日志系统设计方案

# 1、 背景

目前我们项目部署的方式为群集模式，错误日志只会输出到其中的一台机器中，导致我们收集错误日志困难。同时为了支持以后大量的访问日志收集，我们需要设计一套新的日志系统来支撑未来不断增加的日志写入需求。

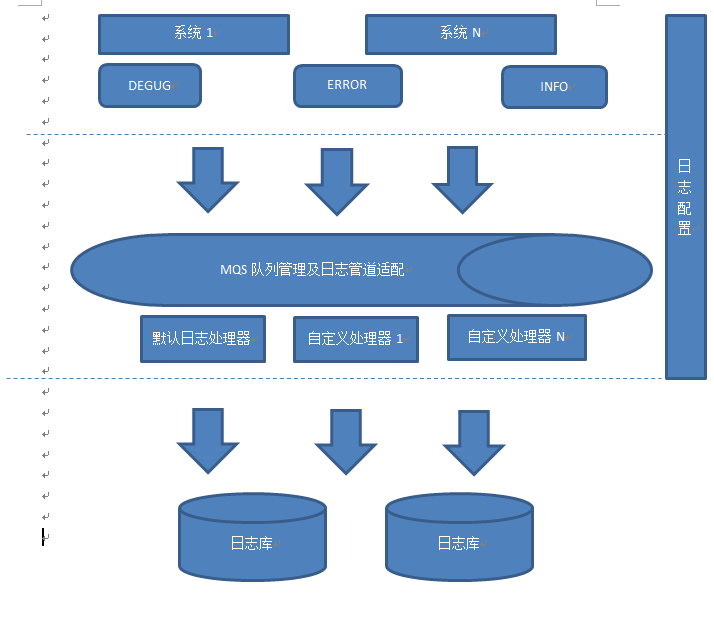
# 2、业务需求

* 日志支持按项目、模块、业务分类收集
* 日志支持实时启停某分类的日志收集功能
* 精简日志级别，只支持DEGUG、ERROR、INFO
* 支持日志标签功能，提供纵向划分日志。
* 提供统一的界面查询日志。
* 支持统一方式扩展日志写入与处理
* 支持高并发动态分流处理

# 3、日志框架概述

日志系统采用阿里云的MQS消息队列作为日志传输载体，采用分布式架构，将各个系统的日志写入需求保存到指定的位置。日志框架分为日志分发器、日志处理服务、日志运维平台三个部分组成，日志分发器是日志写入的入口，各项目只需要调用日志写入接口就可以把日志写入各个日志数据库中。日志分发器支持多通道写入，也支持写入通道根据业务特性动态调整。日志处理服务的消息处理为并发管道模式，每个管道会独立处理一批日志消息，同时日志处理管道支持定制化，方便扩展各种日志的写入及处理。各项目中只需要按照业务写入日志（特殊日志类型需要配合专门的处理管道处理），然后在日志运维后台按分类、按标签、按级别查询即可。

## 3.1、系统架构图



# 4、日志分发器

日志分发器是根据平均负载法将写入请求分发到一组默认的通道上，也可以根据配置将对应业务的日志发送到匹配的通道上。同时监听配置变化，根据配置变化内容调整写入策略。日志分发器采用异步发送模式，考虑到Web项目的IIS应用程序池会定时回收，导致后台线程会被强制关闭，后台线程中处理的内存数据也会丢失，为了避免数据丢失，利用本地MSMQ队列作为缓冲组件，将写入请求缓冲到本机的MSMQ队列中，再由日志分发器将日志消息从MSMQ队列中取出发送到对应的通道上去。（利用MSMQ为缓存组件会降低一部分处理速度，如果不考虑内存数据丢失，可以使用BlockingCollection阻塞式队列，支持并发入队和出对，性能高）

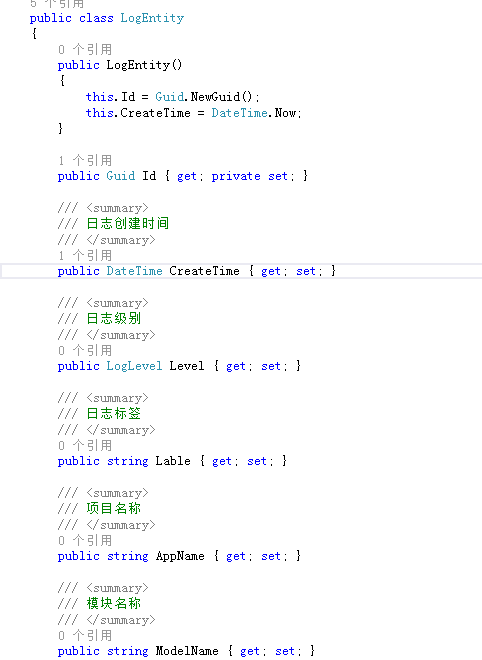
## 4.1、日志写入接口

日志写入接口支持DEBUG、ERROR、INFO，同时支持按项目、模块、业务、标签等维度写入日志。配合日志配置内容将符合写入要求的日志发送到日志处理服务器处理。接口定义如下：



