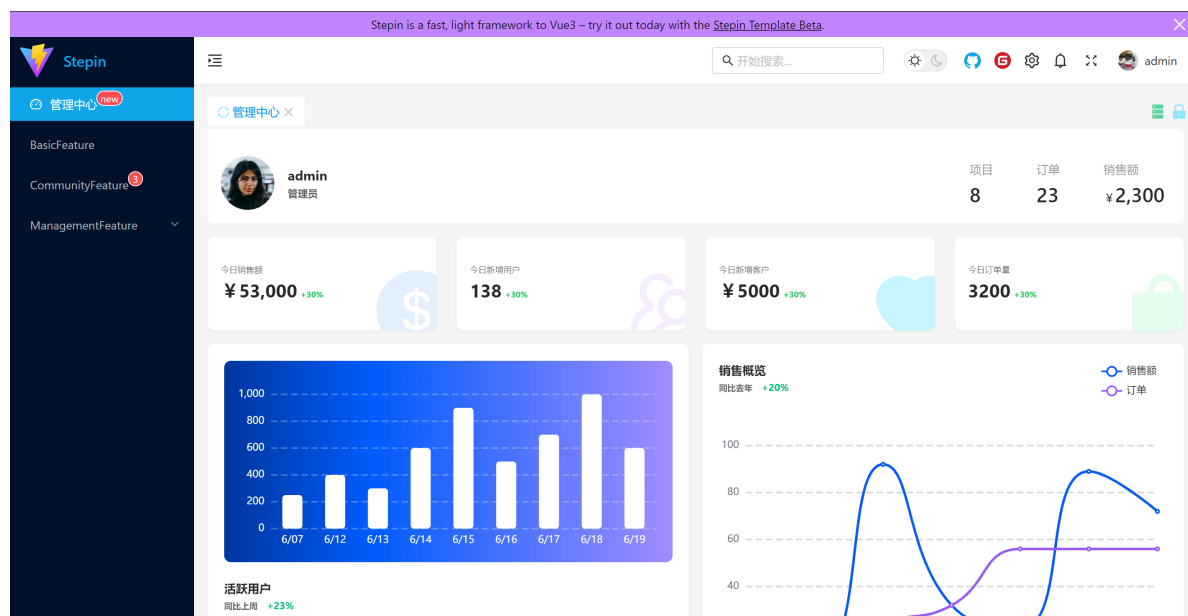


管理员界面部分说明&用户身份确认

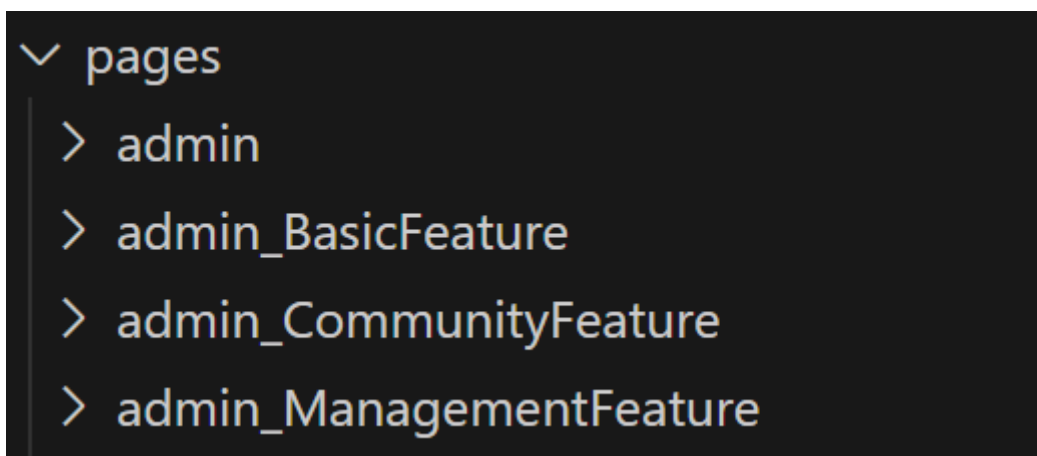
页面说明

管理员用户名: admin 密码: 888888 (六个8)

