2019-2-23

EAIDK实验室管理系统

架构设计文档

OPEN AI LAB

变更记录  
(Revision Record)

|  |  |  |  |
| --- | --- | --- | --- |
| 日期  (Date) | 版本  (Rev) | 说明  (Change Description) | 作者  (Author) |
| 2019.2.23 | V0.1.0 | 初始版本 | Danny Zhan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录(**catalog**)

[1 文档目的 3](#_Toc1834378)

[2 文档用户（Audience） 3](#_Toc1834379)

[3 术语解释（Term） 3](#_Toc1834380)

[4 架构描述 3](#_Toc1834381)

[4.1 系统总体概述 4](#_Toc1834382)

[4.2 前端框架选择 5](#_Toc1834383)

[4.3 浏览器兼容性 6](#_Toc1834384)

[4.4 后端开发语言选择 6](#_Toc1834385)

[4.5 后端Web框架选择 6](#_Toc1834386)

[4.6 服务器端推送(Push) 8](#_Toc1834387)

[4.7 数据库 8](#_Toc1834388)

[4.8 Web服务器及系统部署 9](#_Toc1834389)

[4.9 系统可维护性描述 9](#_Toc1834390)

[4.10 系统架构概述 9](#_Toc1834391)

[4.11 各模块详述 10](#_Toc1834392)

[5 风险评估与对策 10](#_Toc1834393)

# 文档目的

本文档目的是描述EAIDK实验室管理系统的架构设计，从技术角度概述系统设计的总体纲领。

本文档基于EAIDK实验室管理系统需求规格说明书1.0.

# 文档用户（Audience）

本文档主要面向软件开发团队成员。

# 术语解释（Term）

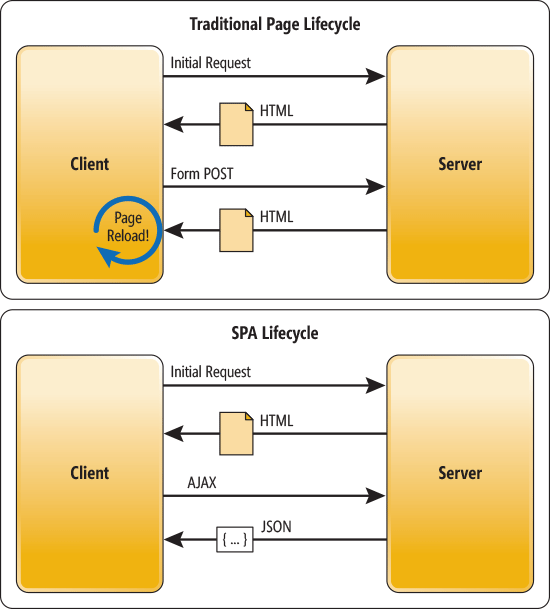
|  |  |
| --- | --- |
| 术语 | 解释 |
| EAIDK | Embedded AI Development Kit，由Open AI Lab开发的基于Arm SoC的面向行业和教育的嵌入式AI开发套件 |
| Single Page Application | 单页应用，只有一个静态页面，通过用户交互动态更新Web页面的一种技术 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

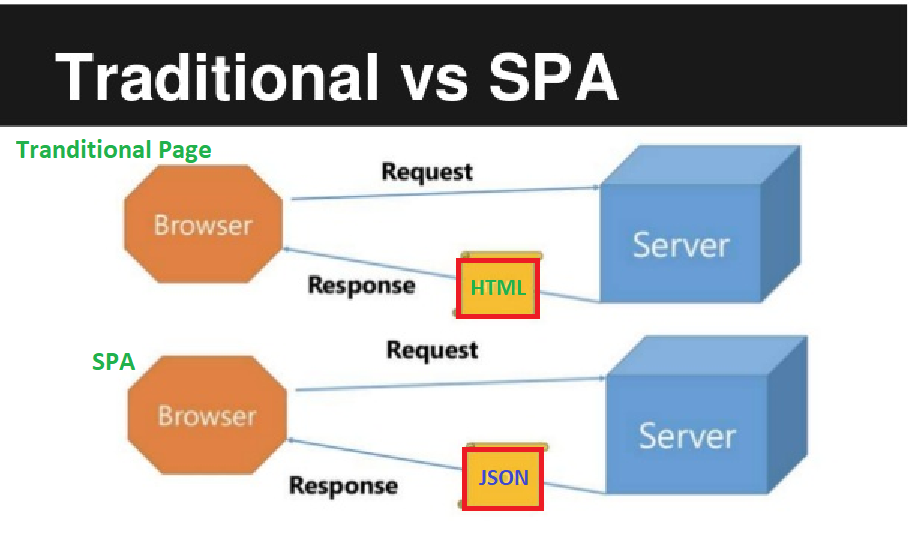
# 架构描述

## 系统总体概述

本系统是一个网站，采用Browser-Server架构。

随着网络速度的提升和浏览器技术的进步，单页应用(Single Page Application, 简称SPA)受到越来越多用户的青睐。其好处是显而易见的：接近桌面应用的用户体验，服务器端可以只关注数据而不必关注呈现，这使得同样的API可以供PC浏览器端，不同的手机客户端使用，减少了开发成本。单页应用的缺点可能是首次加载的时候较慢，这可以通过服务器端压缩等技术缓解，至于搜索引擎不友好，则不是本系统需要考虑的，因此本系统也将采用单页应用。





## 前端框架选择

本系统前端通常采用HTML 5 + CSS 3 + Javascript.

