Easydian Common APIs

目录

[1. schema: 2](#_Toc350093320)

[1.1 Shops: 2](#_Toc350093321)

[1.2 SysManger: 3](#_Toc350093322)

[1.3 Userinfo: 3](#_Toc350093323)

[2. Common APIs 4](#_Toc350093324)

[3. API Details 5](#_Toc350093325)

[3.1 shops API 5](#_Toc350093326)

[3.1.1 GET /shops 5](#_Toc350093327)

[3.1.2 GET /shop\_info/:id 5](#_Toc350093328)

[3.1.3 POST /comment/:id 5](#_Toc350093329)

[3.1.4 PUT /visit/:id 6](#_Toc350093330)

[3.1.5 POST /shop 6](#_Toc350093331)

[3.1.6 PUT /field 6](#_Toc350093332)

[3.1.7 GET /validate\_ field 6](#_Toc350093333)

[3.1.8 PUT /goodbad/:id 7](#_Toc350093334)

[3.1.9 GET /gps/:id 7](#_Toc350093335)

[3.2 User APIs 7](#_Toc350093336)

[3.2.1 /new\_user 7](#_Toc350093337)

[3.2.2 GET /user/:id 7](#_Toc350093338)

[3.2.3 PUT /login/:id 8](#_Toc350093339)

[3.2.4 PUT /logout/:id 8](#_Toc350093340)

[3.2.5 POST /register 8](#_Toc350093341)

[3.3 SysManger API 8](#_Toc350093342)

[3.3.1 POST /new\_category 8](#_Toc350093343)

[3.3.2 GET /category 9](#_Toc350093344)

[3.3.3 PUT /category 9](#_Toc350093345)

[3.3.4 PUT /col\_num 9](#_Toc350093346)

[3.3.5 GET /col\_num 9](#_Toc350093347)

[3.3.6 PUT / item\_visit/:id 10](#_Toc350093348)

[3.3.7 GET / item\_visit 10](#_Toc350093349)

[3.3.8 GET /header 10](#_Toc350093350)

[3.3.9 PUT /header 10](#_Toc350093351)

# 1. schema:

## 1.1 Shops:

{

**shopname** : {type:String, default:'fullname'},#name->index, id->index

shopalias : [{type:String, default:'short name'}],

shoptype : [{type:String, default:'type'}],

shopvisit : {type:Number, default:1000},

shoppriority : {type: Number, default:1000},

shopavgconsume : {type: Number, default:10},

shopwebsite: {type:String, default: 'fullurl'},

shopaddress: {type:String, default: 'address'},

shopgps: {“x”:0,”y”:0 },#index

shopphone: {type:Array, default: ['010-22222222']},

shoponbusiness: {type:Boolean, default: true},

shoponadv: {type:Boolean, default: true},

shopweekday:[

{monday: {type:Array, default:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}},

{tuesday: {type:Array, default:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}},

{wednesday:{type:Array, default:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}},

{thurday: {type:Array, default:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}},

{friday: {type:Array, default:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}},

{saturday: {type:Array, default:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}},

{sunday: {type:Array, default:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}}

],

shopweek: [

{weekday: {type:Array, default:[0,0,0,0,0,0,0]}},

{weekdaygood: {type:Array, default:[0,0,0,0,0,0,0]}},

{weekdaybad: {type:Array, default:[0,0,0,0,0,0,0]}}

],

shopdaystats:{type:Array,default:[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]},

shopmonthstats: {type:Array,default:[0,0,0,0,0,0,0,0,0,0,0,0]},

shopgoodtotal: {type:Number, default:0},

shopbadtotal: {type:Number, default:0},

shoplogo: {type:String, default: './images/default.jpg'},

shopcover:{type:Array, default: ['beijing','shanghai']},

shopaccount: {type:Number, default:0},

shopcreatetime: {type:Date, default: Date.now},

shopcomments: [{type:String}],

shopnews: [{type:String}],

shopuserip: [{type:String}]

