

软件设计文档

目录

- 1.软件设计技术2
- 2 代码结构.....2
- 3.源码分析.....4
 - 通过路由分析（通过看路由可以了解思路）4
 - 路由解析5
 - 通过文件分析16
- 4.数据库表19

1. 软件设计技术

Object-Oriented Programming, Aspect-Oriented Programming

采用 nodejs 框架，结合 bootstrap 前端设计(用 jade 代替 html)，和 jquery

2 代码结构

代码 github 地址 <https://github.com/zhoutengteng/circle>

glueCire/

 bower_components/

 config/

 db/

 complex.js

 file.js

 issue.js

 item.js

 mysql.db

 pic.js

 user.js

 readme.txt

lib/

node_modules/

public/

 css/

 js/

 circle_edit.js

 detail.js

 edit.js

header.js

index.js

modify.js

profile.js

signin.js

signup.js

readme.txt

public/

views/

includes/

footer.jade

head.jade

header.jade

header_only.jade

readme.txt

circle_edit.jade

detail.jade

edit.jade

index.jade

layout_header_only.jade

layoutBasic.jade

modify.jade

profile.jade

signin.jade

signnp.jade

welcome.jade

readme.txt

router/

circle_edit.js

detail.js

edit.js
index.js
logout.js
profile.js
signin.js
signup.js
validate.js
welcome.js

app.js
package.json
README.md

3.源码分析

通过路由分析（通过看路由可以了解思路）

```
app.get('/', require('./router/index').getMethod);  
app.get('/index', require('./router/index').getMethod);  
app.post('/logout', require('./router/logout').postMethod);  
app.get('/signup', require('./router/signup').getMethod);  
app.post('/signup', require('./router/signup').postMethod);  
app.get('/signin', require('./router/signin').getMethod);  
app.post('/signin', require('./router/signin').postMethod);  
app.get('/*', require('./router/validate').getMethod);  
app.get('/profile/detail/[0-9]+', require('./router/detail').getMethod);  
app.post('/circle/comment/[0-9]+', require('./router/detail').getComment);  
app.get('/circle/edit', require('./router/circle_edit').getMethod);  
app.post('/circle/comment', require('./router/detail').postComment);  
app.get('/profile/[A-Za-z0-9]+', require('./router/profile').getMethod);  
app.post('/circle/edit', require('./router/circle_edit').postMethod);
```

```
app.post('/circle/update', require('./router/circle_edit').updateMethod);  
app.post('/profile/[A-Za-z0-9 ]+', require('./router/profile').postMethod);  
app.post('/count', require('./router/profile').click);
```

路由解析

```
app.get('/', require('./router/index').getMethod);  
app.get('/index', require('./router/index').getMethod);
```

左小角为一个 slide，可以轮播，右上角有注册登录，对应界面

/public/views/index.jade

对应路由处理/router/index.js



对应的文件有：

/public/view/index.jade

/public/js/index.js

/public/views/header_only.jade

/public/js/header.js

加上 include ./includes/head.jade 包括的文件

其中 head.jade

包括 **bootstrap.min.css**

jquery.min.js

bootstrap.min.js

```
app.get('/signup', require('./router/signup').getMethod);  
app.post('/signup', require('./router/signup').postMethod);
```

localhost:3000/signup

应用 从 Safari 中导入 ubuntu

Sign in

注册一个账号

User Name
 数字和字母的组合

Sex
☐ 男 ☐ 女 选择一项

password
 数字和字母的组合

password repeat
 和密码输入要对应一致

Email address
 数字和字母以及@的组合和

telephone
 11位数字

submit

对应的路由处理文件为/router/signup.js,

这个 get 路由是拉取注册界面/public/views/signup.jade,, 在输入过程中会动态检验合理性

当点击提交按钮, 触发/public/js/signup.js 中 ajax 的 post 请求, 同时会检测合理性, 然后从/router/signup.js 中调用 mysql 数据库中的 users_tb 的表插入操作

