

# JAVADOCS y OPENAPI

Vladimir Edwin Alaro



# Que es JAVADOC?

**Javadoc** es la herramienta de java para generar documentación básica para el programador a partir del código fuente

```
1  /**
2   * HelloWorld
3   */
4  public class HelloWorld {
5
6      public static void main(String[] args) {
7          System.out.println("Hello World!");
8      }
9  }
```

PACKAGE

CLASS

USE

TREE

DEPRECATED

INDEX

HELP

PREV CLASS

NEXT CLASS

FRAMES

NO FRAMES

SUMMARY: NESTED | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

com.baeldung.javadoc

Class SuperHero

java.lang.Object

com.baeldung.javadoc.Person

com.baeldung.javadoc.SuperHero

public class **SuperHero**

extends Person

Hero is the main entity we will be using to ...

Author:

Captain America

Field Summary

Fields

| Modifier and Type        | Field and Description  |
|--------------------------|--|
| private int              | defense  |
| private int              | health   |
| private java.lang.String | heroName<br>The public name of a hero that is common knowledge |
| private java.lang.String | uniquePower  |

# EJEMPLO DE JAVADOC

[illegible]**blankString**

```
public static boolean blankString(String string)
```

Check if value is empty, regardless of whether it is a string with pure spaces. Examples

- input: null, output: true
- input: "", output: true
- input: " ", output: true
- input: " x ", output: false

### Parameters:

string - value to process

**Returns:**

## Boolean

# Tags

| Tag         | Descripción  | Uso   | Versión |
|-------------|--|---|---------|
| @author     | Nombre del desarrollador.  | nombre_autor  | 1.0     |
| @version    | Versión del método o clase.  | versión   | 1.0     |
| @param      | Definición de un parámetro de un método, es requerido para todos los parámetros del método.                                      | nombre_parametro descripción  | 1.0     |
| @return     | Informa de lo que devuelve el método, no se puede usar en constructores o métodos "void".  | descripción   | 1.0     |
| @throws     | Excepción lanzada por el método, posee un sinónimo de nombre @exception  | nombre_clase descripción  | 1.2     |
| @see        | Asocia con otro método o clase.  | referencia (#método());<br>clase#método(); paquete.clase;<br>paquete.clase#método()). | 1.0     |
| @since      | Especifica la versión del producto   | indicativo numerico   | 1.2     |
| @serial     | Describe el significado del campo y sus valores aceptables. Otras formas válidas son @serialField y @serialData                  | campo_descripcion   | 1.2     |
| @deprecated | Indica que el método o clase es antigua y que no se recomienda su uso porque posiblemente desaparecerá en versiones posteriores. | descripción   | 1.0     |

<https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html#tag>

# Cómo se generan los javadocs en maven y gradle?

```
apply plugin: 'java'

javadoc {
    options.charSet = 'UTF-8'
    options.encoding = 'UTF-8'
}
```

```
task javadocArtifact(type: Zip) {
    dependsOn javadoc
    classifier 'javadoc'
    from 'build/docs/javadoc/'
}

artifacts {
    archives javadocArtifact
}
```

**GRADLE** > ./gradlew javadoc

**MAVEN** > mvn clean package javadoc:jar

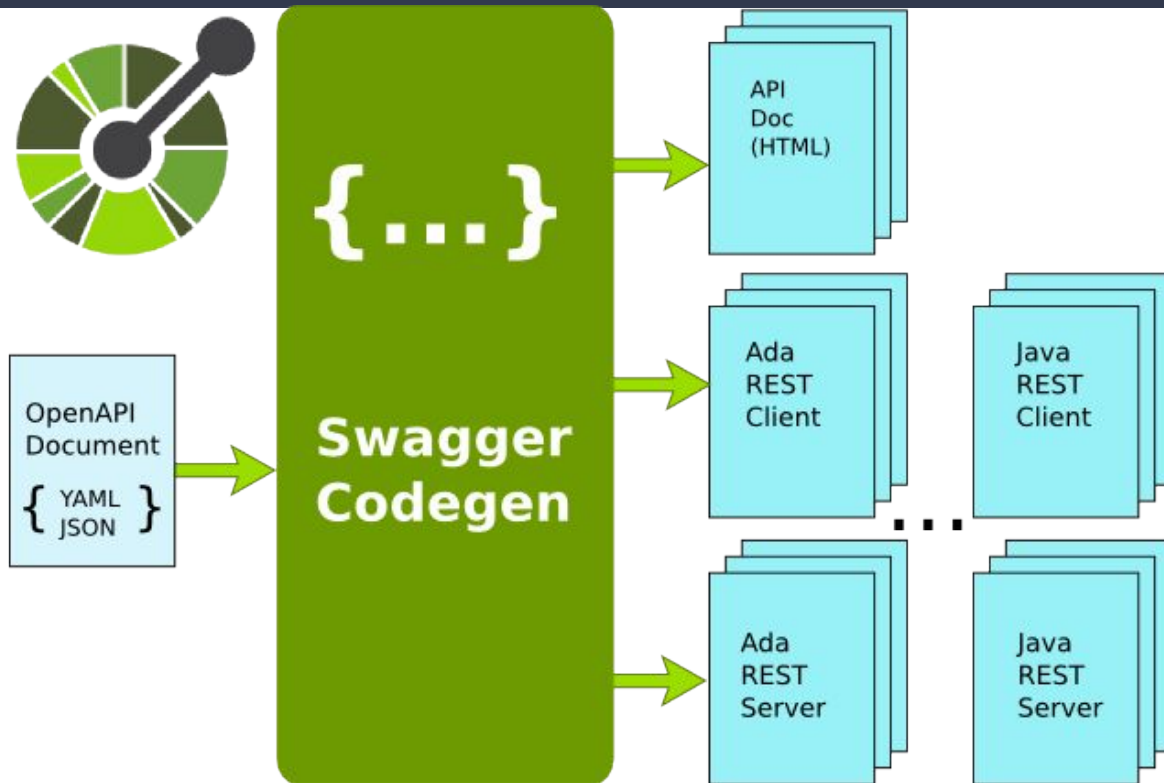
# La importancia



# OPENAPI



# Generación





# DEMO

## MicroProfile Starter

Generate MicroProfile Maven Project with Examples

groupId \*

com.example.demo

artifactId \*

demo-openapi

MicroProfile Version

MP 3.3



Java SE Version

Java 8



### Project Options

MicroProfile Runtime \*

Payara Micro



Examples for specifications [Select All](#) [Clear All](#)

☐

Config

☐

Fault Tolerance

☐

JWT Auth

☐

Metrics

☐

Health Checks

☐

OpenAPI

☐

OpenTracing

☐

TypeSafe Rest Client

DOWNLOAD