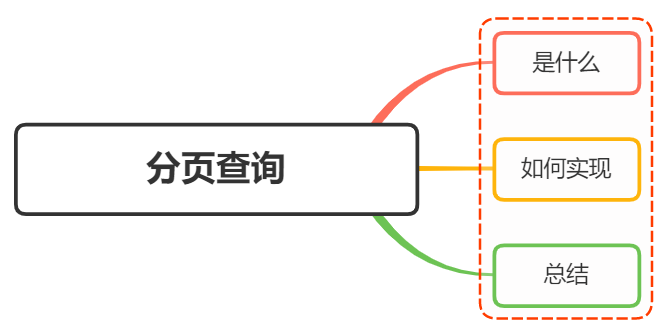
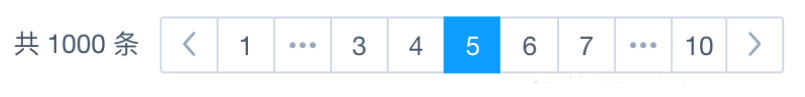
# 面试官：如果让你来设计一个分页功能, 你会怎么设计? 前后端如何交互?



## 一、是什么

在我们做数据查询的时候，如果数据量很大，比如几万条数据，放在一个页面显示的话显然不友好，这时候就需要采用分页显示的形式，如每次只显示10条数据



要实现分页功能，实际上就是从结果集中显示第110条记录作为第1页，显示第1120条记录作为第2页，以此类推

因此，分页实际上就是从结果集中截取出第M~N条记录

## 二、如何实现

前端实现分页功能，需要后端返回必要的数据，如总的页数，总的数据量，当前页，当前的数据

{  
 "totalCount": 1836, // 总的条数  
 "totalPages": 92, // 总页数  
 "currentPage": 1 // 当前页数  
 "data": [ // 当前页的数据  
 {  
 ...  
 }  
]

后端采用mysql作为数据的持久性存储

前端向后端发送目标的页码page以及每页显示数据的数量pageSize，默认情况每次取10条数据，则每一条数据的起始位置start为：

const start = (page - 1) \* pageSize

当确定了limit和start的值后，就能够确定SQL语句：

const sql = `SELECT \* FROM record limit ${pageSize} OFFSET ${start};`

上诉SQL语句表达的意思为：截取从start到start+pageSize之间（左闭右开）的数据

关于查询数据总数的SQL语句为，record为表名：

SELECT COUNT(\*) FROM record

因此后端的处理逻辑为：

* 获取用户参数页码数page和每页显示的数目 pageSize ，其中page 是必须传递的参数，pageSize为可选参数，默认为10
* 编写 SQL 语句，利用 limit 和 OFFSET 关键字进行分页查询
* 查询数据库，返回总数据量、总页数、当前页、当前页数据给前端

代码如下所示：

router.all('/api', function (req, res, next) {  
 var param = '';  
 // 获取参数  
 if (req.method == "POST") {  
 param = req.body;  
 } else {  
 param = req.query || req.params;  
 }  
 if (param.page == '' || param.page == null || param.page == undefined) {  
 res.end(JSON.stringify({ msg: '请传入参数page', status: '102' }));  
 return;  
 }  
 const pageSize = param.pageSize || 10;  
 const start = (param.page - 1) \* pageSize;  
 const sql = `SELECT \* FROM record limit ${pageSize} OFFSET ${start};`  
 pool.getConnection(function (err, connection) {  
 if (err) throw err;  
 connection.query(sql, function (err, results) {  
 connection.release();  
 if (err) {  
 throw err  
 } else {  
 // 计算总页数  
 var allCount = results[0][0]['COUNT(\*)'];  
 var allPage = parseInt(allCount) / 20;  
 var pageStr = allPage.toString();  
 // 不能被整除  
 if (pageStr.indexOf('.') > 0) {  
 allPage = parseInt(pageStr.split('.')[0]) + 1;  
 }  
 var list = results[1];  
 res.end(JSON.stringify({ msg: '操作成功', status: '200', totalPages: allPage, currentPage: param.page, totalCount: allCount, data: list }));  
 }  
 })  
 })  
});

## 三、总结

通过上面的分析，可以看到分页查询的关键在于，要首先确定每页显示的数量pageSize，然后根据当前页的索引pageIndex（从1开始），确定LIMIT和OFFSET应该设定的值：

* LIMIT 总是设定为 pageSize
* OFFSET 计算公式为 pageSize \* (pageIndex - 1)

确定了这两个值，就能查询出第 N页的数据

## 参考文献

* https://www.liaoxuefeng.com/wiki/1177760294764384/1217864791925600
* https://vue3js.cn/interview/