**XXX project**

**Outline Design Specification**

**Fiction:**

**Review:**

**approve:**

**Zhongkesoft Technology Co., Ltd.**

**File change history**

serial number: Serial number:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| serial number | date | change content | Edition | drafter or reviser | approver |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Table of contents**

[1 Introduction 4](#_Toc250466929)

[1.1 Purpose of writing 4](#_Toc250466930)

[1.2 Background 4](#_Toc250466931)

[1.3 Definition 4](#_Toc250466932)

[1.4 Design Guidelines 4](#_Toc250466933)

[1.5 Reference 4](#_Toc250466934)

[2Overall design 4](#_Toc250466935)

[2.1 Overall structure 4](#_Toc250466936)

[2.2 Process Design 4](#_Toc250466937)

[2.3 Performance Design 4](#_Toc250466938)

[2.4 Component manufacturing, purchase and reuse analysis 5](#_Toc250466939)

[2.5 Unresolved Issues 5](#_Toc250466940)

[3 System Function Design 5](#_Toc250466941)

[4 System data structure design 5](#_Toc250466942)

[4.1 Key points of logical structure design 5](#_Toc250466943)

[4.2 Key points in physical structure design 5](#_Toc250466944)

[5 Interface Design 5](#_Toc250466945)

[5.1External interface 5](#_Toc250466946)

[5.2 Internal interface 5](#_Toc250466947)

[6System error handling design 6](#_Toc250466948)

[6.1 Error message 6](#_Toc250466949)

[6.2 Remedies 6](#_Toc250466950)

[6.3 System maintenance design 6](#_Toc250466951)

[7 Cutting Instructions 6](#_Toc250466952)

# introduction

## Purpose of writing

[ Description: The purpose of writing this summary design specification is to indicate the intended audience. You can list the issues between modules that this document will focus on solving for developers based on the requirements specification, the impact on detailed design, etc. ]

## background

[Description: The name of the software system to be developed; list the task proposer, developer, user of this project, the environment in which the software will run, and the relationship with other related systems, etc. 】

## definition

[List the definitions of technical terms used in this document and the original phrases of the foreign initials].

## design guidelines

[List the criteria for software outline design, such as: compatibility, high cohesion and low coupling, technological advancement, technological maturity, etc. 】

## References

[List relevant reference documents, such as:

1. The approved mission statement or contract for this project, and the approval document from the superior authority;
2. Other published documents belonging to this project;
3. The documents and materials cited throughout this document include the software development standards to be used. List the titles, document numbers, publication dates and publishing units of these documents, and indicate the sources from which these documents can be obtained. 】

# overall design

## Design Principles

[Describe the structural design principles and overall design guiding ideology of this system, mainly considering the goals of system design implementation, such as: data real-time description, scalability description, maintainability, advancement, data integrity, security, etc. describe. 】

## design specifications

[Describe the design specifications or national standards or specifications for software development used, mainly including: (not limited to the following, nor specified)

* naming convention

Specify the preparation specifications for system and subsystem names, program names, database table (file) names, data names, variable names, etc.

* Interface convention

Specify the overall layout of the screen interface, such as menu rows, display bodies, icon buttons, prompt messages, error messages, etc. to standardize and unify the style.

* Program writing specifications

According to the characteristics of the programming tools used, standardized requirements are formulated to make the program easy to read, maintainable and portable.

The specific selected specifications can be listed as long as they are helpful to the design. The number and related specification titles can be decided by yourself. 】

## The overall structure

[Describe the overall architecture and framework of the system. Requires illustrations with diagrams and explanations with text. Explain the division of system elements (modules at each level, subroutines, utility programs, etc.) of this system, briefly explain the identifier and function of each system element, and give the control and controlled relationship between each element hierarchically. 】

## Process Design

[Explain the basic design concepts and processing procedures of this system, using diagrams as much as possible. 】

## Performance design

[Describe the main performance requirements for this system, including requirements for security, stability, scalability, etc. 】

## Component manufacturing, purchasing, and reuse analysis

[Analyze and evaluate whether each system component (subsystem, functional module, etc.) is developed by itself, purchased from others, or reused with existing functions. 】

## unresolved issues

[Describe various issues that have not been resolved during the outline design process and that the designer believes must be resolved before the system is completed. 】

# System functional design

[Systematically describe the functional design of each subsystem and each module. 】

The message center mainly serves as an independent third-party message communication framework, as a distribution center for message processing within the site, and is the basis for dialogue between the system and users. Through this distribution center, the system can communicate to users information including but not limited to the following:

In-site messages: refer to the message content triggered by some activities of the system itself, including system announcements, to-do reminders, etc.; used to convey to users the actions they need to perform in the current system, or the information they need to know. This type of information is limited to the system and does not interact with external systems for information or actions.

