## **1.安装**

### **1.1.安装mysql**

* [mysql](https://dev.mysql.com/downloads/mysql/)

### **1.2.安装redis**

* [redis](https://redis.io/)

### **1.3.安装nginx**

* [nginx](http://nginx.org/en/download.html)

## **2. 配置nginx**

### **2.1 nginx.conf**

http {

include mime.types;

default\_type application/octet-stream;

log\_format main '$time\_iso8601 - - $remote\_addr $http\_host $status $request\_time $request\_length $body\_bytes\_sent 15d04347-be16-b9ab-0029-24e4b6645950 - - 9689c3ea-5155-2df7-a719-e90d2dedeb2c 937ba755-116a-18e6-0735-312cba23b00c - - $request\_uri $http\_user\_agent - sample=-&\_UC\_agent=-&device\_id=-&- - - -';

access\_log logs/access.log main;

server {

listen 80;

server\_name localhost;

if ($time\_iso8601 ~ "^(\d{4})-(\d{2})-(\d{2})T(\d{2}):(\d{2})") {

set $year $1;

set $month $2;

set $day $3;

set $hour $4;

set $minute $5;

}

access\_log logs/$year$month-$day-$hour-$minute.log main;

location / {

root html;

index index.html index.htm;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

| **字段** | **含义** |
| --- | --- |
| $time\_iso8601 | 服务器时间的ISO 8610格式 |
| $remote\_addr | 客户端地址 |
| $http\_host | 主机名 |
| $status | HTTP响应代码 |
| $request\_time | 处理客户端请求使用的时间 |
| $request\_length | 请求的长度 |
| $body\_bytes\_sent | 传输给客户端的字节数 |
| $request\_uri | 包含一些客户端请求参数的原始URI |
| $http\_user\_agent | 客户端用户代理 |

## **3. 下载项目**

git clone https://gitee.com/zhufengpeixun/monitor.git

## **4.埋点上报**

### **4.1 安装**

cd monitor/sdk

cnpm install

### **4.2 配置**

window.dt && dt.set({

pid: 'template', // [必填]项目id,

uuid: 'uuid', // [可选]设备唯一id, 用于计算uv数&设备分布. 一般在cookie中可以取到, 没有uuid可用设备macidfa/imei替代. 或者在storage的key中存入随机数字, 模拟设备唯一id.

ucid: 'ucid', // [可选]用户ucid, 用于发生异常时追踪用户信息, 一般在cookie中可以取到, 没有可传空字符串

is\_test: true, // 是否为测试数据, 默认为false(测试模式下打点数据仅供浏览, 不会展示在系统中)

record: {

time\_on\_page: true, // 是否监控用户在线时长数据, 默认为true

performance: true, // 是否监控页面载入性能, 默认为true

js\_error: true, // 是否监控页面报错信息, 默认为true

// 配置需要监控的页面报错类别, 仅在js\_error为true时生效, 默认均为true(可以将配置改为false, 以屏蔽不需要报的错误类别)

js\_error\_report\_config: {

ERROR\_RUNTIME: true, // js运行时报错

ERROR\_SCRIPT: true, // js资源加载失败

ERROR\_STYLE: true, // css资源加载失败

ERROR\_IMAGE: true, // 图片资源加载失败

ERROR\_AUDIO: true, // 音频资源加载失败

ERROR\_VIDEO: true, // 视频资源加载失败

ERROR\_CONSOLE: true, // vue运行时报错

ERROR\_TRY\_CATCH: true, // 未catch错误

// 自定义检测函数, 上报前最后判断是否需要报告该错误

// 回调函数说明

// 传入参数 =>

// desc: 字符串, 错误描述

// stack: 字符串, 错误堆栈信息

// 返回值 =>

// true : 上报打点请求

// false : 不需要上报

checkErrrorNeedReport: **function**(desc, stack){

**return** true

}

}

},

// 业务方的js版本号, 会随着打点数据一起上传, 方便区分数据来源

// 可以不填, 默认为1.0.0

version: '1.0.0',

// test.com/detail/1.html

// test.com/detail/2.html

// test.com/detail/3.html

// 这种页面来说, 虽然url不同, 但他们本质上是同一个页面

// 因此需要业务方传入一个处理函数, 根据当前url解析出真实的页面类型

getPageType: **function**(location){ **return** `${location.host}${location.pathname}` }

})

### **4.3 监控类型**

#### **4.3.1 time\_on\_page(用户在线时长统计)**

* t\_r\_duration\_distribution

// 用户在线时长统计**const** OFFLINE\_MILL = 15 \* 60 \* 1000 // 15分钟不操作认为不在线**const** SEND\_MILL = 5 \* 1000 // 每5s打点一次

**let** lastTime = Date.now()window.addEventListener('click', () => {

// 检查是否监控用户在线时长

**const** isTimeOnPageFlagOn = \_.get(

commonConfig,

['record', 'time\_on\_page'],

\_.get(DEFAULT\_CONFIG, ['record', 'time\_on\_page'])

)

**const** isOldTimeOnPageFlagOn = \_.get(commonConfig, ['online'], false)

**const** needRecordTimeOnPage = isTimeOnPageFlagOn || isOldTimeOnPageFlagOn

**if** (needRecordTimeOnPage === false) {

debugLogger(`config.record.time\_on\_page值为false, 跳过停留时长打点`)

**return**

}

**const** now = Date.now();

**const** duration = now - lastTime;

**if** (duration > OFFLINE\_MILL) {

lastTime = Date.now()

} **else** **if** (duration > SEND\_MILL) {

lastTime = Date.now()

debugLogger('发送用户留存时间埋点, 埋点内容 => ', { duration\_ms: duration })

// 用户在线时长

log.product(10001, { duration\_ms: duration });

}

}, false)

