网页的发展历史:

1. 后端渲染
2. 后端渲染，后端路由：

早期的网页 都是由后端语言编写的（没有js） 后端接收到浏览器请求的URL，对

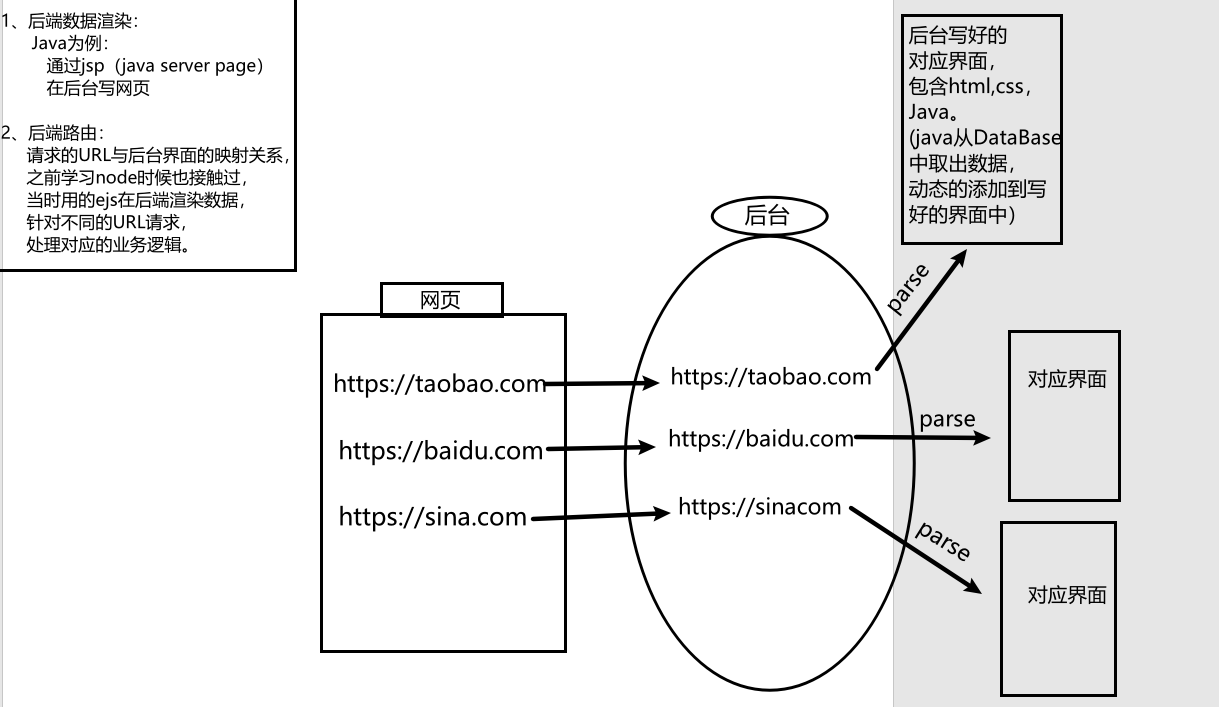
URL进行解析，再把对应的请求界面渲染到前端。 这些请求界面都是提前在后端写好的界面，包含html、css、后台语言， 后台语言的作用是从DataBase中读取数据，动态的添加到写好的界面中。（界面在浏览器展示前就已经写好定型了）

例如JAVA中的jsp（java server page）

缺点:

整个页面和后台，都是由后台开发人员完成的。

Html、css、后台语言，参杂在一起，后期维护艰难。



1. 前端渲染（前后端分离）：

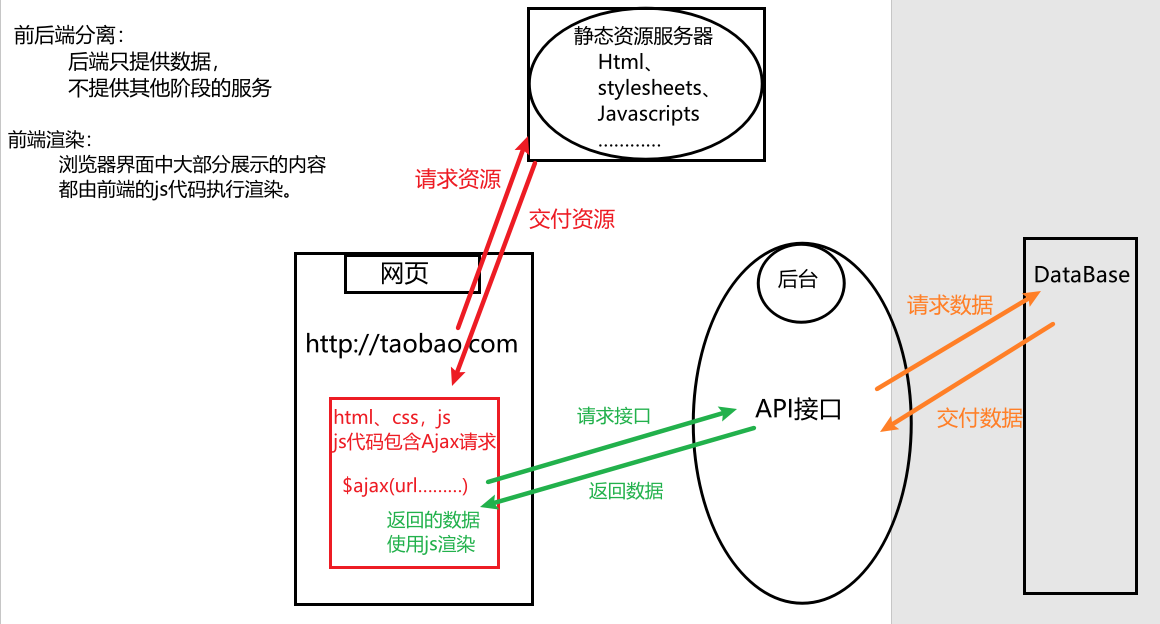
出现新角色（静态资源服务器），有些静态资源服务器是和后台服务器结合在一起的，一般后台服务器只提供API接口，静态资源服务器用来存放网页渲染的资源，html、css、javascript等。

当网页发送Url请求时，首先请求 静态资源服务器，得到html、css、js

Html、css能够直接渲染，但是js中包含Ajax请求，Ajax向后台请求对应的接口，

后台再向DataBase请求数据。

Ajax得到请求的数据，再通过js进行渲染。



优点:

随着Ajax的出现，有了前后端分离的开发模式。

后端只提供API来返回数据，前端通过Ajax获取数据，并且可以通过js将数据渲染到界面上。

前后端责任清晰，后端专注数据处理，前端负责界面展示和交互。

当移动端（Android/IOS）出现后，后端不需要进行任何处理，依然使用之前的API即可，数据的渲染都是通过前端Ajax请求动态渲染的。

1. 前端路由（单页面富应用SPA）

网页发展的第三个阶段：SPA(Single Page Application )单页面富应用

SPA就是在前后端分离的基础上添加了一层前端路由，抽离的资源与URL形成映射关系。