其中 CSS样式部分将采用目前最为的流行的bootstrap框架：<https://getbootstrap.com/>

关于Javascript框架，目前有三大主流前端框架：Angular, React.JS和Vue.JS。

Angular最为复杂庞大，学习曲线陡峭，React.JS其次，Vue.JS最为简单。

考虑到OpenAI Lab内部已有使用Vue.JS的先例和它的上手难度较小，决定使用Vue.JS作为前端框架。

## 浏览器兼容性

目前主流浏览器有Chrome, Firefox, Edge，我们的系统将兼容这三种浏览器的最近两个主要版本，同时兼顾IE 11, 本系统将不支持其它IE浏览器（微软早已停止对IE 6~10的支持， IE 11不支持目前广泛使用的Javascript标准ECMAScript 6）。

## 后端开发语言选择

目前Web后端语言非常多，常见的有PHP, Java, Python, C#（Asp.Net), Javascript(Node.JS)等。

本系统采用Node.JS作为后端设计的平台。这是由于Node.JS可以兼容不同OS，且采用Javascript语言，相对简单。

## 后端Web框架选择

选定了Javascript(Node.JS)作为后端开发语言之后，还需要为网站选择一种Web框架。

Express: <https://github.com/expressjs/express>

<https://www.npmjs.com/package/express>

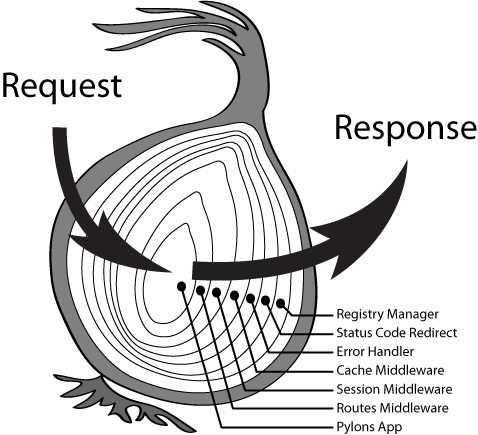
Node.JS社区成熟度最高的Web框架，几乎成了标准的Web框架，但由于缺乏对Promise， async/await的支持，核心团队成员离开，新版V5拖延数年还处于alpha状态，发展堪忧。

Express最被人诟病的地方在于嵌套callback以及由此带来的异常处理困难。

Koa: <https://github.com/koajs/koa>

<https://www.npmjs.com/package/koa>

由前Express团队成员TJ创立的轻量级JS库，剥离了非核心代码，拥抱最新的Web技术标准。整个框架需要借助中间件完成：<https://github.com/koajs/koa/wiki>



Egg.JS: <https://github.com/eggjs/egg>

<https://www.npmjs.com/package/egg>

Egg.JS孵化于阿里巴巴但不与任何阿里巴巴业务耦合，其创立者也是Koa的团队成员。使用koa时，你要写一个项目，要往里面加很多中间件，要写脚本加载routes文件夹下面的所有路由以及model文件夹下面的所有sequelize模型，koa仅仅是一个骨架，其他的都是你来完成，自由度高，但集成度低，每创建一个新项目都要做很多重复工作。egg.js封装了一套koa，可以理解成大礼包版koa，集成度高，可以轻松创建一个项目而不用做很多繁琐的初期工作，解放生产力，更可贵的是有一套现成的规范提供给我们，不需要我们自己再去探索一套规范。

基于以上介绍及理由，本系统将使用Egg.js作为后端Web框架。

## 服务器端推送(Push)

传统网站不具备服务器端推送能力，因而在需要获取服务器端的变化及消息通知时，只能采取定期查询(轮询， Pull)这种消耗资源且不及时的方式进行。在HTML 5中引入的WebSocket改变了这一传统，使得网站有了和桌面应用相媲美的双向通讯能力。本系统也将采用WebSocket技术实现服务器端向客户端推送消息。

基于Node.Js，我们将采用Socket.IO这一流行库。

<https://github.com/socketio/socket.io>

## 数据库

MySQL是目前流行度最高的开源数据库，然而MongoDB开创了NoSQL数据库的先河，并迅速流行起来。和传统关系型数据库相比，MongoDB的数据模型更加灵活，不需要像关系型数据库那样提前把数据库Schema定义好，且可以用嵌入式文档来部分替代关系型数据库的外键。 MongoDB内存直接存储类似JSON的BSON格式，很适合网站。MongoDB还可以把具有派生关系的父子对象存储到同一个集合中。由于使用类JSON的数据存储，其数据模型更加简单，开发效率也有一定的提升。除非高度事务性的系统或者对于数据处理极度复杂的BI应用，MongoDB适合绝大部分应用场景。

本系统将使用MongoDB作为数据库, 使用MongoDB官方的Node.JS驱动。

参考：

数据库流行度排名：<https://db-engines.com/en/ranking>

MongoDB下载：<https://www.mongodb.com/download-center/community>

MongoDB的Node.JS驱动：<https://github.com/mongodb/node-mongodb-native>

Mongoose （MongoDB的对象模型化工具）: <https://github.com/Automattic/mongoose>

## Web服务器及系统部署

由于本系统可能面向学校或者培训机构，系统必须适应各种部署环境，因此要尽可能跨平台。我们将使用nginx作为Web服务器。

## 系统可维护性描述

本系统需要输出日志以方便排错。

## 系统架构概述

下图为本系统使用的主要技术栈

浏览器

服务器

Egg.JS

CSS

Vue.JS

HTML

Node.JS

WebSocket

Browsers(Chrome/Firefox/Edge/IE11)

MongoDB

Web Server(Nginx)

下图为系统模块架构图

Domain Layer（业务逻辑）

Data Access （数据库数据访问层）

Entities（定义数据库存储的对象模型）

## 各模块详述

暂无

# 风险评估与对策

暂无。