{

"d": {

"type": "product",

"code": 10001,

"detail": {

"duration\_ms": 21553

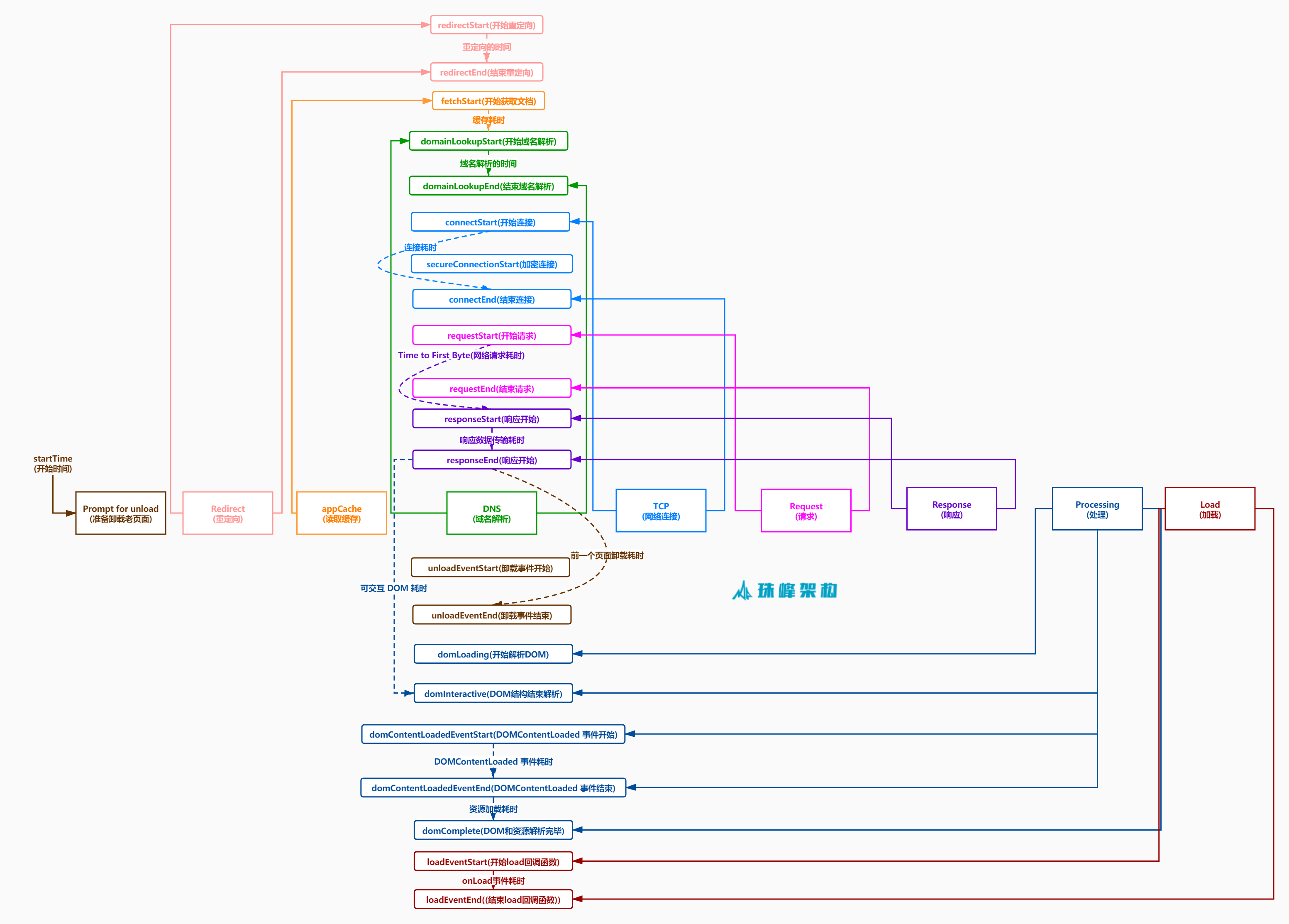
},

"extra": {}

}

}

#### **4.3.2 performance(页面载入性能)**



#### **4.3.2.1 sdk\src\index.js**

sdk\src\index.js

window.onload = () => {

// 检查是否监控性能指标

**const** isPerformanceFlagOn = \_.get(

commonConfig,

['record', 'performance'],

\_.get(DEFAULT\_CONFIG, ['record', 'performance'])

)

**const** isOldPerformanceFlagOn = \_.get(commonConfig, ['performance'], false)

**const** needRecordPerformance = isPerformanceFlagOn || isOldPerformanceFlagOn

**if** (needRecordPerformance === false) {

debugLogger(`config.record.performance值为false, 跳过性能指标打点`)

**return**

}

**const** performance = window.performance

**if** (!performance) {

// 当前浏览器不支持

console.log('你的浏览器不支持 performance 接口')

**return**

}

**const** times = performance.timing.toJSON()

debugLogger('发送页面性能指标数据, 上报内容 => ', {

...times,

url: `${window.location.host}${window.location.pathname}`

})

log('perf', 20001, {

...times,

url: `${window.location.host}${window.location.pathname}`

})

}

{

"d": {

"type": "perf",

"code": 20001,

"detail": {

"connectStart": 1592020165386,

"navigationStart": 1592020165383,

"loadEventEnd": 0,

"domLoading": 1592020165401,

"secureConnectionStart": 0,

"fetchStart": 1592020165386,

"domContentLoadedEventStart": 1592020165630,

"responseStart": 1592020165393,

"responseEnd": 1592020165394,

"domInteractive": 1592020165630,

"domainLookupEnd": 1592020165386,

"redirectStart": 0,

"requestStart": 1592020165387,

"unloadEventEnd": 1592020165398,

"unloadEventStart": 1592020165397,

"domComplete": 1592020165630,

"domainLookupStart": 1592020165386,

"loadEventStart": 1592020165630,

"domContentLoadedEventEnd": 1592020165630,

"redirectEnd": 0,

"connectEnd": 1592020165386,

"url": "localhost:9000/"

},

"extra": {}

}

}

#### **4.3.3 js\_error(页面报错)**

| **错误类型** | **含义** |
| --- | --- |
| ERROR\_RUNTIME | js运行时报错 |
| ERROR\_SCRIPT | js资源加载失败 |
| ERROR\_IMAGE | 图片资源加载失败 |
| ERROR\_AUDIO | 音频资源加载失败 |
| ERROR\_VIDEO | 视频资源加载失败 |
| ERROR\_CONSOLE | vue运行时报错 |
| ERROR\_TRY\_CATCH | 未catch错误 |

##### **4.3.3.1 ERROR\_RUNTIME**

window.addEventListener('error', **function** (event) {

// 过滤 target 为 window 的异常，避免与上面的 onerror 重复

**var** errorTarget = event.target

**if** (errorTarget !== window && errorTarget.nodeName && LOAD\_ERROR\_TYPE[errorTarget.nodeName.toUpperCase()]) {

handleError(formatLoadError(errorTarget))

} **else** {

// onerror会被覆盖, 因此转为使用Listener进行监控

**let** { message, filename, lineno, colno, error } = event

handleError(formatRuntimerError(message, filename, lineno, colno, error))

}

}, true)

**function** **formatRuntimerError** (message, source, lineno, colno, error) {

**return** {

type: ERROR\_RUNTIME,

desc: message + ' at ' + source + ':' + lineno + ':' + colno,

stack: error && error.stack ? error.stack : 'no stack'

}

}

console.log(a.b);

{

"d": {

"type": "error",

"code": 7,

"detail": {

"error\_no": "页面报错\_JS\_RUNTIME\_ERROR",

"url": "localhost:9000/"

},

"extra": {

"desc": "Uncaught ReferenceError: a is not defined at http://localhost:9000/:53:21",

"stack": "ReferenceError: a is not defined\n at http://localhost:9000/:53:21"

}

}

}

##### **4.3.3.2 LOAD\_ERROR**

<script src="/error.js"></script>

<link rel="stylesheet" href="/error.css">

<img src="/error.gif" alt="">

<audio src="/error.mp3"></audio>

<video src="/error.mp4"></video>

**function** **formatLoadError** (errorTarget) {

**return** {

type: LOAD\_ERROR\_TYPE[errorTarget.nodeName.toUpperCase()],

desc: errorTarget.baseURI + '@' + (errorTarget.src || errorTarget.href),

stack: 'no stack'

}

}

{

"d": {

"type": "error",

"code": 7,

"detail": {

"error\_no": "页面报错\_SCRIPT\_LOAD\_ERROR",

"url": "localhost:9000/"

},

"extra": {

"desc": "http://localhost:9000/@http://localhost:9000/error.js",

"stack": "no stack"

}

}

}

{

"d": {

"type": "error",

"code": 7,

"detail": {

"error\_no": "页面报错\_CSS\_LOAD\_ERROR",

"url": "localhost:9000/"

},

"extra": {

"desc": "http://localhost:9000/@http://localhost:9000/error.css",

"stack": "no stack"

