Code Inspections

IntelliLang contains a set of inspections that validate the correct use of the supplied annotations (or custom configured ones). These inspections can be configured through the regular Settings | Errors configuration dialog.

Language injection

The inspections in this category apply to the language injection feature related to the `@Language` annotation.

Unknown language id

This inspection provides validation for the use of non-existing Language-IDs. It flags usages of incorrect values of the `@Language` value attribute, such as `@Language("NonExistingID")`.

Language mismatch

Validation for using references to elements that are annotated as containing different languages or are not annotated at all. The inspection offers a Quick-Fix to annotate such elements with the right annotation for the expected language.

Injection not applicable

This inspection checks whether an @Language or any derived annotation is used for anything other than elements of type `String` or `String[]`.

Pattern validation

This category contains inspections about validating the use of the @Pattern or its derived annotations.

Validate annotated patterns

This inspection validates that expressions (String literals, as well as other compile-time constant or substituted expressions) match the pattern required by the `@Pattern` annotation. The inspection has an option to ignore non-constant expressions that contain non-substituted references and offers a Quick-Fix to add a substitution where applicable.

Pattern annotation not applicable

Checks whether a pattern-validation annotation (`@Pattern` or derived ones) is valid to be applied to the annotated element. Only elements of type `String` may be annotated.

Non-annotated method overrides @pattern method

This inspection checks whether a method without any `@Pattern` or derived annotation overrides an annotated method from its base classes. This is not necessary for the error-highlighting inside the editor, however the runtime-check instrumentation doesn't pick up annotations from base-class methods.

A Quick-Fix is provided to add an annotation that matches the one from the base-class method. This ensures that the runtime-check instrumentation works correctly.

See Also

Concepts:

- Code Inspection

Web Resources:

- Developer Community