After the message data enters the message center, it will be triggered to relevant users through its distribution mechanism. Users can view the specific message content in the "site message" module on the front end.

The main functions that the message center needs to include are:

1. An interface for triggering messages called externally (external here refers to the message center, which can be called through the HTTP protocol open interface). It can support custom message templates (support HTML templates, and can directly generate messages containing <a> tags, providing jump);
2. Message center: used for message distribution, is a centralized processing module for message data, and is responsible for triggering messages to the corresponding user groups.

When the message source requests to send a message notification, the request generally contains: template ID, request time, receiving object, and message content (variables defined in the template will be replaced). The receiving object can mark the user group by organization ID, position or role, etc., or it can trigger the message only for the specified user ID.

The message center generally contains a pool to be sent and a pool that has been sent. The pool to be sent is generally the original request from the message source, and the sent pool contains the target user ID of the message data.

Messages in the sent pool can be queried by users in the "Site Message" module, and include two types of status: read messages and unread messages. When the user clicks on an unread message, it will be converted into a read message. There is no reversible setting for the time being. operate.

As a front-end module, "site messaging" is the only way to present message data to users. There are two main actions: push and pull to obtain message data. Currently, we ignore the "push" action for the time being and only deal with the "pull" action, that is, the in-site messages regularly pull new messages from the message center, usually marked with a timestamp, and only the new messages after the timestamp are obtained each time. Data and message centers must provide corresponding interfaces.

# System data structure design

## physical structure design

[Give the name and identifier of each data structure used in this system, as well as the identification, definition, and length of each data item, record, file, and system among them, as well as the hierarchical or tabular interactions between them. relation. 】

### 4.1.1 library storage

[Estimate the storage volume and growth trend of each table, estimate the data volume, and plan the storage structure and distribution. 】

### 4.1.2 table storage

[Determine the storage method and parameters of the table based on the frequency of addition, deletion, modification, and query operations, data volume, and usual operations of each table. 】

### 4.1.3 index

### 4.1.4 other

## Data Dictionary

[Various code tables (including values). 】

## Precautions

[Things to note about database management and maintenance. 】

## Database table design

[Give the storage requirements, access method, access unit, physical relationship of access (index, device, storage area), design considerations and confidentiality conditions for each data item in each data structure used in this system. 】

| Logical table name | Message producer message storage table | | Physical table name | | ProducerMessage | |
| --- | --- | --- | --- | --- | --- | --- |
| primary key | i d | | index | |  | |
| Logical field name | Physical field name | type of data | null value | default value | | Remark |
| ID |  | int |  |  | |  |
| producerid | message producer id | int |  |  | |  |
| consumerid | Message consumer id | int |  |  | |  |
| messagecontent | Message content | varchar |  |  | |  |
| consumerjudge | Message consumer type | int |  |  | | The number issued to independent users is 0, the department is 1, the position is 2, and all are 3 |
| producerdate | message time | datetime |  |  | |  |
| messagelink | Link | varchar |  |  | |  |
| send | send | int |  |  | | 0 not sent, 1 sent |
| senddate | Schedule message sending time | datetime |  |  | |  |

| Logical table name | Message consumer message storage table | | Physical table name | | Consumer Message | |
| --- | --- | --- | --- | --- | --- | --- |
| primary key | i d | | index | |  | |
| Logical field name | Physical field name | type of data | null value | default value | | Remark |
| ID |  | int |  |  | |  |
| producerid | message producer id | int |  |  | |  |
| consumerid | Message consumer id | int |  |  | |  |
| messagecontent | Message content | varchar |  |  | |  |
| producerdate | message time | datetime |  |  | |  |
| readjudge | Read unread | int |  |  | | 0 unread, 1 read |
| messagelink | Link | varchar |  |  | |  |