}

}

}

{

"d": {

"type": "error",

"code": 7,

"detail": {

"error\_no": "页面报错\_IMAGE\_LOAD\_ERROR",

"url": "localhost:9000/"

},

"extra": {

"desc": "http://localhost:9000/@http://localhost:9000/error.gif",

"stack": "no stack"

}

}

}

{

"d": {

"type": "error",

"code": 7,

"detail": {

"error\_no": "页面报错\_VIDEO\_LOAD\_ERROR",

"url": "localhost:9000/"

},

"extra": {

"desc": "http://localhost:9000/@http://localhost:9000/error.mp4",

"stack": "no stack"

}

}

}

{

"d": {

"type": "error",

"code": 7,

"detail": {

"error\_no": "页面报错\_AUDIO\_LOAD\_ERROR",

"url": "localhost:9000/"

},

"extra": {

"desc": "http://localhost:9000/@http://localhost:9000/error.mp3",

"stack": "no stack"

}

}

}

##### **4.3.3.2 LOAD\_ERROR**

console.error('vue error');

{

"d": {

"type": "error",

"code": 7,

"detail": {

"error\_no": "页面报错\_CONSOLE\_ERROR",

"url": "localhost:9000/"

},

"extra": {

"desc": "vue error"

}

}

}

##### **4.3.3.3 ERROR\_TRY\_CATCH**

**function** **exec**(){

**throw** **new** Error('未捕获错误');

}window.dt.tryJS.wrap(exec);

exec.\_wrapped();

{

"d": {

"type": "error",

"code": 7,

"detail": {

"error\_no": "页面报错\_TRY\_CATCH\_ERROR",

"url": "localhost:9000/"

},

"extra": {

"desc": "未捕获错误",

"stack": "Error: 未捕获错误\n at Function.exec (http://localhost:9000/:56:17)\n at Function.func.\_wrapped (webpack-internal:///./src/js-tracker/try.js:31:21)\n at http://localhost:9000/:59:14"

}

}

}

#### **4.3.4 手工上报**

//错误上报**export** **const** Elog = log.error = (code, detail, extra) => {

**return** log('error', code, detail, extra)

}//产品数据上报**export** **const** Plog = log.product = (code, detail, extra) => {

**return** log('product', code, detail, extra)

}//普通信息上报**export** **const** Ilog = log.info = (code, detail, extra) => {

**return** log('info', code, detail, extra)

}

window.dt.product(19999, {}, {})

{

"d": {

"type": "product",

"code": 19999,

"detail": {},

"extra": {}

}

}

## **5.数据处理**

### **5.1 安装server**

cd server

npm install

### **5.2 配置**

#### **5.2.1 配置mysql**

server\src\configs\mysql.js

**const** development = {

host: '127.0.0.1',

port: '3306',

user: 'root',

password: '5f8b8a5d650637f8',

database: 'platform'

}

#### **5.2.2 配置redis**

server\src\configs\redis.js

**const** development = {

host: '127.0.0.1',

port: '6379'

}

### **5.3 数据解析**

### **5.3.1 数据存储**

### **5.3.1.1 表名说明**

* 所有表都以t\_开头
* 原始表添加\_o后缀，即t\_o\_
* 结果表添加-r后续，即t\_r\_
* 表名、字段名默认使用下划线方式命名，不区分大小写
* 数据库编码字符集为utf8mb4
* 记录ID用unsigned bigint
* 如果字段名有关键字，需要加\_c前缀
* 所有表中必须有update\_time 和create\_time字段

### **5.3.1.2 数据库表**

t\_o\_project 项目名

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '项目id' |  |
| display\_name varchar(50) NOT NULL DEFAULT '' COMMENT '项目名称' |  |
| project\_name varchar(50) NOT NULL DEFAULT '' COMMENT '项目代号(替代项目id, 方便debug)' |  |
| c\_desc varchar(100) NOT NULL DEFAULT '' COMMENT '备注信息' |  |
| rate int(10) NOT NULL DEFAULT '10000' COMMENT '数据抽样比率' |  |
| is\_delete tinyint(1) NOT NULL DEFAULT '0' COMMENT '是否已删除(1 => 是, 0 => 否)' |  |
| create\_ucid varchar(20) NOT NULL DEFAULT '' COMMENT '创建人ucid' |  |
| update\_ucid varchar(20) NOT NULL DEFAULT '' COMMENT '更新人ucid' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_behavior\_distribution

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| code varchar(50) NOT NULL DEFAULT '' COMMENT '用户行为标识' |  |
| name varchar(50) NOT NULL DEFAULT '' COMMENT '用户点击名称' |  |
| url varchar(200) NOT NULL DEFAULT '' COMMENT '用户点击页面' |  |
| total\_count int(10) NOT NULL DEFAULT '0' COMMENT '点击总量' |  |
| count\_at\_time varchar(30) NOT NULL DEFAULT '' COMMENT '统计日期,格式根据统计尺度不同有三种可能, hour => YYYY-MM-DD\_HH, day => YYYY-MM-DD, month => YYYY-MM' |  |
| count\_type varchar(20) NOT NULL DEFAULT 'day' COMMENT '统计尺度(hour/day/month)' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布详情记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_duration\_distribution 用户停留时间表

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| total\_stay\_ms bigint(20) NOT NULL DEFAULT '0' COMMENT '总停留毫秒数' |  |
| total\_uv int(10) NOT NULL DEFAULT '0' COMMENT '总uv数(从uv表中获取)' |  |
| count\_at\_time varchar(30) NOT NULL DEFAULT '' COMMENT '统计日期,格式根据统计尺度不同有两种可能, day => YYYY-MM-DD, month => YYYY-MM' |  |
| count\_type varchar(20) NOT NULL DEFAULT 'day' COMMENT '统计尺度(day/month)' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布详情记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_http\_error\_distribution

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| total\_count int(10) NOT NULL DEFAULT '0' COMMENT 'http响应码总数' |  |
| http\_code\_2xx\_count int(10) NOT NULL DEFAULT '0' COMMENT 'http响应码2xx总数' |  |
| http\_code\_3xx\_count int(10) NOT NULL DEFAULT '0' COMMENT 'http响应码3xx总数' |  |
| http\_code\_4xx\_count int(10) NOT NULL DEFAULT '0' COMMENT 'http响应码4xx总数' |  |
| http\_code\_5xx\_count int(10) NOT NULL DEFAULT '0' COMMENT 'http响应码5xx总数' |  |
| http\_code\_other\_count int(10) NOT NULL DEFAULT '0' COMMENT '其他http响应码总数' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布详情记录id' |  |
| count\_at\_time varchar(30) NOT NULL DEFAULT '' COMMENT '统计日期,格式根据统计尺度不同有三种可能, hour => YYYY-MM-DD\_HH, day => YYYY-MM-DD, month => YYYY-MM' |  |
| count\_type varchar(20) NOT NULL DEFAULT 'hour' COMMENT '统计尺度(hour/day/month)' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |

t\_r\_page\_view

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| total\_count int(10) NOT NULL DEFAULT '0' COMMENT 'pv数' |  |
| count\_at\_time varchar(30) NOT NULL DEFAULT '' COMMENT '统计日期,格式根据统计尺度不同有三种可能, hour => YYYY-MM-DD\_HH, day => YYYY-MM-DD, month => YYYY-MM' |  |
| count\_type varchar(20) NOT NULL DEFAULT 'hour' COMMENT '统计尺度(hour/day/month)' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_system\_browser

