

综合实验-清软论坛

最后编辑时间：2020/08/22 20:35

1、作业简述

论坛是一个允许用户交流沟通的开放平台，百度贴吧、Reddit、NGA游戏社区、虎扑社区都是相对知名度较高的论坛。在本次综合练习中，我们将要求同学们熟练运用前端课程中所学的知识，搭建一个论坛的前端网页，使得用户可以使用该前端方便轻松地使用论坛服务。

为了合理控制工作量、便于评测，助教会提供一个公开的论坛后端-清软论坛，该后端具有登录、登出、发帖、回帖等论坛应具有的基本API。详细API细节会在文档后续部分中说明。在本次综合实验中，所有同学的论坛前端都应该依托于该后端进行。

每位同学会提供一个登录账号（与学号相同）以及相应的登录密码，以网络学堂作业的形式下发，请留意并注意保护自己的密码。

2、作业要求

基础要求（80分）：

- 使用Vue或React等现代前端Web框架脚手架（推荐使用Vue）如 `vue-cli` 或 `create-react-app` 实现一个论坛前端页面（在文档最后会以 `vue-cli` 为例简单介绍如何使用脚手架来创建一个前端工程并接入助教提供的后端），可以正确与助教提供的后端进行交互(推荐使用axios)。合理处理并提示用户后台返回的错误信息（30分）；
- 用户可以进行登录登出，应设计相应页面跳转与提示，在关闭浏览器后仍能保留登录状态（10分）；
- 用户可以浏览帖子列表（应正确显示标题、发帖人、发帖时间等信息，合理处理分页），并且可以主动发帖/回帖或编辑自己发布的帖子与回帖（10分）；
- 用户可以查看帖子详情以及相应的回帖，正确处理详情中的自定义格式，可以查看指定用户的发帖记录（10分）；
 - 对于帖子以及回帖，后台的记录都是一个string，如果想在其中添加自己的自定义格式，可以考虑参照富文本的实现，也就是使用HTML标签定义内容
 - 如果想实现HTML标签定义之外的内容，可以考虑使用自定义标签或者其他标记并在渲染时妥善处理。进阶要求中的部分项目就可以使用自定义标签或者其他标记实现。
 - 注意你只需要处理你自己定义好的内容，只要能实现“如果一个帖子是用你的前端发的，那么用你的前端查看这个帖子的时候，这个帖子能正确显示”就是对的。
- 可以正确显示不同用户的发帖记录，并实现合理的跳转（10分）；
- 页面应美观大方，符合现代主流网页审美，用户体验良好（10分）。

进阶要求（20分+5分）：

- 实现bbs中楼中楼的显示来区分回复主楼与回复某一楼层，并合理实现回帖列表的前端分页（5分）；
- 自定义标签来实现表情与代码段的插入（5分）；
- 实现外部图片链接的插入与合理展示（如长图）（5分）；

- 实现“只看楼主”、“收藏”、“浏览记录”功能（5分）；
- （额外加分项）其他可以前端实现的亮点，也可以是在某一基础或进阶要求上做出突出特色。要求工作量与其他进阶要求条目相当，具体分值由助教组评定，请注意在文档中说明你的亮点（1-5分）。

其他要求：

- 禁止上传不符合国家相关法律法规、攻击侮辱他人或其他对他人造成困扰的内容；
- 上传富文本内容不得包含可执行js代码来进行xss攻击；
- 每位同学应妥善保管自己的密码，不得交予他人使用；
- 发现上述行为，本次综合实验计0分。

3、提交说明

每位同学应当提交一个zip压缩包，命名格式为 `xxx_201801xxxx.zip`（例：清小软_2018010001.zip）。压缩包内容包含一个code文件夹容纳作业代码以及一份pdf或markdown格式的说明文档。

代码应当包含所有必要文件，至少保证助教能够使用该代码顺利启动工程。同时注意剔除 **node_modules**、**dist**等不必要的内容。脚手架生成的项目中 `.gitignore` 文件内提到的文件均不应出现的提交代码包中。

