



# CS131a

## Introduction to JavaDocs

Dimokritos Stamatakis

Spring 2021, Brandeis University

# What is JavaDocs

- Tool to create standard documentation of Java code
  - HTML-based documentation
- Others can understand what your code does
- Easy to write and generate
  - Use specific syntax and rules
- Java Library API uses Java Docs
  - You can do it too!

# How to create Java Docs

- JavaDocs look like regular comments but start with `/**`
- Every next line starts with `*`
  - JavaDoc generator ignores the first `*` of each line
- They end with `*/`
- You can use HTML tags like `<b>`, `<i>`
- Use JavaDocs tags to specify attributes

# JavaDoc tags

- `@author`
  - Information about the author of this code (name, address)
- `@version`
  - The version number of this code
- `@param`
  - The parameter name and description of a method or constructor
- `@return`
  - The return value of a method
- `@throws`
  - Exceptions that are thrown by a method or constructor
- `@deprecated`
  - The class field, or method is deprecated and shouldn't be used
- `@see`
  - Adds references to Java Library Docs or local JavaDocs

# Examples – class User

```
/**  
 * Holds information about a user  
 * @author dimos  
 * @version 1.1.9  
 */  
public class User {
```

# Examples – members of class User

```
/**
 * The name of the user
 */
private String name;
/**
 * The email of the user
 */
private String email;
/**
 * The ID of the user
 */
private long id;
/**
 * The last time that this user logged in
 */
private Date lastLogin;
/**
 * The URL of the preferences file of this user
 */
private String preferencesFile;
```

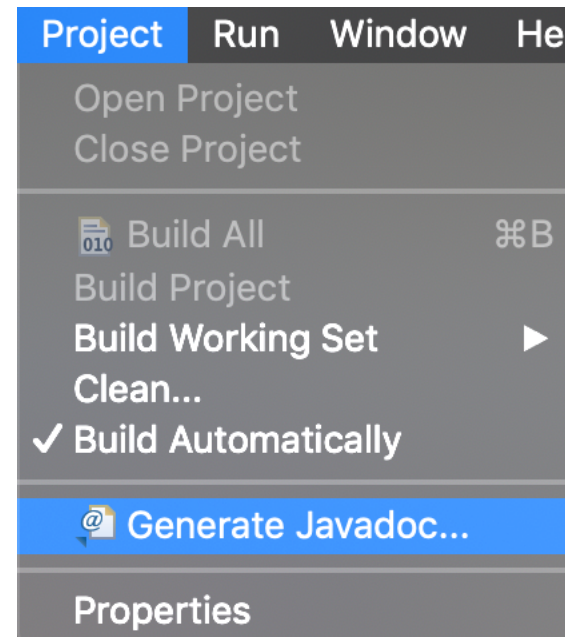
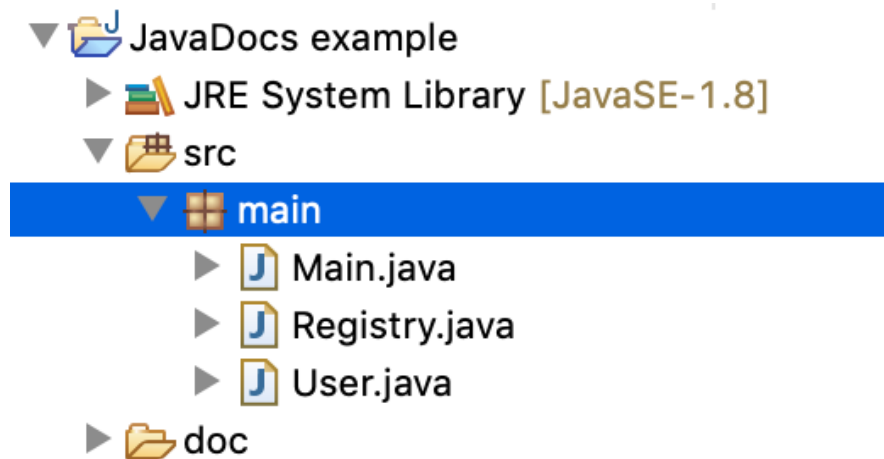
# Examples – constructor and methods

```
/**
 * Creates a new user
 * @param name the name of the user
 * @param email the email of the user
 * @param id the id of the user
 */
User(String name, String email, long id){
    this.name = name;
    this.email = email;
    this.id = id;
}

/**
 * Opens the file specified in <i>filePath</i>, reads the preferences and
 * stores within the preferences member of this user.
 * @param filePath the specified file path to read the preferences from
 * @see #loadPreferences()
 * @throws IOException When there is an error opening the specified file
 */
public void loadPreferences(String filePath) throws IOException {
    File f = new File(filePath);
    BufferedReader br = new BufferedReader(new FileReader(f));
    while(br.ready()) {
        this.preferences += br.readLine();
    }
    br.close();
}
```

# Generating JavaDocs in Eclipse

- Click on the desired project on the left of Package Explorer
- Project -> Generate Javadoc...






# Generating JavaDocs in Eclipse

Javadoc command:

/Library/Java/JavaVirtualMachines/jdk1.8.0\_144.jdk/Contents/Home/bin/javadoc Configure...

