**1.SSR、同构、同构组件**

参考： [React 中同构（SSR）原理脉络梳理](https://juejin.im/post/5bc7ea48e51d450e46289eab)， [聊一聊前端「同构」](https://juejin.im/entry/5b1631085188257d492adc9e)

**2.fetch**

参考：[优雅地请求: fetch, node-fetch, isomorphic-fetch](https://zhuanlan.zhihu.com/p/22336630)

**3. fe\_client的ssr流程分析，以rank页面为例，jupiter-cli-service dev启动服务**

输入/c/rank（地址可以从根目录配置，entry对应jupiter.config.js里配置的entry）

         进入入口文件page里的rank.js（在jupiter.config.js里配置的entry指定）

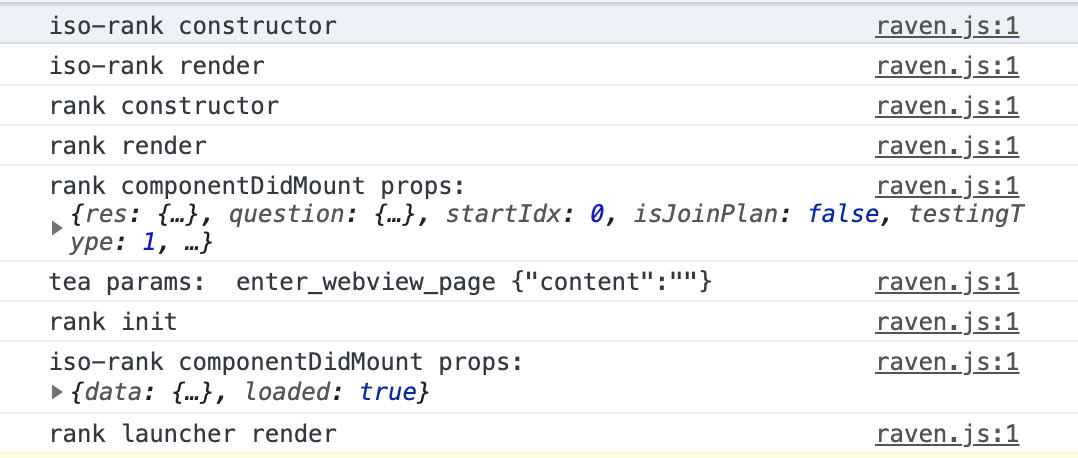
  rank.js export launch.render() （launch是对浏览器端进行处理，进行组件（这里的组件是已经同构完的）挂载、渲染，给同构组件传入loaded，data）

launch.render()返回containers下的rank/index.js

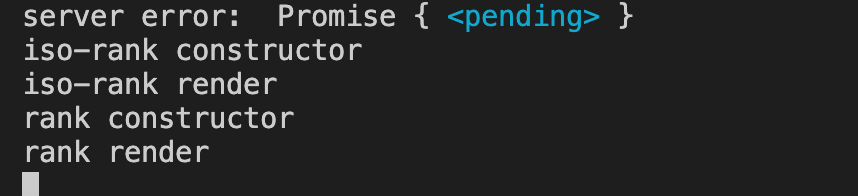
rank/index.js调用createIsoComponent生成Rank同构组件，同构组件根据传入的loaded，data来判断是否去执行prefetch（如果data为null，则loaded为false，同构组件在componentDidMount时就会去prefetch，然后在render()中将获取到的data，loaded状态以props形式传给真正的业务组件）

执行流程: node server接受到请求，执行multipages中间件->1.执行prefetch，获取数据，2.生成reactDom，传入prefetch获取的参数，并调用reactDom的renderToString()生成html字符串，将html字符串插入模版（模版已链入js），再插入<template id="REACT\_SSR\_HYDRATED\_DATA">{data}</template >标签（带入prefetch获取的数据）->浏览器端获取到html，进行渲染，运行js，发现html中有REACT\_SSR\_HYDRATED\_DATA标记，于是从template标签里获取data，完成reactDom的渲染

浏览器端：



node端：



**4.React生命周期**

参考：[图解ES6中的React生命周期](https://juejin.im/post/5a062fb551882535cd4a4ce3)

**5.深入了解nodejs的事件循环机制**

参考：[深入了解nodejs的事件循环机制](https://juejin.im/post/5bae19ebe51d450e7633241a)

**6.Charles使用教程**

1.下载地址

不支持在 Doc 外粘贴 block

2.注册

打开Charles，  进入 Help -> Register

输入

Registered name:

Bytedance Ltd.