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| browser varchar(20) NOT NULL DEFAULT '' COMMENT '浏览器品牌' |  |
| browser\_version varchar(50) NOT NULL DEFAULT '' COMMENT '浏览器详情' |  |
| total\_count int(10) unsigned NOT NULL DEFAULT '0' COMMENT '数量总和' |  |
| count\_at\_month varchar(15) NOT NULL DEFAULT '' COMMENT '统计时间段, YYYY-MM格式 => 2018-09' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_system\_runtime\_version

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| runtime\_version varchar(50) NOT NULL DEFAULT '' COMMENT '应用版本号' |  |
| total\_count int(10) unsigned NOT NULL DEFAULT '0' COMMENT '数量总和' |  |
| count\_at\_month varchar(15) NOT NULL DEFAULT '' COMMENT '统计时间段, YYYY-MM格式 => 2018-09' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_system\_device

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| device\_vendor varchar(50) NOT NULL DEFAULT '' COMMENT '设备制造商' |  |
| device\_model varchar(50) NOT NULL DEFAULT '' COMMENT '设备详情' |  |
| total\_count int(10) unsigned NOT NULL DEFAULT '0' COMMENT '数量总和' |  |
| count\_at\_month varchar(15) NOT NULL DEFAULT '' COMMENT '统计时间段, YYYY-MM格式 => 2018-09' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_system\_os

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| os varchar(50) NOT NULL DEFAULT '' COMMENT '操作系统名' |  |
| os\_version varchar(50) NOT NULL DEFAULT '' COMMENT '操作系统版本' |  |
| total\_count int(10) unsigned NOT NULL DEFAULT '0' COMMENT '数量总和' |  |
| count\_at\_month varchar(15) NOT NULL DEFAULT '' COMMENT '统计时间段, YYYY-MM格式 => 2018-09' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_unique\_view

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| total\_count int(10) NOT NULL DEFAULT '0' COMMENT 'uv数' |  |
| count\_at\_time varchar(30) NOT NULL DEFAULT '' COMMENT '统计日期,格式根据统计尺度不同有三种可能, hour => YYYY-MM-DD\_HH, day => YYYY-MM-DD, month => YYYY-MM' |  |
| count\_type varchar(20) NOT NULL DEFAULT 'hour' COMMENT '统计尺度(hour/day/month)' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_o\_alarm\_config

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT |  |
| project\_id bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '要报警项目id' |  |
| owner\_ucid varchar(20) NOT NULL DEFAULT '' COMMENT '项目所有人id' |  |
| error\_type varchar(20) NOT NULL DEFAULT '' COMMENT '报警错误类型' |  |
| error\_name varchar(255) NOT NULL DEFAULT '' COMMENT '要报警错误名字' |  |
| time\_range\_s int(20) unsigned NOT NULL DEFAULT '0' COMMENT '报警时间范围\_秒' |  |
| max\_error\_count int(20) unsigned NOT NULL DEFAULT '0' COMMENT '报警错误数阈值' |  |
| alarm\_interval\_s bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '报警时间间隔\_秒' |  |
| is\_enable tinyint(1) unsigned NOT NULL DEFAULT '1' COMMENT '是否开启本条报警配置1：是0：否' |  |
| note varchar(255) NOT NULL DEFAULT '' COMMENT '配置说明' |  |
| is\_delete tinyint(1) unsigned NOT NULL DEFAULT '0' COMMENT '是否删除(1 => 是, 0 => 否)' |  |
| create\_ucid varchar(20) NOT NULL DEFAULT '' COMMENT '创建此记录的人' |  |
| update\_ucid varchar(20) NOT NULL DEFAULT '' COMMENT '更新此记录的人' |  |
| create\_time bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '创建此记录的时间' |  |
| update\_time bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '更新此记录的时间' |  |

t\_o\_user

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| ucid varchar(50) NOT NULL DEFAULT '' COMMENT '贝壳ucid' |  |
| account varchar(50) NOT NULL DEFAULT '' COMMENT '账户名,不能重复' |  |
| email varchar(50) NOT NULL DEFAULT '' COMMENT '邮箱' |  |
| password\_md5 varchar(32) NOT NULL DEFAULT '' COMMENT 'md5后的password' |  |
| nickname varchar(20) NOT NULL DEFAULT '' COMMENT '昵称' |  |
| role varchar(50) NOT NULL DEFAULT 'dev' COMMENT '角色(dev => 开发者,admin => 管理员)' |  |
| register\_type varchar(20) NOT NULL DEFAULT 'site' COMMENT '注册类型(site => 网站注册, third => 第三方登录)' |  |
| avatar\_url varchar(200) NOT NULL DEFAULT '<http://ww1.sinaimg.cn/large/00749HCsly1fwofq2t1kaj30qn0qnaai.jpg'> COMMENT '头像地址, 默认使用贝壳logo' |  |
| mobile varchar(20) NOT NULL DEFAULT '' COMMENT '手机号' |  |
| is\_delete tinyint(1) unsigned NOT NULL DEFAULT '0' COMMENT '是否删除(1 => 是, 0 => 否)' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_o\_project\_member

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT |  |
| project\_id bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| ucid varchar(20) NOT NULL DEFAULT '' COMMENT '项目参与者ucid' |  |
| role varchar(20) NOT NULL DEFAULT '' COMMENT '参与者角色(owner => 组长, dev => 成员)' |  |
| need\_alarm tinyint(1) unsigned NOT NULL DEFAULT '0' COMMENT '是否需要报警(0 => 不需要, 1 => 需要)' |  |
| is\_delete tinyint(1) unsigned NOT NULL DEFAULT '0' COMMENT '是否已删除(0 => 未删除, 1 => 已删除)' |  |
| create\_ucid varchar(20) NOT NULL DEFAULT '' COMMENT '创建者ucid' |  |
| update\_ucid varchar(20) NOT NULL DEFAULT '' COMMENT '更新者ucid' |  |
| create\_time bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '创建此记录的时间' |  |
| update\_time bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '更新此记录的时间' |  |

t\_r\_new\_user\_summary

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '项目id' |  |
| total\_count int(10) NOT NULL DEFAULT '0' COMMENT '数量总和' |  |
| count\_at\_time varchar(20) NOT NULL DEFAULT '' COMMENT '统计日期,格式根据统计尺度不同三两种可能, hour => YYYY-MM-DD\_HH, day => YYYY-MM-DD, month => YYYY-MM' |  |
| count\_type varchar(10) NOT NULL DEFAULT 'day' COMMENT '统计尺度(hour/day/month)' |  |
| city\_distribute\_id bigint(10) NOT NULL DEFAULT 0 COMMENT '城市分布详情记录id' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '更新时间' |  |

t\_r\_alarm\_log

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| project\_id bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '要报警项目id' |  |
| config\_id bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '报警配置id' |  |
| send\_at bigint(20) NOT NULL DEFAULT '0' COMMENT '报警时间戳' |  |
| error\_type varchar(20) NOT NULL DEFAULT '' COMMENT '错误类型' |  |
| error\_name varchar(255) NOT NULL DEFAULT '' COMMENT '错误名字' |  |
| message varchar(500) NOT NULL DEFAULT '' COMMENT '报警信息' |  |
| create\_time bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '创建时间' |  |
| update\_time bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '更新时间' |  |