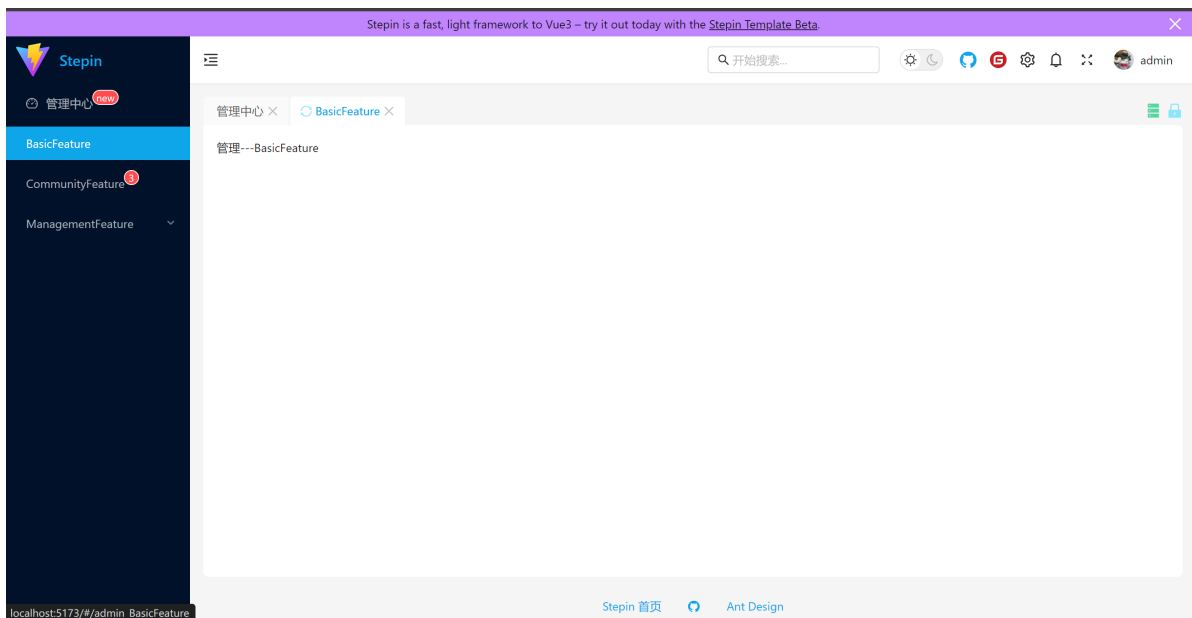
目前只设置了一个, 如果有需要再改或者后端手动添加权限即可。



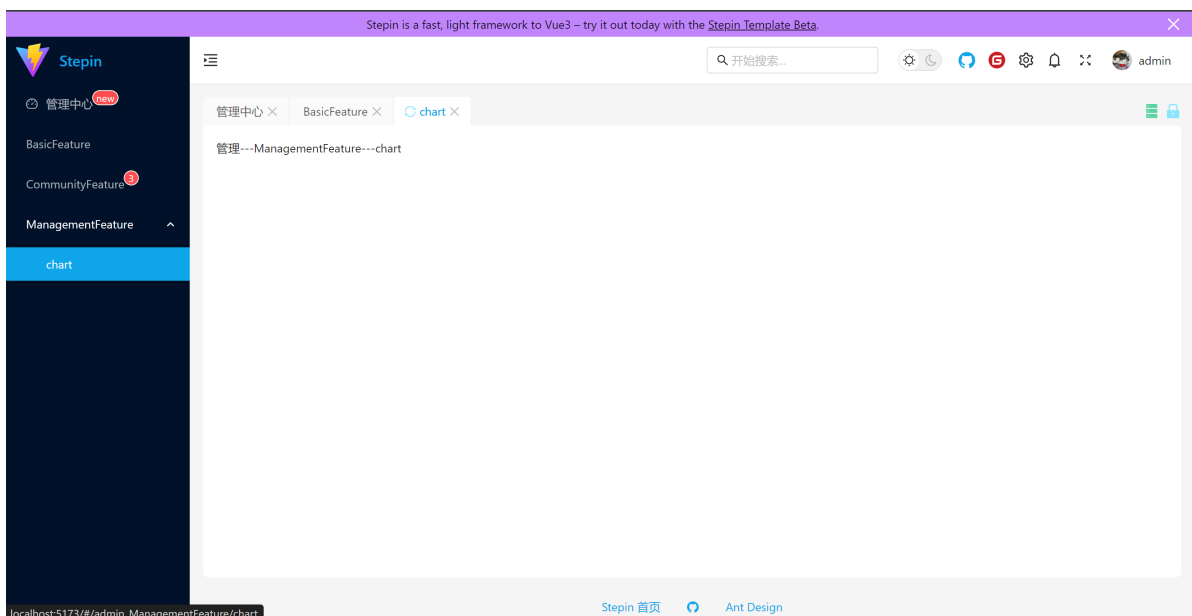
登录成功后有三个区域（管理中心沿用了之前的工作台），每个区域新建了以 `admin_` 为开头的文件夹作为区分，按照cjh之前的划分方法，大家自行添加内容。