说明文档应包括但不限于这些内容：实验说明、启动/编译方式说明、功能实现情况、你所认为的亮点与难点。鼓励大家在文档中附有截图说明。同时，也需要提交至少一个帖子URL，用于评判你的帖子详情显示。建议同学们在提交前创建一个帖子预先编辑部分内容来展示自己的工作。助教只会评测助教本人使用作业提交前端发出以及当前待评测同学给出相关的帖子与其中回复，无需处理其他人所发帖子与回复中的标签等自定义信息。

请留意，不合规的提交会被酌情扣分。

4、API列表

本章会列举助教所提供的所有API信息，使用这些API即可实现全部实验要求。若没有特殊说明，所有Request和Response Body类型均为 `application/json`。API可能会存在测试不充分的情况，如遇bug请在课程群内联系助教，助教会妥善处理。

登录状态使用Json Web Token记录，在需要登录身份的API Request中(除"测试API-无身份验证"与"登录")，应该在 Request Header中增加Authorization字段，其值为登录时返回的jwt token值。同学们应当自行考虑如何保存登录时返回的token值。在本次实验中，token的有效期为24小时，登出操作会使得token过期。当登录状态不合规或过期时，后台会返回HTTP Error 401，请同学们注意处理。关于JWT的相关介绍以及原理，可以参照后附的相关链接。

论坛后端域名：simplebbs.iterator-traits.com

例如，下面一个API的地址是：<http://simplebbs.iterator-traits.com/api/v1/hello>

测试API-无身份验证：

```
Method: GET
URL: /api/v1/hello
Response:
{
  'message': 'Hello world'
}
```

测试API-有身份验证

```
Method: GET
URL: /api/v1/hello-user
Response:
{
  'message': 'Hello world, USERNAME' # USERNAME是登录者的学号
}
```

登录:

```
Method: PATCH
URL: /api/v1/login
Request:
{
  'username': '',
  'password': ''
}
Response:
{
  'username': '',
  'nickname': '',
  'jwt': ''
}
```

登出:

```
Method: PATCH
URL: /api/v1/logout
Response:
{
  'message': 'ok'
}
```

获取当前登录账号个人信息

```
Method: GET
URL: /api/v1/user
Response:
{
  "id": 1,
  "username": "201801****",
  "nickname": "清小软",
  "created": "2020-08-14T23:15:49+08:00"
}
```

获取用户昵称

特别注意:

此API为后续版本新增API, 同学们可根据兴趣选用, 不计入评分。

```
Method: GET
URL: /api/v1/user/:userId
Response:
{
  "id": 1,
  "nickname": "清小软",
  "created": "2020-08-14T23:15:49+08:00"
}
```

获取帖子列表(更新时间降序):

特别注意:

orderByReply参数以及返回的lastRepliedUserId、lastRepliedNickname、lastRepliedTime字段为后续增补API, 同学们可根据兴趣选用, 不计入评分。

```
Method: GET
URL: /api/v1/post
QueryParam:
{
  'page': 1, # 可选, 默认获取第一页
  'size': 10, # 可选, 默认为10
  'userId': 1, # 可选, 默认获取所有用户帖子
  'orderByReply': true # 可选, 默认表示按照主贴更新时间降序, 为true表示按照最新回复时间降序 (不评分)
}
# Example: GET /api/v1/post?page=1&size=10&orderByReply=true
Response:
{
  'page': 1,
  'size': 10,
  'total': 1,
  'posts': [
```

```
{
  "id": 1,
  "userId": 1,
  "nickname": "清小软",
  "title": "hello, world", # 帖子标题
  "content": "welcome to simplebbs", # 帖子内容, 为富文本
  "lastRepliedUserId": 2, # 最新回复用户id, 默认为发帖人, 非评分要求字段
  "lastRepliedNickname": "清大软", # 最新回复用户昵称, 默认为发帖人, 非评分要求字段
  "lastRepliedTime": "2020-08-22T20:18:19+08:00", # 最新回复时间, 非评分要求字段
  "created": "2020-08-14T00:00:00+08:00",
  "updated": "2020-08-14T00:00:00+08:00"
}
```