(即文件/db/user.js)。如果成功会跳转到登录界面

包括的文件有/public/views/signup.jade

/public/js/signup.js

/router/signup.js

/db/user.js

/public/views/header_only.jade

/public/js/header.js

加上 include ./includes/head.jade 包括的文件

其中 head.jade

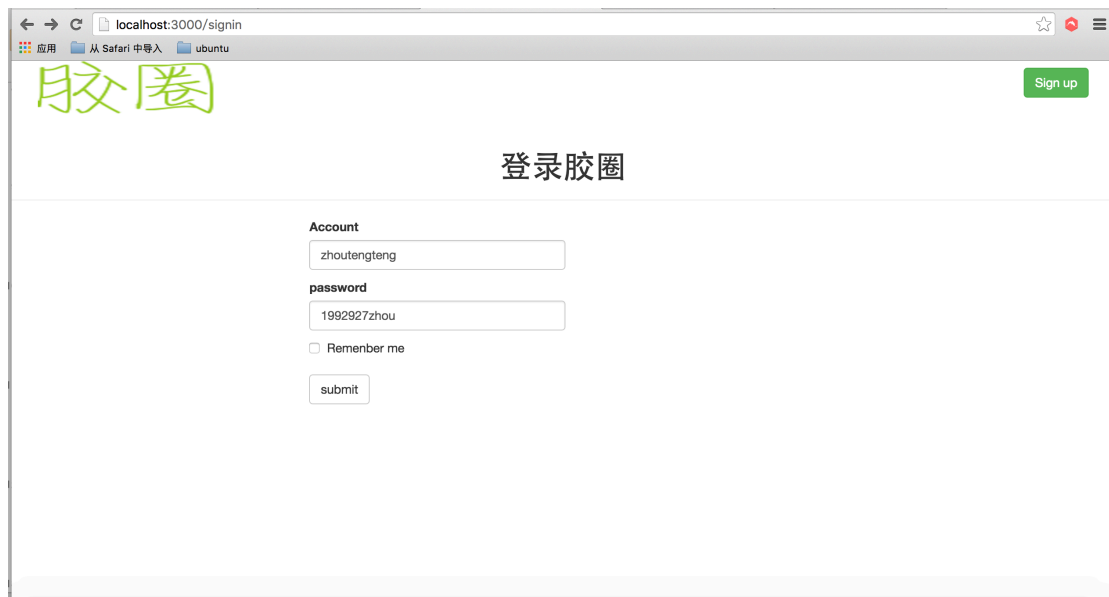
包括 bootstrap.min.css

jquery.min.js

bootstrap.min.js

```
app.get('/signin', require('./router/signin').getMethod);
```

```
app.post('/signin', require('./router/signin').postMethod);
```



对应的路由处理文件为/router/signin.js,

这个 get 路由是拉取注册界面/public/views/signin.jade,

当点击提交按钮, 触发/public/js/signin.js 中 ajax 的 post 请求, 同时会检测合理性, 然后从/router/signin.js 中调用 mysql 数据库中的 users_tb (即文件/db/user.js) 进行验证操作

如果成功会跳转到用户主页面

包括的文件有/public/views/signin.jade

/public/js/signin.js

/router/signin.js

/db/user.js

/public/views/header_only.jade

/public/js/header.js

加上 include ./includes/head.jade 包括的文件

其中 head.jade

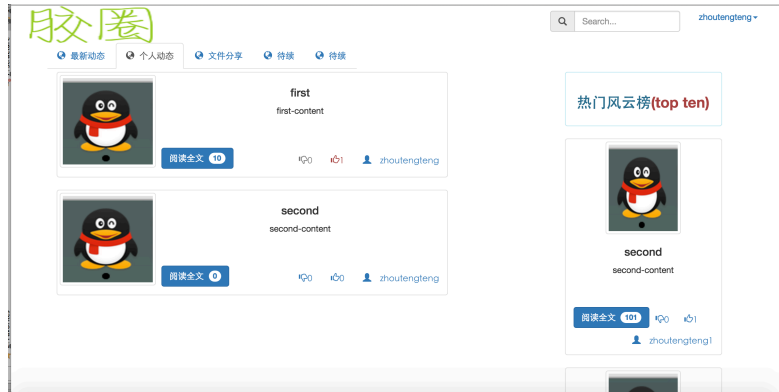
包括 **bootstrap.min.css**

jquery.min.js

bootstrap.min.js

app.get('/profile/[A-Za-z0-9]+', require('./router/profile').getMethod);