可以直接添加在页面里：



也可以添加子页面（方法见 [docs/router.md](#)）：



代码说明

权限分配

我们使用的模板有一套权限分配逻辑，具体在 [docs/authority.md](#) 里。管理员已经配置了'admin'权限，可以访问包含admin权限的页面。分配方法如下（用户的页面同样适用）：

1. 先在 **menu.js** 里的页面中设置相应的permission。

```

{
  id: 13,
  name: 'adminCenter',
  title: '管理中心',
  icon: 'DashboardOutlined',
  badge: 'new',
  target: '_self',
  path: '/admin',
  component: '@/pages/admin',
  renderMenu: true,
  parent: null,
  permission: 'admin',
  cacheable: true,
},

```

2. 在account.ts里给指定用户分配权限

目前是在前端根据用户名直接手动分配的，比较推荐后端能直接返回一个permissions数组，包含所有权限，不要在前端一点点设置

```

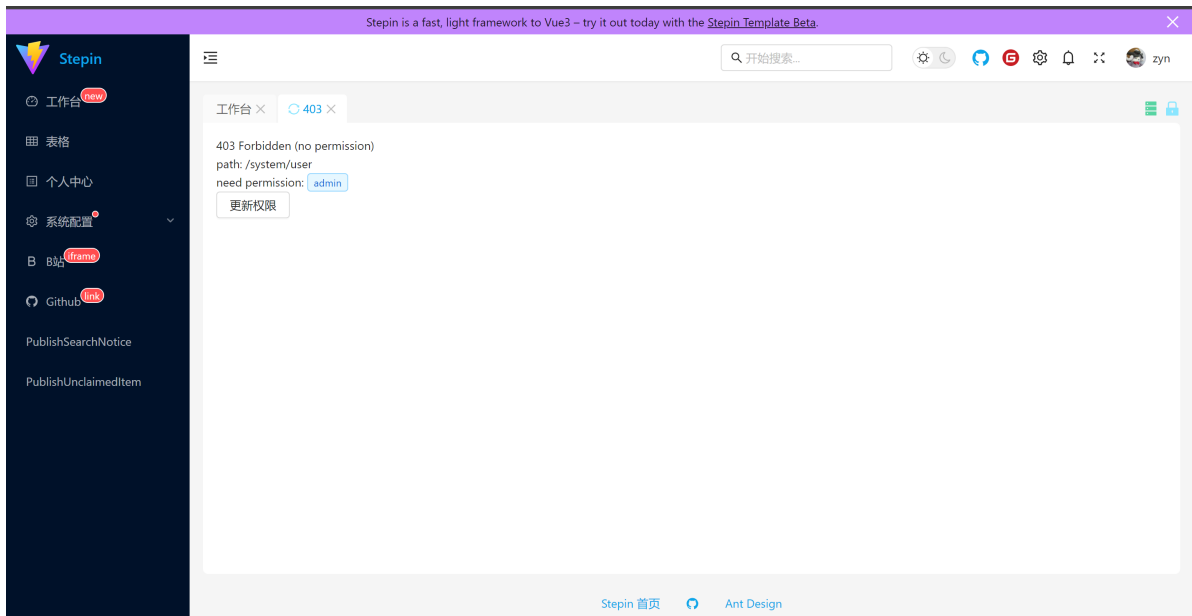
if(username === 'admin'){
  this.permissions.push('admin');
}
else{
  this.permissions.push('user');
}

```

3. 用setAuthorities插件添加权限

```
useAuthStore().setAuthorities(this.permissions);
```

github的分支里这一页没设置权限，只演示：



菜单设置

直接给管理员在 `nemu.js` 里设置了一个新的菜单，登录的时候根据权限分配的不同加载不同菜单，名称请自行更改~

```
const adminList = [
  {
    id: 3,
    name: 'personal',
    title: '个人中心',
    path: '/personal',
    icon: 'ProfileOutlined',
    permission: null,
    component: '@/pages/personal',
    renderMenu: false,
    parent: null,
  },
  {
    id: 13,
    name: 'adminCenter',
    title: '管理中心',
    icon: 'DashboardOutlined',
    badge: 'new',
    target: '_self',
    path: '/admin',
    component: '@/pages/admin',
    renderMenu: true,
    parent: null,
    permission: 'admin',
    cacheable: true,
  },
  {
    id: 14,
    name: 'BasicFeature',
    title: 'BasicFeature',
    icon: '',
```