License key:

abb23b8b770c915352    (Charles 3)

64c7e0adbbe56e3abf     (Charles 4)

3.本地boe环境测试

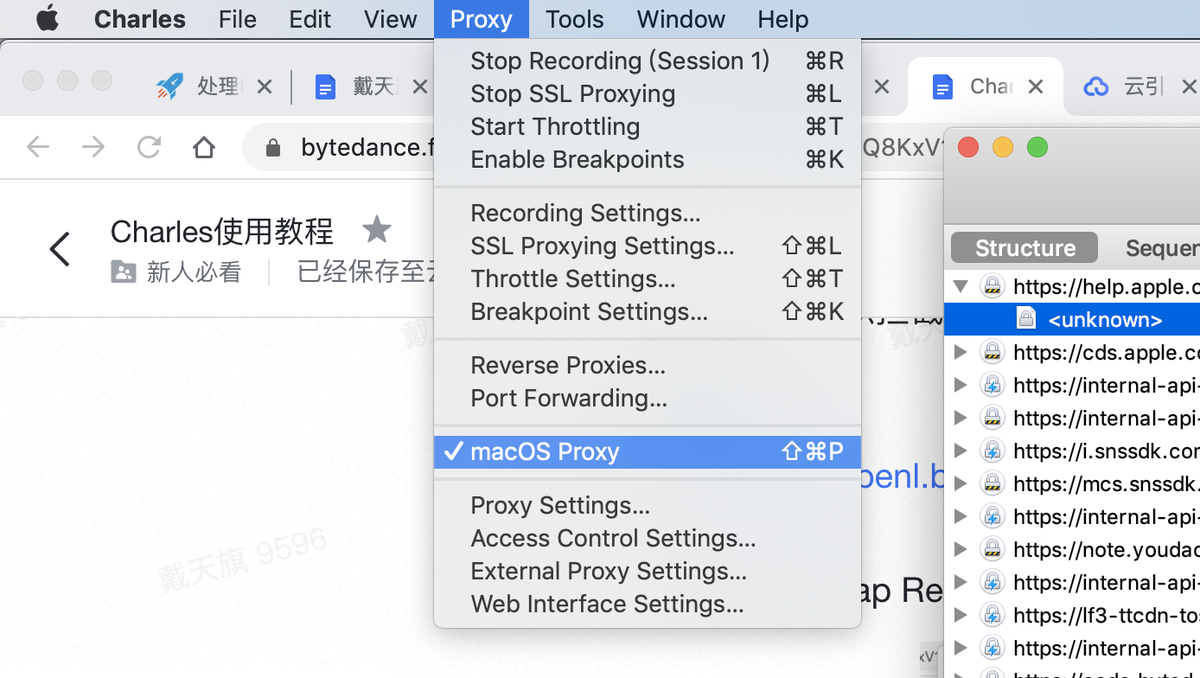
3.1原理

Charles可以拦截浏览器请求，将请求的文件换成本地起的服务的文件。

