Abstract

Annotations are descriptive declarative information that can be associated with program elements. They can be used to represent metadata. This is useful in many cases like providing documentation, connecting to database etc. These annotations are later read by the execution environment and appropriate action will be taken. For custom annotations, a process called
Reflection is used to take necessary action. Although Java introduced this feature recently, attributes / annotations have been a topic of interest for researchers. There are tools and applications which make use of annotations. This paper, we will first have a look at annotations and annotation types. Then we go on to discuss in detail how annotations support injecting dependencies into resources like data sources, mail sources, environment entries, EJBs, web services, and so on. The new features are targeted to shift the responsibility of writing the boilerplate code from the programmer to the compiler or other tools. The resulting code is less likely to be bug-prone.

Reference

- Sun Documentation on Annotations http://java.sun.com/j2se/1.5.0/docs/guide/language/annotations.html
- Sun Documentation on Reflection http://java.sun.com/docs/books/tutorial/reflect
- "Web Tier to Go With Java EE 5: A Look at Resource Injection"
- Chris Male, David Pearce, Alex Potanin, and Constantine Dymnikov” Java bytecode verification for @NonNull types”, In Compiler Construction: 14th International Conference, CC 2008, pages 229–244, Budapest, Hungary, April 3–4, 2008.


**Index Terms**

<table>
<thead>
<tr>
<th>Computer Science</th>
<th>Software Engineering</th>
</tr>
</thead>
</table>

**Key words**

- Annotations
- Security
- Dependency Injection
- Transactions