controls

DOCUMENTATION

This sample demonstrates the work of the WProofreader in HTML Editable Controls.

WProofreader can be easily enabled and used with HTML editable controls such as **input** or **textarea** and elements with contenteditable attribute set to 'true'.

WProofreader in Textarea

This sampl text demonstrates the work of a new WProofreader. Just start typing here and you will see your spelling and and grammar mistakes.

WProofreader in Iframe

This sampl text demonstrates the work of a new WProofreader. Just start typing here and you will see your spelling and and grammar mistakes.

WProofreader in Div

This sampl text demonstrates the work of a new WProofreader. Just start typing here and you will see your spelling and and grammar mistakes.

7. Further steps

As soon as backend components are up-and-running, you need to integrate WebSpellChecker frontend components (plugins, JavaScript libraries) into your web application.

Option A. Integration guides

Visit [Products and integrations guides](#) to learn more about how to get started with WebSpellChecker products, one of them being [WProofreader](#).

WProofreader

[WProofreader](#) is a multilingual proofreading tool for web applications. It enables grammar and spell check while you type or work with your text in a dedicated dialog. It detects editable fields in focus, be it HTML editable controls or WYSIWYG editors, and enables spelling and grammar checking there. View the [full list of supported integrations](#).

WProofreader benefits:

- Available as a standalone secure solution on your server or as an annual subscription to the service hosted by WebSpellChecker.
- Out of the box offer provides spell check for 150 languages and basic grammar check for over 20 languages. Find out more in [Supported languages](#).
- All modern [web browsers are supported](#), including the latest stable versions of Chrome, Firefox, Safari, MS Edge, and Internet Explorer 11, and mobile web browsers such as Chrome for Android and iOS Safari;
- [Company custom dictionary](#) managed by system admins and intended for all WProofreader users;
- [User custom dictionaries](#) for end-users;
- [WAI-compliant](#), up to [WCAG 2.1](#) and [Section 508 accessibility standards](#);
- Operation and installation via HTTPS protocol is supported to fulfill strict security requirements.

With the cloud-based version of WProofreader you can access and use its features without server-side installation. In the server-based version, all the application data is securely processed on your own server. Refer to the [Get started with WProofreader Server \(autoSearch\)](#) guide for more details.

WProofreader allows users to see and correct spelling and grammar problems on the fly while they are typing directly in the text input field. Every misspelled word found will be underlined with a red wavy line and every grammar problem detected will be marked with a green one. The user just needs to hover a marked word or phrase and replace it with a proper one from the list of given corrections.

Option B. Demo samples

Depending on the type of elements, namely HTML native elements or rich text editors where the spelling and grammar check is needed, you may choose and use the demo samples that are shipped with the package.

To access the default samples, go to [http\(s\)://your_host_name/wscservice/samples/](http(s)://your_host_name/wscservice/samples/) where port is the **port** of your web server or Java application server. The default samples contain all the necessary samples and instructions that will help with the integration of the WebSpellChecker products. Moreover, each sample contains detailed development documentation with a list of methods and parameters available.