

Kiuwan for Developers for Microsoft Visual Studio Code

This section will show you how to integrate the Kiuwan IDE Plug-In into Visual Studio Code.



The Kiuwan IDE Plug-In for Visual Studio Code is only available in Viewer Mode.

It has been successfully tested with **VS Code 1.33.1**

Contents:

- [Kiuwan IDE Plug-In](#)
- [Installation](#)
- [Configuration](#)
 - [Connection Settings](#)
 - [Mapping your VS Code folder or workspace to your Kiuwan Application](#)
 - [Source of Defects](#)
 - [Limiting and filtering Defects](#)
 - [VS Code commands](#)
- [Viewing Kiuwan defects in VS Code](#)
- [Support and Troubleshooting](#)

Kiuwan IDE Plug-In

Kiuwan allows for a true **shift-left** approach by integrating with all the main IDEs.

Kiuwan for Developers is a **plug-in for development IDEs** that facilitates and automates compliance with **security** normative, **quality** standards and **best practices** for several languages.

It provides the following benefits:

- **Security Vulnerabilities Detection** - The plug-in allows developers to detect and fix security vulnerabilities, such as Injection (SQL, XML, OS, etc), XSS, CSRF, etc., directly within their development IDEs.
- **Adoption of Security and Coding Standards** - The plug-in helps to ensure compliance to standards (CVE, OWASP, CERT-Java/C/C++, SANS-Top25, WASC, PCI-DSS, NIST, MISRA, BIZEC, ISO/IEC 25000 and ISO/IEC 9126) by automating the work. This plug-in connects with Kiuwan and harnesses the power of its quality models to prevent errors and automatically standardize the code.
- **Automatic Error Prevention** - The plug-in implements and monitors compliance to coding standards at the time the code is entered. Thus you can avoid errors and reduce the time and cost of debugging and testing activities.

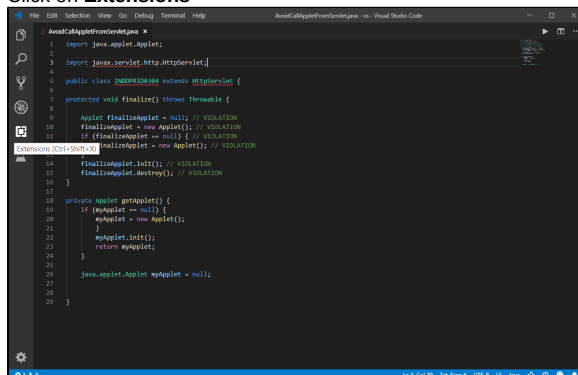
The Kiuwan IDE Plug-in monitors and reports on the security, quality, and efficiency of your code at the point that it is written. This immediate feedback provides you with the opportunity to improve your code before it is delivered.

Installation

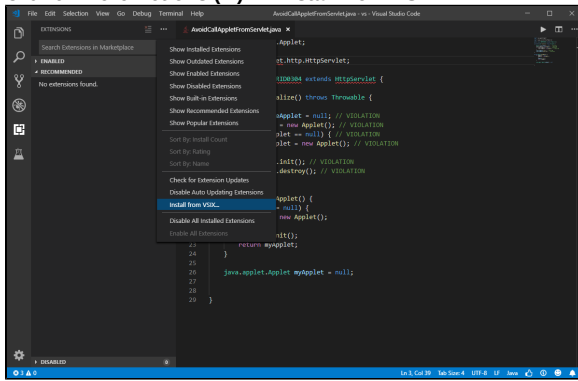


Before starting the installation, you must download **k4d-vscode.vsix** from <https://static.kiuwan.com/download/vscode/k4d-vscode.vsix>

