目录

[谈谈你做过什么数据处理相关的操作 1](#_Toc58780342)

[讲一下Cookie是Http哪个字段 1](#_Toc58780343)

[三、缓存常用字段 1](#_Toc58780344)

[Redux 2](#_Toc58780345)

[二面：ajax、axios 等和后台交互的问题 3](#_Toc58780346)

[二分查找、快速排序 3](#_Toc58780347)

[写一个正则匹配不同的文件类型 3](#_Toc58780348)

手写正则表达式判断电话号码

vue3.0新增了什么

手写快排（有延伸）

手写斐波那契 递归 动态规划（有延伸）

# 谈谈你做过什么数据处理相关的操作

# 讲一下Cookie是Http哪个字段

三、缓存常用字段

1、http1.0时期的缓存方案

注意：

（1）如果使用了Pragma: 'no-cache'的话，再设置Expires或者Cache-Control，就没有用了，说明Pragma的权值比后两者高。

 （2）如果设置了Expires之后，客户端在需要请求数据的时候，首先会对比当前系统时间和这个Expires时间，如果没有超过Expires时间，则直接读取本地磁盘中的缓存数据，不发送请求。

2、http1.1 时期的缓存方案

2.1、Cache-Control 字段

2.1.1、Cache-Control 作为请求头字段

（1）Cache-Control: no-cache

使用no-cache指令的目的是为了防止从缓存中返回过期的资源。 客户端发送的请求中如果包含 no-cache 指令，则表示客户端将不会接收缓存的资源。每次请求都是从服务器获取资源，返回304。

（2）Cache-Control: no-store

使用no-store 指令表示请求的资源不会被缓存，下次任何其它请求获取该资源，还是会从服务器获取，返回 200，即资源本身。

2.1.2、Cache-Control 作为响应头字段

Cache-Control: public

当指定使用 public 指令时，则明确表明其他用户也可利用缓存。

 Cache-Control: private

当指定 private 指令后，响应只以特定的用户作为对象，这与 public 指令的行为相反。 缓存服务器会对该特定用户提供资源缓存的服务，对于其他用户发送 过来的请求，代理服务器则不会返回缓存。

 Cache-Control: no-cache

如果服务器返回的响应中包含 no-cache 指令，每次客户端请求，必需先向服务器确认其有效性，如果资源没有更改，则返回304.

Cache-Control: no-store

不对响应的资源进行缓存，即用户下次请求还是返回 200，返回资源本身。

Cache-Control: max-age=604800（单位：秒）

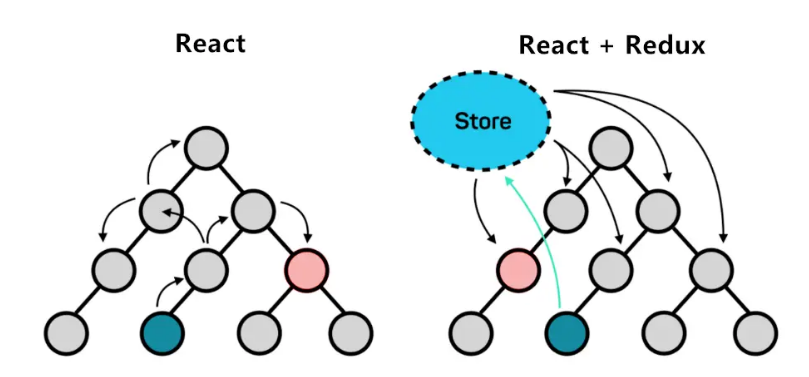
资源缓存在本地浏览器的时间，如果超过该时间，则重新向服务器获取。

# Redux

问题来了，React 中一个组件里面维护数据只需要 state 和 setState 就可以轻松搞定。假如多个组件都需要维护这一份数据怎么办呢？

Redux 有一个非常核心的部分就是 Store，Store 中管理的数据独立于 React 组件之外，如果 React 某个组件中的某个数据在某个时刻改变了（可以称之为状态改变了），就可以直接更改这个 Store 中管理的数据，这样其他组件想要拿到此时的数据直接拿就行了，不需要传来传去。

Store 通常要和 Reducer 来配合使用，Store 存数据，Reducer 是个纯函数，它接收并更新数据。



webpack用过么？原理明白吗？为什么 可以 import css文件

ES6、7、9

react开发用的是hook么？ 不是，那说一下生命周期

react-router如何工作？如果输入 '/a/b/c/index.html'应该怎么处理

# 二面：ajax、axios 等和后台交互的问题

# 二分查找、快速排序

# 写一个正则匹配不同的文件类型