# Insights Components

This section introduces you to the Components tab in Kiuwan Insights.

**Contents**:



## Components Inventory

Kiuwan Insight analyzes your application software, discovering all external dependencies, and builds a **co mponents inventory** that lets you track any external piece of code that could be part of your application.

Go to **Insights** > **Components** to access the components inventory.

### Supported languages and resources

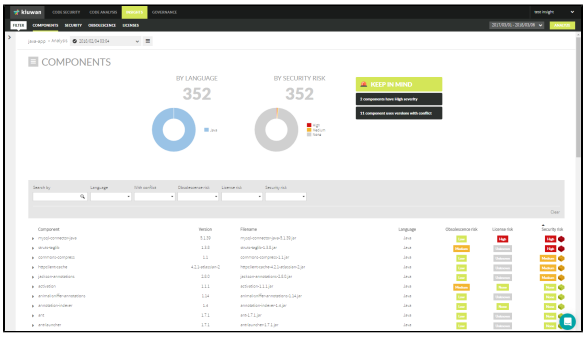
Kiuwan Insights uses the following resources to extract information on 3rd party dependencies.

| Supported languages | Supported repositories | Supported build systems | Repositories Used | Database Vulnerabilities Used | Licenses extract from |
|---|---|---|---|---|---|
| Go | • GitHub | • go.mod<br>• Gopkg.lock | GitHub: https://github.com/ | • NVD: https://nvd.nist.gov/ | • GitHub |
| Java | • Maven<br>• Gradle | • Ant (*.xml files)<br>• Maven (pom.xml files)<br>• Gradle (*.gradle files)<br>• *.jar, *.war, *.ear files | Maven (central or others configured in settings.xml or pom.xml files):<br><br>https://repo.maven.apache.org/maven2/ | • NVD: https://nvd.nist.gov/ | • pom.xml<br>• License file into jar file. |
| Javascript | • Npm<br>• Bower | • Npm (package.json files)<br>• Bower (bower.json files)<br>• Yarn (package.json files) | Npm: https://www.npmjs.com/ | • NVD: https://nvd.nist.gov/ | • NPM Rest services. |
| Kotlin | • Maven<br>• Gradle<br>• Ant | • Ant (*.xml files)<br>• Maven (pom.xml files)<br>• Gradle (*.gradle and *.gradle.kts files) | Maven (central or others configured in settings.xml or pom.xml files):<br><br>https://repo.maven.apache.org/maven2/ | • NVD: https://nvd.nist.gov/ | • Maven services |
| .Net | • Nuget | • Nuget (*.csproj, project.json, global.json, *.vbproj files) | Nuget: https://www.nuget.org/ | • NVD: https://nvd.nist.gov/ | • Nuget Rest services. |

| | | | | | |
|---|---|---|---|---|---|
| Php | • Packagist | • Composer (composer.json, composer.lock files) | Packagist: https://packagist.org/ | • NVD: https://nvd.nist.gov/ | • NVD: https://nvd.nist.gov/ |
| Python | • PyPI<br>• GitHub | • PyPI (setup.py files)<br>• Requirements (txt file with declared dependencies) | PyPI: https://pypi.org/ | • NVD: https://nvd.nist.gov/ | • PyPI Rest services |
| Ruby | • RubyGems | • Gemfile, Gemfile.lock and *.gemspec files | RubyGems: https://rubygems.org/ | • NVD: https://nvd.nist.gov/<br>• Bundler audit database | • License and obsolescence pending |
| Scala | • Maven | • SBT (build.sbt) | Maven (central or others configured in settings.xml or pom.xml files):<br><br>https://repo.maven.apache.org/maven2/ | • NVD: https://nvd.nist.gov/ | • pom.xml. |
| Swift | • Cocoapods<br>• GitHub | • Podspec (*.podspec, Podfile.lock files)<br>• Package (Package.swift files) | Repository Podspec in Github:<br><br>https://github.com/CocoaPods/Specs | • NVD: https://nvd.nist.gov/ | • podspec.json of the component. |

The **Components Inventory** section shows the following information:

1. Overall Information on Components – aggregated information on number and type of components
2. List of Components – detailed listing of components
3. Component detail – detailed information on selected component




## Overall Information on Components


- Number of components by language
- Number of components by Security Risk level (High, Medium, Low and None)
- Alerts :

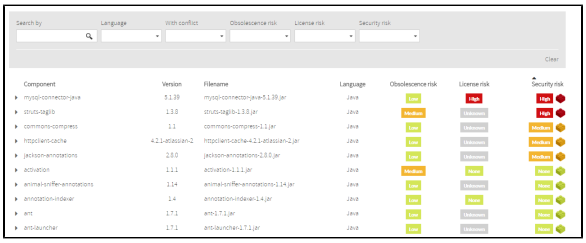
- Components with High-Security Risk
- Components being used with different versions that might cause conflicts
- Etc.

## List of Components

Kiuwan Insights provides a full list of all those components being used by your application.

For every 3rd party component, you will have access to detailed component information such as:

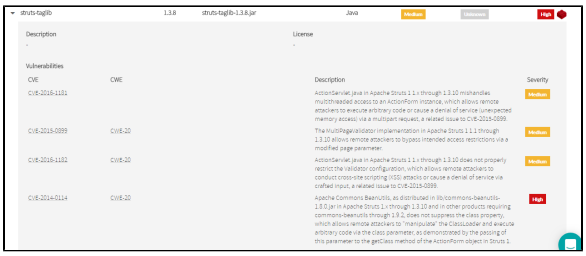
| Name | Description |
|---|---|
| Component name | Name of the component |
| Version | The version(s) in use |
| Filename | The physical container (.jar, .dll, .js, etc) |
| Language | The programming language it is written in. |
| Obsolescence risk | A component's ***Obsolescence Risk*** is a measure of the risk level relative to:<br>• the antiquity of your version respect to the latest version, and<br>• how active is the component<br><br>Both values are combined in the Obsolescence Risk to provide a value of the risk associated with using outdated or "dead" components.<br><br>Please visit Obsolescence Risk for further information. |
| License risk | A component's **License Risk** is a measure of the risk level relative to the legal implications of used components' licenses.<br><br>Please visit Insights Licenses for further information. |
| Security risk | A component's **Security Risk** is based on CVSS v2 **Base Scores** (**Severities)** of its vulnerabilities:<br>• If the selected component has more than one vulnerability, Kiuwan will label the component with the highest severity value of all the vulnerabilities of the component.<br>• If the selected component has only one vulnerability, the Severity of that vulnerability will be the Security Risk of the component. |




## Component details

By clicking on a component, you will have access to the following information:

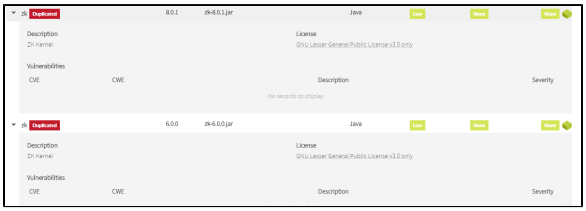
- The description of the component
- The license of the component
- Found vulnerabilities of the selected component:
  - CVE identifier, and link to NIST National Vulnerability Database desc page
  - CWE type, and link to MITRE Common Weakness Enumeration desc page
  - Vulnerability description
  - Severity (more on this at Security Risk )




## Duplicated components

With Kiuwan Insights you can identify different versions of the same component used by your application.

The below example shows that the analyzed application is incorporating two different versions of ZKoss common library: 8.0.1 and 6.0.0




Most probably, this component duplication is not intended, and it's something that would produce maintainability headaches when upgrading to a newer version of the library.