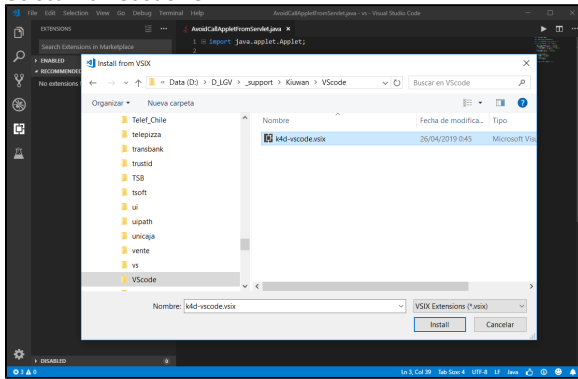
1. Click on **Extensions**



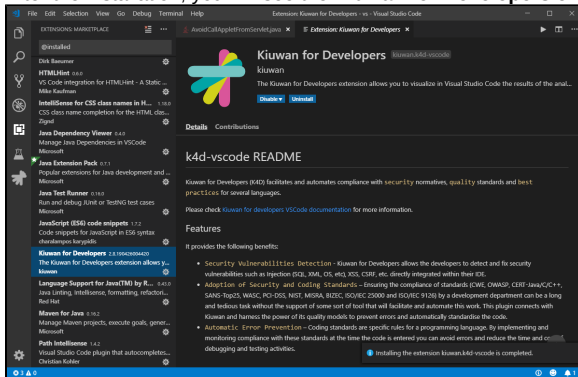
2. Click on **More Actions (...)** >> **Install from VSIX**



3. Select **k4d-vscode.vsix**



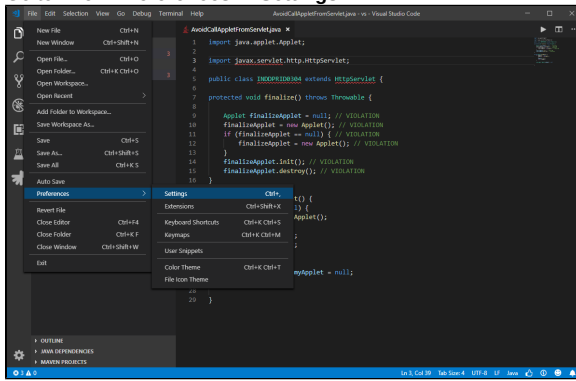
4. After the installation, you will see the **Kiuwan for Developers** extension.



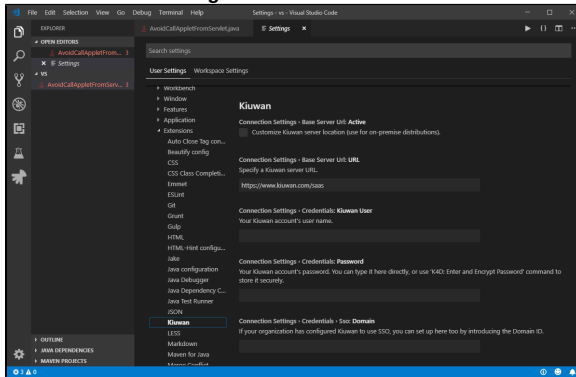
Configuration

After installation, you need to **configure the Kiuwan IDE Plug-In** to connect to Kiuwan. Please remember that you need to have a valid Kiuwan Account.

1. Go to File > Preferences > Settings



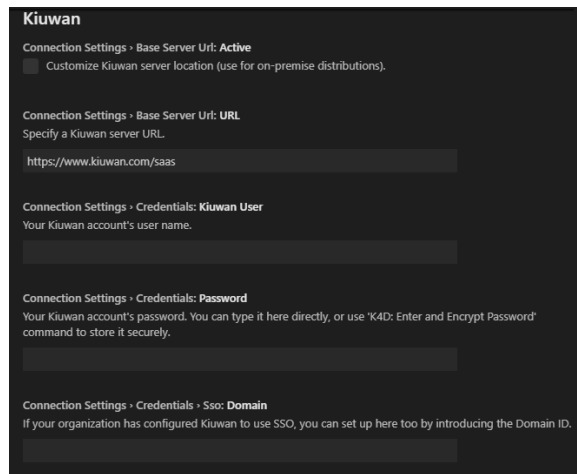
2. and select User Settings > Extensions > Kiuwan



Connection Settings

You can find the connection settings at **User Settings > Extensions > Kiuwan**

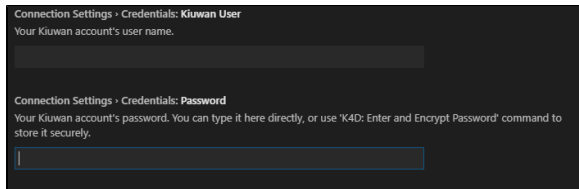
Please remember that you need to have a valid Kiuwan Account.



The **Kiuwan server URL** comes preconfigured (leave it with default value).

- This field only needs to be modified in case you are using Kiuwan On-Premises (KOP).
- If you need to modify it (to set your KOP server URL, check **Customize Kiuwan Server location**)

Fill in the **User** and **Password** fields with your Kiuwan account credentials.