| Logical table name | User information storage table | | Physical table name | | User | |
| --- | --- | --- | --- | --- | --- | --- |
| primary key | i d | | index | |  | |
| Logical field name | Physical field name | type of data | null value | default value | | Remark |
| ID |  | int |  |  | |  |
| consumerid | Message consumer id | int |  |  | |  |
| department id | department id | int |  |  | |  |
| position id | Position id | int |  |  | |  |

| Logical table name | Message template storage table | | Physical table name | | MessageTemplate | |
| --- | --- | --- | --- | --- | --- | --- |
| primary key | i d | | index | |  | |
| Logical field name | Physical field name | type of data | null value | default value | | Remark |
| ID |  | int |  |  | |  |
| messagetemplateid | template id | int |  |  | |  |
| templateownerid | template owner id | int |  |  | | 0 public |
| consumerid | Message consumer id | int |  |  | |  |
| messagecontent | Message content | varchar |  |  | |  |
| consumerjudge | Message consumer type | int |  |  | | The number issued to independent users is 0, the department is 1, the position is 2, and all are 3 |
| messagelink | Link | varchar |  |  | |  |
| send | send | int |  |  | | 0 not sent, 1 sent |
| senddateparamesnumber | Number of parameters | int |  |  | |  |

# Interface design

## external interface

[Explain that the arrangement of all interfaces between this system and the outside world includes the interfaces between software and hardware, the interfaces between this system and various supporting software, and the human-computer interaction interface relationship. 】

|  |  |  |  |
| --- | --- | --- | --- |
| Interface ID | Interface detailed description | Interface Type | Corresponding component name |
|  |  |  |  |

## internal interface

[Describe the arrangement of interfaces between various system elements within this system. 】

|  |  |  |  |
| --- | --- | --- | --- |
| Interface ID | Interface detailed description | Interface Type | Corresponding component name |
| 1 | Message producers generate messages | post | producermessage/add |

### **Message producers generate messages**

### 5. 2. 1.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 1.2 Function description \_

Message producers generate messages ;

The client saves the return information into shared parameters.

### 5.2.1.3 Scope of application \_ \_

Android , IOS , web

### 5.2.1.4Interface address \_ \_ \_ \_

  /producermessage/add

### 5. 2. 1.5 Interface type \_

POST

### 5. 2. 1.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| producerid | yes | String | Message producer ID |
| consumerid | yes | String | Message consumer id |
| messagecontent | no | String | Message content |
| consumerjudge | yes | String | Message consumer properties |
| producerdate | yes | String | Dateyyyy -MM-dd HH:mm:ss |
| messagelink | no | String | Link |
| senddate | yes | String | Scheduled delivery time |

### 5.2.1.7 Request Protocol \_ \_ \_

{

" producerid ": " 111 ",

" consumerid ": " 222 ",

" messagecontent ": " You logged in on 2022. 12. 22 " ,

" consumerjudge ": " 0 ",

" producerdate ": " 2022-12-22 1 : 1 : 1 ",

" messagelink ": " https://123.1.1 " ,

" senddate ": " 2022-12-22 1 : 1 : 1 "

}

### 5.2.1.8 Response protocol \_ \_ \_

{

" msg ": " Published successfully ",

"code": 0

}

{

" msg ": " Publishing failed ",

"code": 1

}

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| msg |  | Prompt message |
| code |  | 0 normal , 1 failed |

### **Query all messages sent by the message producer**

### 5. 2. 2.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 2.2 Function description \_

### Query all messages sent by the message producer

### 5.2.2.3 Scope of application \_ \_

Android , IOS , web

### 5. 2. 2.4 Interface address \_

  /producermessage/selectByproducerid

### 5. 2. 2.5 Interface type \_

GET

### 5. 2. 2.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| producerid | yes | String | Message producer ID |

### 5.2.2.7 Request Protocol \_ \_ \_

{

" producerid ": " 111 "

}

### 5.2.2.8 Response protocol \_ \_ \_

[{

"id": 5,

"producerid": 111,

"consumerid": 333 ,

"messagecontent": " Sent you a message ",

"consumerjudge": 1,

"producerdate": "2022-11-11 12:34:50",

"messagelink": "https://blog.csdn.net/" ,

" senddate ": " 2022-12-22 1 : 1 : 1 "

