Creating Annotations



Afzaal Ahmad Zeeshan
DEVELOPER ADVOCATE

@afzaalvirgoboy www.afzaalahmadzeeshan.com

Overview



Creating a custom annotation

Scenario: Adding version labels

Explore annotation initialization

Provide default values

The "value" field



Creating @Version Annotation

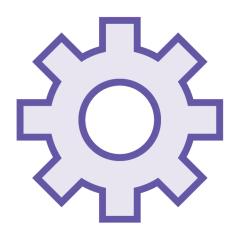


Question:

Create a Java Annotation to provide version information for Java types. Handle the default value for the version and allow developers to pass extra arguments such as author name.



Use-case of Java Annotation



Configuration
Provide details to configure system



Readability

Make the code more Java and less XML



Metadata

Additional details for a

Java type



Demo



Create @Version Annotation

Define fields



Interface and Annotation Types



Why Interface?



Provide a contract of metadata to be provided or generated



Utilize the interface features, such as default values



Annotations do not contain business logic, only metadata



Normal Annotations



Normal annotations accept an optional list of parameters with the annotation name.

Most used annotations in Java SDK

@Version(major=1, minor=2, release=true)

Marker Annotations

Has an annotation name but does not accept parameters or arguments

Is used to label the types

- @Deprecated
- @Documented

It is possible to ignore the brackets



Single-element Annotations

Has an annotation name and accepts a single argument in construction

Array-values are accepted

- @Platform(name="Pluralsight")
- @Languages({ "Java", "Kotlin" })
- @Author(@User(firstName="Afzaal Ahmad", lastName="Zeeshan"))

At least one field named "value" and all other fields have default values



Demo



Customizing annotations

Providing Default values



Demo



Adding a value field

Default values and value field



Summary



Created a Java Annotation

- @Version

Types of Annotations

Defaults values

Value field



Up Next: Annotating Java Annotations

