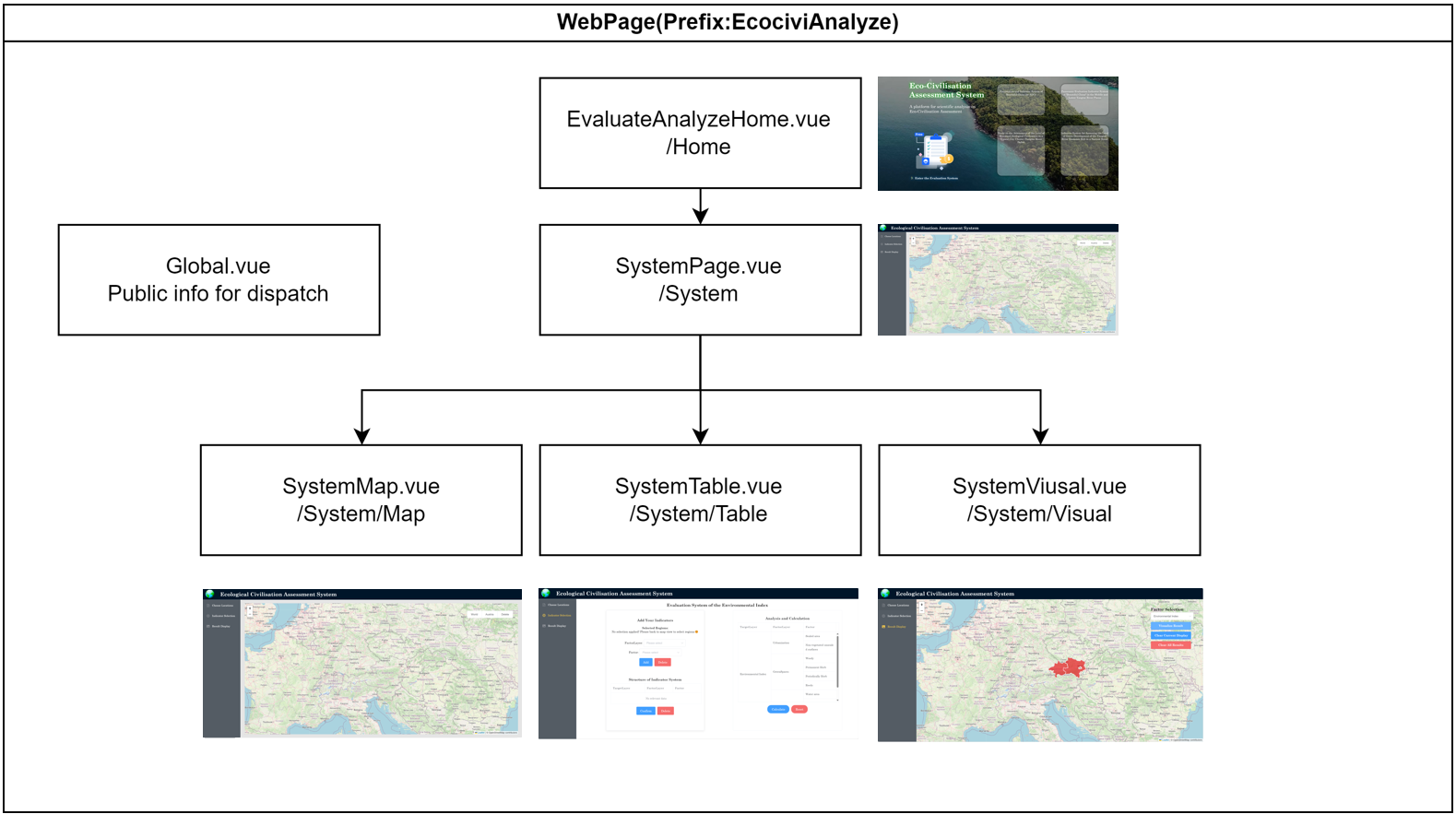
# 1 design decision

总体：本项目采用Nodejs+Vue2 +JavaScript框架实现了可供用户自定义的生态环境评估和评估结果可视化的网页地图应用程序设计。

1. 项目主题？选择原因？预计实现的功能

2. 预期的页面组织和设计（具体每个页面大概是做什么的？）

需补充：页面1提供了Austria和国家地图，此二者在点选时有何差异？



3.使用到的技术或亮点：

0. a dash board with navigator and router control

1. integrate additional data sources （json数据，外部图片，外部环境因子数据）

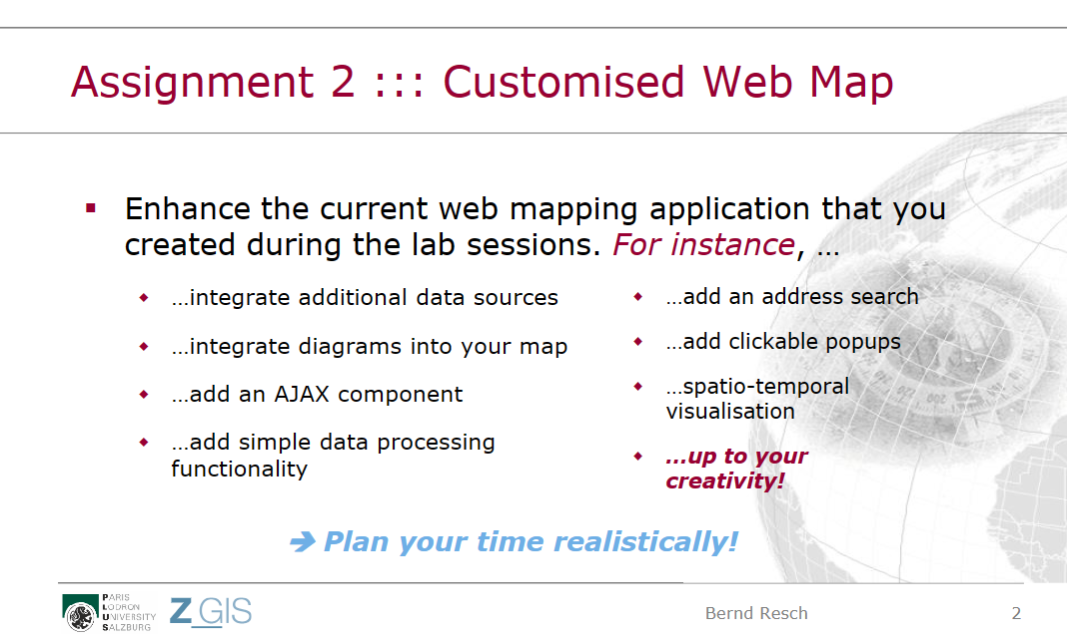
2. add an AJAX component （构建了后端服务器，前端通过axios技术，使用GET方法获取了GeoJson数据）

3. add data processing functionality （添加了很多数据获取和处理逻辑，特别是table页面，添加了数据互动，以计算环境指数）

4. add clickable popups （map展示页面和结果展示页面都有popup）

5. spatio-temporal visualization （结果展示页面通过分级色彩可视化呈现结果）

