

Template Data Languages

IntelliJ IDEA supports using template data languages. For the template data files, IntelliJ IDEA provides dedicated file types, coding assistance, syntax highlighting, inspections and intention actions.

In this section:

- [Template Data Languages](#)
 - [File types](#)
 - [Code completion](#)
 - [Inspections and intention actions](#)
- [Specifying Template Data Languages for Templates](#)

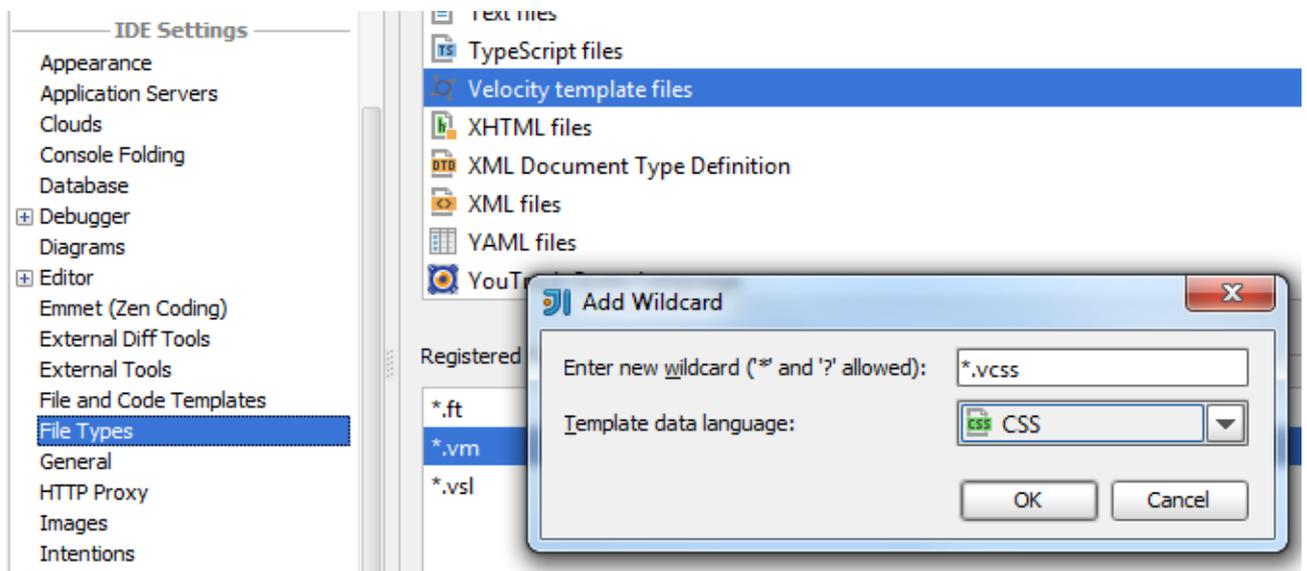
File types

IntelliJ IDEA can recognize Velocity and FreeMarker templates by file extension (for example, *.vm or *.ftl), and even by double file extension (for example, *.java.vm or *.css.ft)

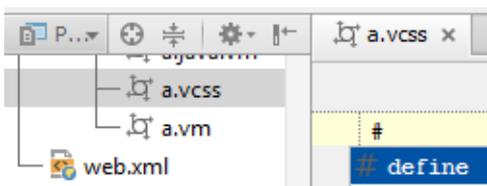
Besides that, it is possible to map files to the various template languages by defining own registered patterns.

To create a new file extension pattern for a template language, follow these steps

1. In the [File Types](#) page of the Settings dialog, select the desired template language type, for example, Velocity template files, or FreeMarker template files.
2. In the **Registered Patterns** section, click **+**.
3. In the **Add Wildcard** dialog box, enter the desired pattern, and then from the **Template data language** drop-down list, select the desired data language.

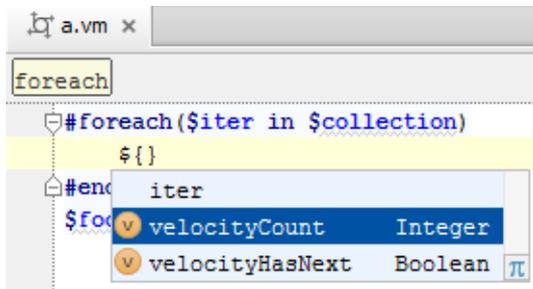
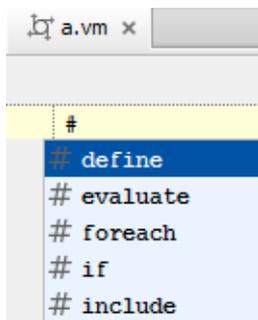