根据标签名进行选择返回的数据，然后用/public/js/profile.js 动态渲染生成 DOM 添加到网页中。



包括的文件有/public/views/profile.jade

/public/js/profile.js

/public/css/profile.css

/db/user.js

/db/complex.js

/db/item.js

/db/pic.js

/public/views/header_only.jade

/public/js/header.js

加上 include ./includes/head.jade 包括的文件

其中 head.jade

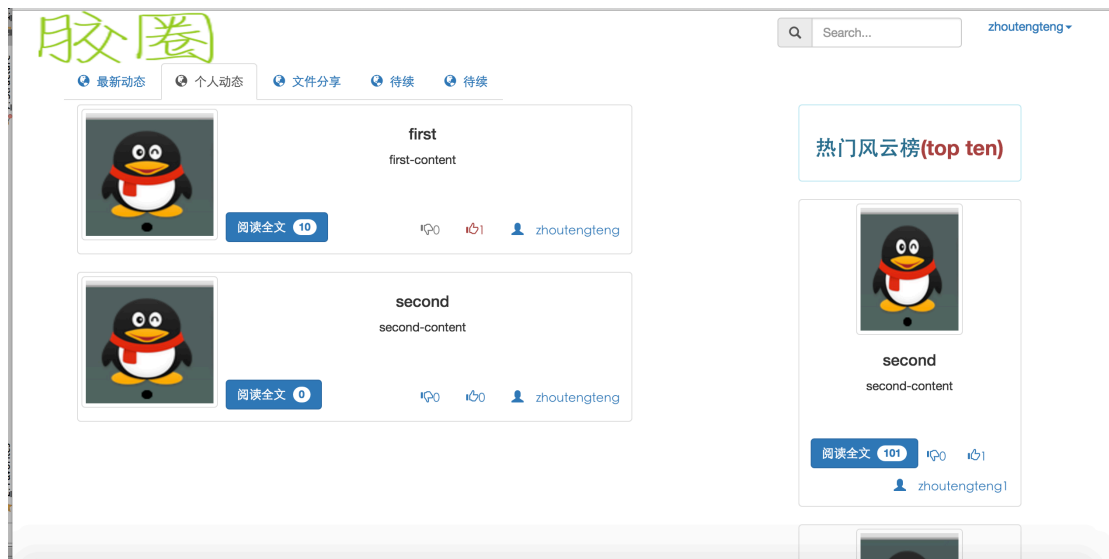
包括 **bootstrap.min.css**

jquery.min.js

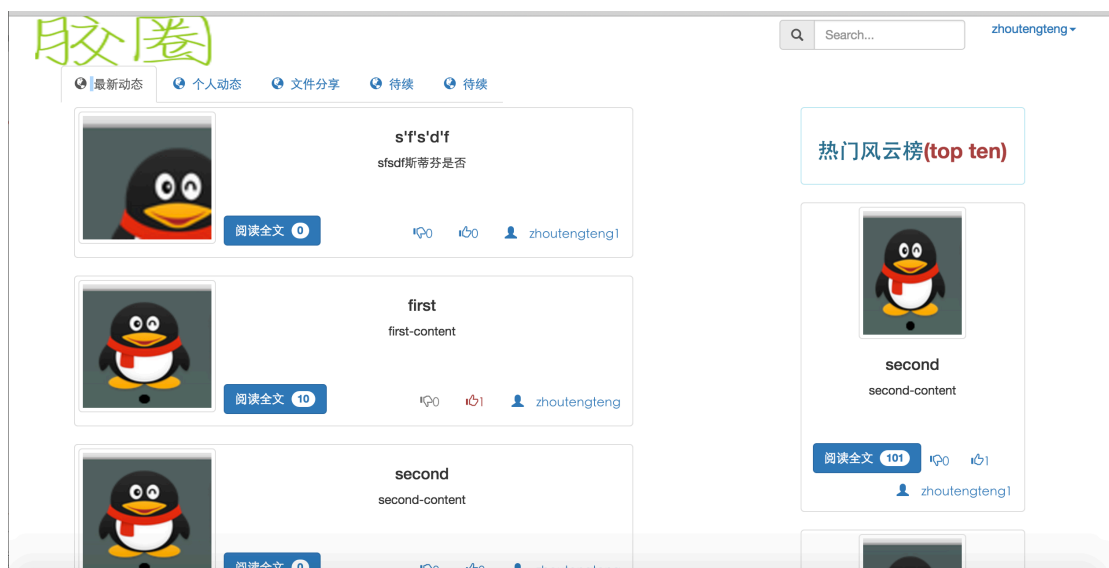
bootstrap.min.js

```
app.post('/profile/[A-Za-z0-9 ]+', require('./router/profile').postMethod);
```