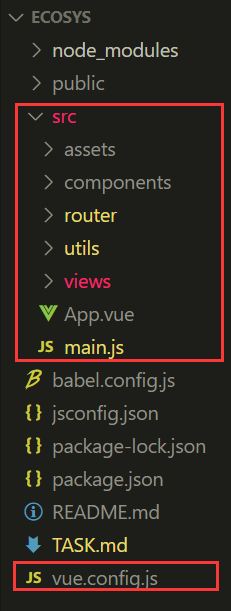
6. customization functionality （在region选择页面和table页面，添加了大量可用户自定义的输入，提高了程序的互动性和独特性）



# 2 项目程序结构

前端+后端

前端：核心程序如下红框，vue.config.js主要配置端口号（8099）和public文件夹访问目录



App.vue是整个前端页面的入口;

main.js用于vue项目的初始化，挂载外部库和自定义功能

router/index.js用于配置页面间切换的路由………

assets用于存放各类资源（但两个data还是放在后端）

component用于存放可复用的map；utils中主要负责地图的基本配置

views-> EvaluateAnalyze是核心页面

1. js
2. views
   1. global: 用于存放全局数据，供xx页面调用，如…，…，
   2. SystemPageComponent文件夹+ SystemPage.vue（dashboard的主题，分别具体解释）
3. EvaluateAnalyzeHome.vue+ index.vue：欢迎页面

后端（略）：存储两个数据，配置body-parser，cors和express库实现和前端的跨域通讯。在server.js中，配置服务器端口为8050，同时配置了两个get请求，地址为/api/world和/api/at.

# 3 最终实现效果（举例）