

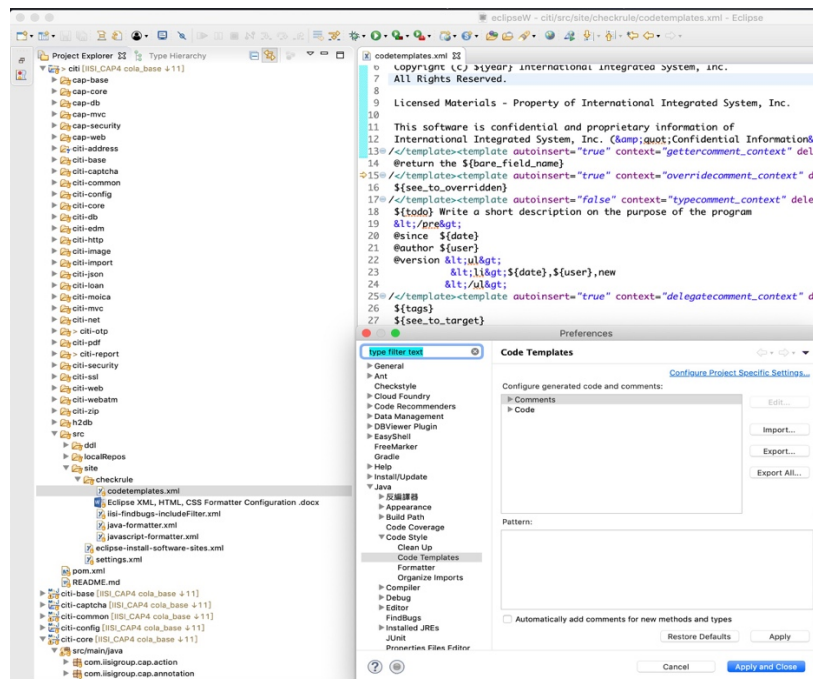
如何撰寫及產生 Javadoc

一、如何撰寫 Javadoc

1. 匯入 codetemplates.xml

- 路徑：citi/src/site/checkrule/codetemplates.xml
- Eclipse 匯入步驟：

Preferences > Java > Code Templates > Import > 指向 codetemplates.xml
位置 > Apply



2. 於 java 上進行註解標記

(1) 註解標記說明

標記	說明
@version	指定版本
@since	指定最早出現的版本
@author	指定作者
@see	生成參考其它 Javadoc 的連結
@link	類似於@see 的功能，可為注釋中的特定詞彙產生連結
@deprecated	表示已無使用或將被廢棄
@param	描述參數值
@return	描述傳回值
@throws	描述擲出的例外，或觸發該例外的條件

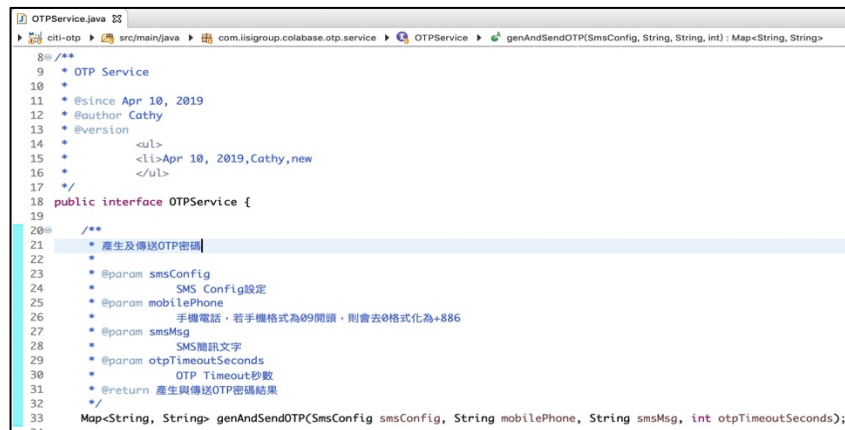
(2) Java 程式註解

1. 加入註解符號規則：

1. 將「註解開始符號」 `/**` 縮排，對齊欲註解的程式碼。
 2. 從第二行開始撰寫註解內容（第一行為「註解開始符號」），每行註解均以星號 `*` 作為開頭，並於「註解開始符號」的第一個符號對齊。
 3. 在說明文字與標籤列表之間插入一行空白註解。
 4. 最後一行（不包括註解內容）為「註解結束符號」 `*/`。
- * 以上規則若有匯入 `codetemplates.xml`，於 Eclipse Code formatter 熱鍵（`Ctrl+Shift+F`、`Ctrl+Shift+O`），格式則會自動排版。