t\_o\_monitor\_1\_202005

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| error\_type varchar(20) NOT NULL DEFAULT '' COMMENT '异常类型(目前是四种,分别对应前端的四种报错类型: api\_data =>前端数据结构报警, start\_process => 启动过程异常, load\_wv => Url加载空服务报警, api\_code => 请求接口异常报警)' |  |
| error\_name varchar(255) NOT NULL DEFAULT '' COMMENT '错误名/错误代码,用于细分错误类别' |  |
| http\_code int(10) NOT NULL DEFAULT '0' COMMENT 'http状态码, 没有则为0' |  |
| monitor\_ext\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '详情id' |  |
| during\_ms int(10) unsigned NOT NULL DEFAULT '0' COMMENT '接口请求时长, 单位毫秒' |  |
| request\_size\_b int(10) unsigned NOT NULL DEFAULT '0' COMMENT '接口请求体积, 单位b' |  |
| response\_size\_b int(10) unsigned NOT NULL DEFAULT '0' COMMENT '接口响应体积, 单位b' |  |
| url varchar(255) NOT NULL DEFAULT '' COMMENT '发生异常的url' |  |
| country varchar(10) NOT NULL DEFAULT '' COMMENT '所属国家' |  |
| province varchar(15) NOT NULL DEFAULT '' COMMENT '所属省份' |  |
| city varchar(15) NOT NULL DEFAULT '' COMMENT '所属城市' |  |
| log\_at bigint(20) NOT NULL DEFAULT '0' COMMENT '日志记录时间' |  |
| md5 char(32) NOT NULL DEFAULT '' COMMENT '记录生成MD5' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_o\_monitor\_ext\_1\_202005

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| ext\_json text COMMENT '异常记录扩展信息' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_o\_uv\_record\_1\_202005

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '项目id' |  |
| uuid varchar(50) NOT NULL DEFAULT '' COMMENT '设备唯一标识' |  |
| country varchar(10) NOT NULL DEFAULT '' COMMENT '所属国家' |  |
| province varchar(15) NOT NULL DEFAULT '' COMMENT '所属省份' |  |
| city varchar(15) NOT NULL DEFAULT '' COMMENT '所属城市' |  |
| visit\_at\_hour varchar(20) NOT NULL DEFAULT '' COMMENT '访问日期, 数据格式为 YYYY-MM-DD\_HH demo => 2018-08-02\_23' |  |
| pv\_count int(10) NOT NULL DEFAULT '0' COMMENT '时间段内同一uuid访问次数, 用于计算总pv' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_city\_distribution\_1\_202005

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| city\_distribute\_json text COMMENT '扩展字段' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_r\_performance\_1\_202005

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| sum\_indicator\_value bigInt(10) NOT NULL DEFAULT '0' COMMENT '性能指标求和' |  |
| pv bigInt(10) NOT NULL DEFAULT '0' COMMENT '性能指标pv数,用于计算平均时长' |  |
| indicator varchar(50) NOT NULL DEFAULT '' COMMENT '性能指标:DNS响应时间/TCP时间/404数量/etc' |  |
| url varchar(255) NOT NULL DEFAULT '' COMMENT '页面url' |  |
| city\_distribute\_id bigint(20) NOT NULL DEFAULT '0' COMMENT '城市分布详情记录id' |  |
| count\_at\_time varchar(20) NOT NULL DEFAULT '' COMMENT '统计日期,格式根据统计尺度不同有四种可能, minute => YYYY-MM-DD\_HH:mm, hour => YYYY-MM-DD\_HH, day => YYYY-MM-DD, month => YYYY-MM' |  |
| count\_type varchar(10) NOT NULL DEFAULT 'minute' COMMENT '统计尺度(minute/hour/day/month)' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '更新时间' |  |

t\_o\_system\_collection\_1

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| uuid varchar(50) NOT NULL DEFAULT '' COMMENT '设备唯一标识' |  |
| browser varchar(50) NOT NULL DEFAULT '' COMMENT '浏览器品牌' |  |
| browser\_version varchar(100) NOT NULL DEFAULT '' COMMENT '浏览器版本详情' |  |
| engine varchar(100) NOT NULL DEFAULT '' COMMENT '内核名称' |  |
| engine\_version varchar(100) NOT NULL DEFAULT '' COMMENT '内核版本详情' |  |
| device\_vendor varchar(100) NOT NULL DEFAULT '' COMMENT '手机品牌' |  |
| device\_model varchar(100) NOT NULL DEFAULT '' COMMENT '手机型号' |  |
| os varchar(50) NOT NULL DEFAULT '' COMMENT '操作系统' |  |
| os\_version varchar(50) NOT NULL DEFAULT '' COMMENT '操作系统详情' |  |
| country varchar(10) NOT NULL DEFAULT '' COMMENT '所属国家' |  |
| province varchar(15) NOT NULL DEFAULT '' COMMENT '所属省份' |  |
| city varchar(15) NOT NULL DEFAULT '' COMMENT '所属城市' |  |
| runtime\_version varchar(50) NOT NULL DEFAULT '' COMMENT '应用版本' |  |
| visit\_at\_month varchar(20) NOT NULL DEFAULT '' COMMENT '访问日期, 数据格式为 YYYY-MM demo => 2018-09' |  |
| log\_at bigint(20) NOT NULL DEFAULT '0' COMMENT '日志记录时间' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '数据库更新时间' |  |

t\_o\_user\_first\_login\_at\_1

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| ucid varchar(20) NOT NULL DEFAULT '' COMMENT '用户id' |  |
| first\_visit\_at bigint(20) NOT NULL DEFAULT '0' COMMENT '用户首次访问时间' |  |
| country varchar(10) NOT NULL DEFAULT '' COMMENT '所属国家' |  |
| province varchar(15) NOT NULL DEFAULT '' COMMENT '所属省份' |  |
| city varchar(15) NOT NULL DEFAULT '' COMMENT '所属城市' |  |
| create\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '创建时间' |  |
| update\_time bigint(20) NOT NULL DEFAULT '0' COMMENT '更新时间' |  |

t\_r\_error\_summary\_1\_202005

| **字段** | **备注** |
| --- | --- |
| id bigint(20) unsigned NOT NULL AUTO\_INCREMENT COMMENT '记录id' |  |
| error\_type varchar(20) NOT NULL DEFAULT '' COMMENT '错误类型' |  |
| error\_name varchar(255) NOT NULL DEFAULT '' COMMENT '错误名字' |  |
| url\_path varchar(255) NOT NULL DEFAULT '' COMMENT '错误URL\_PATH' |  |
| city\_distribution\_id bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT '城市分布详情id' |  |
| count\_at\_time varchar(20) NOT NULL DEFAULT '' COMMENT '统计时间(按类型不同,day:YYYY-MM-DD;hour:YYYY-MM-DD\_HH;minute:YYYY-MM-DD\_HH:mm)' |  |
| count\_type varchar(10) NOT NULL DEFAULT 'day' COMMENT '统计类型(day天,hour小时,minute分)' |  |
| error\_count int(10) NOT NULL DEFAULT '0' COMMENT '数量总和' |  |
| create\_time int(20) NOT NULL DEFAULT '0' COMMENT '创建时间' |  |
| update\_time int(20) NOT NULL DEFAULT '0' COMMENT '更新时间' |  |

### **5.3.2 指令**

#### **5.3.2.1 指令帮助**

node dist/fee.js -hUsage:

command [arguments] [options]

Global Options:

--env Set NODE\_ENV before running the commands

--no-ansi Disable colored output

Available Commands:

