Server Module-API

Detailed design specification

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Reviser | Instructions |
| V1.0 | 2020-9-13 | 张震宇 | Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

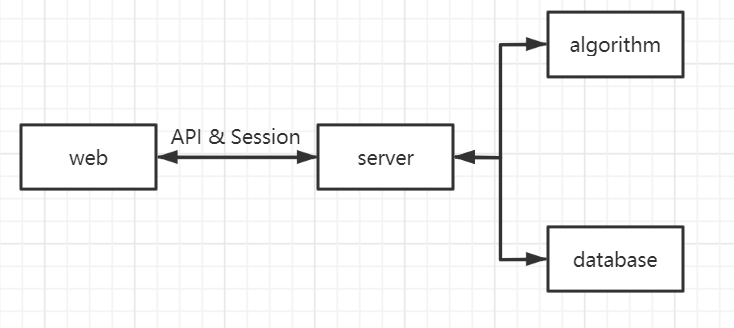
1. **Introduction**
   1. **Purpose**

The purpose of this manual is to explain the technical scheme of the overall design of the system to users and relevant personnel of software development. From the point of view of system design, this manual explains the overall architecture, processing flow, module division, function allocation, interface design, operating environment, data structure design and error handling design of the system. To provide an overall description of how the logic and data functions of the program system are implemented throughout the design period, thus serving as a basis for detailed program design and coding. This document will be the core document for the design phase. The applicable readers of this summary design specification are: customers, system architects, system developers, testers.

* 1. **References**

1. <https://www.runoob.com/js/js-cookies.html>
2. <https://www.jianshu.com/p/4350065bdffe>
3. <https://www.cnblogs.com/liuye1990/p/9663474.html>
4. <http://c.biancheng.net/view/7534.html>
5. <http://c.biancheng.net/view/7550.html>
6. https://www.layui.com/demo/admin.html
7. **Over Design**

The main function of the module is to provide the API for web module. At the web side we us ajax to send the data to server module, at the server module, we us Django library to manage the data. The overall Design can be described at below picture.



**3.API document**

**(1)** **http://114.55.107.174:8000/login: user want to login the system**

**Input:({“username”:””,”password”:””})**

**Output:({“success”:False/True,”error”:””})**

**(2) http://114.55.107.174:8000/usernameExist: judge whether this username is exist in DB**

**Input:{“username”:””}**

**Output:{“success”:False/True}**

**(3) http://114.55.107.174:8000/addUserInformation: Add the user information after user register.**

**Input:{“username”:””,”password”:””,”label”:””}**

**Output:{“success”:False/True,”error”:””}**

**(4) http://114.55.107.174:8000/register: Ask for the register html**

**Output: ‘newregister.html’**

**(5) http://114.55.107.174:8000/Recommendsong: recommend song for user**

**Input:{“username”:””}**

**Output:(‘newPersonalRecommend.html’,data)**

**(6) http://114.55.107.174:8000/personCenter: show the user’s personal Center**

**Input:{“username”:””}**

**Output:(‘newPersonCenter.html’)**

**(7) http://114.55.107.174:8000/searchSong: Ask for searchSong page**

**Output:(‘searchSong.html’)**

**(8) http://114.55.107.174:8000/search: User search song by song name**

**Input:{“songname”:””}**

**Output:{“song”:””}**

**(9) http://114.55.107.174:8000/addSongToCenter: add song to user’s Center**

**Input:({“data”:””}**

**Output:{“success”:False/True,”error”:””}**

**(10) http://114.55.107.174:8000/deleteSongFromCenter: delete song from user’s personal Center**

**Input:{“songid”:””}**

**Output:{“success”:False/True,”error”:””}**

**(11) http://114.55.107.174:8000/exit: exit from system**

**Output:{‘newlogan.html’}**