Vue2Project DAY09

实现排片计划列表业务

需求： 在放映厅列表页中，点击小眼睛， 跳转到排片计划列表页面。显示当前已经配置好的排片计划。

实现思路：

1. 准备一个排片计划列表页面：`cinema/ShowingonPlanList.vue`，配置路由，使得访问：`/home/showingon-play-list?id=xxxx`可以看到它。传递放映厅的ID。
2. 在排片计划列表页面中，显示放映厅的基本信息。
3. 通过放映厅ID，加载并显示排片计划列表即可。

控制排片计划的发布状态

通过两个接口来处理发布状态：

```
publish
no-publish
```

请求响应的统一业务异常处理

```
// 针对instance处理统一的业务异常    基于响应拦截器
instance.interceptors.response.use((response)=>{
  // 统一异常处理  400业务状态码
  if(response.data.code==400){
    let errmsg = response.data.msg.details[0].message
    console.warn('请求参数有误，请检查:', errmsg)
    Notification.error({
      title:'注意',
      message:'系统开小差了，等会试试吧！'
    })
  }
  return response;
})
```

设计描述座位模板的字符串：

```
[
    [
        {x:0,y:0,name:'一排一号',type:'1'}, {x:0,y:1}, {}, {}.....
    ],
    [],
    [],
    .....
]
```

```
AAAAAAAAANNAAAAA
AAAAAAAAANNAAAAA
AAAAAAAAANNAAAAA
AAAAAAAAANNAAAAA
NNNNNNNNNNNNNNNN
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA
```

实现登录业务

需求： 进入登录页面，输入账号密码，执行登录业务。成功后，进入首页。不登录不可看到项目内部的页面。不同权限的账号，看到的内容应该是不一样的。

实现步骤：

1. 准备一个登录页面：`views/user/Login.vue`。配置路由：

   /user/login时，看到该登录页面。

   ```
   insert into admin
       (username, password, nickname, phone, email)
   values
       ('admin', MD5('123456'),'管理员亮亮','13312341234', 'liang@qq.com');
   ```

2. 测试接口。
3. 执行登录业务，登录成功后，跳转到首页。

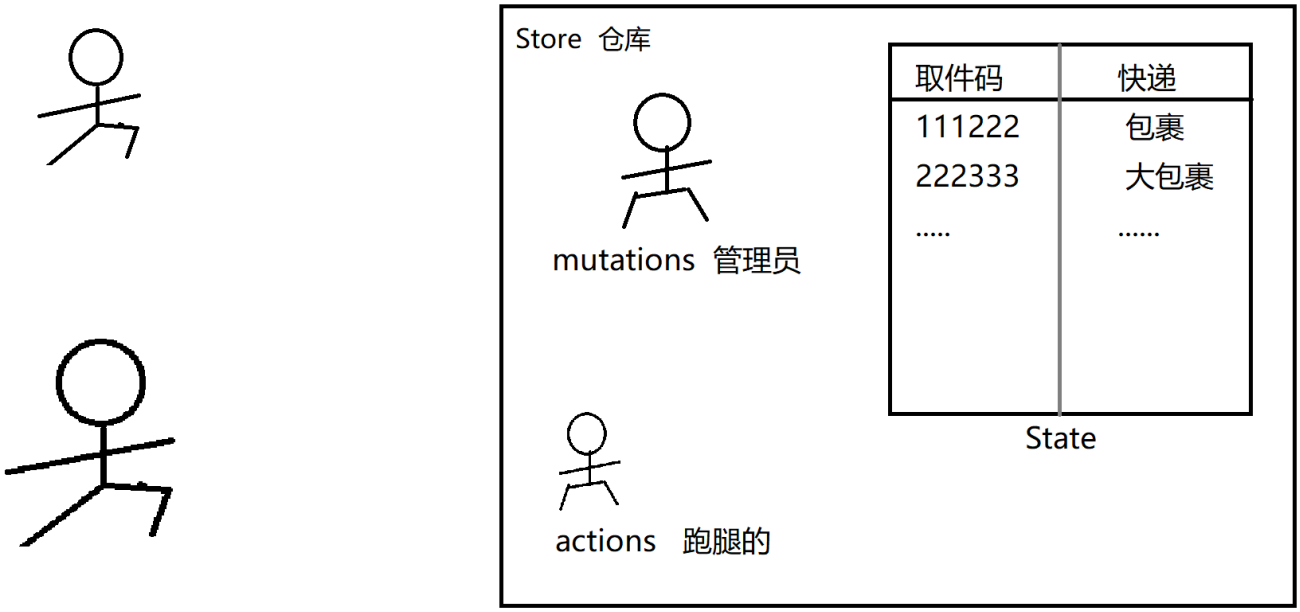
登录成功后，在首页显示用户信息

核心需求：将在登录页面中获取到的登录成功信息（用户数据），在首页中显示。（实现跨组件的数据共享）。此处可以使用Vuex来进行用户信息的管理。

Vuex Vue官方推荐的全局状态管理器

在脚手架项目的实现过程中，避免不了需要将一些数据存入“全局共享区域”，方便在任何组件中直接获取并且使用。这种需求就非常契合 `vuex` 的功能。 `vuex` 主要用于实现将一些数据全局存储，需要使用的使用直接访问 `vuex` 即可获取这些数据。

https://v3.vuex.vuejs.org/zh/guide/state.html



Vuex的核心概念

```
new Vuex.Store({
  state: {    },
  getters: {   },
  mutations: {   },
  actions: {    },
  modules: {    }
})
```

1. **state：单一状态树。用于存储全局共享的数据信息。**

   ```
   state: {
       cityname: '北京',
       user: {
           username: 'admin',
           nickname: '管理员·亮'
       }
   }
   ```

   在组件中，使用如下代码即可访问vuex中state里定义的状态数据：

   ```
   this.$store.state.cityname
   this.$store.state.user.nickname
   ```

2. **mutations: 用于声明一些修改state数据的方法**

```
mutations: {
    updateUser(state, payload){
        state.user = payload
    }
},
```

需要修改user时，可以通知vuex，执行updateUser方法，更新用户信息：

```
this.$store.commit('updateUser', user)
```

3. actions：用于定义一些方法，执行异步任务后得到结果，让后更新state。（这次更新state，需要调用mutations中声明的方法更新state，而不是直接更新）。

```
actions: {
    login(store, payload){
        // 此处 发登录请求  payload就是传来的账号与密码
        // 得到登录结果后，将user对象，更新state.user
        store.commit('updateUser', user对象)
    }
}
```

如下代码，即可通知vuex执行actions：

```
this.$store.dispatch('login', {username:'admin', password:'123456'})
```