框架目标：通过一个特殊格式的文档生成整个web应用。

关于前后端分离的思考：信息管理类应用并发数一般都低于100，对于性能的要求极低。所以性能上没有前后分离的必要。开发上要求周期短，迭代快，统一的后端管理更符合要求。

完整的应用由三部分组成：由jinjia编写的模板，由python编写的路由注册，由python编写的功能模块。 上述废弃

完整的框架也应该能生成三部分文件。框架的本质是一个可以生成文件的类，根据一个完整的数据模板，生成三部分互相配合的软件。

采用flask-restful 和vue 配合的前后端分离能减少开发难度。由框架同时生成表示层和资源注册，代码结构更统一。

统一数据规范

全数据：包含路由映射，样式设定，核心数据

{# 全数据模板 d = {  
 "id":'employeelist',  
 "name": 'employeelist',  
 "css":'table table-hover table-bordered',  
 "colnames": colnames,  
 "lang":chi,  
 "step": 5,  
 "pages": 1,  
 "data":data  
 } #}

表示层数据：样式设定

引擎层数据：路由映射

底层数据：核心数据，包含数据实体和数据关系

逻辑三层：

表示层：由jinjia模板，JS，CSS构成

引擎层：由python编写 路由注册，数据库接口，其他底层模块

数据层：数据库，配置文档

前端页面：

Base.Html 框架

Layout[N].html 样式页面N种