发帖:

```
Method: POST
URL: /api/v1/post
Request:
{
  "title": "欢迎使用清软论坛", # 帖子标题
  "content": "请同学们畅所欲言" # 帖子内容
}
Response:
{
  'message': 'ok'
}
```

编辑当前用户发布的帖子:

```
Method: PUT
URL: /api/v1/post/:postId
# :postId 为帖子的id, 需要是本人所发帖才能修改
Request:
{
  "title": "欢迎使用清软论坛", # 帖子标题
  "content": "请同学们畅所欲言!" # 帖子内容
}
Response:
{
  'message': 'ok'
}
```

获取帖子详情与回帖列表:

```
Method: GET
```

```
URL: /api/v1/post/:postId
# :postId 为帖子的id
Response:
{
  "id": 1,
  "userId": 1,
  "nickname": "清小软",
  "title": "hello, world", # 帖子标题
  "content": "welcome to simplebbs", # 帖子内容
  "created": "2020-08-14T00:00:00+08:00",
  "updated": "2020-08-14T00:00:00+08:00",
  "lastRepliedTime": "2020-08-22T20:18:19+08:00", # 最新回复时间, 非评分要求字段
  "reply": [ # 回帖列表, 创建时间升序
    {
      "id": 1,
      "userId": 1,
      "nickname": "清小软",
      "postId": 1,
      "replyId": 0, # 回复主帖
      "content": "Hello, Everyone!",
      "created": "2020-08-15T11:31:41+08:00",
      "updated": "2020-08-15T11:31:41+08:00"
    },
    {
      "id": 2,
      "userId": 1,
      "nickname": "清小软",
      "postId": 1,
      "replyId": 1, # 回复上一条回帖
      "content": "reply to the first reply",
      "created": "2020-08-15T11:33:23+08:00",
      "updated": "2020-08-15T11:33:23+08:00"
    }
  ]
}
```

回帖:

```
Method: POST
URL: /api/v1/post/:postId/reply
# :postId 为回复帖子的id
Request:
{
  "content": "", # 回帖内容
  "replyId": 1 # 可选, 回复目标回复Id, 不提供表示回复主楼
}
Response:
{
  'message': 'ok'
}
```

编辑当前用户发布的回帖:

```
Method: PUT
URL: /api/v1/post/:postId/reply/:replyId
# :postId 为回复帖子的id
# :replyId 为修改回复的id
Request:
{
  content": "", # 修改回帖内容
}
Response:
{
  'message': 'ok'
}
```

5、参考链接

[Axios文档](#)

[JWT文档](#)

6、vue-cli使用

[官方中文文档](#)

Install

```
npm install -g @vue/cli
# OR
yarn global add @vue/cli
# 使用国内镜像加速安装
npm install -g @vue/cli --registry=https://registry.npm.taobao.org
```

Create project

```
# 名字可以自己起一个，这里以simplebbs-frontend为例
```

```
vue create simplebbs-frontend
```

注：Vue3目前还没有正式Release版本但已经处于RC阶段，建议大家先使用Vue2，有兴趣的同学也可以尝试Vue3的新feature

Add Config

同学们需要在 `package.json` 平级添加一个文件 `vue.config.js`，内容如下：

```
module.exports = {
  devServer: {
    proxy: {
      '/api': {
        target: 'https://server_url',
        ws: true,
        changeOrigin: true
      }
    }
  }
}
```

其中target的值应该是助教所提供的后端域名，这样同学们在发送请求时就可以用形如

`Axios.get('/api/v1/hello')` 来请求到相应的内容。dev server会将带有api前缀的请求转发到助教提供的后端，来避免直接请求带来的CORS问题。

Start

```
cd simplebbs-frontend
npm run serve
```

如果看到如下内容，表示启动dev server成功，可以在浏览器中打开<http://localhost:8080/>查看

```
DONE   Compiled successfully in 2563ms

App running at:
- Local:   http://localhost:8080/
- Network: unavailable

Note that the development build is not optimized.
To create a production build, run npm run build.
```

Enjoy!