开发人员指南

# 开发环境搭建

## Dev tools

• JDK 1.8

• Eclipse JEE

• Git

• Maven 3.6.0

• Tomcat 9.0.X

• Database：DB2

## 项目工程

Git地址：https://134.251.81.37/gitlab/groups/

开发分支：development

由台湾项目PM在gitlab上创建账号，账号下来后需要修改密码

参考：

1. https://www.jianshu.com/p/142b3dc8ae15

2. https://www.jianshu.com/p/1cda2620a5ba

### 工程说明

taroko-authbatch https://134.251.81.37/gitlab/taroko/taroko-authbatch.git

taroko-authintf https://134.251.81.37/gitlab/taroko/taroko-authintf.git

taroko-batch https://134.251.81.37/gitlab/taroko/taroko-batch.git

taroko-common <https://134.251.81.37/gitlab/taroko/taroko-common.git>

所有太魯閣專案(web,batch)所需的會被共同使用的jar檔、類別、介面皆需放置於此，如hsm-util

taroko-http https://134.251.81.37/gitlab/taroko/taroko-http.git

taroko-web <https://134.251.81.37/gitlab/taroko/taroko-web.git>

web專案共用物件及jar檔放置於此(僅有web會使用到的如busi等)

taroko-web-ecs <https://134.251.81.37/gitlab/taroko-web/taroko-web-ecs.git>

所有web程式(除了共用)皆放置於此，依赖taroko-common, taroko-web工程

### 專案相依性

taroko-common <- taroko-web <- taroko-web-ecs

因此假設開發taroko-web-ecs則會需要taroko-common, taroko-web, taroko-web-ecs

### 编译注意事项

工程编译运行需要DXC内网，maven的settings文件不要配置mirror，由工程pom文件中指定的私有镜像

## Test Environment

Web site : http://134.251.81.31:7001/ECS/MainControl

使用者: 0000 密码：

### 数据库配置

Database: DB2

databasename: DEMODB

Schema: DB2SA02

ip: 134.251.81.31

port: 50001

user name: db2sa02

password: 1qaz2wsx

Database: DB2

DBName: UT

Schema: ECSUTDB

ip: 134.251.81.31

port: 55000

user name: ecsutdb

password: 1qaz2wsx

### 菜單創建

secm0012 (作業程式資料維護)

### 復核主管

id: DXC

pwd: 002

## 运行调试

### taroko-web-ecs工程配置文件：

WebContent\EcsWeb\data\parm\TarokoParameter.txt

1. 复制WebContent目录下的EcsWeb目录到本机根目录下或者报错的路径下，
2. 然后在 \EcsWeb\data\parm 的TarokoParameter.txt 裡面加一行ECS\_LOGIN=Y
3. 在tomcat中添加Datasource的JNDI，配置信息见后面的JNDI配置，复制db2jcc4.jar到tomcat的lib目录
   1. 直接启动tomcat，修改conf目录下的context.xml；
   2. Eclipse中修改Servers下当前Server的context.xml；

### JNDI配置

<Resource

name=*"jdbc/Taroko\_SIT"*

auth=*"Container"*

type=*"javax.sql.DataSource"*

username=*"utap1"*

password=*"1qaz2wsx"*

driverClassName=*"com.ibm.db2.jcc.DB2Driver"*

url=*"jdbc:db2://134.251.81.31:55000/UT:currentSchema=ECSUTDB;currentFunctionPath=ECSUTDB;"*

maxTotal=*"3"*

maxIdle=*"1"*

maxWaitMillis=*"-1"*

validationQuery=*"select 1 from sysibm.sysdummy1"*

testOnBorrow=*"true"* />

# 开发规范

## 开发注意事项

1. 对于在已有功能上做修改，禁用IDE提供的Source format功能对源码进行格式化，会影响代码merge。对于要删除的代码使用注释方式，不要直接删除。完全自己开发的代码不在此限制范围内，但要注意代码美观，缩进统一。
2. Java代码顶部需要有modification log，每次修改前需要加注一笔修改大纲，见后面的“提交物注意事项”

## 提交物注意事项

因为客户方在台湾地区，所有提交物的中文必须使用正体字，即繁体。

### 代码

* Java代码顶部需要有modification log，每次修改前需要加注一笔修改大纲，格式如下：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*                                                                            \*

\*                              MODIFICATION LOG                              \*

\*                                                                            \*

\* DATE       Version   AUTHOR      DESCRIPTION                               \*

\* ---------  --------  ---------- ------------------------------------------ \*

\* 106-05-05  V1.00.00  yash           program initial                        \*

\* 108-10-04  V1.00.01  Alex           page bug fixed                         \*

\* 108-12-17  V1.00.02  Zuwei          remove field“???” for TCB project    \*

\*                                                                            \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