用户主目录，对应的渲染文件为/public/views/profile.jade



这是个人动态（都是自己发布的）



这是最新动态（所有人的动态，按照时间顺序排序）

右边表示点击量最高的动态条目。我们可以点赞，可以阅读全文进行点赞。

个人动态和最新动态的切换，采用 jquery 的 ajax 的方法局部刷新，切换的 js 文件为/public/js/profile.js.通过 ajax 的 post 到路由/profile/[A-Za-z0-9]+

```
app.get('/circle/edit', require('./router/circle_edit').getMethod);
```



有主页 `profile.jade` 点击阅读全文按钮触发，操作数据库进行计数加一，同时返回渲染 `detail.js` 或者 `modify.js` 根据是否是登陆者是作业来决定

```
app.post('/logout', require('./router/logout').postMethod);
```



登录注销，点击届满中的 Logout，会触发 `/public/js/header.js` 中的 ajax 的 post，从而 `/router/logout.js` 清除客户端 cookies

```
app.get('/*', require('./router/validate').getMethod);
```

此路由为验证客户端 cookie，从而只让登录的人进行查看其它活动。相当于 get 网页之前，进行一层拦截

```
app.get('/circle/edit', require('./router/circle_edit').getMethod);
```

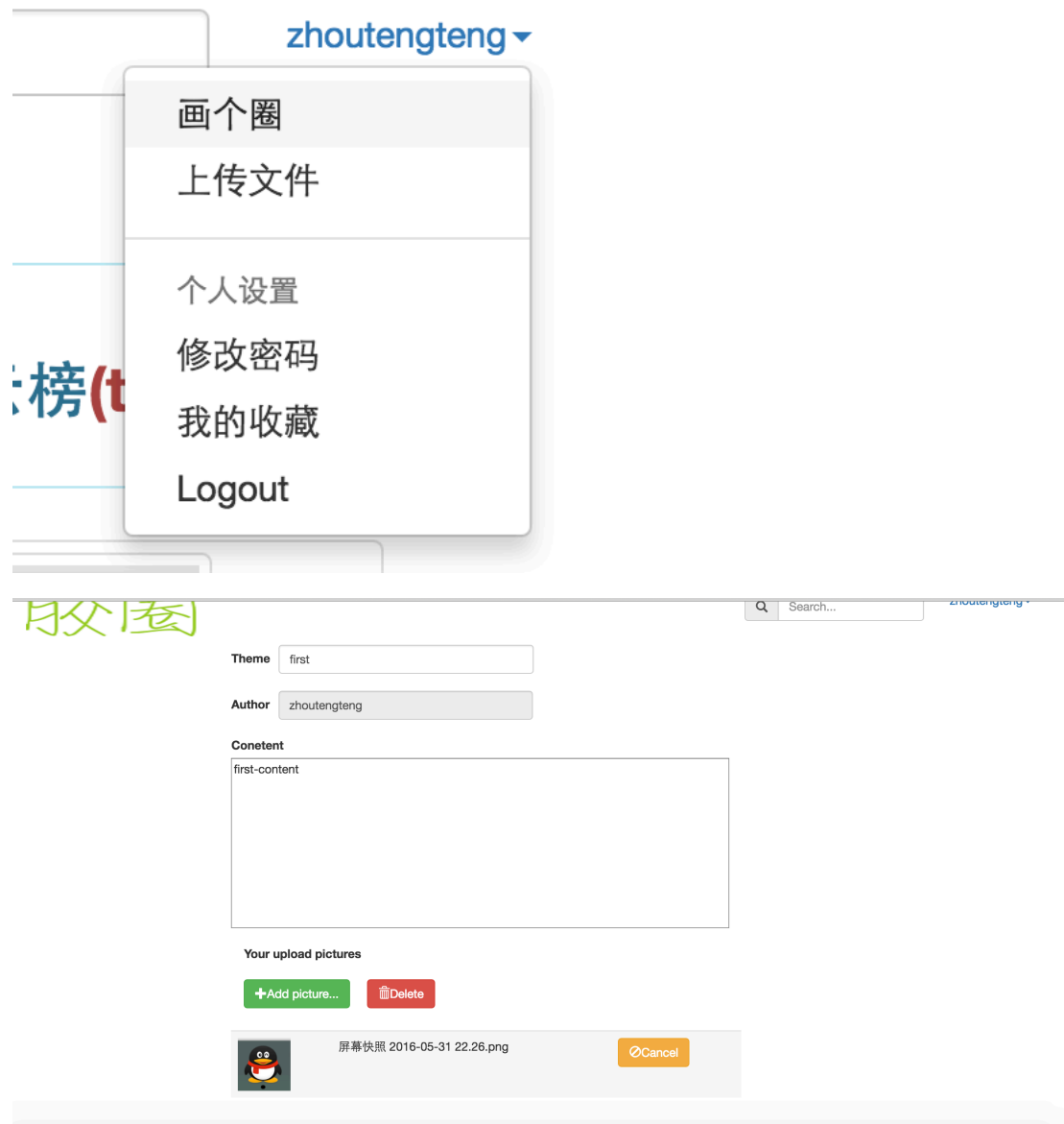
获得条目编辑界面/public/views/circle_edit.jade

```
app.post('/circle/edit', require('./router/circle_edit').postMethod);
```

