文件服务器设计文档

作者：Lance

修订：2020.08.23

目 录

[一、 背景 3](#_Toc49108558)

[二、 总体架构图 3](#_Toc49108559)

[1. 用户界面 3](#_Toc49108560)

[2. 逻辑处理层 3](#_Toc49108561)

[3. 数据持久层 3](#_Toc49108562)

[三、 表结构设计 3](#_Toc49108563)

[1. 文件表 3](#_Toc49108564)

[2. 分类表 4](#_Toc49108565)

[3. 接入密钥表 4](#_Toc49108566)

[四、 接口设计 4](#_Toc49108567)

[1. 获取所有文件信息 4](#_Toc49108568)

[2. 无条件限制分页获取文件信息 5](#_Toc49108569)

[3. 有条件限制分页获取文件信息 5](#_Toc49108570)

[4. 新增文件 5](#_Toc49108571)

[5. 删除文件 5](#_Toc49108572)

[6. 修改文件 5](#_Toc49108573)

[7. 下载文件 5](#_Toc49108574)

[8. 获取所有分类信息 6](#_Toc49108575)

[9. 无条件限制分页获取分类信息 6](#_Toc49108576)

[10. 新增分类 6](#_Toc49108577)

[11. 删除分类 6](#_Toc49108578)

[12. 修改分类 6](#_Toc49108579)

[13. 获取所有密钥信息 6](#_Toc49108580)

[14. 新增密钥 6](#_Toc49108581)

[15. 删除密钥 7](#_Toc49108582)

[16. 无条件限制分页获取密钥信息 7](#_Toc49108583)

[17. 密钥下载 7](#_Toc49108584)

[五、 可行性分析 7](#_Toc49108585)

[六、 性能分析 7](#_Toc49108586)

# 背景

Magic box 需要文件服务器来承载项目中所需要持久化的文件，以便于后续持续访问。

# 总体架构图



## 用户界面

前端采用element-ui搭建，旨在快速成型，主要提供4个页面：

* 文件列表
* 分类管理
* 文件上传
* AccessKey管理

## 逻辑处理层

后端采用java语言实现，选用框架为springboot，快速构建后端功能，选用mybatis-plus便捷操作数据模型。

## 数据持久层

文件信息持久采用mysql，文件持久暂采用本地方式，后续再考虑迁移。

# 表结构设计

## 文件表

create table if not exists `files` (

`id` char(19) not null comment 'File ID',

`cate\_id` char(19) not null comment 'Category ID',

`name` varchar(50) not null comment 'File Name',

`path` varchar(100) not null comment 'File Path',

`file\_desc` varchar(200) not null comment 'File Describe',

`is\_deleted` tinyint(1) unsigned not null default '0' comment 'logic delete 0: (false) exist, 1: (true) deleted',

`created\_time` datetime not null comment 'create time',

`modified\_time` datetime not null comment 'update time',

primary key (`id`),

unique key `uk\_name` (`path`)

)engine=innodb default charset=utf8mb4 comment 'files';

## 分类表

create table if not exists `categories` (

`id` char(19) not null comment 'Category ID',

`name` varchar(20) not null comment 'Category Name',

`cate\_desc` varchar(200) not null comment 'Category describe',

`is\_deleted` tinyint(1) unsigned not null default '0' comment 'logic delete 0: (false) exist, 1: (true) deleted',

`created\_time` datetime not null comment 'create time',

`modified\_time` datetime not null comment 'update time',

primary key (`id`),

unique key `uk\_name` (`name`)

)engine=innodb default charset=utf8mb4 comment 'categories';

## 接入密钥表

create table if not exists `accesskey`(

`id` char(19) not null comment 'Accesskey ID',

`accesskey` char(19) not null comment 'Accesskey',

`is\_deleted` tinyint(1) unsigned not null default '0' comment 'logic delete 0: (false) exist, 1: (true) deleted',

`created\_time` datetime not null comment 'create time',

primary key (`id`),

unique key `accesskey` (`accesskey`)

)engine=innodb default charset=utf8mb4 comment 'categories';

# 接口设计

文件接口

## 获取所有文件信息

* URL: /filemgt/files/all
* Method: GET
* RequestBody: {}
* ResponseBody: {}

## 无条件限制分页获取文件信息

* URL: /filemgt/files/page/{current}/{limit}
* Method: GET
* RequestBody: {}
* ResponseBody: {}

## 有条件限制分页获取文件信息

* URL: /filemgt/files/page/{current}/{limit}
* Method: GET
* RequestBody: {“conditionKey”: “conditionValue”}
* ResponseBody: {}

## 获取单个文件信息

* URL: /filemgt/files/fileinfo/{fileId}
* Method: GET
* RequestBody: {“fileId”: “fileId”}
* ResponseBody: {}

## 页面上传单个文件

* URL: /filemgt/files/file
* Method: POST
* RequestBody: {“MultiPartFile”: “file”}
* ResponseBody: {}

## 新增文件信息

* URL: /filemgt/files/fileinfo
* Method: POST
* RequestBody: {“fileInfoKey”: “fileInfoValue”}
* ResponseBody: {}

## 删除文件

* URL: /filemgt/files/{fileId}
* Method: DELETE
* RequestBody: {}
* ResponseBody: {}

## 修改文件

* URL: /filemgt/files/{fileId}
* Method: PUT
* RequestBody: {“fileInfoKey”: “fileInfoValue”, “MultiPartFile”: “file”}
* ResponseBody: {}

## 下载文件

* URL: /filemgt/files/{fileId}
* Method: GET
* RequestBody: {“fileId”: “fileId”}
* ResponseBody: {}

分类接口

## 获取所有分类信息

* URL: /filemgt/categories/all
* Method: GET
* RequestBody: {}
* ResponseBody: {}

## 无条件限制分页获取分类信息

* URL: /filemgt/categories/page/{current}/{limit}
* Method: GET
* RequestBody: {}
* ResponseBody: {}

## 新增分类

* URL: /filemgt/categories/category
* Method: POST
* RequestBody: {“CategoryInfoKey”: “CategoryInfoValue”}
* ResponseBody: {}

## 删除分类

* URL: /filemgt /categories/{categoryId}
* Method: DELETE
* RequestBody: {}
* ResponseBody: {}

## 修改分类

* URL: /filemgt /categories/{categoryId}
* Method: DELETE
* RequestBody: {“fileInfoKey”: “fileInfoValue” }
* ResponseBody: {}

认证密钥接口

## 获取所有密钥信息

* URL: /filemgt/accesskey/all
* Method: GET
* RequestBody: {}
* ResponseBody: {}

## 新增密钥

* URL: /filemgt/accesskey/category
* Method: POST
* RequestBody: {}
* ResponseBody: {}

## 删除密钥

* URL: /filemgt /accesskey/{akId}
* Method: DELETE
* RequestBody: {}
* ResponseBody: {}

## 无条件限制分页获取密钥信息

* URL: /filemgt/accesskey/page/{current}/{limit}
* Method: GET
* RequestBody: {}
* ResponseBody: {}

## 密钥下载

* URL: /filemgt/accesskey/{akId}/{ak}
* Method: GET
* RequestBody: {}
* ResponseBody: {}

# 可行性分析

本设计采用SpringBoot，Vue主流框架作为前、后端承载，mysql作为信息存储，本地卷作为文件存储，无技术盲点，整体方案可行。

# 性能分析

该服务作为承载magic-box所有微服务的文件存储服务器，逻辑读取以及列表获取无压力，但文件下载存在一定的压力，后续需要更专业的分布式文件存储服务来承载，就目前业务量而言，无风险。