总设计

页面总体由几部分组成：

1. Navigator导航栏部分
2. SlideShow 幻灯片模块
3. Main 核心内容部分

…

# Navigator架构设计

页面上的导航栏，一个navigatorLoader对象作为核心对象，之后通过navigatorCell对象声名导航格，每一个导航格的声明占一篇代码，统一写到一个文件夹里好维护。一篇声明代码的标准格式如下：

// navigator-cell-xxx.ts

const navigatorCell = new NavigatorCell("content", [], function () {  
 console.log("这里是回调函数触发事件");  
})  
  
export default navigatorCell;

## Navigator-loader.ts

创建一个类用于提供多种管理导航格的方法，例如添加，删除，修改，查看等等，维护一个列表用于储存当前正在被管理的导航格对象，从而动态生成导航。

动态生成导航需要根据添加的项异步刷新导航（setTimeout(xxx,0)）

## Navigator-cell.ts

每一个导航格的具体单元的类，这个类的实例对象表示一个导航格。这个类的实例对象要维护以下信息：导航格的内容，导航格的二级导航，导航格的响应回调函数

## Navigator-adapter.ts

作为将navigatorLoader对象显示在html文档中的中介而存在，是一个显示的适配器，主要根据html文档需要调整输出策略，需要维护navigatorLoader实例对象，提供一个display()方法

# SlideShow

在主页面上显示的图片轮播

此处注意，所有图片轮播图片必须预加载

## SlideShow.js

标准图片轮播提供类

# Main模块

需要考虑两点：跳转到页面之后会根据所选中的人物再跳转。

一个HTML页面有多个<template>之后根据当前页面需要显示的东西实现显示和隐藏

多个同样的页面，例如某人的详细信息，应该用类接管

## 简介：

无需使用JS生成，直接使用HTML和CSS简易布局即可

## 社区荣誉：

给定一个空页面即可

## 网格管理

贴张图片就OK

需要图片解析：图像解析算法流程如下：

输入：一个png或者jpg图像用于解析，一个json文件用于记录一个颜色块和其信息的对应关系，包含name，desc等属性

1. 使用offscreencanvas载入图像，将图像大小固定在内容容器之内且不失真，使用2d上下文
2. 使用像素处理
3. 遍历像素，根据json文件描述对各个块进行分类，将位置信息存入数组
4. 通过最大最小值算出数组边界值，确定区块范围
5. 为每一个颜色区块分别建立一个canvas，canvas尺寸为区块大小，并为其设置一个id，id为当前颜色区块对应信息关系中的name属性
6. 将建成的canvas放入容器中，使用容器内的绝对定位，定位位置为区块的左上方位置，并使用CSS将canvas中透明的部分修剪掉
7. 在其他模块中通过获取id的方法来获取对应的区块

注意：这里一定要显示一个让用户等待的标识

## 两委成员：

需要构建两个页面，分别显示所有信息和详情信息，一个页面就是一个类，每个类需要对应的template

详情信息中给定一个返回按钮

注册一个container元素，用于作为多页面容器，每一个页面必须要给定一个render()方法用于渲染至父元素，每一个页面必须实现SubPage接口，此接口定义一个render()方法。

Html：一堆template和一个container

JS：

没个两委的具体信息可以根据一个或多个json文件给出

Commissioner类：相当于DAO类，维护每一个委员的个人信息，有name，image，info等属性

CommissionerListEvent类：是CommissionerList类的父类，用于编写事件

CommissionerList类：用于维护两委列表页面的信息，需要基本属性：commissionerList: Commissioner[]，currentPage:number，itemInAPage:number，totalPage:number等，需要维护基本方法render();

CommissionerDetails类：用于维护详情信息页面，需要维护基本方法render();

## 为民服务：

空页面即可

## 事项清单：

和两委一样

## 证据出具：

和两委一样

给出网络接口以获取证据出具文件，html格式即可