提交编辑条目

```
app.post('/circle/update', require('./router/circle_edit').updateMethod);
```

更新条目



```
app.post('/circle/comment', require('./router/detail').postComment);
```

提交评论

```
app.post('/circle/comment/[0-9]+', require('./router/detail').getComment);
```

从服务器拉取评论

```
app.post('/count', require('./router/profile').click);
```

更新

您的评论

剩余可输入700字

提交评论

Comment

1# zhoutengteng	评论	sdfsdfsfs	评论
2# zhoutengteng	回复zhoutengteng	sdfs	评论
3# zhoutengteng	回复zhoutengteng	sdfs	评论

包括的文件有

- /public/views/circle_edit.jade
- /router/circle_edit.js
- /public/js/circle.edit.js
- /db/pic.js
- /db/user.js
- /db/issue.js
- /db/item.js
- /public/views/header.jade
- /public/js/header.js

加上 `include ./includes/head.jade` 包括的文件

其中 `head.jade`

包括 `bootstrap.min.css`

`jquery.min.js`

`bootstrap.min.js`

通过文件分析

1. bower_components/

nodejs 安装 bower 后，用 bower 安装 bootstrap 和 less（后面未使用）和 jquery

2. config/

config.js 配置端口，和可能的全局配置

dbconfig.js 连接 mysql 数据库的必要设置文件，方便使用数据库操作的时候，更方便

3. db/

complex.js 为从客户端 post 到来的“top10”和“最新动态”拉取数据，里面数据库中的多个表的操作

file.js 文件目录表操作，待做

issue.js 评论表操作，存放对应每个用户对条目的评论信息

item.js 条目表操作，用户发起的条目，都保存在这个表中

mysql.db 这是 mysql 的数据库原型语句，可以直接生成 mysql 数据库表

pic.js 图片表操作，存放图片的目录，存的是 base64

user.js 用户表操作，记录用户的信息，用户的新建，登录验证等等

readme.txt

4. lib/

这是 jquery 插件库，只用到了上传图片的界面 UI 文件，上传功能为自己手写

5. node_modules/

nodejs 下载的本地模块，可在 package.json 找到