Command

Command:Demo 解析nginx日志, 分析pv

CreateCache

CreateCache:UpdatePerOneMinute [每10分钟执行一次] 主动调用方法, 更新Redis缓存, 每10分钟更新一次

Parse

Parse:Device [按天] 解析nginx日志, 分析指定时间范围Device

Parse:MenuClick [按天] 解析nginx日志, 用户点击情况

Parse:Monitor [按分钟] 解析nginx日志, 分析Monitor

Parse:Performance [按小时] 解析nginx日志, 分析分钟级别的指定时间范围内的性能指标

Parse:TimeOnSiteByHour [按小时] 解析nginx日志, 分析记录指定时间范围内用户停留时长

Parse:UV [按小时] 解析nginx日志, 分析记录指定时间范围内的uv

Parse:UserFirstLoginAt [按天] 解析nginx日志, 记录用户首次登陆时间

SaveLog

SaveLog:Nginx 每一分钟读取Nginx日志文件，并解析

Summary

Summary:Error [按分钟/按小时/按天] 根据历史数据, 汇总分析错误数

Summary:HttpError [按天/按月] 基于数据表做统计, 统计http error分布情况

Summary:NewUser [按小时/按天/按月] 根据历史数据, 汇总分析记录指定时间范围内的新增用户数

Summary:Performance [按小时/按天/按月] 根据历史数据, 汇总分析记录指定时间范围内的性能指标数据

Summary:SystemBrowser [按月] 基于数据库统计浏览器占比

Summary:SystemDevice [按月] 基于数据库统计设备占比

Summary:SystemOS [按月]基于数据库统计操作系统占比

Summary:SystemRuntimeVersion [按月] 基于数据库统计浏览器占比

Summary:TimeOnSite [按天/按月] 根据历史数据, 汇总分析记录指定时间范围内用户停留时长

Summary:UV [按小时/按天/按月] 根据历史数据, 汇总分析记录指定时间范围内的uv

Task

Task:Manager 任务调度主进程, 只能启动一次

Utils

Utils:CleanOldLog 只保留当前月内数据, 每月20号之后自动删除上个月数据

Utils:GenerateSQL 生成项目在指定日期范围内的建表SQL

Utils:TemplateSQL 生成项目在指定日期范围内的建表SQL

Utils:Test 专业粘贴调试代码

Utils:TestUC 测试UC接口

WatchDog

WatchDog:Alarm [根据报警配置] 监测每一条报警配置对应的项目错误

WatchDog:Saas [按分钟] 检查最近5分钟内错误数是否超出阈值, 自动报警

#### **5.3.2.2 初始化数据库**

* 创建platform数据库
* npm run watch
* node dist/fee.js Utils:GenerateSQL 1 '2020-06' '2020-06' > init.sql

//id, 抽样比率, 项目名(展示), 项目id, 负责人信息

REPLACE INTO `t\_o\_project` (`id`, `rate`, `display\_name`, `project\_name`, `c\_desc`, `is\_delete`, `create\_ucid`, `update\_ucid`, `create\_time`, `update\_time`) VALUES (1, 100, '示例项目', 'template', '负责人', 0, '', '', 0, 0);

#### **5.3.2.3 SaveLog:Nginx**

* 每一分钟读取Nginx日志文件，并解析

server\src\library\nginx\index.js

**let** logPath = appConfig.absoluteLogPath

server\src\configs\app.js

**const** development = {

name: 'fee监控平台开发环境',

port: 3000,

proxy: false,

absoluteLogPath: path.resolve(\_\_dirname, '../../', 'log')

}

node dist/fee.js SaveLog:Nginx

收到数据, 当前共记录1/1条数据

清洗后的日志路径 C:\afirst\monitor\server\log\nginx\raw\month\_202006\day\_13\16\14.log

清洗后的日志路径 C:\afirst\monitor\server\log\nginx\json\month\_202006\day\_13\16\14.log

{

"type": "perf",

"code": 20001,

"detail": {

"connectStart": 1592036791666,

"navigationStart": 1592036791361,

"loadEventEnd": 0,

"domLoading": 1592036791677,

"secureConnectionStart": 0,

"fetchStart": 1592036791362,

"domContentLoadedEventStart": 1592036791976,

"responseStart": 1592036791669,

"responseEnd": 1592036791670,

"domInteractive": 1592036791976,

"domainLookupEnd": 1592036791365,

"redirectStart": 0,

"requestStart": 1592036791667,

"unloadEventEnd": 1592036791674,

"unloadEventStart": 1592036791674,

"domComplete": 1592036791976,

"domainLookupStart": 1592036791365,

"loadEventStart": 1592036791976,

"domContentLoadedEventEnd": 1592036791976,

"redirectEnd": 0,

"connectEnd": 1592036791667,

"url": "localhost:9000/"

},

"extra": {},

"common": {

"pid": "template",

"uuid": "uuid2",

"ucid": "ucid",

"is\_test": true,

"record": {

"time\_on\_page": true,

"performance": true,

"js\_error": true,

"js\_error\_report\_config": {

"ERROR\_RUNTIME": true,

"ERROR\_SCRIPT": true,

"ERROR\_STYLE": true,

"ERROR\_IMAGE": true,

"ERROR\_AUDIO": true,

"ERROR\_VIDEO": true,

"ERROR\_CONSOLE": true,

"ERROR\_TRY\_CATCH": true

}

},

"version": "1.0.0",

"test": "b47ca710747e96f1c523ebab8022c19e9abaa56b",

"timestamp": 1592036791977,

"runtime\_version": "1.0.0",

"sdk\_version": "1.0.40",

"page\_type": "localhost:9000/"

},

"md5": "382eecc357bb7c6629e139eb7eb6509e",

"project\_id": 1,

"project\_name": "template",

"time": 1592036792,

"ua": {

"ua": "Mozilla/5.0 (iPhone; CPU iPhone OS 13\_2\_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.0.3 Mobile/15E148 Safari/604.1",

"browser": {

"name": "Mobile Safari",

"version": "13.0.3",

"major": "13"

},

"engine": {

"name": "WebKit",

"version": "605.1.15"

},

"os": {

"name": "iOS",

"version": "13.2.3"

},

"device": {

"vendor": "Apple",

"model": "iPhone",

"type": "mobile"

},

"cpu": {}

},

"ip": "59.109.146.215",

"country": "中国",

"province": "北京",

"city": "北京"

}

#### **5.3.2.4 Parse:Performance**

* [按小时] 解析nginx日志, 分析分钟级别的指定时间范围内的性能指标

node dist/fee.js Parse:Performance 2020-06-13\_16:26 2020-06-13\_16:26

{

[项目ID,url,指标,汇总的时间]:{

[国家,省,市],value：{指标和:614,pv和:1}

}

}

{

[1,"localhost:9000/","tcp\_connect\_ms","2020-06-13\_16:26"]:{

["中国","北京","北京"],value：{"sum\_indicator\_value":614,"pv":1}

}

}

#### **5.3.2.5 Summary:Performance**

* [按小时/按天/按月] 根据历史数据, 汇总分析记录指定时间范围内的性能指标数据

node dist/fee.js Summary:Performance "2020-06-13 16" hour

node dist/fee.js Summary:Performance 2020-06-13 day

node dist/fee.js Summary:Performance 2020-06 month

#### **5.3.2.5 Parse:TimeOnSiteByHour**

* [按小时] 解析nginx日志, 分析记录指定时间范围内用户停留时长
* t\_r\_city\_distribution\_1\_202006
* t\_r\_duration\_distribution

node dist/fee.js SaveLog:Nginx

node dist/fee.js Parse:TimeOnSiteByHour "2020-06-13 22:30" "2020-06-13 22:30"

