

主題 3：制定開發規範

Java Coding Standard

適用範圍	參考規則		舉例					
package	1. 根據系統或架構等名稱做識別。 2. 名稱全部小寫，連續單字直接寫在一起。		com.fubonlife.mytest.common.model.productspec					
檔案名稱	用途	CLASS 種類	名稱(大寫駝峰式)	package 語句				
	Rest API Controller	class	後綴文字 Controller	com . fubonlife . 系統名稱 . api . controller				
	配置文件	class	後綴文字 Configuration	com . fubonlife . 專案名稱 . common . config				
	服務的介面	interface	後綴文字 Service	com . fubonlife . 系統名稱 . common . service				
	服務代理者	class	後綴文字 Impl	com . fubonlife . 專案名稱 . common . service . impl				
	常駐(static)工具集	class	後綴文字 Util	com . fubonlife . 專案名稱 . common . util				
	資料表的資料封裝	class	資料表名稱	com . fubonlife . 專案名稱 . common . entity . 用途名稱				
	資料庫資料CRUD	interface	後綴文字 Repository	com . fubonlife . 專案名稱 . common . repository . 用途名稱				
	傳輸型的資料封裝	class	後綴文字 Dto	com . fubonlife . 系統名稱 . common . model				
	特定畫面資料封裝	class	後綴文字 Vo	com . fubonlife . 系統名稱 . common . model				
	列舉	enum	後綴文字 Enum	com . fubonlife . 專案名稱 . common . enumS				
	過濾器	class	後綴文字 Filter	com . fubonlife . 專案名稱 . common . filter				
	自定義的共用元件	class	自定義	com . fubonlife . 專案名稱 . common . 功能名稱				
	定時排程	class	後綴文字 Schedule	com . fubonlife . 專案名稱 . batch . schedule				
	測試	class	後綴文字 Test					
【補充說明】								
• 當程式內容有包含標記「@Autowired」，建議命名成 XxxService								
Class 名稱 Interface 名稱 Enum 名稱	1. 建議是名詞OR動名詞。 2. 避免用頭字語或是縮寫（除非縮寫的形式被用的更廣泛例如 URL 或 HTML）		例如：ProductSpecService					
基本型變數名稱	1. 需命名成有意義的名稱 2. 存取範圍應宣告為【private】		應避免之示範：String a; int orz; boolean flag; 較好的示範：String userAddress; boolean isFubonEmployee; ※ 部門、科別編號變數命名：departmentCode、sectionCode					
常數型變數名稱	1. 命名以全部大寫的英文字表示 2. 單字與單字間使用底線【 _ 】分隔單字。 3. 存取範圍應宣告為【private】		例如：BATCH_SERVER 或 CHANNEL_CODE_AGENCY					
方法名稱前綴	用途	新增	單筆查詢	多筆查詢	修改	刪除		
	Rest API Controller、服務的介面	add	search	list	modify	remove		

字串與字串變數 OR 字串與數字變數 結合	<ol style="list-style-type: none">1. 只有一個變數時，可直接使用【+】來進行串接。2. 超過一個變數時，改透過【String.format 或 StringBuilder】來串接。	例如： <pre>String message = String.format("Hello %s !!", "Samon.Tsai"); StringBuilder content = new StringBuilder(); for (Object item : items) { content.append(item); }</pre>
變數的存取範圍	<ol style="list-style-type: none">1. 針對基本型 / 常數型變數，應特別指定為【private】2. 針對物件型變數名稱，應特別指定為【private】<ul style="list-style-type: none">• 若需透過Spring容器依賴注入，其存取範圍建議不用特別指定	
變數的資料型態	<ol style="list-style-type: none">1. 若非特殊情況，宜選用【boolean】而少用【Boolean】2. 若有要為型態為 BigDecimal 的變數賦值 0 或 1，需用 BigDecimal.ZERO 或 BigDecimal.ONE	

其他

適用範圍		參考規則
IDE	自動排版功能	<ol style="list-style-type: none">1. 縮排（indentation）是把程式碼依照結構作整理，使其有更高的可讀性2. 若程式碼要進行縮排對齊，請用 4 格空白
資料庫	主鍵的命名	原生資料表名稱，續接【_ID】為後綴文字
	CRUD語法	在 JPQL 語法或 SQL 語法中，若需宣告資料表別名，建議用底線分隔且取字首來命名

方法名稱前綴	用途		新增	查詢		
	資料庫資料 CRUD	命名式	—	findBy	—	
		舉例	—	<pre>Account findById(String employeeId); List<Department> findByDeletedFalse();</pre>		—
		聲明式 (JPQL、SQL)	—	selectXXX selectXXXBy000		update update
		舉例JPQL	—	<pre>@Query("select x from AccountAgent x where x.endDateDatetime <= :now and enabled = '1' ") List<AccountAgent> selectByExpiredRecordForEndDatetime(LocalDateTime now);</pre>		@ @ V
		舉例SQL	—	<pre>@Query(value = "select COUNT(*) from QUARTER_PLAN_M",nativeQuery = true) Integer countQuarterPlanMForHeadDpt();</pre>		(缺點)
	【補充說明】 <ul style="list-style-type: none">JPA 框架：Spring Data JPA					
URL	組成規則	限全小寫英文、數字，以及特殊符號【 - 】組合而成				
Git	Branch	1. 開發 <ul style="list-style-type: none">從Branch=dev，創建New Branch，命名為【 /feature/AD帳號/功能】完成後，發PR合併到Branch=dev 2. 測試 <ul style="list-style-type: none">從Branch=dev，創建New Branch，命名為【 /release-版號 】 3. 測試時除蟲 <ul style="list-style-type: none">從Branch=/release-版號，創建New Branch，命名為【 /bugfix/AD帳號 】只能修bug，不能偷渡埋入新功能完成後，發PR合併到Branch=dev，以及Branch=/release-版號 4. 上線 <ul style="list-style-type: none">發PR，讓Branch=/release-版號能合併到Branch=master				

	<div><div><div>Commit</div><div>Message</div></div><div><div><div>1. 格式 [type][issue編號] message</div><div>2. type 請選用下列清單</div><div><div><div><div>• feature</div><div>新增OR修改功能</div></div><div><div>• fix</div><div>除蟲</div></div><div><div>• amend</div><div>微調程式OR樣式（不影響程式碼執行的變動）</div></div><div><div>• refactor</div><div>重構功能</div></div><div><div>• optimize</div><div>優化效能</div></div><div><div>• config</div><div>改變設定檔</div></div><div><div>• test</div><div>編修測試程式</div></div><div><div>• build</div><div>改變 build 程序OR工具</div></div><div><div>• 3rd</div><div>增加OR刪除第三方套件</div></div><div><div>• delete</div><div>刪除檔案</div></div><div><div>• add</div><div>增加檔案</div></div></div></div><div><div><div>3. 建議 message 不只記錄做了「什麼」異動，最好也能寫下「為什麼」要這樣作</div><div>4. 若需前(後)端配合測試，請於 message 末端加註「待X端完成才可測試」</div><div>5. 除蟲時，不應該一次 commit 所有異動，應該獨立 commit 每個不同意義的異動，這樣的資訊能讓日後進行維護人員更快進入狀況</div></div></div></div></div></div>
--	--