Select types for which Javadoc will be generated:

<input checked="" type="checkbox"/>  JavaDocs example	
<input type="checkbox"/>  pat-1	

Create Javadoc for members with visibility:

☒ Private ☐ Package ☐ Protected ☐ Public

Private: Generate Javadoc for all classes and members.

☒ Use standard doclet

Destination: /Users/dimos/eclipse-tutorial/JavaDocs example/doc Browse...

☐ Use custom doclet

Doclet name:

Doclet class path:



< Back

Next >

Cancel

Finish

# Generating JavaDocs in Eclipse

Loading source files for package main...

Constructing Javadoc information...

Standard Doclet version 1.8.0\_144

Building tree for all the packages and classes...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/Main.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/Registry.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/User.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/package-frame.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/package-summary.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/package-tree.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/constant-values.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/class-use/Registry.html...

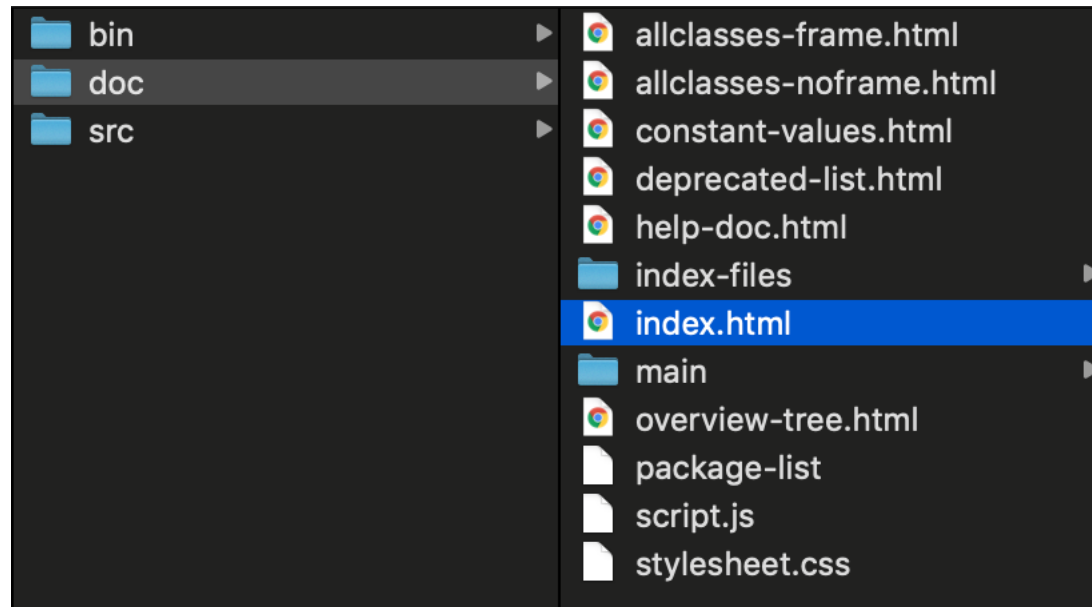
Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/class-use/Main.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/class-use/User.html...

Generating /Users/dimos/eclipse-tutorial/JavaDocs example/doc/main/package-use.html...

Building index for all the packages and classes...

# Viewing JavaDocs



# Viewing JavaDocs – all classes

← → ↺ ⓘ File | /Users/dimos/eclipse-tutorial/JavaDocs%20example/doc/index.html

All Classes

Main  
Registry  
User

PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

## Package main

Class Summary

Class	Description
<b>Main</b>	The Main class performing operations on the registry of users
<b>Registry</b>	Holds the registry of users
<b>User</b>	Holds information about a user

PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV PACKAGE NEXT PACKAGE FRAMES NO FRAMES

# Viewing JavaDocs – class User

main

## Class User

java.lang.Object  
main.User

```
public class User  
extends java.lang.Object
```

Holds information about a user

**Version:**

1.1.9

**Author:**

dimos

### Field Summary

#### Fields

Modifier and Type	Field and Description
private java.lang.String	<b>email</b> The email of the user
private long	<b>id</b> The ID of the user
private java.util.Date	<b>lastLogin</b> The last time that this user logged in
private java.lang.String	<b>name</b> The name of the user
private java.lang.String	<b>preferences</b> The preferences of this user

# Viewing JavaDocs – constructor and methods

## Constructor Summary

### Constructors

#### Constructor and Description

**User**(java.lang.String name, java.lang.String email, long id)

Creates a new user

## Method Summary

### All Methods

### Instance Methods

### Concrete Methods

#### Modifier and Type

#### Method and Description

java.lang.String

**getEmail()**

Returns the email of this user

long

**getID()**

Returns the id of this user

java.util.Date

**getLastLogin()**

Gets the last time that this user logged in

# Viewing JavaDocs – methods detailed

## loadPreferences

```
public void loadPreferences(java.lang.String filePath)
                        throws java.io.IOException
```

Opens the file specified in *filePath*, reads the preferences and stores within the preferences member of this user.

### Parameters:

`filePath` – the specified file path to read the preferences from

### Throws:

`java.io.IOException` – When there is an error opening the specified file

### See Also:

[loadPreferences\(\)](#)

Questions?