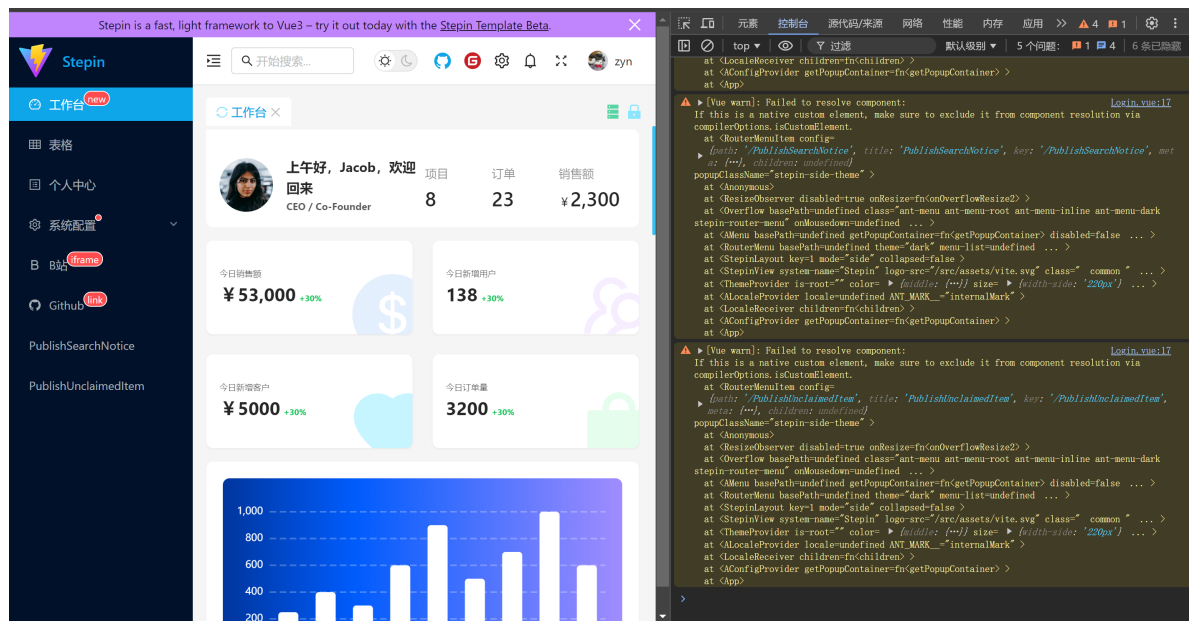
```

        badge: null,
        target: '_self',
        path: '/admin_BasicFeature',
        component: '@/pages/admin_BasicFeature',
        renderMenu: true,
        parent: null,
        permission: 'admin',
        cacheable: true,
    },
    {
        id: 15,
        name: 'CommunityFeature',
        title: 'CommunityFeature',
        icon: '',
        badge: 3,
        target: '_self',
        path: '/admin_CommunityFeature',
        component: '@/pages/admin_CommunityFeature',
        renderMenu: true,
        parent: null,
        permission: 'admin',
        cacheable: true,
    },
    {
        id: 16,
        name: 'ManagementFeature',
        title: 'ManagementFeature',
        icon: '',
        badge: null,
        target: '_self',
        path: '/admin_ManagementFeature',
        component: '@/pages/admin_ManagementFeature',
        renderMenu: true,
        parent: null,
        children : [
            {
                id: 17,
                name: 'chart',
                title: 'chart',
                icon: '',
                badge: null,
                target: '_self',
                path: '/admin_ManagementFeature/chart',
                component: '@/pages/admin_ManagementFeature/chart',
                renderMenu: true,
                parent: 'ManagementFeature',
                permission: 'admin',
                cacheable: true,
            },
        ],
        permission: 'admin',
        cacheable: true,
    },
];

```

ps: 如果要添加在登录之后显示的页面（不论是管理员还是用户），不要直接在 `routes.ts` 里添加，添加到 `menu.js` 更方便做个性化处理。

登录之后有warning：路由还没有注册，不用管，确认之后再统一注册即可。



之前提到的用户身份确认问题：

在 `account.ts` 里有以下属性，通过直接调用store可以获得（role和logged如果没用可以删掉）。

```
account: {
  userName: '',
  userId: '',
  contact: '',
} as Account,
permissions: [],
role: '',
logged: true,
```

调用：

```
import { useAccountStore } from '@/store/account';
const {account, permissions} = useAccountStore();
account.userName //获取用户名
permissions.include('admin') //检查是否拥有某一权限
```

也可以在需要单独获取用户全部信息，或者登录之后有信息更改要更新的时候调用 `profile()` 函数（在登录的时候已经调用了，没有信息更新可以直接调用`accountStore`里的某一属性，有更新的话建议调用下）：

```
import { useAccountStore } from '@/store/account';
const {profile} = useAccountStore(); //或者用const accountStore = useAccountStore();
profile(); //或者accountStore().profile();
```

同样的，目前只根据用户名获取ID和contact，建议能有permissions的返回。