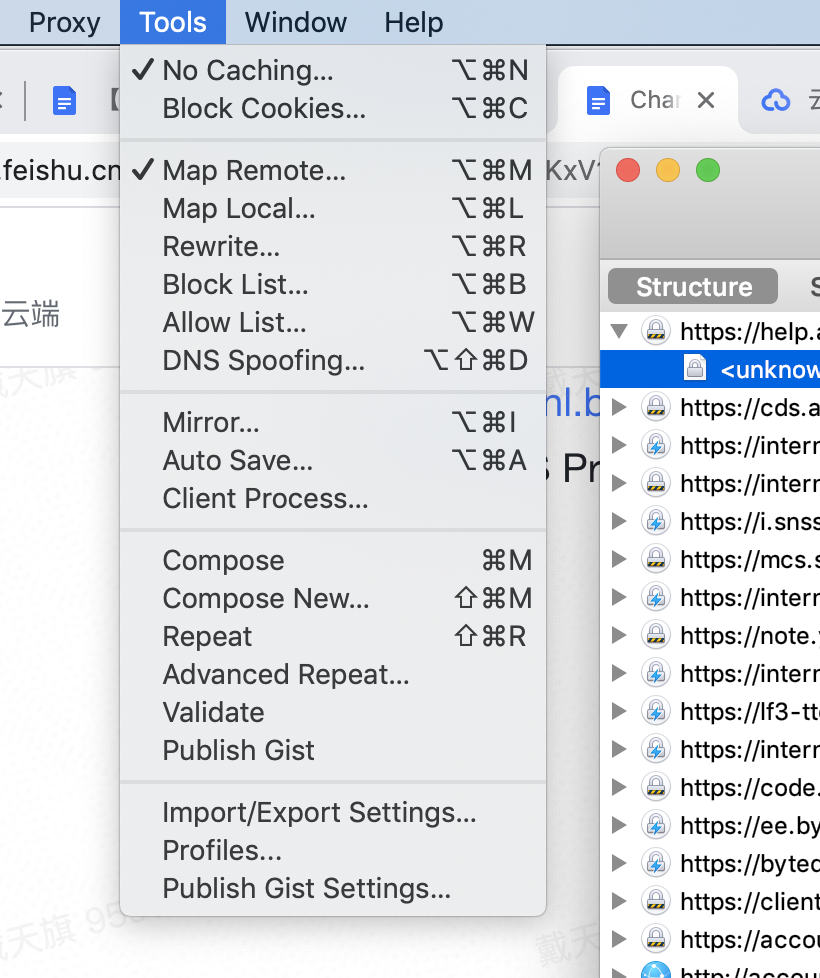
3.2配置

以<https://openl.bytedance.net/>为例

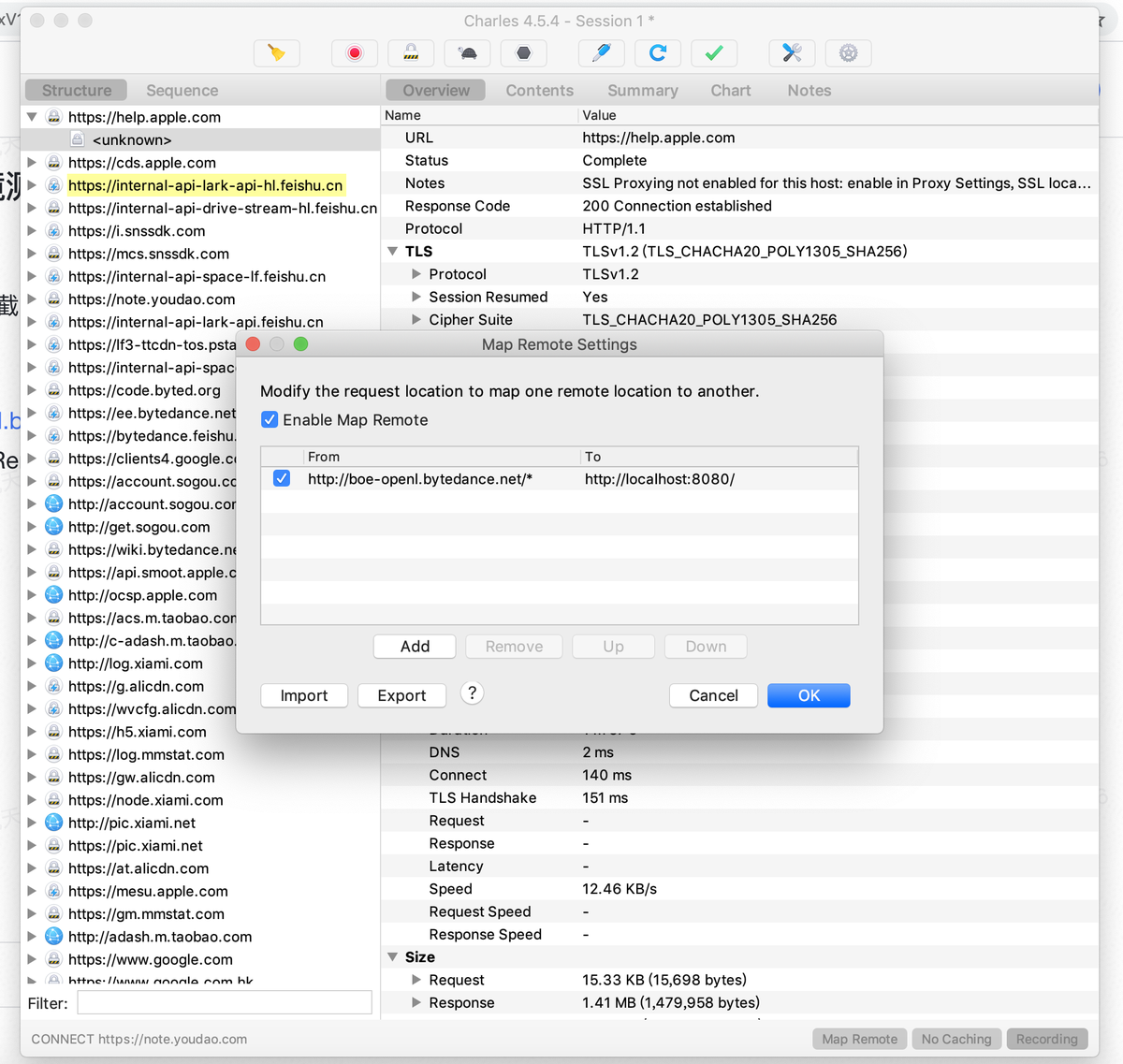
1.开启macOS Proxy



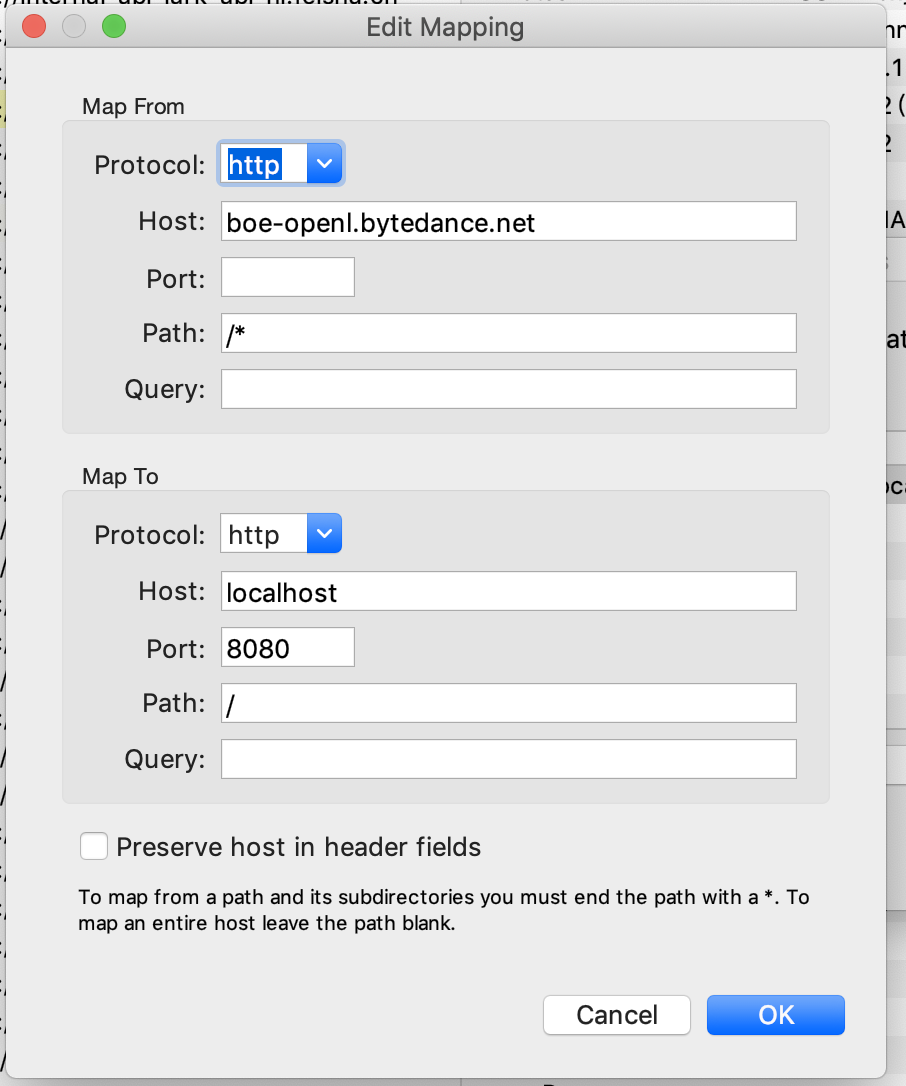
2.开启noCaching，防止浏览器缓存本地服务器的文件



3.Tools -> Map Remote



点击 Add 添加新的替换规则



protocol：协议（boe测试一般是http）；host：域名；Port：端口（远程服务器一般不用写端口）；Path：拦截的路径（\*表示多个文件或目录，可以只指定一个文件）