},

{

"id": 6,

"producerid": 111,

"consumerid": 222,

"messagecontent": " Sent you a message ",

"consumerjudge": 1,

"producerdate": "2022-11-11 12:34:50",

"messagelink": "https://blog.csdn.net/" ,

" senddate ": " 2022-12-22 1 : 1 : 1 "

}]

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| producerid | Message producer ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| consumerjudge | consumer attributes | The number issued to independent users is 0, the department is 1, the position is 2, and all are 3 |
| producerdate | date | yyyy-MM-dd HH:mm:ss |
| messagelink | Link |  |
| senddate | Scheduled delivery time | yyyy-MM-dd HH:mm:ss |

### **Query the entire message producer message object based on the primary key**

### 5. 2. 3.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 3.2 Function description \_

### Query the entire message producer message object based on the primary key

### 5.2.3.3 Scope of application \_ \_

Android , IOS , web

### 5. 2. 3.4 Interface address \_

  /producermessage/selectByPrimaryKey

### 5. 2. 3.5 Interface type \_

GET

### 5. 2. 3.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| ID | yes | String | primary key |

### 5.2.3.7 Request Protocol \_ \_ \_

{

" id ": " 1 "

}

### 5.2.3.8 Response protocol \_ \_ \_

{

"id": 5,

"producerid": 111,

"consumerid": 333 ,

"messagecontent": " Sent you a message ",

"consumerjudge": 1,

"producerdate": "2022-11-11 12:34:50",

"messagelink": "https://blog.csdn.net/"

}

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| producerid | Message producer ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| consumerjudge | consumer attributes | The number issued to independent users is 0, the department is 1, the position is 2, and all are 3 |
| producerdate | date | yyyy-MM-dd HH:mm:ss |
| messagelink | Link |  |
| senddate | Scheduled delivery time | yyyy-MM-dd HH:mm:ss |

### **Message producer deletes message**

### 5. 2. 4.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 4.2 Function description \_

### Message producer deletes message

### 5.2.4.3 Scope of application \_ \_

Android , IOS , web

### 5. 2. 4.4 Interface address \_

  /producermessage/ delete

### 5. 2. 4.5 Interface type \_

POST

### 5. 2. 4.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| ID | yes | String | primary key |

### 5.2.4.7 Request Protocol \_ \_ \_

{

" id ": " 1 "

}

### 5.2.4.8 Response protocol \_ \_ \_

True / false

### **Query all read information of consumers**

### 5. 2. 5.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 5.2 Function description \_

### Query all read information of consumers

### 5.2.5.3 Scope of application \_ \_

Android , IOS , web

### 5. 2. 5.4 Interface address \_

  /consumermessage/selectread1

### 5. 2. 5.5 Interface type \_

GET

### 5. 2. 5.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| consumerid | yes | String | Message consumer id |

### 5.2.5.7 Request Protocol \_ \_ \_

{

" consumerid ": " 111 "

}

### 5.2.5.8 Response protocol \_ \_ \_

[{

"id": 1,

"producerid": 111,

"consumerid": 111,

"messagecontent": "111",

"producerdate": "2022-01-01 01:01:01",

"readjudge": 1,

"messagelink": "https://blog.csdn.net/"

},

{

"id": 2,

"producerid": 222,

"consumerid": 111,

"messagecontent": "222",

"producerdate": "2022-10-10 01:01:01",

"readjudge": 1,

"messagelink": "https://blog.csdn.net/"

}]

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| producerid | Message producer ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| producerdate | date | yyyy-MM-dd HH:mm:ss |
| readjudge | Read unread | 0 unread , 1 read |
| messagelink | Link |  |

### **Query all unread information of consumers**

### 5. 2. 6.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 6.2 Function description \_

### unread information of consumers

### 5.2.6.3 Scope of application \_ \_

Android , IOS , web

### 5. 2. 6.4 Interface address \_

  /consumermessage/selectunread0

### 5. 2. 6.5 Interface type \_

GET

### 5. 2. 6.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| consumerid | yes | String | Message consumer id |

### 5.2.6.7 Request protocol \_ \_ \_

{

" consumerid ": " 111 "

}

### 5.2.6.8 Response protocol \_ \_ \_

[{

"id": 1,

"producerid": 111,

"consumerid": 111,

"messagecontent": "111",

"producerdate": "2022-01-01 01:01:01",

"readjudge": 0 ,

"messagelink": "https://blog.csdn.net/"