}

## 1.2 SysManger:

{

header\_image: [{type:String}],

header\_context: [{type:String}],

category: {type:Array, default: [‘canyin’,’banjia’, ‘zhuangxiu’]},

canyin:[

{shopnum: { type:Number, default:10}}

{column: { type:Number, default:4}},

{items: {type:Array, default:[ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}}

],

banjia:[

{shopnum: { type:Number, default:10}}

{column: { type:Number, default:4}},

{items: {type:Array, default:[ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}}

],

zhuangxiu:[

{shopnum: { type:Number, default:10}}

{column: { type:Number, default:4}},

{items: {type:Array, default:[ 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]}}

],

}

## 1.3 Userinfo:

{

username: {type:String, default:’fullname’}, #email

useraddress: {type:String, default:’address’},

userpassword: {type:String, default:’md5’}

usertargetaddress: {type:String, default:’address’},

useraddressgps: {“x”:0,”y”:0 },

usertargetaddressgps: {“x”:0,”y”:0 },

userphone: {type:Array, default: [‘010-22222222’]},

logintime: [{type:Date, default: Date.now}],

belogin: {type:Boolean, default: true},

}

# Common APIs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| For shops | | | 1 | |
| **URL** | **Method** | **parameters** | | **note** |
| /shops | get | Limit, type | | Return the basic info of a group of shops |
| /shop\_info/:id | get |  | | Get the detail info |
| /comment/:id | post |  | | Insert a comment to a shop |
| /visit/:id | PUT |  | | Update the visit num of a shop |
| / shop | post | Type, value of instance | | Insert a shop to the db |
| / field | put | Fieldpath, value | | Update the value of a given field |
| /validate\_ field | get | Fieldpath, value | | Validate the given field |
| / goodbad/:id | put | type | | Update the good bad comment |
| / gps/:id | get | Type | | Return the gps location |

|  |  |  |  |
| --- | --- | --- | --- |
| For User | | 2 | |
| **URL** | **Method** | **parameters** | **note** |
| / register | post | userinfo | Insert a user info to database |
| /validate\_field | get | Fieldpath, value | The same in shops |
| /user/:id | get |  | Return the user info |
| /login/:id | put |  |  |
| /logout/:id | put |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| For SysManager | | 1 | |
| **URL** | **Method** | **parameters** | **note** |
| /new\_category | post | category | Insert a category to the db |
| / category | get |  | Return all the category |
| / category | put |  | Update the category’s name |
| /col\_num | put | num |  |
| / col\_num | get |  |  |
| / item\_visit | get |  |  |
| / item\_visit/:id | put | num |  |
| / header | get |  |  |
| / header | put | Image\_paths, text |  |

# 3. API Details

## 3.1 shops API

### 3.1.1 GET /shops

Get the shops registered in easydian

Priority: must

Request Parameters:

Category: specify the category of the shops

Start(optional): start index (sorted by priority, good, bad, and account), 0 base

End(optional): end index

Limit(optional): limit the number of items

Request body: N/A

Response Status: 200, 404, 500

Response Body: An array of shops, JSON format, including: shoplogopath, shopavgconsume, shopwebaddress, shopgood/bad count, shopvisittoatal,

### 3.1.2 GET /shop\_info/:id

GET the shop detail

Prority: must

Request Parameters:

Category: specify the category of the shops

Request body: N/A

Response Status: 200, 404, 500

Response Body: shop detail: all the statistics info: day week, month,comment for draw picture

### 3.1.3 POST /comment/:id

Insert a new comment to a shop

Prority: should

Request Parameters:

Category: specify the category of the shops

Request body: {comment:value}

Response Status: 200, 404, 500

Response body: {comment:value}

### 3.1.4 PUT /visit/:id

Update the visit num of shop

Priority: should

Request Parameters:

Category: specify the category of the shops

Request body: N/A

Response Status: 200, 404, 500

Response body: {visit:value}

### 3.1.5 POST /shop

Insert a new shop to db

Priority: must

Request Parameters:

Category: specify the category of the shops

Request body:

Instance:A json configuration for a shop

Response Status: 200, 404, 500

Response body: {success:boolean}

### 3.1.6 PUT /field

Update value of a field

Priority: should

Request Parameters:

