**IUAP平台使用指南**

**业务日志组件**

用友网络科技股份有限公司

二○一八年五月

**此文档为保密文档，未经用友软件股份有限公司书面同意，不得向任何单位或个人提供、转让本文档中的任何内容，用友软件股份有限公司将保留对泄漏文档内容的起诉权利。**

**IUAP平台使用指南**

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [**√**]正在修改  [] 正式发布 | 文件标识： |  |
| 当前版本： |  |
| 作者： |  |
| 完成日期： |  |

版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 作者 | 参与者 | 起止日期 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

审核记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 姓名 | 职位 | 审阅日期 | 备注 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

[第一章 概述 1](#_Toc515617135)

[1.1 业务场景 1](#_Toc515617136)

[1.2 开发方案 1](#_Toc515617137)

[第二章 示例配置 2](#_Toc515617138)

[2.1 开发环境配置 2](#_Toc515617139)

[2.2 示例代码配置 2](#_Toc515617140)

[第三章 代码开发示例 3](#_Toc515617141)

[3.1 设置busilog-systemConfig.properties 3](#_Toc515617142)

[3.2 业务Bean注入Spring容器，并符合拦截方法 3](#_Toc515617143)

[3.3 为业务方法WorkorderService.save添加业务日志annotation——@BusiLogConfig("workorder\_info\_save") 3](#_Toc515617144)

[3.4 配置业务方法与日志模板映射 3](#_Toc515617145)

# 概述

## 业务场景

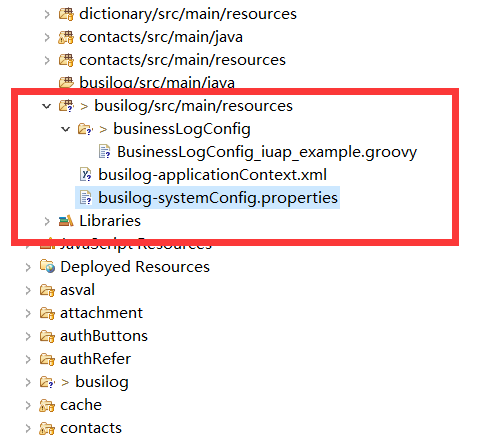
## 开发方案

iuap-busilog组件提供采用Spring AOP拦截器机制，实现将记录日志与具体的业务方法分离开。同时，日志模板采用groovy脚本配置方式，实现拦截业务方法参数等，日志的输出用户可自行定义，也可使用已经提供的方式，输出到控制台或者输出到数据库中。