➤ 註解範例：

若有分 Interface 介面 和 Implement 實作時，將 method 的註解寫在 Interface 即可，於 Interface method 打 `/**` 自動產生註解撰寫各標記內容。



```
8 /**
9  * OTP Service
10 *
11 * @since Apr 10, 2019
12 * @author Cathy
13 * @version
14 *
15 * <ul>
16 * <li>Apr 10, 2019, Cathy, new
17 * </li>
18 * </ul>
19 */
20
21 public interface OTPService {
22
23     /**
24      * 產生及傳送OTP密碼
25      *
26      * @param smsConfig SMS Config設定
27      * @param mobilePhone 手機電話，若手機格式為09開頭，則會去0格式化為+886
28      * @param smsMsg SMS簡訊文字
29      * @param otpTimeoutSeconds OTP Timeout秒數
30      * @return 產生與傳送OTP密碼結果
31      */
32     Map<String, String> genAndSendOTP(SmsConfig smsConfig, String mobilePhone, String smsMsg, int otpTimeoutSeconds);
33 }
```

➤ Implement 註解範例：

Implement 實作時僅需連結 Interface 介面註解，於 Implement method 打 `/**` 自動產生連結 Interface 註解，如下圖 58~62 行。



```
30 /**
31  * OTP Service Implement
32  *
33  * @since Apr 16, 2019
34  * @author Cathy
35  * @version
36  *
37  * <ul>
38  * <li>Apr 16, 2019, Cathy, new
39  * </li>
40  * </ul>
41 */
42
43 @Service
44 public class OTPServiceImpl implements OTPService {
45
46     private static final DecimalFormat OTP_DECIMAL_FMT = new DecimalFormat("000000");
47     private final Logger logger = LoggerFactory.getLogger(getClass());
48
49     // OTP回傳參數
50     /** OTP Key值 */
51     public static final String OTP_KEY = "otp";
52     /** 傳送OTP簡訊文字訊息 */
53     public static final String OTP_SMS_MSG = "otpSmsMsg";
54     /** 傳送OTP次數 */
55     public static final String OTP_RETRY_MSG = "retryMsg";
56     /** 是否傳送OTP次數 */
57     public static final String IS_MAX_RETRY = "isMaxRetry";
58     /** 是否傳送OTP */
59     public static final String IS_SEND_OTP = "isSendOTP";
60
61     /*
62      * (non-Javadoc)
63      * @see com.iisigroup.colabase.otp.service.OTPService#genAndSendOTP(com.iisigroup.colabase.otp.model.SmsConfig, java.lang.String, java.lang.String)
64      */
65     @Override
66     public Map<String, String> genAndSendOTP(SmsConfig smsConfig, String mobilePhone, String smsMsg, int otpTimeoutSeconds) {
67         Map<String, String> otpMap = new HashMap<>();
68         try {
69             if (!CapString.isEmpty(mobilePhone)) {
70                 String otp = generateOTP();
71                 otpMap.put(OTP_KEY, otp);
72                 String otpSmsMsg = MessageFormat.format(smsMsg, otp, otpTimeoutSeconds);
73                 String otpSendOTP = "otpSendOTP";
74                 otpMap.put(otpSendOTP, otpSendOTP);
75             }
76         } catch (Exception e) {
77             logger.error("OTPService genAndSendOTP error: " + e.getMessage());
78         }
79         return otpMap;
80     }
81 }
```