4. Apply changes and close the Settings dialog. Create a new file with the new extension. IntelliJ IDEA recognizes it as a template file, and provides all the necessary assistance:



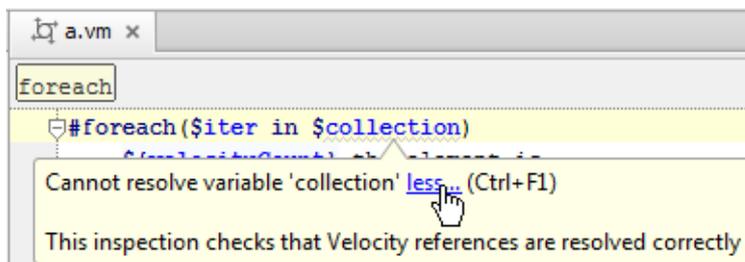
Code completion

IntelliJ IDEA supports [code completion](#) for the template files:

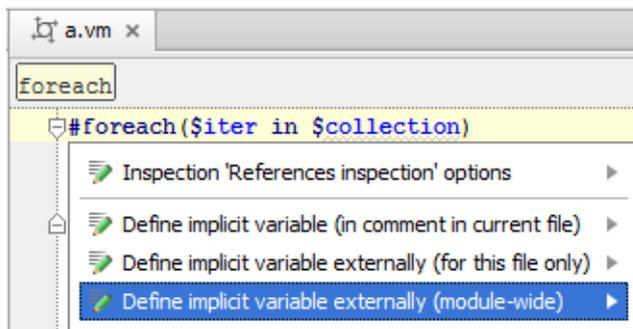


Inspections and intention actions

IntelliJ IDEA provides inspections to recognize unresolved references:

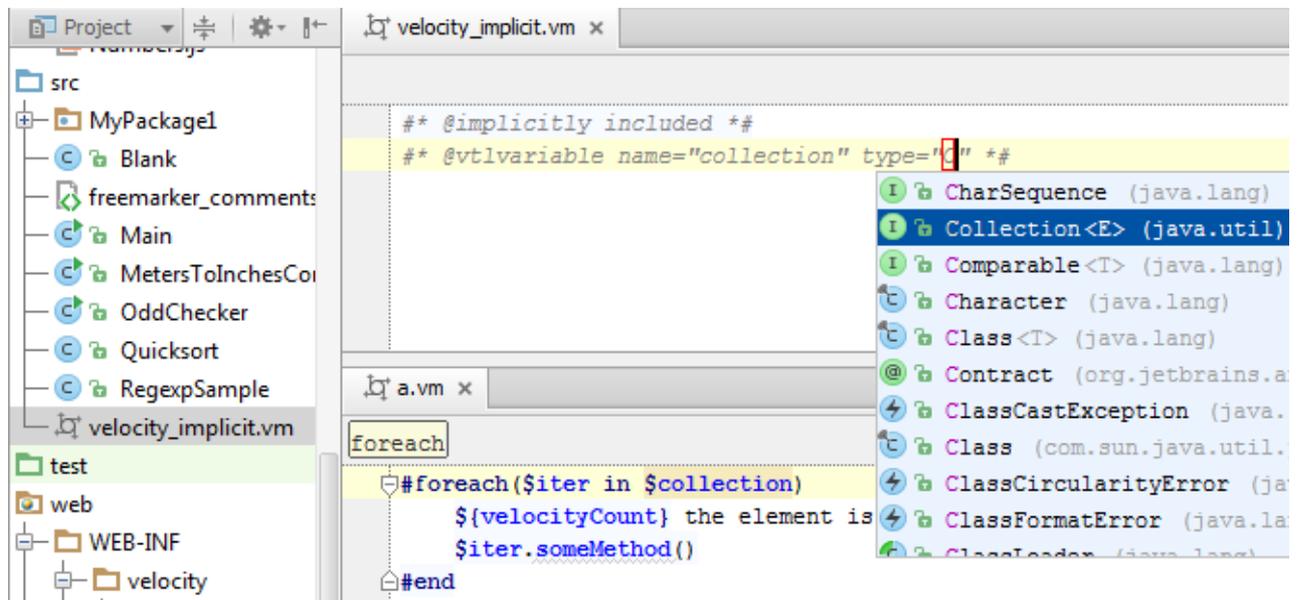


The unresolved references can be fixed in a number of ways, for example, by creating a comment in the same file, or by creating a separate file with comments.



A file with the default name `velocity_implicit.vm` or `freemarker_implicit.ftl` is created under the source root. Such file starts with the comment

```
** @implicitly included **
```



Use code completion to define the reference type.

Further, you can rename this file, or move to a different location within the source root, and the definition of a reference will not be lost.

See Also

Reference:

- [Template Data Languages](#)
- [File Types](#)

Web Resources:

- [Developer Community](#)