If your Kiuwan account is configured to use **Single Sign-On (SSO)**, enter your **Domain ID** (consult your Kiuwan admin and see [How to integrate Kiuwan with SAML SSO](#))

To check the connection, you can use **K4D: Check Connection With Current Settings** to select the delivery. See [Kiuwan VS Code commands](#)

Mapping your VS Code folder or workspace to your Kiuwan Application

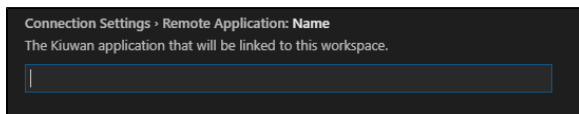
After the Kiuwan IDE Plug-In is installed, you are ready **to map your VS Code workspace or folder to a Kiuwan application**.



This action will allow synchronizing defects and vulnerabilities found by Kiuwan in your source code, getting them ready for fixing.

All the following settings can be configured at **User level** (i.e. they will apply to all folders opened with the user currently logged in the machine), or at **Workspace level** (i.e. you can configure different values for different folders/workspaces); the later is recommended.

To map your VS Code workspace to Kiuwan, type your Kiuwan app name at **Remote Application: Name**

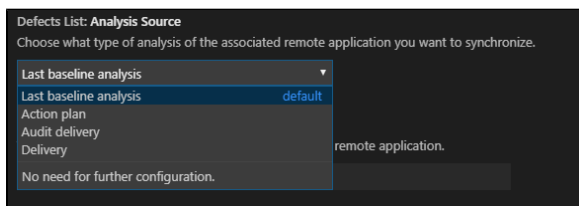


Leaving it blank, you can use **K4D: Pick Remote Application** to select the app.

See [Kiuwan VS Code commands](#)

Source of Defects

Once mapped, you can select the **source of the defects** that will be shown in VS Code.



Depending on your needs, the **source** of server defects could be different:

- **Last baseline analysis**
 - All the defects found during last complete application analysis (i.e. the Application Baseline)
- **Action plan**
 - Defects included within an Action Plan (you must type the plan name)
 - Leaving it blank, you can use **K4D: Pick Action Plan** to select the action plan. See [Kiuwan VS Code commands](#)
- **Audit Delivery**
 - Defects that must be fixed so the Audit of a delivery can be successfull (you must type the delivery name)
 - Leaving it blank, you can use **K4D: Pick Audit Delivery** to select the delivery. See [Kiuwan VS Code commands](#)
- **Delivery**
 - Defects found for the delivery analysis of the mapped application

- Leaving it blank, you can use **K4D: Pick Delivery** to select the delivery. See [Kiuwan VS Code commands](#)

For Audit Delivery and Delivery , you can select a **range** of defects.

Defects List: **Analysis Source**

Choose what type of analysis of the associated remote application you want to synchronize.

Last baseline analysis ▾

Defects List › Analysis Source Action Plan: **Name**

Type the name of an existing action plan for the chosen remote application.

Defects List › Analysis Source Audit Delivery: **Code**

Type the code of an existing audit delivery for the chosen remote application.

Defects List › Analysis Source Audit Delivery: **Pick Range**

This range will be used by the "K4D: Pick Audit Delivery" command to limit the results retrieved from the server.

Last 10 ▾

Defects List › Analysis Source Delivery: **Code**

Type the code of an existing delivery for the chosen remote application.

Defects List › Analysis Source Delivery: **Pick Range**

This range will be used by the "K4D: Pick Delivery" command to limit the results retrieved from the server.

Last 10 ▾

Limiting and filtering Defects

Finally, you can limit how many defects to download from Kiuwan servers (**Defects Limit**), as well as filter the resulting set of defects by **Characteristics**, **File Patterns**, **Language** and **Priority**.

Defects List: **Defects Limit**

Limit number of defects retrieved from Kiuwan (value equal to or less than zero means no limit).

100

Defects List › Filters: **By Characteristic**

Comma-separated list of characteristics (Efficiency, Maintainability, Portability, Reliability, Security) to retrieve from server (leave blank for all).

Defects List › Filters: **By File Pattern**

File path contains the following substring.

Defects List › Filters: **By Language**

Comma-separated list of languages (for a full list of languages available see online documentation) to retrieve from server (leave blank for all).