* source code的主要修改區也請加備註 (ex. 2019/12/18 update ??? )
* 不要物理删除原有代码，用注释方式。因為後續可能有原程式版本修改要再merge 進來的問題，所以原來程式的邏輯若要調整，請用單行註釋的方式，不要physical delete

### 测试报告

* 测试报告需要字体一致，不要用多种字体，维持原文档的字体，如果复制粘贴后字体改变，及时修改回原有的字体。默認字體為“DFKai-SB”
* 如果是从数据库获取数据并展示，需要证明抓取的数据没有问题，在测试报告中贴出DB的Data

## 提交规范

* 每次修改前先Pull最新代码
* 提交过程：commit -> pull -> check代码是否有冲突(如有冲突，解决冲突后再次提交)-> push
* 只允许提交与本次所做功能相关的代码，本地工程相关的文件不允许提交

## 前端规范

* 不需要传入后端的页面输入元素不应该有name属性，只用id属性，特殊情况除外，如button类型
* 与后端交互数据的输入元素尽量只提供name属性而不用id属性，特殊情况除外

## 后端规范

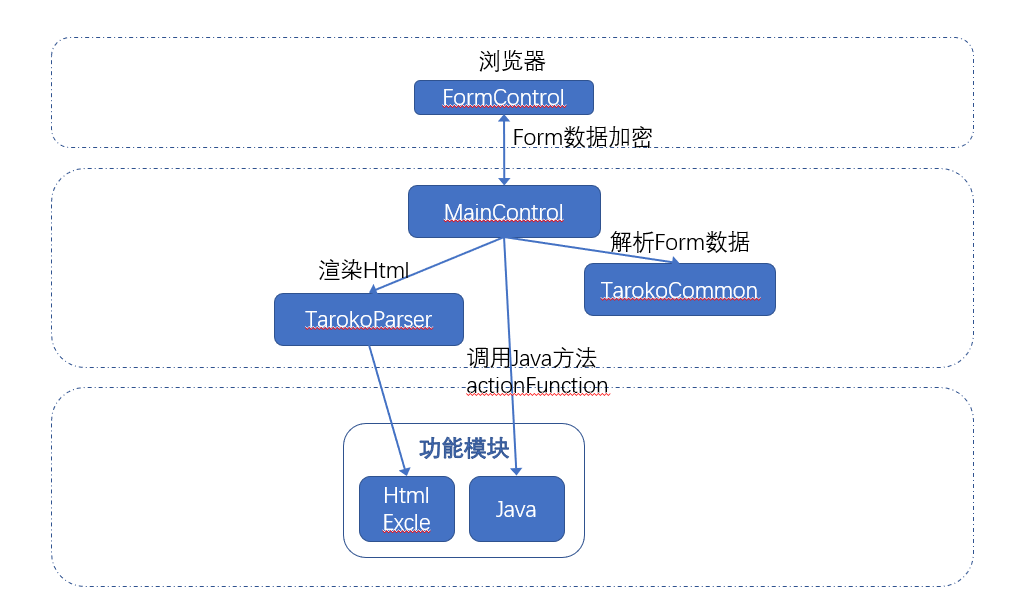
* 类名、变量名、方法名符合Java编码规范，除了TCB自带的无法修改的接口及方法、变量。
* 命名只能使用英文及数字，驼峰格式，方法名及变量名第一个字母消息，类名第一个字母大写。
* 变量名、类名、方法名尽量用有意义的单词定义，尽量避免使用无意义的命名
* 尽量不要使用全局变量
* 除了必要的接口方法尽量使用自定义方法，通过参数传递必要的数据

## 代码结构

## 运行机制

项目架构采用BS结构，Servlet（taroko.com.MainControl）作为主控制类，负责接收页面请求和调用各个模块功能

1. 页面提交form，将form数据加密处理后提交到MainControl
2. MainControl解析提交的数据生成TarokoCommon对象并调用功能模块，功能模块必须实现baseEdit接口的actionFunction方法
3. 调用TarokoParser类读取html页面并渲染（处理页面参数），最后通过Servlet的response输出正常页面
4. TarokoCommon类作为数据载体，包含前端提交的数据和sql执行后查询出的数据，以及数据转换和格式化方法



# 编码指南

## 后端

### 功能模块主类

功能模块主类必须继承ofcapp.baseEdit抽象类

接口方法

// 功能調用入口方法

public void actionFunction(TarokoCommon wr) throws Exception;

