文件编号：

版 本：V0.1

修订状态：新建

**J2EE开发日志规范**

**编制：**

**审核：**

**批准：**

版 本 历 史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本 | 状态 | 作者 | 参与者 | 起止日期 | 描述 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

状态：新建，增加，修改，删除

目 录

[1. 引言 4](#_Toc519008517)

[1.1. 目的 4](#_Toc519008518)

[1.2. 范围 4](#_Toc519008519)

[2. 日志规范 4](#_Toc519008520)

[2.1. 业务审计日志 4](#_Toc519008521)

[2.1.1. 内容 4](#_Toc519008522)

[2.1.2. 存储 5](#_Toc519008523)

[2.1.3. 范围 6](#_Toc519008524)

[2.1.4. 语言 7](#_Toc519008525)

[2.2. 系统运行日志 7](#_Toc519008526)

[2.2.1. 存储 7](#_Toc519008527)

[2.2.2. 级别 7](#_Toc519008528)

[2.2.3. 规范 8](#_Toc519008529)

[2.2.4. 语言 9](#_Toc519008530)

[2.2.5. 配置 9](#_Toc519008531)

# 引言

## 目的

本文档是使用JAVA开发应用系统时的日志记录规范，便于开发人员写出统一的业务审计日志和系统运行日志，满足系统运行监控和业务安全审计的需要。

## 范围

本文档的适用范围是所有类型项目，包括合同项目和产品研发。

# 日志规范

## 系统访问日志

系统访问日志用于记录用户成功或失败地登录、主动退出、被动退出、超时退出等活动。

### 内容

系统访问日志要记录的内容包括：

* **操作时间**
* **客户端IP**
* **操作员名**：操作用户登录名
* **操作**：登录、主动登出、被动登出（被踢掉）、超时登出
* **操作结果**：成功/失败
* **失败原因**：登录/登出失败原因。
* **在线时长**：单位秒

### 存储

* **存储方式：**系统访问日志存储在数据库中，表结构参考示例如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | 系统访问日志表SYS\_ACCESS\_LOG | | | |
| 字段名 | 描述 | 数据类型（精度范围） | 空/非空 | 约束条件 |
| ID | 主键ID | bigint | not null | PK |
| OPERATE\_TIME | 操作时间 | datetime | not null |  |
| CLIENT\_IP | 客户端IP | varchar(50) |  |  |
| OPERATE\_ACCOUNT | 操作员名 | nvarchar(100) |  |  |
| ACTION | 操作 | varchar(50) |  | login/logout/kickout/timeout |
| OPERATE\_RESULT | 操作结果 | tinyint(1) |  | 1:成功 0：失败 |
| REASON\_CODE | 失败原因代码 | varchar(50) |  |  |
| ONLINE\_TIME | 在线时长(秒) | int |  |  |
| 补充说明 | 注：   1. 操作（ACTION）值为kickout是指用户被踢掉的情况； 2. 在线时长（ONLINE\_TIME）只针对logout/kickout/timeout的情况 | | | |

### 范围

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **事件** | **失败原因** | **在线时长** | **说明** |
| 1 | login | 登录失败时，记录失败原因 | （空） | 登录 |
| 2 | logout | （空） | 记录在线时长 | 主动登出 |
| 3 | kickout | （空） | 记录在线时长 | 被动登出（被踢掉了） |
| 4 | timeout | （空） | 记录在线时长 | 超时登出 |

### 语言

* 系统访问日志不允许使用中文，且要求做规范化处理，便于展现给用户时的国际化；

## 业务审计日志

### 内容

业务审计日志的内容包括基本信息和详细信息两部分。涉及业务数据变化的操作，如业务数据增加、修改、删除等，建议这两部分信息都记录，并对这两部分信息进行关联，以便查询和审计；不涉及业务数据变化的操作，如登录、登出、业务查询等，只需要记录基本信息。

注：记录业务审计日志的详细信息不是强制要求，各项目可根据项目要求和性能综合进行考虑。

#### 基本信息

* **操作时间**
* **客户端IP**
* **操作员ID**：登录用户的ID
* **操作员名**：操作用户登录名
* **操作对象**：被操作的资源
* **操作**：如增加、删除、修改、查询等
* **操作结果**：成功/失败
* **操作内容**：不记录详细操作内容，只记录关键信息。如新建查验任务时，可只需记录新任务的ID（主键）、任务号等主要信息。