},

{

"id": 2,

"producerid": 222,

"consumerid": 111,

"messagecontent": "222",

"producerdate": "2022-10-10 01:01:01",

"readjudge": 0 ,

"messagelink": "https://blog.csdn.net/"

}]

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| producerid | Message producer ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| producerdate | date | yyyy-MM-dd HH:mm:ss |
| readjudge | Read unread | 0 unread , 1 read |
| messagelink | Link |  |

### **Query all read information of consumers after a specific time point**

### 5. 2. 7.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 7.2 Function description \_

### Query all read information of consumers after a specific time point

### 5.2.7.3 Scope of application \_ \_

Android , IOS , web

### 5.2.7.4Interface address \_ \_ \_ \_

  /consumermessage/selecttimeread1

### 5. 2. 7.5 Interface type \_

GET

### 5. 2. 7.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| consumerid | yes | String | Message consumer id |
| producerdate | yes | String | Dateyyyy -MM-dd HH:mm:ss |

### 5.2.7.7 Request Protocol \_ \_ \_

{

" consumerid ": " 111 " ,

“producerdate”=”2022-9-9 1:1:1”

}

### 5.2.7.8 Response protocol \_ \_ \_

[{

"id": 1,

"producerid": 111,

"consumerid": 111,

"messagecontent": "111",

"producerdate": "2022-1 0 -01 01:01:01",

"readjudge": 1 ,

"messagelink": "https://blog.csdn.net/"

},

{

"id": 2,

"producerid": 222,

"consumerid": 111,

"messagecontent": "222",

"producerdate": "2022-10-10 01:01:01",

"readjudge": 1 ,

"messagelink": "https://blog.csdn.net/"

}]

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| producerid | Message producer ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| producerdate | date | yyyy-MM-dd HH:mm:ss |
| readjudge | Read unread | 0 unread , 1 read |
| messagelink | Link |  |

### **Query all unread messages from consumers after a specific point in time**

### 5. 2. 8.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 8.2 Function description \_

### unread messages from consumers after a specific point in time

### 5.2.8.3 Scope of application \_ \_

Android , IOS , web

### 5.2.8.4Interface address \_ \_ \_ \_

  /consumermessage/selecttimeunread0

### 5. 2. 8.5 Interface type \_

GET

### 5. 2. 8.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| consumerid | yes | String | Message consumer id |
| producerdate | yes | String | Dateyyyy -MM-dd HH:mm:ss |

### 5.2.8.7 Request Protocol \_ \_ \_

{

" consumerid ": " 111 " ,

“producerdate”=”2022-9-9 1:1:1”

}

### 5.2.8.8 Response protocol \_ \_ \_

[{

"id": 1,

"producerid": 111,

"consumerid": 111,

"messagecontent": "111",

"producerdate": "2022-1 0 -01 01:01:01",

"readjudge": 0 ,

"messagelink": "https://blog.csdn.net/"

},

{

"id": 2,

"producerid": 222,

"consumerid": 111,

"messagecontent": "222",

"producerdate": "2022-10-10 01:01:01",

"readjudge": 0 ,

"messagelink": "https://blog.csdn.net/"

}]

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| producerid | Message producer ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| producerdate | date | yyyy-MM-dd HH:mm:ss |
| readjudge | Read unread | 0 unread , 1 read |
| messagelink | Link |  |

### **Query the entire message consumer message object based on the primary key**

### 5. 2. 9.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 9.2 Function description \_

### Query the entire message consumer message object based on the primary key

### 5.2.9.3 Scope of application \_ \_

Android , IOS , web

### 5.2.9.4Interface address \_ \_ \_ \_

  /consumermessage/selectByPrimaryKey

### 5. 2. 9.5 Interface type \_

GET

### 5. 2. 9.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| ID | yes | String | primary key |

### 5.2.9.7 Request Protocol \_ \_ \_

{

" i d ": " 1 " ,

}

### 5.2.9.8 Response protocol \_ \_ \_

{

"id": 1,

"producerid": 111,

"consumerid": 111,

"messagecontent": "111",

"producerdate": "2022-1 0 -01 01:01:01",

"readjudge": 0 ,

"messagelink": "https://blog.csdn.net/"