#### **5.3.2.6 Summary:TimeOnSite**

* [按小时] 解析nginx日志, 分析记录指定时间范围内用户停留时长
* t\_r\_duration\_distribution

node dist/fee.js Summary:TimeOnSite "2020-06 13" day

node dist/fee.js Summary:TimeOnSite "2020-06" month

#### **5.3.2.7 Parse:Monitor**

* [按分钟] 解析nginx日志, 分析Monitor
* t\_o\_monitor\_1\_202006
* t\_o\_monitor\_ext\_1\_202006

node dist/fee.js Parse:Monitor "2020-06-13 22:35" "2020-06-13 22:35"

#### **5.3.2.8 Summary:Error**

* [按分钟/按小时/按天] 根据历史数据, 汇总分析错误数
* t\_r\_error\_summary\_1\_202006

node dist/fee.js Summary:Error "2020-06-13 22:35" minute

node dist/fee.js Summary:Error "2020-06-13 22" hour

node dist/fee.js Summary:Error "2020-06-13" day

#### **5.3.2.9 Parse:Device**

* [按天] 解析nginx日志, 分析指定时间范围Device
* t\_o\_system\_collection\_1

node dist/fee.js Parse:Device "2020-06-13 22:58" "2020-06-13 22:58"

projectMap={

1:{

"2020-06":{

"uuid":{

"browser":"Mobile Safari"

}

}

}

}

#### **5.3.2.10 Summary**

* Summary:SystemBrowser [按月] 基于数据库统计浏览器占比
* Summary:SystemDevice [按月] 基于数据库统计设备占比
* Summary:SystemOS [按月]基于数据库统计操作系统占比

Summary:SystemRuntimeVersion [按月] 基于数据库统计浏览器占比

t\_r\_system\_browser

* t\_r\_system\_device
* t\_r\_system\_os
* t\_r\_system\_runtime\_version

node dist/fee.js Summary:SystemBrowser "2020-06" month

node dist/fee.js Summary:SystemDevice "2020-06" month

node dist/fee.js Summary:SystemOS "2020-06" month

node dist/fee.js Summary:SystemRuntimeVersion "2020-06" month

## **6.计划任务**

* server\src\commands\task\manage.js

### **6.1 任务执行周期**

1. 每分钟一次(准实时)

1. 原始数据入库

1. 错误数据入库(延迟2分钟)

2. 按分钟统计

1. 错误数据统计(延迟2分钟)2. 每10分钟一次

1. 原始数据入库

1. uv

2. 页面性能指标

3. 用户停留时长

2. 按小时统计

1. uv

2. 新用户数

3. 页面性能指标

4. 错误数据统计3. 每小时一次

1. 原始数据入库

1. 设备数据

2. 用户点击

3. 首次登陆用户

2. 按天统计(当天)

1. uv

2. 新用户数

3. 页面性能指标

4. 错误数据统计

5. 用户停留时长4. 每六小时一次

1. 按天统计(昨日)

1. uv

2. 新用户数

3. 页面性能指标

4. 错误数据统计

5. 用户停留时长

2. 按月统计

1. uv

2. 新用户数

3. 页面性能指标

4. 错误数据统计

5. 用户停留时长

6. 操作系统分布

7. 设备分布

8. 浏览器分布

### **6.2 执行任务**

* [cron](https://cron.qqe2.com/)

node dist/fee.js Task:saveLog

**async** handle (args, options) {

**let** that = **this**

schedule.scheduleJob('0 \*/1 \* \* \* \* \*', **function** () {

that.log('registerTaskRepeatPer1Minute 开始执行')

that.execCommand('SaveLog:Nginx', []);

})

}

\* \* \* \* \* \*

┬ ┬ ┬ ┬ ┬ ┬

│ │ │ │ │ |

│ │ │ │ │ └ day **of** week (0 - 7) (0 or 7 is Sun)

│ │ │ │ └───── month (1 - 12)

│ │ │ └────────── day **of** month (1 - 31)

│ │ └─────────────── hour (0 - 23)

│ └──────────────────── minute (0 - 59)

└───────────────────────── second (0 - 59, OPTIONAL)

## **7.前台展示**

### **7.1 启动接口**

node dist/app.js

需要登录,也需要检验项目权限(Method: **get**) =>/api/behavior/menu

需要登录,也需要检验项目权限(Method: **get**) =>/api/behavior/online

需要登录,也需要检验项目权限(Method: **get**) =>/api/os

需要登录,也需要检验项目权限(Method: **get**) =>/api/browser/list

需要登录,也需要检验项目权限(Method: **get**) =>/api/browser

需要登录,也需要检验项目权限(Method: **get**) =>/api/browser/

需要登录,也需要检验项目权限(Method: **get**) =>/api/runtimeVersion

需要登录,也需要检验项目权限(Method: **get**) =>/api/error/viser/area/

需要登录,也需要检验项目权限(Method: **get**) =>/api/error/log/list

需要登录,也需要检验项目权限(Method: **get**) =>/api/error/distribution/url

需要登录,也需要检验项目权限(Method: **get**) =>/api/error/distribution/

需要登录,也需要检验项目权限(Method: **get**) =>/api/error/distribution/

需要登录,也需要检验项目权限(Method: **get**) =>/api/error/distribution/

不需要登录(Method: **get**) =>/api/log/content/test

需要登录,也需要检验项目权限(Method: post) =>/api/alarm/config/add

需要登录,也需要检验项目权限(Method: **get**) =>/api/alarm/config/query

需要登录,也需要检验项目权限(Method: **get**) =>/api/alarm/config/list

需要登录,也需要检验项目权限(Method: **get**) =>/api/alarm/config/delete

需要登录,也需要检验项目权限(Method: post) =>/api/alarm/config/update

需要登录,也需要检验项目权限(Method: **get**) =>/api/alarm/config/error\_name\_list

需要登录,也需要检验项目权限(Method: **get**) =>/api/alarm/config/error\_type\_list

需要登录,也需要检验项目权限(Method: **get**) =>/api/alarm/log

需要登录,也需要检验项目权限(Method: **get**) =>/api/alarm/log/line

需要登录,但不需要检验项目权限(Method: **get**) =>/api/user/detail

需要登录,但不需要检验项目权限(Method: **get**) =>/api/user/search

需要登录,但不需要检验项目权限(Method: post) =>/api/user/update

需要登录,但不需要检验项目权限(Method: **get**) =>/api/user/delete

需要登录,但不需要检验项目权限(Method: **get**) =>/api/user/search\_uc

不需要登录(Method: post) =>/api/user/register

需要登录,但不需要检验项目权限(Method: post) =>/api/user/modify/password

需要登录,但不需要检验项目权限(Method: post) =>/api/user/modify/msg

需要登录,但不需要检验项目权限(Method: **get**) =>/api/user/destroy

不需要登录(Method: post) =>/api/login/site

不需要登录(Method: post) =>/api/login/uc

不需要登录(Method: **get**) =>/api/logout

不需要登录(Method: post) =>/api/login/normal

不需要登录(Method: **get**) =>/api/login/type

不需要登录(Method: post) =>/api/login

需要登录,但不需要检验项目权限(Method: **get**) =>/api/project/item/detail

需要登录,但不需要检验项目权限(Method: post) =>/api/project/item/add

需要登录,但不需要检验项目权限(Method: **get**) =>/api/project/item/list

需要登录,但不需要检验项目权限(Method: **get**) =>/api/project/item/delete

需要登录,但不需要检验项目权限(Method: post) =>/api/project/item/update

需要登录,也需要检验项目权限(Method: post) =>/api/project/member/add

需要登录,也需要检验项目权限(Method: **get**) =>/api/project/member/list

需要登录,也需要检验项目权限(Method: **get**) =>/api/project/member/delete

需要登录,也需要检验项目权限(Method: post) =>/api/project/member/update

需要登录,也需要检验项目权限(Method: **get**) =>/api/project/summary/new\_user/distribution\_line

