Inferring Nullity

IntelliJ IDEA makes it possible to analyse the source code for the elements that can become null, and annotate them, provided that annotations are available in the project sources.

To automatically annotate nullable and non-null elements

1. Make sure that `annotations.jar` is added to your project. If it is not the case, IntelliJ IDEA will suggest to configure the annotations first. This can be done either manually (**File | Settings | Project Settings - Inspections - Probable bugs - Constant conditions and exceptions**), or automatically.
2. On the main menu, choose **Analyze | Infer Nullity**.
3. In the Specify Infer Nullity Scope dialog box, do the following:
   - Select the scope where you want to infer nullity: the entire project, the current file, etc.
   - If you want to perform nullity analysis for the test sources as well, and annotate local variables, select the corresponding check boxes.

   Click **OK**. IntelliJ IDEA adds import statement for annotations if required, and annotates parameters and variables.

Example

Consider the following code:

```
      public Color myMethod(){
Color color = null;
return color;
}
```

Infer nullity for this code, with the check box **Annotate local variables** selected. IntelliJ IDEA annotates the method and local variable with the `@Nullable` annotation:

```
      @Nullable
      public Color myMethod(){
@Nullable Color color = null;
return color;
}
```

However, if in the initial code the variable is initialized with some value, rather than null, the method and the local variable will be annotated with the `@NotNull` annotation:

```
      @NotNull
      public Color myMethod(){
@NotNull Color color = new Color(255);
return color;
}
```

See Also

Reference:

- Specify Dependency Analysis Scope Dialog

External Links:

- Auto-infer @Nullable/@NotNull Annotations

Web Resources:

- Developer Community