}

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| producerid | Message producer ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| producerdate | date | yyyy-MM-dd HH:mm:ss |
| readjudge | Read unread | 0 unread , 1 read |
| messagelink | Link |  |

### **Message consumer modifies read and unread**

### 5. 2. 10.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 10.2 Function description \_

### Message consumer modifies read and unread

### 5.2.10.3 Scope of application \_ \_

Android , IOS , web

### 5 . 2 . 10.4 interface address

  /consumermessage/ readjudge

### 5. 2. 10.5 Interface type \_

POST

### 5. 2. 10.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| ID | yes | String | primary key |
| readjudge | yes | String | Read is 1 , Unread is 0 |

### 5.2.10.7 Request Agreement \_ \_ \_

{

" i d ": " 1 " ,

" readjudge ":"0"

}

### 5.2.10.8 Response protocol \_ \_ \_

### True/false

### **Message consumer deletes message**

### 5. 2. 11.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 11.2 Function description \_

### Message consumer deletes message

### 5.2.11.3 Scope of application \_ \_

Android , IOS , web

### 5.2.11.4Interface address \_ \_ \_ \_

  /consumermessage/ delete

### 5. 2. 11.5 Interface type \_

POST

### 5. 2. 11.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| ID | yes | String | primary key |

### 5.2.11.7 Request Protocol \_ \_ \_

{

" i d ": " 1 " ,

}

### 5.2.11.8 Response protocol \_ \_ \_

### True/false

### **Generate message template**

### 5. 2. 12.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 12.2 Function description \_

### Generate message template

### 5.2.12.3 Scope of application \_ \_

Android , IOS , web

### 5. 2. 12.4 interface address \_

  /messagetemplate/ add

### 5. 2. 12.5 Interface type \_

POS T

### 5. 2. 12.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| messagetemplateid | yes | String | Template ID |
| templateownerid | yes | String | Template owner ID |
| consumerid | yes | String | Message consumer id |
| messagecontent | no | String | Message content |
| consumerjudge | yes | String | Message consumer properties |
| messagelink | no | String | Link |
| parametersnumber | yes | String | Number of parameters |

### 5.2.12.7 Request Protocol \_ \_ \_

{

" messagetemplateid ": " 111 ",

" templateownerid ": " 111 ",

" consumerid ": " 222 ",

" messagecontent ": " You logged in on 2022. 12. 22 " ,

" consumerjudge ": " 0 ",

" messagelink ": " https://123.1.1 " ,

" paramesnumber ": " 2 "

}

### 5.2.12.8 Response protocol \_ \_ \_

{

" msg ": " Published successfully ",

"code": 0

}

{

" msg ": " Publishing failed ",

"code": 1

}

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| msg |  | Prompt message |
| code |  | 0 normal , 1 failed |

### **Update message template**

### 5. 2. 13.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 13.2 Function description \_

### Update message template

### 5.2.13.3 Scope of application \_ \_

Android , IOS , web

### 5.2.13.4Interface address \_ \_ \_ \_

  /messagetemplate/ update

### 5. 2. 13.5 Interface type \_

POS T

### 5. 2. 13.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| ID | yes | String | primary key |
| messagetemplateid | yes | String | Template ID |
| templateownerid | yes | String | Template owner ID |
| consumerid | yes | String | Message consumer id |
| messagecontent | no | String | Message content |
| consumerjudge | yes | String | Message consumer properties |
| messagelink | no | String | Link |
| parametersnumber | yes | String | Number of parameters |

### 5.2.13.7 Request Protocol \_ \_ \_

{

" id ": " 11 ",

" messagetemplateid ": " 111 ",

" templateownerid ": " 111 ",

" consumerid ": " 222 ",

" messagecontent ": " You logged in on 2022. 12. 22 " ,

" consumerjudge ": " 0 ",

" messagelink ": " https://123.1.1 " ,

" paramesnumber ": " 2 "

}

### 5.2.13.8 Response protocol \_ \_ \_

{

" msg ": " Published successfully ",

"code": 0

}

{

" msg ": " Publishing failed ",

"code": 1

}

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| msg |  | Prompt message |
| code |  | 0 normal , 1 failed |