Class 註解

常數註解

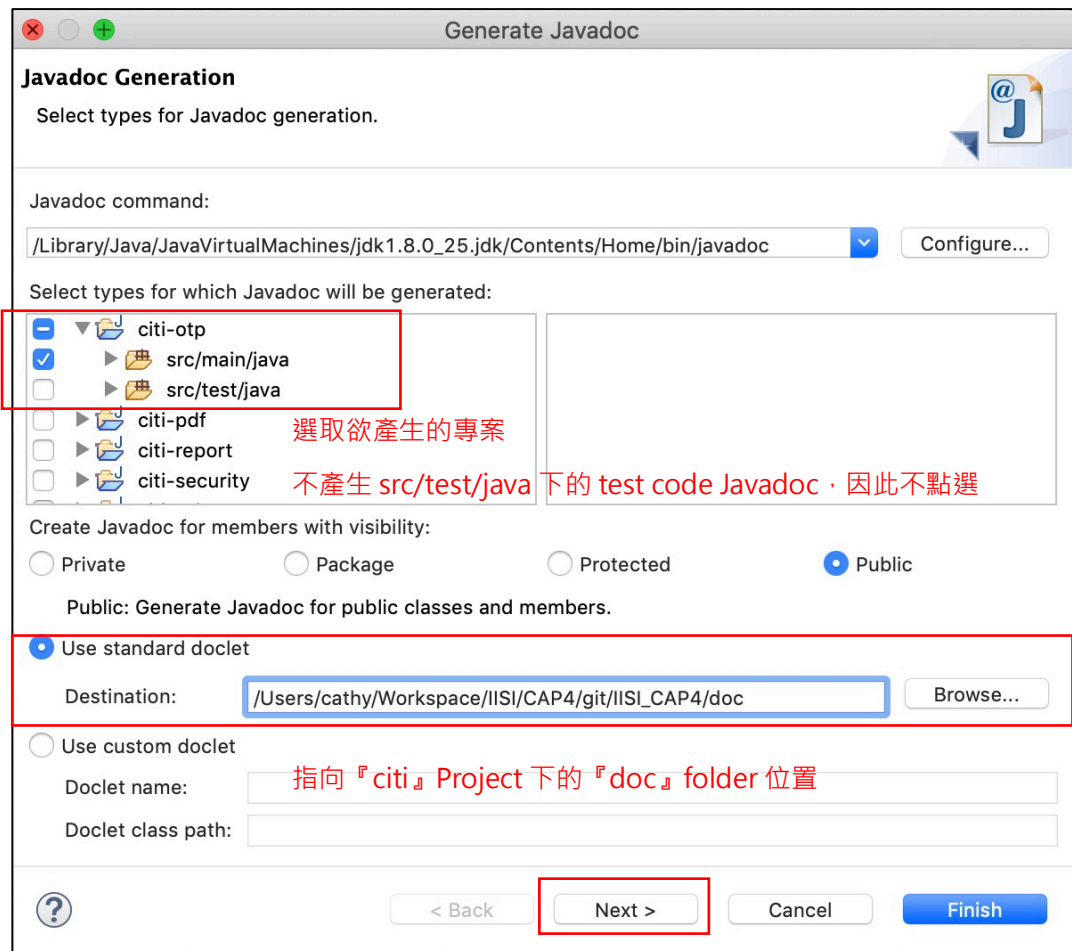
實作註解

二、How to use Eclipse generate Javadoc

1. Eclipse 產生 Javadoc 步驟：

(1). Project > Generate Javadoc...

(2). Javadoc 放置於『citi』Project 下的『doc』folder，於 Use standard doclet Destination 指向路徑位置



Generate Javadoc

Javadoc Generation
Configure Javadoc arguments for standard doclet.

☐ Document title: Cola Base - OTP Module

Basic Options

- ☒ Generate use page
- ☒ Generate hierarchy tree
- ☒ Generate navigator bar
- ☒ Generate index
- ☒ Separate index per letter

Document these tags

- ☒ @author
- ☒ @version
- ☒ @deprecated
- ☒ deprecated list

Select referenced archives and projects to which links should be generated:

- ☐ antlr-2.7.7.jar - not configured
- ☐ aopalliance-1.0.jar - not configured
- ☐ aspectjrt-1.8.9.jar - jar:file:/Users/cathy/.m2/repository/org/aspectj/aspectjrt/1.8.9/aspectjrt-1.8.9.jar
- ☐ aspectjweaver-1.8.9.jar - jar:file:/Users/cathy/.m2/repository/org/aspectj/aspectjweaver/1.8.9/aspectjweaver-1.8.9.jar
- ☐ cdi-api-1.1-PFD.jar - jar:file:/Users/cathy/.m2/repository/javax/enterprise/cdi-api/1.1/cdi-api-1.1-PFD.jar
- ☐ charsets.jar - https://docs.oracle.com/javase/8/docs/api/
- ☐ classes - not configured

☐ Style sheet:

➤ 避免中文編碼產生亂碼，於 VM Options 輸入 -charset UTF-8 -encoding UTF-8

Generate Javadoc

Javadoc Generation
Configure Javadoc arguments.

☐ Overview:

VM options (prefixed with '-J', e.g. -J-Xmx180m for larger heap space):

-charset UTF-8 -encoding UTF-8

Extra Javadoc options (path names with white spaces must be enclosed in quotes):

JRE source compatibility: 1.8

☐ Save the settings of this Javadoc export as an Ant script:

Ant Script:

☐ Open generated index file in browser