6. public/

● css/

基本没用到自己的 css，留下这个可以扩展自己喜欢的样式

● js/

circle_edit.js

对应用户发起的条目的 js 文件，负责提交

detail.js

用户浏览其他的问的条目对应的前端文件的 js 文件，负责拉取题目信息和提

交评论

`edit.js`

多余文件

`header.js`

处理网页最顶层对应的用户退出和触发那些用户发起条目的链接

`index.js`

项目主页的展示信息

`modify.js`

用户浏览自己发起的条目的时候，会渲染的 `modify.jade`, 包括的 `modify.js` 文件负责提交更新条目。

`profile.js`

负责拉取最新动态，个人动态和 top10 等

`signin.js`

负责登录验证

`signup.js`

负责注册验证

`readme.txt`

- `public/`

`views/` 视图

`includes/` 一些公用的模板

`footer.jade` 网页脚部，未加，可能以后加

`head.jade` 网页头部引用的样式和 js

`header.jade` 网页头部视图

`header_only.jade` 网页头部的缩减视图

`readme.txt`

`circle_edit.jade` 编辑用户发起条目的视图

`detail.jade` 用户浏览其他人条目的视图

`edit.jade` 未使用

`index.jade` 项目主页的视图

layout_header_only.jade 引用模板的模板文件

layoutBasic.jade 引用模板的模板文件

modify.jade 用户修改自己发起的条目的视图文件

profile.jade 用户主页浏览条目的信息界面

signin.jade 登录界面

signnp.jade 注册界面

welcome.jade 未使用，忽略

readme.txt

- router/

circle_edit.js 对用户提交条目的路由处理

detail.js 对用户查看条目和用户评论的处理

edit.js 未使用

index.js 对主页的路由处理

logout.js 对用户退出的路由处理

profile.js 对用户主浏览页的路由处理

signin.js 对用户登录的路由处理

signup.js 对用户注册的路由处理

validate.js 对路由进行中间操作的路由处理，可拦截，（用与安全信息）

welcome.js 未使用

filter.js 未使用

- app.js

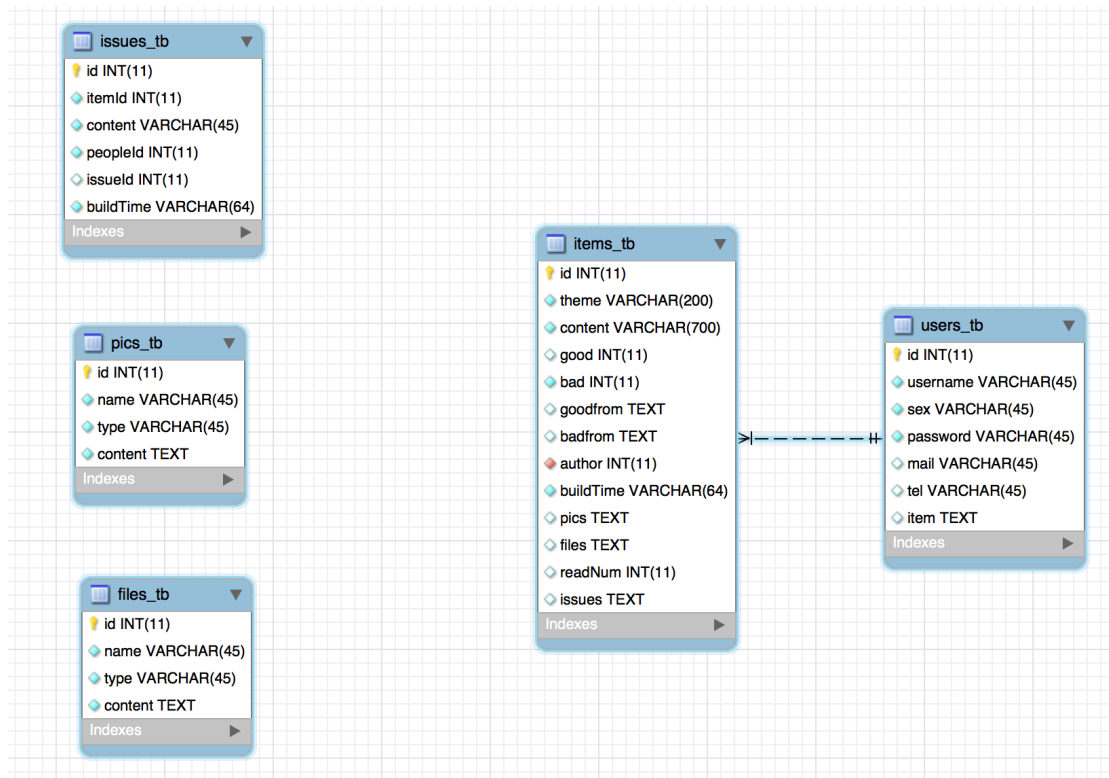
对项目的启动文件

- package.json

nodejs 包文件

README.md

4.数据库表



因为 items_tb 中的 pics，其实是引用多个 pics_tb 中的 id，然后用\t 分开，组成 TEXT 格式，所以没有单纯的外键什么的，可以看成一种无形的外键

同理队里 files_tb 和 issues_tb

然后 issues_tb 中的 issuId 是自己的外键