### **Delete message template**

### 5. 2. 14.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 14.2 Function description \_

### Delete message template

### 5.2.13.3 Scope of application \_ \_

Android , IOS , web

### 5.2.13.4Interface address \_ \_ \_ \_

  /messagetemplate/ delete

### 5. 2. 13.5 Interface type \_

POS T

### 5. 2. 13.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| ID | yes | String | primary key |

### 5.2.13.7 Request Protocol \_ \_ \_

{

" id ": " 11 ",

}

### 5.2.13.8 Response protocol \_ \_ \_

true/false

### **Query templates by owner**

### 5. 2. 15.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 15.2 Function description \_

### Query templates by owner

### 5.2.15.3 Scope of application \_ \_

Android , IOS , web

### 5.2.15.4Interface address \_ \_ \_ \_

  /producermessage/selectBytemplateownerid

### 5. 2. 15.5 Interface type \_

GET

### 5. 2. 15.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| templateownerid | yes | String | Template owner ID |

### 5.2.15.7 Request Protocol \_ \_ \_

{

" templateownerid ": " 111 "

}

### 5.2.15.8 Response protocol \_ \_ \_

[ { " id ": " 11 ",

" messagetemplateid ": " 111 ",

" templateownerid ": " 111 ",

" consumerid ": " 222 ",

" messagecontent ": " You logged in on 2022. 12. 22 " ,

" consumerjudge ": " 0 ",

" messagelink ": " https://123.1.1 " ,

" paramesnumber ": " 2 "

} ,

{

" id ": " 12 ",

" messagetemplateid ": " 111 ",

" templateownerid ": " 111 ",

" consumerid ": " 222 ",

" messagecontent ": " You logged in on 2022. 12. 22 " ,

" consumerjudge ": " 0 ",

" messagelink ": " https://123.1.1 " ,

" paramesnumber ": " 2 "

}]

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| messagetemplateid | template id |  |
| templateownerid | Template owner ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| consumerjudge | consumer attributes | The number issued to independent users is 0, the department is 1, the position is 2, and all are 3 |
| messagelink | Link |  |
| parametersnumber | Number of parameters |  |

### **Query templates based on template id**

### 5. 2. 2.1 Modification records \_

|  |  |  |
| --- | --- | --- |
| Modifier | Change the time | illustrate |
| Zhang Ruichen | 2023-1-9 | Create new interface |

### 5. 2. 2.2 Function description \_

### Query templates based on template id

### 5.2.2.3 Scope of application \_ \_

Android , IOS , web

### 5. 2. 2.4 Interface address \_

  /producermessage/selectBymessagetemplateid

### 5. 2. 2.5 Interface type \_

GET

### 5. 2. 2.6 Parameter description \_

|  |  |  |  |
| --- | --- | --- | --- |
| parameter name | Required | type | illustrate |
| messagetemplateid | yes | String | Template ID |

### 5.2.2.7 Request Protocol \_ \_ \_

{

" messagetemplateid ": " 111 "

}

### 5.2.2.8 Response protocol \_ \_ \_

{ " id ": " 11 ",

" messagetemplateid ": " 111 ",

" templateownerid ": " 111 ",

" consumerid ": " 222 ",

" messagecontent ": " You logged in on 2022. 12. 22 " ,

" consumerjudge ": " 0 ",

" messagelink ": " https://123.1.1 " ,

" paramesnumber ": " 2 "

}

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Frontend display name** | **illustrate** |
| ID |  | primary key |
| messagetemplateid | template id |  |
| templateownerid | Template owner ID |  |
| consumerid | Message consumer id |  |
| messagecontent | Message content |  |
| consumerjudge | consumer attributes | The number issued to independent users is 0, the department is 1, the position is 2, and all are 3 |
| messagelink | Link |  |
| parametersnumber | Number of parameters |  |

# System error handling design

## error message

[Use a list to describe the form, meaning and processing method of the system output information when each possible error or malfunction occurs. 】

## Remedies

[Describe possible workarounds after the fault occurs. 】

## System maintenance design

[Describe the arrangements made in the internal design of the program for the convenience of system maintenance, including specifically arranging checkpoints and special modules in the program for system inspection and maintenance. 】

# Cutting Instructions

1. The contents of the above chapters can be added according to the needs of the project, or the chapter structure can be modified according to the actual situation.

2. Chapter 4 can form a separate "Database Design Instruction".

3. Chapter 5 can form a separate "Interface Design Specification".