示例应用将从以上几个方面来对使用业务日志服务进行介绍。

示例应用为busilog，主要代码工作为“配置”，代码结构如下：

* 代码结构：



# 示例配置

## 开发环境配置

1. 安装部署iuap-saas-busilog-service

首先完成iuap-saas-busilog-service应用工程的部署，此项工作已在IUAP一键式部署中实现，在此不再做详述。

1. 在工程中添加iuap- busilog组件的Maven依赖，具体如下：

<dependency>

<groupId>com.yonyou.iuap</groupId>

<artifactId>iuap-busilog</artifactId>

<version>${iuap.modules.version}</version>

</dependency>

其中，${iuap.modules.version} 为平台在maven私服上发布的组件的version

## 示例代码配置

1. 从示例工程中复制配置文件

busilog-applicationContext.xml

busilog-systemConfig.properties

具体修改方法参见**第三章 代码开发示例**

# 代码开发示例

## 设置busilog-systemConfig.properties

1. 打开业务日志开关

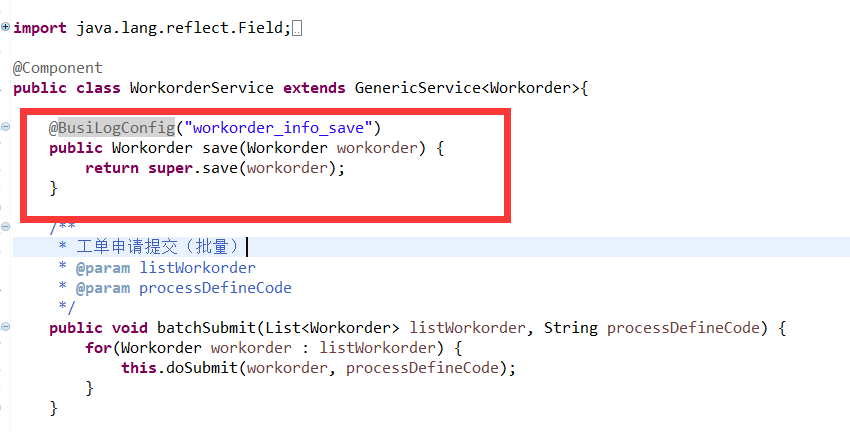
yonyou.businesslog.enable=true

1. 指定业务日志的拦截方法

pointcut=execution(public \* \*..service.\*.\*(..))

## 业务Bean注入Spring容器，并符合拦截方法

## 为业务方法WorkorderService.save添加业务日志annotation——@BusiLogConfig("workorder\_info\_save")



## 配置业务方法与日志模板映射

映射文件采用groovy文件。目前支持两种配置方式：

* **单文件配置**

在类路径下加入'BusinessLogConfig.groovy'

* **多文件配置**

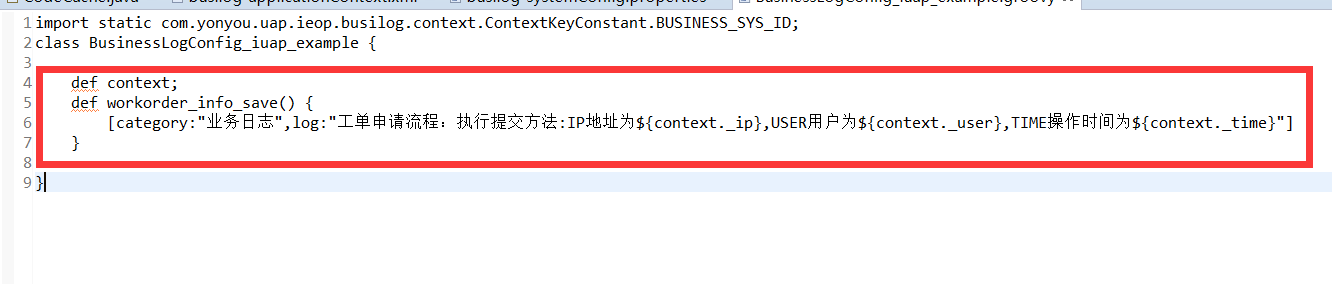
1. 在类路径中加入'businessLogConfig'文件夹
2. 在该文件夹中加入日志映射配置文件

BusinessLogConfig\_iuap\_example.groovy

其中，文件名任意，但必须符合Groovy类文件命名规范

注：多文件配置方式与单文件配置方式不兼容。在此业务日志组件中，单文件配置方式优先。businessLogConfig'文件夹中的所有以'.groovy'结尾的文件都将被作为日志配置文件。

* **groovy文件内容示例**



1. 声明context

业务日志组件将把使用了 @BusiLogConfig 的业务方法返回值、参数、当前用户等信息写入上下文context中，以便于记录业务日志使用。其具体提供的信息内容见下表：

|  |  |
| --- | --- |
| key | value |
| \_methodReturn | 业务方法返回值 |
| \_param | 业务方法的参数, \_param0代表第一个参数 \_param1代表第二个参数，依此类推 |
| \_executeError | 业务方法执行失败的异常信息 |
| \_businessMethod | 业务方法 |
| \_user | 业务方法操作人 |
| \_time | 业务方法操作时间 |
| \_ip | ip地址 |

1. 声明业务日志映射

“def workorder\_info\_save(){…}” 中方法名应该与annotation的声明保持一致，均为“workorder\_info\_save”。