**点评CAT之java客户端集成**

# 客户端集成

CAT推出多种语言的客户端，基本覆盖了主流开发语言。

CAT目前支持：：Java、C、Python、node、Go等语言的接入。

# Java客户端

## 1、创建Maven项目

在pom.xml中添加cat客户端依赖

```

<dependency>

<groupId>com.dianping.cat</groupId>

<artifactId>cat-client</artifactId>

<version>3.0.0</version>

</dependency>

```

## 2、初始化

### （1）、创建 /data/appdatas/cat 目录

在你创建的项目的根目录下创建且确保你具有这个目录的读写权限。

在这里我的项目是创建在E盘下的,因此这些操作我需要在E盘下完成。

### （2）、创建 /data/applogs/cat 目录 (可选)

这个目录是用于存放运行时日志的，这将会对调试提供很大帮助，同样需要读写权限。

### （3）、创建 /data/appdatas/cat/client.xml，内容如下

```

<?xml version="1.0" encoding="utf-8"?>

<config xmlns:xsi="http://www.w3.org/2001/XMLSchema" xsi:noNamespaceSchemaLocation="config.xsd">

<servers>

<server ip="<cat server ip address>" port="2280" http-port="8080" />

</servers>

</config>

```

不要忘记把 <cat server IP address> 替换成你自己的服务器地址哦！

### （4）、创建 src/main/resources/META-INF/app.properties 文件

然后你需要在你的项目中创建 src/main/resources/META-INF/app.properties 文件, 并添加如下内容:

```

app.name={appkey}

```

appkey 只能包含英文字母 (a-z, A-Z)、数字 (0-9)、下划线 (\_) 和中划线 (-)

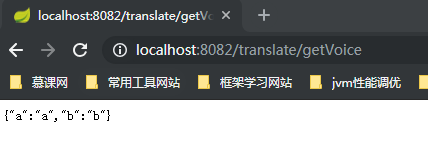


## 3、快速入门（代码演示）

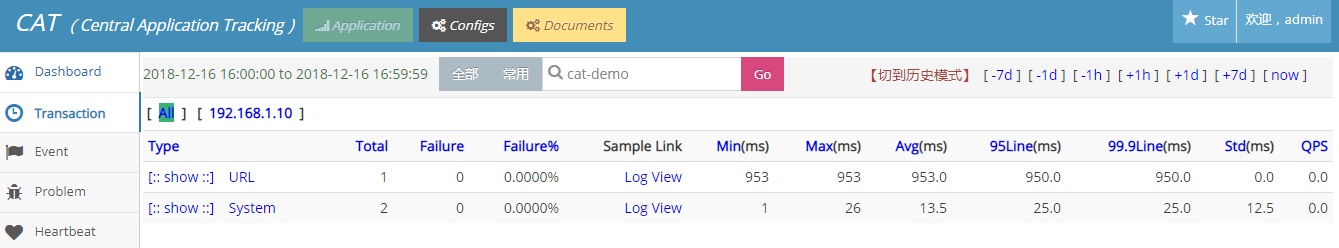
### （1）、案例代码



### （2）、访问接口



### （3）、观察cat页面



OK 至此CAT的java客户端完成

## 4、API List

### （1）、Transaction

为了避免忘记关闭 Transaction, 建议在 finally 代码块中执行 complete。

```

Transaction t = Cat.newTransaction("URL", "pageName");

try {

yourBusiness();

t.setStatus(Transaction.SUCCESS);

} catch (Exception e) {

t.setStatus(e);

Cat.logError(e);

} finally {

t.complete();

}

```

CAT提供了一系列API来对Transaction进行修改

**A: addData**

示例代码:

```

t.addData("testCase接口业务代码执行完成","值");

```

效果图



**B: setStatus**

示例代码

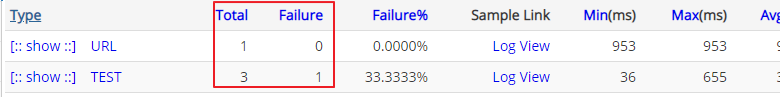
```

t.setStatus(Message.SUCCESS); // 表示成功

t.setStatus(e); // 表示失败

```

效果图





**C: complete**

t.complete(); // 完成Transaction

注意：不要忘记完成 transaction！否则你会得到一个毁坏的消息树以及内存泄漏！

### （2）、Event

**A: Cat.logEvent 记录一个事件**

# Log a event with success status and empty data.

Cat.logEvent("URL.Server", "serverIp");

# Log an event with given status and given data.

Cat.logEvent("URL.Server", "serverIp", Event.SUCCESS, "ip=${serverIp}");

**B: Cat.logErroe**

记录一个带有错误堆栈信息的 Error。

Error 是一种特殊的事件，它的 type 取决于传入的 Throwable e.

如果 e 是一个 Error, type 会被设置为 Error。

如果 e 是一个 RuntimeException, type 会被设置为 RuntimeException。

其他情况下，type 会被设置为 Exception。

同时错误堆栈信息会被收集并写入 data 属性中。

```

try {

1 / 0;

} catch (Throwable e) {

Cat.logError(e);

}

```