操作结果是否需要也由各项目根据项目要求来决定。

#### 详细信息

* **数据项**：业务数据字段名
* **操作前内容**：操作前值。增加时，操作前内容为空
* **操作后内容**：操作后值。删除时，操作后内容为空

增加业务数据时，记录所有新增非空字段的值；修改业务数据时，只需要记录内容有变化字段的值，包括修改前和修改后的值；删除业务数据时，只需要记录被删除业务数据的主键和关键字段的值（如删除查验任务时，可只记录被删除查验任务的ID、任务号等主要字段信息）。

### 存储

* **存储方式：**业务审计日志要求存储在数据库中。业务审计日志表参考示例如下：

1. 业务审计日志基本信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | 基本信息表SYS\_AUDIT\_LOG | | | |
| 字段名 | 描述 | 数据类型（精度范围） | 空/非空 | 约束条件 |
| ID | 主键ID | bigint | not null | PK |
| OPERATE\_TIME | 操作时间 | datetime | not null |  |
| CLIENT\_IP | 客户端IP | varchar(50) |  |  |
| OPERATOR\_ID | 操作员ID | varchar(50) |  |  |
| OPERATE\_ACCOUNT | 操作员名 | nvarchar(100) |  |  |
| OPERATE\_OBJECT | 操作对象 | varchar(100) |  |  |
| ACTION | 操作 | varchar(50) |  |  |
| OPERATE\_RESULT | 操作结果 | tinyint(1) |  | 1:成功 0：失败 |
| OPERATE\_CONTENT | 操作内容 | nvarchar(2000) |  |  |
| 补充说明 |  | | | |

1. 业务审计日志详细信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表名 | 基本信息表SYS\_AUDIT\_LOG\_DETAIL | | | |
| 字段名 | 描述 | 数据类型（精度范围） | 空/非空 | 约束条件 |
| ID | 主键ID | bigint | not null | PK |
| AUDIT\_LOG\_ID | 基本审计日志ID | bigint | not null |  |
| FIELD\_NAME | 数据项 | datetime | not null |  |
| VALUE\_BEFORE | 操作前内容 | nvarchar(2000) |  |  |
| VALUE\_AFTER | 操作后内容 | nvarchar(2000) |  |  |
| 补充说明 | 1. 数据项建议采用“实体名.属性名”的形式存储，便于展示时的国际化 | | | |

* **存储期限：**要求可根据用户需要（或业务需要）设置业务审计日志的存储期限。超期的业务审计日志数据，由系统自动删除，基本信息和详细信息一起删。自动删除前是否需要归档，也可由用户进行设置。**一定不能为任何用户提供手动删除或修改业务审计日志的功能。**

建议引导用户设置合理的业务审计日志的存储期限，最好不要超过六个月。如用户要求业务审计日志的存储期限较长，会导致业务审计日志表中数据特别多，影响性能，可采用表分区的方式来解决，如每个月建一个分区。

归档时，建议按生成日志的先后顺序，采用纯文本格式导出业务审计日志数据，并进行压缩，以便解压后可直接查看和搜索。归档日志的文件名使用XXX-YYYYMMDD-YYYYMMDD.log.zip和XXX-YYYYMMDD-YYYYMMDD.detail.log.zip格式，其中：XXX为系统名，第一个YYYYMMDD为日志的起始日期，第二个YYYYMMDD为日志的截止日期。

### 范围

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **事件** | **是否记录详细信息** | **说明** |
| 1 | 增、删、改 | 是 | 用户在系统中执行的所有增、删、改操作，都需要记录业务审计日志，包括基本信息和详细信息。 |
| 2 | 查询 | 否 | 1. 用户查询操作也建议记录业务审计日志，不需要记录详细信息，但建议在操作内容中记录查询条件和查询结果的行数（含分页相关信息）。翻页操作按查询操作一样记录业务审计日志。 2. 进入某个查询页面后，系统根据默认条件自动进行的查询操作也建议记录业务查询日志。 3. 系统自动进行的查询操作，如查询并加载字典项、用户执行某个业务操作需要执行的查询等不记录业务审计日志。 |
| 3 | 其它业务操作 | 是 | 通过系统界面，用户执行的所有操作，如造成业务数据发生变化的都需要记录业务审计日志，包括基本信息和详细信息。 |
| 4 | 后台服务 | 否 | 1. 后台服务，如定时任务，如造成系统业务数据发生变化，也需要记录业务审计日志。根据后台服务对业务数据的处理情况，有的需要按批记录业务审计日志（如归档时，使用一条业务审计日志记录归档了从哪一天到哪一天的日志），有点也需要每条业务数据单独记录一条业务审计日志。 2. 记录业务审计日志时，客户端IP和操作员ID为空，操作员名固定为daemon。 |