Defects List › Filters: **By Priority**

Comma-separated list of priorities (Very high, High, Normal, Low, Very low) to retrieve from server (leave blank for all).

Defects List › Order: **By**

Choose a criteria to order defects retrieved.

Priority ▾

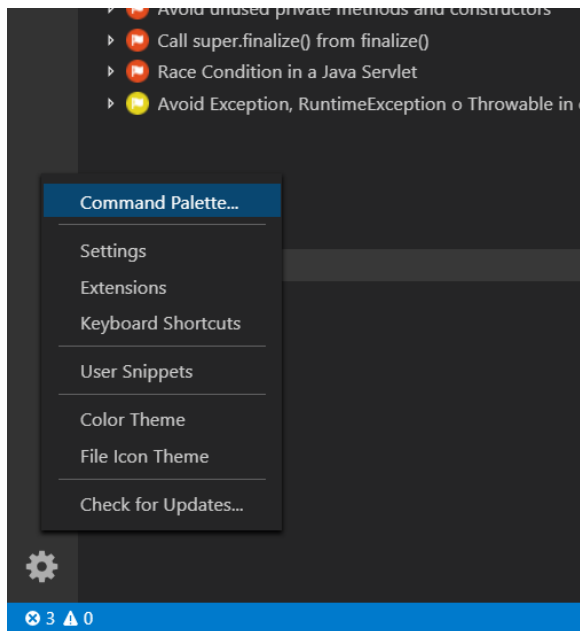
Defects List › Order: **Direction**

Choose ordering direction.

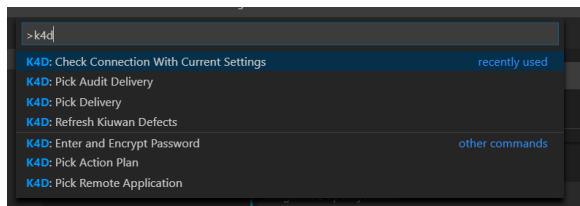
Descending ▾

VS Code commands

Select **Command Palette**..



to use the following list of **Kiuwan VS Code commands**



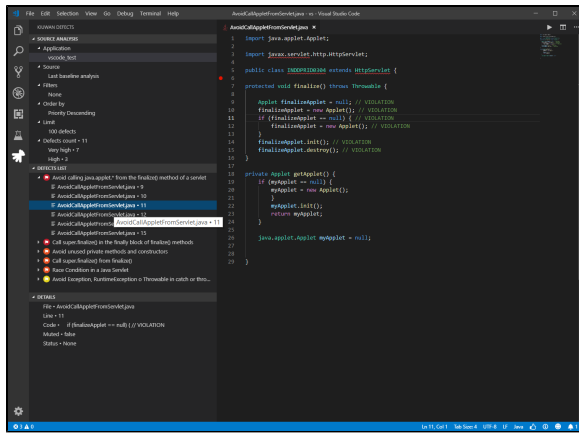
For example, if you select Delivery as the source for defects, you can select the right delivery by running K4D: Pick Delivery and selecting among the available ones.

Viewing Kiuwan defects in VS Code

Once configured, just click on the Kiuwan icon to see the defects.

This 'tree of defects' is structured in two or three levels:

1. **Rule:**
 - The first level represents 'the rule' which generated the defect.
 - If you select it, the bottom section **Details** will refresh its contents, showing important information about that rule.
 - You can also right-click on it and select **Show rule documentation** in Kiuwan and K4D will open a new tab of your system web browser, pointing to Kiuwan, to show you all existing details about the rule.
2. **Defect:**
 - The second level is populated with defects found of the selected rule.
 - The **Details** section will now show information that affects only selected defect, and K4D will try and find the reported file and line among your local sources, to open it in a new editor tab.
3. **Propagation path:**
 - The last level will show you all the locations of the code crossed by a security vulnerability, so you can track it, and neutralize it.



Support and Troubleshooting

If you experience problems with the Kiuwan plugin for VS Code, read the Kiuwan [Documentation](#) to find a solution, or if you prefer you can collect troubleshooting information and send it to us.



Support Information

Important information for troubleshooting is located in the log file.

To make this process easier find log file at **`$USER_HOME/.optimyth/k4d-vscode.log`** and submit to technical support team.

Visit [Contact Kiuwan Technical Support](#) on how to contact us. We will address your problem as soon as possible.