|  |
| --- |
|  |

Contents

[1 Introduction 3](#_Toc499111740)

[2 Test scope 3](#_Toc499111741)

[3 Code structure 4](#_Toc499111742)

[3.1 /lib 4](#_Toc499111743)

[3.2 /src/test/java/csptest 4](#_Toc499111744)

[3.2.1 cucumberOptions 4](#_Toc499111745)

[3.2.2 stepDefinitions 4](#_Toc499111746)

[3.2.3 FunctionFactory 4](#_Toc499111747)

[3.2.4 tools 5](#_Toc499111748)

[3.3 resources 5](#_Toc499111749)

[4 Common Method Introduction 5](#_Toc499111750)

[4.1 Common Step 5](#_Toc499111751)

[4.1.1 tc 5](#_Toc499111752)

[4.1.2 tmp 5](#_Toc499111753)

[4.1.3 tem 6](#_Toc499111754)

[4.1.4 ihu 6](#_Toc499111755)

[4.2 Common Method 6](#_Toc499111756)

[4.2.1 postgres/cassandra 6](#_Toc499111757)

[4.2.2 http访问接口 6](#_Toc499111758)

[4.2.3 mqtt访问接口 7](#_Toc499111759)

[4.2.4 base64编解码 7](#_Toc499111760)

[4.3 tem-simulator 7](#_Toc499111761)

[4.3.1 主动上报消息 7](#_Toc499111762)

[4.3.2 应答消息 7](#_Toc499111763)

[5 Code and how to do test 7](#_Toc499111764)

[5.1 Code 7](#_Toc499111765)

[5.2 How to do test 8](#_Toc499111766)

[5.2.1 开发工具 8](#_Toc499111767)

[5.2.2 服务器 8](#_Toc499111768)

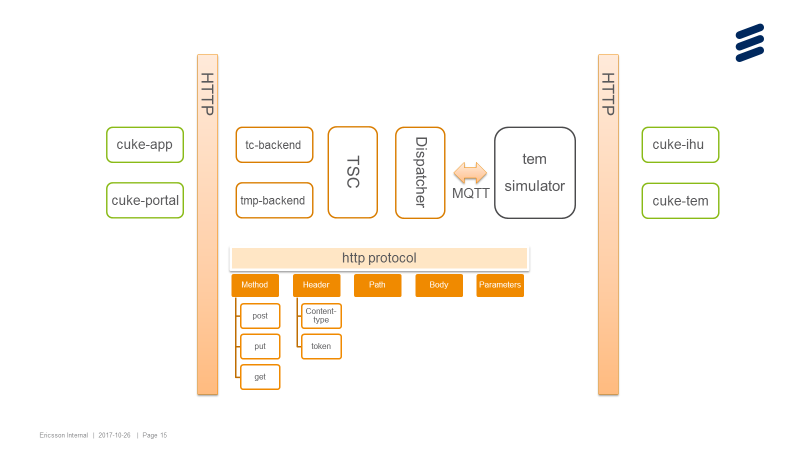
# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Rev | Date | By | Reasons |
| PA1 | 2017.11.15 | Deco Wu | Initial draft |

# Introduction

介绍cucumber的代码框架，以及公共方法代码介绍

# Test scope



# Code structure

## /lib

放在本地的依赖包（注意与其他第三方依赖区别），目前主要是tsc相关实现asn1编解码功能

## /src/test/java/csptest

注意测试代码不在/src/main，而是在/src/test下

### cucumberOptions

cucumber 的main函数功能，定义features的路径与step的路径

**csptest.cucumberOptions.RunCukeTest:**

features = **"test-classes/features/test"**,

glue = **"csptest.stepDefinitions"**,

### stepDefinitions

步骤定义类，按模块划分，如TCStep，TMPStep等

### FunctionFactory

功能实现类，按模块划分，如TC，TMP等

已TC模块为例

TC/api：具体功能的实现方法

TC/messages：下发消息json串的实体类

此外还有公共方法类common，目前实现的方法有

1. 数据库访问
2. Mqtt接口访问
3. http接口访问
4. 配置读取
5. Base64编解码

### tools

一些公共使用的小工具，如全局变量

## resources

可外部编辑的资源，如配置文件，feature文件

# Common Method Introduction

## Common Step

各模块可直接使用的公共步骤，但具体业务需要开发者自己实现

### tc

**app登录**

**RemoteControl\_test.feature:**

**Given** I login the mobile app

@Given(**"^I login the mobile app$"**)  
**public void** login()

**Tc端发起的发送类（POST/PUT）请求**

**RemoteControl\_test.feature:**

**When** I tap "**Cliamte start**" button

@When(**"^I tap \"(.\*?)\" button$"**)  
**public void** AppSendMessage(String service)

**Tc端发起的查询类（GET）请求**

**RemoteControl\_test.feature:**

**Then** I can get latest "**Cliamte start**" status

@Then(**"^I can get latest \"(.\*?)\" status$"**)  
**public void** GetRVS(String service)