## 4.2、日志实体

日志系统定义了基本的日志实体。基本属性为日志级别、日志标签等，框架根据日志实体的相关属性来处理，例如启停控制、通道选择等，自定义日志实体需要从基本日志实体派生，实体定义如下：



## 4.3、配置监听

实时监听日志运维后台的配置信息，根据配置驱动控制日志的写入，包括启停日志及通道分配等。（实时监听方案有两个方向，一个是Memcached，一个是传统XML文件）。

## 4.4、错误处理

如果日志发送失败会重试，达到一定重试次数仍然失败会输出到本地日志中。同时处理错误信息，保证业务线程稳定。

# 5、日志处理服务

日志处理服务包括消息监听器、消息处理管道两部分。消息监听器负责监控MQS队列消息，每一个队列为一个日志处理通道，日志处理通道个数支持配置。消息处理管道负责消息在各个管道中的协调处理，包括消息和管道的匹配。

## 5.1、消息监听器

日志处理服务的一个通道配备一个消息监听器，负责日志的抓取，一个消息监听器包含一个或者多个工作线程。可以根据通道的情况实时调整工作线程，以增加处理速度。

## 5.2、消息处理管道

消息监听器抓取到消息后会依次调用管道组，消息只能进入匹配成功的管道内处理。默认情况下系统提供默认消息处理管道，该管道会将日志消息存入指定的数据库中。

## 5.3、自定义处理管道

日志框架提供消息处理管道的虚基类，支持配置方式加载自定义管道，自定义处理管道需要实现消息匹配逻辑，批处理消息的个数，消息处理是否成功，该消息是否交给其他管道处理。消息监听器自动负责消息的抓取和删除。

# 6、日志运维平台

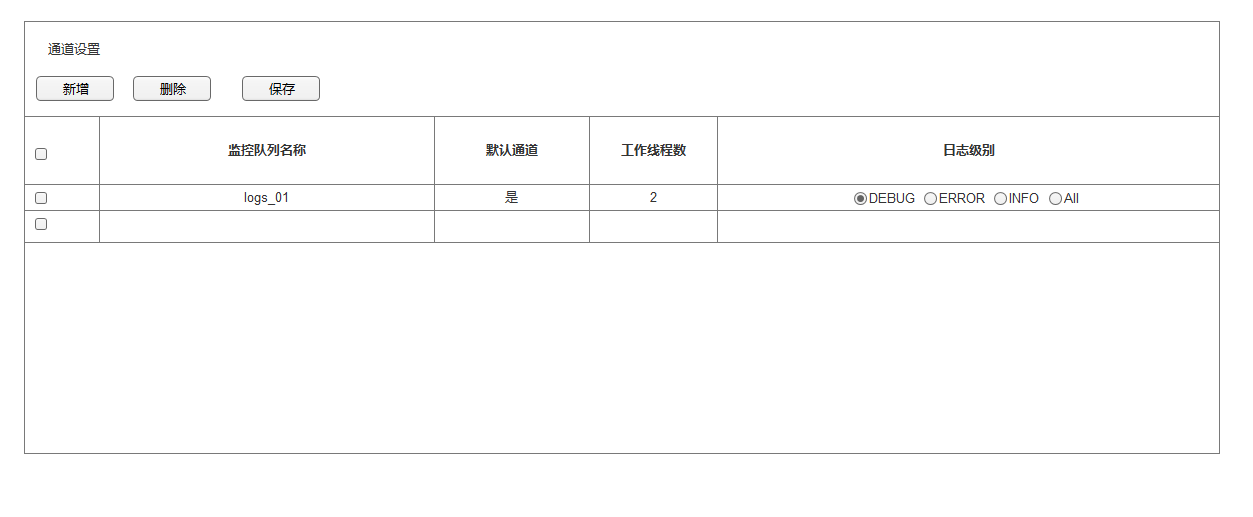
日志运维平台包含日志配置和日志查询两个模块组成。

## 6.1、日志配置

日志配置包括日志基础配置、日志写入配置、日志管道配置组成。日志分发器会定时根据日志配置内容作出反应。

### 6.1.1、日志基础配置

日志基础配置可以设置日志通道个数。包括监控队列名称、消息监听器线程数等。原型界面如下：



### 6.1.2、日志写入配置

日志写入配置可以控制日志分发器的写入规则，日志分发器接受到一个日志写入请求后，会根据日志的项目名称、模块名称、业务名称等相关属性来确定日志是否写入以及写入的通道。原型界面如下：



