

Vue3Project DAY03

当滚动条滚到底后，自动加载当前类别的下一页数据（触底加载）

```
<van-list v-if="movies && movies.length>0"
  v-model:loading="loading"
  :finished="finished"
  finished-text="没有更多了"
  @load="onLoad">

  <movie-item :movie="item"
    v-for="item in movies" :key="item.id">

  </movie-item>

</van-list>
```

移动端web应用的缓存设计方案

当应用需要数据时，就应该发请求，找服务端获取。但是有一些数据：它们的数据量不小，全局基本没有什么变化，访问量还挺频繁，这些数据如果每次都发请求去服务端获取，将造成服务端不小的压力。这时就可以考虑以下使用**缓存方案**来优化程序。

客户端缓存的业务流程：

1. 客户端发送请求，获取这一组数据，拿到数据后，记得把数据向缓存中存一份。
2. 但下次发送请求时，先去缓存中找一找，以前存过没有；存过的话就直接拿来用。如果没有存过再发请求。

考虑缓存何时更新？

如果缓存永远不更新，则每次从缓存中获取的数据都是旧数据。所以应该给应用的数据提供一个更加合理的缓存更新机制。

1. 将缓存存入sessionStorage（关了浏览器就没了），下次访问时就是新数据。
2. 将缓存存入localStorage（永久存储），每次加载时判断是否符合缓存更新的条件，如果符合则更新缓存。例如：
 1. 音乐列表场景下，**上一次缓存的数据是不是当天早上十点以后**
 2. 定时器：每隔5分钟更新一次。
 3. **实现列表的下拉刷新。**

实现下拉刷新，更新缓存

```
<van-pull-refresh success-text="加载完成"
  :success-duration="1500"
  v-model="refreshing">
```

```

        @refresh="onRefresh">
<van-list v-if="movies && movies.length>0"
    v-model:loading="loading"
    :finished="finished"
    finished-text="我是有底线的"
    @load="onLoad">

    <movie-item :movie="item"
        v-for="item in movies" :key="item.id">
    </movie-item>

</van-list>
</van-pull-refresh>

```

制作全局等待框

在移动端app中的每一个请求都应该弹出等待框，获取响应后让等待框消失。

基于拦截器实现：

```

/** 设置请求拦截器，每次请求弹出等待框 */
instance.interceptors.request.use((config)=>{
    console.log('loading...')
    // 每次发请求，都会执行该请求拦截器
    showLoadingToast({
        duration:0,           // 持续时长设置为0，永不自动消失
        message: '加载中...', // 弹窗文案
        forbidClick: true     // 是否禁用背景点击
    })
    return config;
})
/** 在响应拦截器中，关闭弹窗 */
instance.interceptors.response.use((config)=>{
    closeToast()
    return config;
})

```

实现电影详情页的显示

业务需求：当用户在首页点击其中某一个列表项时，跳转到详情页。并且携带选中项的电影ID一起跳转。到详情页后显示该电影的详细信息。

实现步骤：

1. 准备好一个电影详情页的静态页面：`views/index/MovieDetail.vue`。
2. 配置路由：当访问：`/movie-detail/:id` 时，跳转到详情页。
3. 在详情页中，接收参数ID，带着id发请求，加载电影详情数据。
4. 完成页面的数据展示。

git workflow

<https://pan.baidu.com/s/1eqTPvU39a1D99Z4SW0NKTQ>

weth