Category: specify the category of the shops

Request body:

newvalue: A json for new value: {path:value}

Response Status: 200, 404, 500

Response body: {success:boolean}

### 3.1.7 GET /validate\_ field

Validates the given field’s value

Priority: should

Request Parameters:

Path: the field’s path

Value: the field’s value

Request body: N/A

Response Status: 200, 404, 500

Response body: {success:boolean}

### 3.1.8 PUT /goodbad/:id

Update the value of goodbad comment

Priority: must

Request Parameters:

Category: specify the category of the shops

Type: good/bad

Request body: N/A

Response Status: 200, 404, 500

Response body: {bgt:num}

### 3.1.9 GET /gps/:id

Priority: optional

Request Parameters:

Category: specify the category of the shops

Request body: N/A

Response Status: 200, 404, 500

Response body: a json file contains x,y

## 3.2 User APIs

### 3.2.1 /new\_user

Insert a new user to db

Priority: should

Request Parameters: N/A

Request body: A json file of all the user’s information

Response Status: 200, 404, 500

Response body: {success:boolean}

### 3.2.2 GET /user/:id

Get the user’s detail information

Priority: should

Request Parameters: N/A

Request body: N/A

Response Status: 200, 404, 500

Response body: A json file contains user’s information

### 3.2.3 PUT /login/:id

Send user login info to server

Priority: should

Request Parameters: N/A

Request body: a json file of user login info

Response Status: 200, 404, 500

Response body: A json file contains basic user info, / error info

### 3.2.4 PUT /logout/:id

Send user logout info to server

Priority: should

Request Parameters: N/A

Request body: N/A

Response Status: 200, 404, 500

Response body: A json file contains basic user info, / error info

### 3.2.5 POST /register

Insert a user info to db

Priority: should

Request Parameters: N/A

Request body: A json file of user’ info

Response Status: 200, 404, 500

Response body: A json file contains basic user info, / error info

## 3.3 SysManger API

### 3.3.1 POST /new\_category

Insert a new category to db

Priority: should

Request Parameters:

name: category name

Request body: N/A

Response Status: 200, 404, 500

Response body: A json file contains basic user info, / error info

### 3.3.2 GET /category

Return all the categories

Priority: must

Request Parameters:

Request body: N/A

Response Status: 200, 404, 500

Response body: A json file contains all the categories

### 3.3.3 PUT /category

Update the category name

Priority: must

Request Parameters:

Category: the new value of the category

Request body: N/A

Response Status: 200, 404, 500

Response body: {category:value, success:boolean}

### 3.3.4 PUT /col\_num

Update the num of column

Priority: must

Request Parameters:

value: the new value of the col\_num

Request body: N/A

Response Status: 200, 404, 500

Response body: { value:value, success:boolean}

### 3.3.5 GET /col\_num

Get the col\_num

Priority: must

Request Parameters:

value: the new value of the col\_num

Request body: N/A

Response Status: 200, 404, 500

Response body: { value y:value, success:boolean}

### 3.3.6 PUT / item\_visit/:id

Update the item\_visit

Priority: must

Request Parameters:

Request body: N/A

Response Status: 200, 404, 500

Response body: { value: new value, success:boolean}

### 3.3.7 GET / item\_visit

Get An array of items visit

Priority: must

Request Parameters:

Start(optional): start index (sorted by priority, good, bad, and account), 0 base

End(optional): end index

Limit(optional): limit the number of items

Request body: N/A

Response Status: 200, 404, 500

Response body: An array of items visit

### 3.3.8 GET /header

Get An array of items visit

Priority: must

Request Parameters: N/A

Request body: N/A

Response Status: 200, 404, 500

Response body: A JSON file of header information

### 3.3.9 PUT /header

Update the header’s information

Priority: must

Request Parameters: N/A

Request body: A json file contains the header’s informtion

Response Status: 200, 404, 500

Response body: {,success: boolean}

# 4. Design

Node.js

mongodb

redis

View

ejs, jquery

controller

model

Crawler, java