### 语言

* 业务审计日志中不允许使用中文；
* 基本信息中的操作对象（OPERATE\_OBJECT）、操作（ACTION），详细信息中的数据项（FIELD\_NAME）要求规范化，便于展现给用户时可做国际化处理；

## 系统运行日志

### 存储

* **存储方式：**系统运行日志以文本文件的方式存储在服务器的本地存储中。
* **存储期限：**要求可根据用户需要设置系统运行日志的存储期限。超期的系统运行日志数据由系统自动删除。
* **文件大小：**建议系统运行日志文件按天进行命名，每天一个或多个日志文件，每个系统运行日志文件不要超过20M。文件名可使用${APP\_NAME}-%d{yyyy-MM-dd}-%i.log格式命名，其中：${APP\_NAME}为系统名称；%d{yyyy-MM-dd}为记录日志的日期；%i为每天日志文件名的顺序号。

### 级别

建议记录ERROR、WARN、INFO、DEBUG四个级别的系统运行日志，不要记录FATAL、TRACE等级别的日志。

1. **ERROR**
   * 只记录系统逻辑出错、异常或者重要的错误信息。
2. **WARN**
   * 用户输入参数错误。前后端分离时，供本系统使用的接口检查出接口参数错误，建议记录WARN日志。
3. **INFO**
   * 对外提供的接口被调用时必须记录INFO日志，含接收到的参数信息和返回结果。
   * 调用其它系统（或子系统）的接口时必须记录INFO日志，含参数信息和返回结果。
   * 后台定时任务启动、停止和开始执行任务、结束执行任务时都必须记录INFO日志，含执行的任务信息。
   * 有可能执行时间较长或有性能压力的操作，建议记录INFO日志。
4. **DEBUG**
   * 建议不要记录TRACE日志。
   * 如果一个操作需要分多步执行，建议每步操作都记录Debug日志。

### 规范

1. 使用简单日志门面（SLF4J）。不要直接使用log4j、logback、log4j2等日志框架的API，而应依赖使用日志框架SLF4J中的API。建议使用SLF4J + logback或SLF4J+log4j2。不要再使用log4j，因为logback是SLF4J的原生实现，且logback与log4j的开发者为同一人，logback有更多的优势。

|  |
| --- |
| **import** org.slf4j.Logger;  **import** org.slf4j.LoggerFactory;  **private static** **final** Logger logger = LoggerFactory.*getLogger*(Abc.**class**); |

1. 异常信息应该包括两类信息：案发现场信息和异常堆栈信息。如果希望将异常堆栈信息完整写入系统运行日志，使用SLF4J Logger的**public void**error(String msg, Throwable t);方法。

|  |
| --- |
| logger.error("Get status code failed.", ex); |

1. 不要重复记录系统运行日志。建议规定错误日志在某一层统一记，避免每层都记日志造成重复记录。不允许记录日志后又抛出异常，因为这样会多次记录日志。
2. 避免重复打印日志，浪费磁盘空间，在日志配置文件中务必设置additivity=false（详见示例配置中的说明）

|  |
| --- |
| <logger name="org.springframework" level="error" additivity="false"></logger> |

1. 不允许使用system.out.println输出系统运行日志。**不允许**使用e.printStackTrace输出系统运行日志。
2. 不要记录没有意义的日志。必须将要记录的内容表述清楚，不能只记录如：insert failed.这样的日志。
3. 使用isXXXEnabled()。不使用isXXXEnabled的情况下，尽量不要使用+号拼接日志，使用占位符（SLF4J支持站位符）；

|  |
| --- |
| **if** (logger.isDebugEnabled()){  logger.debug("Processing task with id: " + id);  }  或  logger.debug("Processing task with id: {}", id); |

### 语言

* 系统运行日志统一采用英文记录，不允许夹杂中文和拼音。

### 配置

* logback的示例配置如下：