点击OK就能进行拦截了。

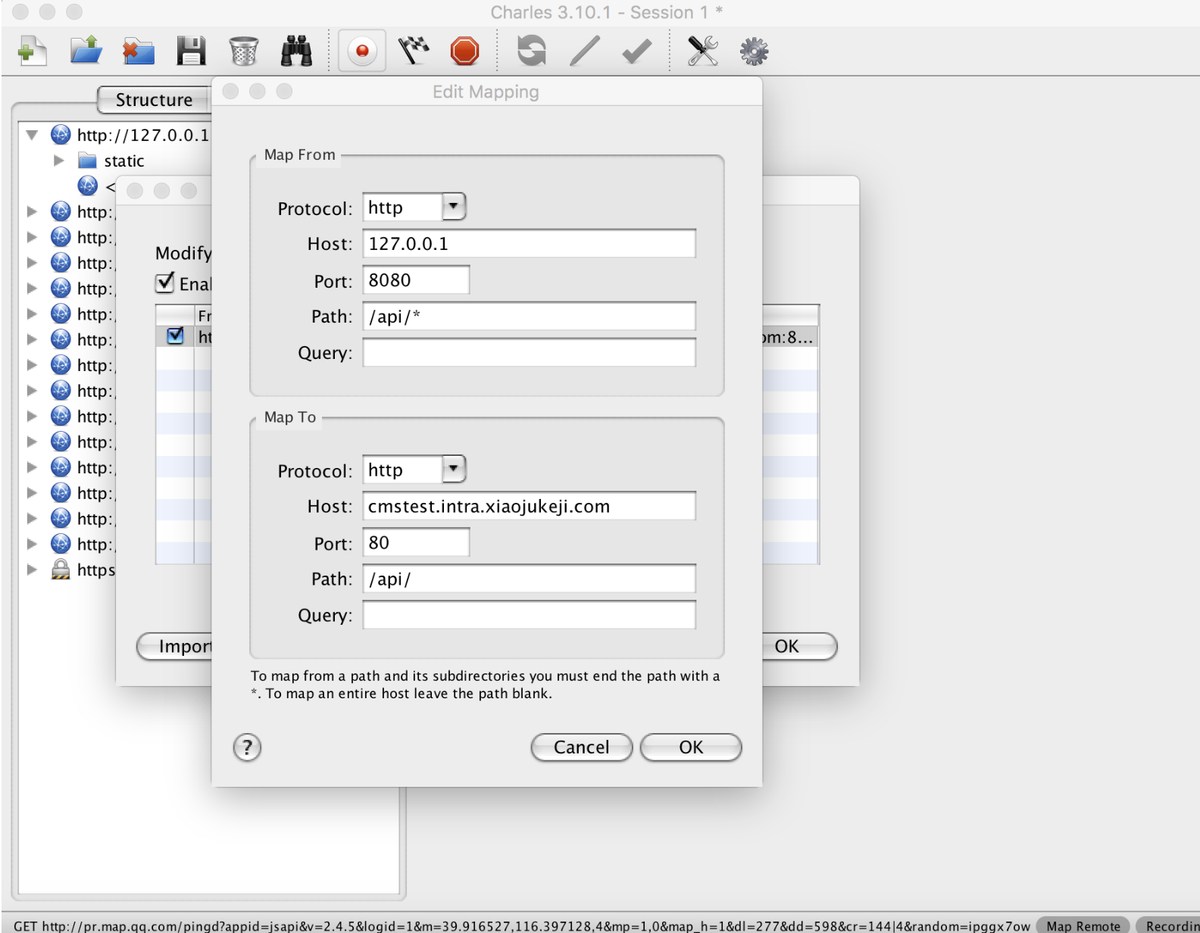
这样我们就能在<https://openl.bytedance.net/>上使用本地开发的页面，然后请求boe测试数据。

3.3注意事项

一定要把本地服务的mock关了，不然数据请求依然会被本地服务器转发到本地。

3.4其他

还有一种思路A2是把Charles当做代理，转发数据请求。



但这种方法并没有上述替换本地页面的方法好，因为A2会碰到权限、跨域等问题。

**7.iview select option key**

这是<Option v-for="option(item,index)  in items" :key="index"中的key的问题导致的,

（也可能是没有写:key="index"，注意，必须写）

key根据键值对定位元素，如果index索引只代表一个元素就不会出现重复，否则就出异常(例如：选中与显示的值不符)。

解决办法key="index"替换成 key="只代表唯一含义的属性"

**8.vscode调试**

参考：[VSCode 调试中 launch.json 配置不完全指南](https://www.barretlee.com/blog/2019/03/18/debugging-in-vscode-tutorial/)

**9.移动端调试之Vconsole**

参考：[移动端使用 vConsole 打印console的信息数据](https://blog.csdn.net/qq_30100043/article/details/79206184)、[vConsole ——开源的前端 console 调试面板](https://www.qianduan.net/vconsole-open-source/)