// 設置頁面按鈕狀態

public void init\_button();

// 記錄詳情查詢方法，insert或update后會調用，詳情查詢的代碼最好放在這裏

public void dataRead() throws Exception;

// 同dataRead，最終會調用dataRead，非必選

public void querySelect() throws Exception;

// 列表查詢，功能首頁的列表查詢

public void queryRead() throws Exception;

// 同queryRead，最終調用queryRead，非必選

public void queryFunc() throws Exception;

// 保存數據方法，會調用功能維護類的方法完成數據的保存

public void saveFunc() throws Exception;

// 頁面下拉框内容查詢設置方法，非必選

public void dddw\_select()

### 功能模块辅助类

实现busi.funcEdit接口，负责数据库表的新增/修改/删除操作

接口方法：

// 数据校验，未被调用，可以不用

dataCheck()

// 查询数据详情，未被调用，可以不用

dataSelect()

// 查询数据列表，未被调用，可以不用

querySelect()

// 删除数据

dbDelete()

// 插入数据

dbInsert()

// 修改数据

dbUpdate()

### 設置保存後是否重新獲取記錄詳情並展示

this.addRetrieve = true;

this.updateRetrieve = true;

### Code转换为中文

一个字段可能有多个不同的值，前端需要显示为中文，后端需要输出对应的中文字段（也可以在前端转换），后端处理又两种方式

1. Sql，在sql查询时就输出中文字段，db2有类似oracle的decode函数，decode(field, value1, returnvalue1, value2, returnvalue2,….., defaultValue)
2. Java方法，zzStr.decode(ss, ",1,2,3,4,5", ",1.正常,2.逾放,3.催收,4.呆帳,5.結清")
3. Java代码，if/else或者switch
4. 前端方法，参见页面显示转换

## 前端

### AJAX調用

### 頁面顯示轉換

TarokoCommon類convertField(String cvtField)方法解析頁面參數，convertFormat方法轉換參數值

支持的格式

{fieldname } 输出fieldname的值

{#YMD:fieldname.YMD} value為字符串，格式為yyyyMMdd或者yyyyMM，輸出格式2019/06/15或者2019/06

{#YM:fieldname} // 不支持

{#TIME:fieldname.TIME} value為字符串，格式為HHmmss或者HHmm，輸出格式15:01:05或者15:01

{#AMT:fieldname} // 不支持

{#JAVA:Classname.Methodname(fieldname)} 以fieldname的取值為參數調用類的方法，類是newInstance的

{#JAVA2:Classname.Methodname(fieldname)} 以fieldname的取值為參數調用類的方法，方法有兩個參數第一個為Connection，第二個為字段取值，類是newInstance的

{#PLUGIN:fieldname} // PLUGIN 為 同 一網頁為其他網頁共用

{#SQL-TT:fieldname} // "no-support: Html-SQL";

{#DECODE:fieldname;fieldvalue1,returnvalue1;fieldvalue2,returnvalue2......} 將field的值與後面的比較，如果等於fieldvalue1則返回returnvalue1，依次類推，返回最後一個滿足條件的returnvalue

{fieldname.(999)} value為數字字符串，可以有,的科學計數法 String.format("%,14.0f", Double.parseDouble(cvtValue)); 輸出格式 123,764

{fieldname.(999.00)} value為數字字符串，可以有,的科學計數法 String.format("%,14.2f", Double.parseDouble(cvtValue)); 輸出格式 123,764.00

{fieldname.(999.000)} value為數字字符串，可以有,的科學計數法 String.format("%,15.3f", Double.parseDouble(cvtValue)); 輸出格式 123,764.000

{fieldname.(999.0000)} value為數字字符串，可以有,的科學計數法 String.format("%,16.4f", Double.parseDouble(cvtValue)); 輸出格式 123,764.0000

# 参考

环境问题：

git clone https://username:password@host:/path/to/repository

//比如:

git clone https://xiaoming:1234@github.com/schacon/example.git

使用https clone，需要设置ssl校验位false

git config --global http.sslVerify false

taroko-web-ecs工程在clone過程中一直出現以下異常，無法clone成功

error: RPC failed; curl 18 transfer closed with outstanding read data remaining

fatal: the remote end hung up unexpectedly

fatal: early EOF

fatal: index-pack failed

網上解決辦法：

一般是项目codebase太大，试下

git config --add core.compression -1

or

git config --global --add core.compression -1

ssh 方式好像被禁，先用https方式拉代码

git@134.251.81.37: Permission denied (publickey,password).