### tmp

**Tmp端发起的发送类（POST/PUT）请求**

**Xcall.feature:**

**When** I send "**Ecall ETA**" to vehicle

@When(**"^I send \"(.\*?)\" to vehicle$"**)  
**public void** TmpOperate(String service)

**Tmp端发起的查询类（GET）请求**

**Xcall.feature:**

**When** I select "**ecall**" page

@When(**"^I select \"(.\*?)\" page$"**)  
**public void** TmpRequery(String service)

**Tmp查询类请求结果检查**

**Xcall.feature:**

**Then** I can get "**ecall vehicle info and location**" of the event

@When(**"^I can get \"(.\*?)\" of the event$"**)  
**public void** TmpQueryResult(String service)

### tem

**Tmp端上报消息**

**Xcall.feature:**

**Given** "**Ecall event**" is triggered by vehicle

@When(**"^\"(.\*?)\" is triggered by vehicle$"**)  
**public void** TemUploadMessage(String service)

### ihu

**ihu端发起的查询类（POST/PUT）请求**

**When** I press "**service**" button on ihu

**##未实现**

**ihu端发起的发送类（GET）请求**

**When** I query "**service**" info on ihu

**##未实现**

## Common Method

### postgres/cassandra

数据库的增删改查，可参考测试代码csptest.stepDefinitions.DBStep

### http访问接口

目前只实现tc端的http访问接口，如tmp，ihu的请求串结构一致，可直接复用

**csptest.FunctionFactory.common.http.httpapi:**

**public** Response tc\_send\_message(String method, String token, String path, String param, Object body)

### mqtt访问接口

目前mqtt由tem-simulator实现，见3.3章节

### base64编解码

用于一般的base64编解码

**csptest.FunctionFactory.common.Helper:**

**public** String ccBase64(String src,**boolean** enc)

## tem-simulator

目前mqtt部分由tem-simulator实现，以下是一些介绍

### 主动上报消息

Xcall，行车日志等业务，需要模拟tem主动上报消息，调用tem-simulator接口，通过tem-simulator发送mqtt消息

**csptest.FunctionFactory.common.http.httpapi:**

**public** Response tem\_send\_message(String path, Object body)

### 应答消息

远程控制等业务，需要由tem-simulator返回应答消息

tem-simulator地址

172.21.38.18: /opt/dispatcher/\_deco/tem-simulator-1.3.5.1-20161125

应答消息为groovy脚本，conf/scripts

消息结构请参考以下附件



# Code and how to do test

## Code

<https://github.com/DecoWu/csptest>

## Develop environment

这里介绍eclipse + maven的环境搭建

### maven安装

在官网<http://maven.apache.org/download.cgi下载安装包，我本地下的3.5.0>

环境变量配置（Control Panel\All Control Panel Items\System\Advanced system settings）：在系统变量中新建一个变量，变量名为M2\_HOME，变量值为Maven的安装目录（例如C:\02Software\apache-maven-3.5.0）。点击确定，接着在系统变量中找到一个名为Path的变量，在变量值的末尾加上%M2\_HOME%\bin; 参考：[Maven学习链接](http://www.cnblogs.com/dcba1112/archive/2011/05/01/2033805.html)

### eclipse安装maven插件

从以下链接下载eclipse，链接: https://pan.baidu.com/s/1boEbobd 密码: e5pg

1. eclipse的菜单栏选择Help->Install New Software
2. 点击Work with后面的add,输入Name，弹出Location输入安装地 址:http://download.eclipse.org/technology/m2e/release下一步会自动完成安装
3. 安装好后重启eclipse，可创建Maven工程
4. 本地会生成.m2的文件夹（本地仓库路径），默认情况下，该文件夹下放置了Maven本地仓库.m2/repository。一般windows用户该路径会安装在C盘Maven工程中存在一个pox.xml配置文件，在这xml文件中添加Maven所依赖的Jar的名称，或添加<dependency></dependency>节点，会自动下载或关联我们需要的包，可参考代码里面的pom.xml

## How to do test

注意，在开发工具调试和服务器上测试，需要修改部分代码及配置

### 开发工具

在IDEA/eclipse等开发工具，可直接test，但需要注意：

pom.xml:需要屏蔽<build></build>

RunCukeTest: features = "src/test/resources/features/test"

csptest.FunctionFactory.common.Helper:配置文件的路径为path = "src/test/resources/Configs/tc.properties";

### 服务器

pom.xml:放开<build></build>

RunCukeTest: features = "test-classes/features/test"

csptest.FunctionFactory.common.Helper:配置文件的路径为path = " test-classes/Configs/tc.properties";

执行命令java -cp ./libs/\*:./test-classes/:. org.junit.runner.JUnitCore csptest.cucumberOptions.RunCukeTest