### Purpose of this article

This article is targeted at the usage explanation, format definition, suggested implementation of Interface Log in applications.

Interface Log itself is very important for each application, with it we can check the request and response for each interface, the status and consumed time cost for each one interface, etc.

With interface Log, we can also collect some statistics data based on it, which can be used as audition or performance checking.

### Oriented Audience

Java Developers

QA

### When to take the Interface Log

As named, the Interface Log is mainly used to log the information related to interactions between different components.

That is to say, when different components want to exchange information based on some kinds of protocols, we might need to record the interface related data.

1. If application is exchanging information via http url/webservice/FTP protocol, etc, the interface log must be recorded;
2. If application itself is money-sensitive and business complex, the interaction between service layer and database should be recorded;
3. If application itself is not money-sensitive, the interaction between service layer and database related to core business scenarios should be recorded

Additionally, the interaction between service layer and database related to core business scenarios could be recorded but not mandatory.

### Specification

#### Interface Log Category

Only one category of interface log is present now

#### Log Level

INFO, DEBUG

#### Log Size and Rotation

The size of each interface log and rotation strategy should be configurable;

Suggested default:

1. Log Size 15M for each file maximum
2. Rotation  rotate every day
3. Keep at most log files within latest 30 days

#### Log File Name

Log path: logs/${Application Name}/

Log file:    interface\_{yyyy-MM-dd}.log.[{number}]

{Application Name} is the abbreviation of each application. Below is the suggested naming convention, for applications that are not covered here, please update the mapping relation in this table before implementation the corresponding interface log

{yyyyMMdd} is the date that served for this interface log

{number} is indicating that there might be multiple interface logs each day if the file size reaching maximum, then a new file will be generated, and the old one will be tagged with this number

The file currently logging should has no {number}, which means interface.log

#### Separator within each log

For each log item within an interface log, the column/field should be separated by some delimiter, generally ‘|’ is used.

In situation there are ‘|’ in each field value, it should be escaped or quoted to avoid confusion.

#### Definition of each column/field

|  |  |  |  |
| --- | --- | --- | --- |
| **Log Column/Field** | **Explanation** | Restriction | Comment |
| Timestamp | Should be human-readable, format as yyyy-mm-dd HH:MM:SS nnn | Mandatory |  |
| Priority | Priority of current log message | Optional |  |
| Logger Class Name | Class Name of logger we used | Optional |  |
| Thread Name | Name or index of thread writing this log | Mandatory |  |
| Invoker Class Name | Class Name that call this logging method | Mandatory |  |
| Invoker Method Name | Method Name inside which call this logging method | Mandatory |  |
| Invoker Method Line | Ling number of above calling method | Mandatory |  |
| Type | Enumeration:  Request, Response | Mandatory |  |
| Status | Enumeration:  Success, Error | Mandatory |  |
| Trace Unique Id | Unique identifier for indicating each request/response pair, could be reused after period of time | Mandatory |  |
| Origin | Sender of the request/response message | Mandatory |  |
| Destination | Receiver of the request/response message | Mandatory |  |
| Interface Name | Name of the interface | Mandatory |  |
| Duration | Duration between the Message sending and receiving | Only available when response is sent back | Unit: millisecond |
| Request | The parameters for specified request | Only available when logging request | The format of parameters key and value should be human-readable |
| Response | Response sent back by  if the interface support such a field | Only available when logging response | The format of response should be human-readable |

### Suggested Scenarios recoding interface log

* Webservice with other applications, including SOAP, RESTFUL, but not restricted
* FTP files uploading or retrieval with other applications or processes
* HTTP messaging via servlet etc, with other applications
* Database process, including both normal jdbc statements or procedures, and wrapped ones by ORM such as ibatis, hibernate