### 6.1.3、日志管道配置

日志管道配置是整个日志框架的扩展点，允许扩展各种日志消息的处理方式。以及派生日志对象的处理等。原型界面如下：



## 6.2、日志查询

日志查询包括默认日志查询及扩展日志查询。

### 6.2.1、默认日志查询

默认日志查询提供日志框架标准的日志记录的查询。可以通过日志级别查询，也可以查询某个时间段的日志。原型界面如下：



### 6.2.2、扩展日志查询

扩展日志是扩展日志管道产生的日志查询。目前只有WebApi项目的请求日志查询，未来估计会有很多查询，都会归类到该模块下。

#### 6.2.2.1 WebApi请求日志查询

WebApi请求日志查询提供方法执行状态查询，用户名查询，请求时间段查询等。原型界面如下：



# 7、数据库设计

数据库设计包含日志配置和日志记录表。

## 7.1、基础日志配置

表名：log\_config\_channel

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 是否为空 | 说明 |
| channel\_id | int | 否 | 数据标识 |
| channel\_name | nvarchar(50) | 否 | 通道名称 |
| channel\_transports\_url | varchar(200) | 否 | 日志传输地址 |
| is\_default | bit | 否 | 是否为默认通道 |
| worker\_count | smallint | 否 | 工作线程数 |
| level | smallint | 否 | 日志级别 |
| create\_time | datetime | 否 | 创建时间 |
| update\_time | datetime | 否 | 更新时间 |
| desc | nvarchar(100) | 是 | 备注 |

## 7.2、日志分发器基础配置

表名：log\_config\_dispatch\_base

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 是否为空 | 说明 |
| dispatch\_base\_id | int | 否 | 数据标识 |
| default\_log\_level | smallint | 否 | 默认日志级别 |
| create\_time | datetime | 否 | 创建时间 |
| update\_time | datetime | 否 | 更新时间 |
| desc | nvarchar(100) | 是 | 备注 |

## 7.3、分发器规则配置

表名：log\_config\_dispatch\_rule

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 是否为空 | 说明 |
| dispatch\_rule\_id | int | 否 | 数据标识 |
| channel\_id | int | 否 | 通道id |
| app\_name | nvarchar(50) | 否 | 项目名称 |
| model\_name | nvarchar(50) | 否 | 模块名称 |
| business\_name | nvarchar(50) | 否 | 业务名称 |
| lable | nvarchar(50) | 否 | 标签 |
| log\_level | smallint | 否 | 日志级别 |
| create\_time | datetime | 否 | 创建时间 |
| update\_time | datetime | 否 | 更新时间 |
| desc | nvarchar(100) | 是 | 备注 |

## 7.4、日志服务管道配置

表名：log\_config\_pipeline

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 是否为空 | 说明 |
| pipeline\_id | int | 否 | 数据标识 |
| pipeline\_name | nvarchar(50) | 否 | 管道名称 |
| order\_id | int | 否 | 排序 |
| pipeline\_type | varchar(200) | 否 | 管道注册类型 |
| is\_enable | bit | 否 | 是否启用 |
| create\_time | datetime | 否 | 创建时间 |
| update\_time | datetime | 否 | 更新时间 |
| desc | nvarchar(100) | 是 | 备注 |

## 7.5、公共日志记录

表名：log\_common

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 是否为空 | 说明 |
| log\_id | uniqueidentifier | 否 | 数据标识 |
| app\_name | nvarchar(50) | 否 | 项目名称 |
| model\_name | nvarchar(50) | 否 | 模块名称 |
| business\_name | nvarchar(50) | 否 | 业务名称 |
| lable | nvarchar(50) | 否 | 标签 |
| level | smallint | 否 | 日志级别 |
| code | int | 否 | 日志码 |
| message | nvarchar(50) | 否 | 日志标题 |
| content | nvarchar(2000) | 否 | 日志内容 |
| create\_time | datetime | 否 | 创建时间 |
| update\_time | datetime | 否 | 更新时间 |
| desc | nvarchar(100) | 是 | 备注 |

# 8、开发计划

|  |  |  |
| --- | --- | --- |
| 功能点 | 时间 | 说明 |
| 日志配置数据表设计 | 32h | 日志配置相关表、默认日志记录表、WebApi请求日志表 |
| 日志处理服务 | 80h | 日志消息处理相关服务，建立在Windows服务上的稳定程序 |
| 日志分发器 | 40h | 开发组件，应用到Webapi项目 |
| 日志运维平台 | 80h | 在运维平台增加相关功能 |