需要登录,也需要检验项目权限(Method: **get**) =>/api/project/summary/new\_user/distribution\_map

需要登录,也需要检验项目权限(Method: **get**) =>/api/performance/url\_list

需要登录,也需要检验项目权限(Method: **get**) =>/api/performance/url/overview

需要登录,也需要检验项目权限(Method: **get**) =>/api/performance/project/overview

需要登录,也需要检验项目权限(Method: **get**) =>/api/performance/url/line\_chart

需要登录,也需要检验项目权限(Method: **get**) =>/api/uv/count

### **7.2 启动前台**

cd client

cnpm i

npm run start

### **7.3 前端应用**

client\src\router\index.js

**const** token = getToken();**if** (!token && to.name !== LOGIN\_PAGE\_NAME) {

// 未登录且要跳转的页面不是登录页

next({

name: LOGIN\_PAGE\_NAME // 跳转到登录页

})

}

server\src\routes\api\user\index.js

**const** register = RouterConfigBuilder.routerConfigBuilder('/api/user/register', RouterConfigBuilder.METHOD\_TYPE\_POST, **async** (req, res) => {

**const** body = \_.get(req, ['body'], {})

}

server\src\model\project\user.js

**async** **function** **register** (account, userInfo) {

**const** tableName = getTableName(); // t\_o\_user

}

### **7.4 页面性能**

<http://localhost:8080/project/1/monitor/performance>

#### **7.4.1 urlList**

* 用来提供某时间范围内的URL性能列表，因为在查看性能指标的时候会按页面的URL来进行，所以在页面的初始阶段会提供一个接口函数来获取 这些性能指标对应的URL列表

client\src\view\performance\index.vue

**async** getUrlList (params = {}) {

**const** {

st,

et,

summaryBy

} = params

**const** res = **await** fetchUrlList({

st: st || +**this**.dateRange[0],

et: et || +**this**.dateRange[1],

summaryBy: summaryBy || 'minute'

})

**this**.urlData = (res.data || []).map((item, index) => ({

name: item

}))

**this**.url = \_.get(**this**, ['urlData', 0, 'name'], '')

**this**.getTimeDetail()

**this**.getTimeLine()

**if** (**this**.urlColumns) {

**this**.urlLoading = false

}

},

client\src\api\performance\index.js

**export** **const** fetchUrlList = (params) => {

**return** axios.request({

url: `project/${getProjectId()}/api/performance/url\_list`,

method: 'get',

params: {

...params

}

})

}

Request URL: http://localhost:3000/project/1/api/performance/url\_list?st=1591977600000&et=1592051809707&summaryBy=minute

server\src\routes\api\performance\index.js

**let** urlList = RouterConfigBuilder.routerConfigBuilder('/api/performance/url\_list', RouterConfigBuilder.METHOD\_TYPE\_GET, **async** (req, res) => {

**let** projectId = \_.get(req, ['fee', 'project', 'projectId'], 0);

console.log('projectId ',projectId);

**let** request = \_.get(req, ['query'], {})

// 获取开始&结束时间

**let** startAt = \_.get(request, ['st'], 0)

**let** endAt = \_.get(request, ['et'], 0)

**const** summaryBy = \_.get(request, 'summaryBy', '')

**if** (\_.includes([DATE\_FORMAT.UNIT.DAY, DATE\_FORMAT.UNIT.HOUR, DATE\_FORMAT.UNIT.MINUTE], summaryBy) === false) {

res.send(API\_RES.showError(`summaryBy参数不正确`))

**return**

}

**const** currentStamp = moment().unix()

**if** (startAt) {

startAt = \_.floor(startAt / 1000)

} **else** {

startAt = currentStamp

}

**if** (endAt) {

endAt = \_.ceil(endAt / 1000)

} **else** {

endAt = currentStamp

}

**let** urlList = **await** MPerformance.getDistinctUrlListInRange(projectId, MPerformance.INDICATOR\_TYPE\_LIST, startAt, endAt, summaryBy)

res.send(API\_RES.showResult(urlList))

}

)

server\src\model\parse\performance.js

**async** **function** **getDistinctUrlListInRange** (projectId, indicatorList, startAt, endAt, countType = DATE\_FORMAT.UNIT.MINUTE) {

**for** (**let** tableName **of** tableNameList) {

console.log('tableName,countType,indicatorList,countAtTimeList',tableName,countType,indicatorList,countAtTimeList);

**let** rawRecordList = **await** Knex

.distinct(['url'])

.from(tableName)

.where({

count\_type: countType

})

.whereIn('indicator', indicatorList)

.whereIn('count\_at\_time', countAtTimeList)

.catch((e) => {

Logger.warn('查询失败, 错误原因 =>', e)

**return** []

})

**for** (**let** rawRecord **of** rawRecordList) {

**if** (\_.has(rawRecord, ['url'])) {

**let** url = \_.get(rawRecord, ['url'])

urlList.push(url)

}

}

}

}// performance distinctUrlList["localhost:9000/"]

#### **7.4.2 lineChartData**

* 在获取 URL之后，根据对应的时间参数提供参数时间范围内的提定URL下面的所有指标折线图

client\src\view\performance\index.vue

**async** getTimeDetail (params = {}) {

**const** {

st,

et,

url,

summaryBy

} = params

**const** res = **await** fetchTimeDetail({

st: st || +**this**.dateRange[0],

et: et || +**this**.dateRange[1],

url: url || **this**.url,

summaryBy: summaryBy || 'hour'

})

**const** dv = **new** DataSet.View().source(res.data)

dv.transform({

type: 'rename',

map: {

dns\_lookup\_ms: 'DNS查询耗时',

response\_request\_ms: '请求响应耗时',

dom\_parse\_ms: 'DOM解析耗时',

response\_transfer\_ms: '内容传输耗时',

load\_resource\_ms: '资源加载耗时',

dom\_ready\_ms: 'DOM\_READY\_耗时',

first\_render\_ms: '首次渲染耗',

first\_response\_ms: '首次可交互耗时',

first\_tcp\_ms: '首包时间耗时',

load\_complete\_ms: '页面完全加载耗时',

ssl\_connect\_ms: 'SSL连接耗时',

tcp\_connect\_ms: 'TCP链接耗时'

}

})

dv.transform({

type: 'fold',

fields: [

'DNS查询耗时',

'请求响应耗时',

'DOM解析耗时',

'内容传输耗时',

'资源加载耗时',

'DOM\_READY\_耗时',

'首次渲染耗',

'首次可交互耗时',

'首包时间耗时',

'页面完全加载耗时',

'SSL连接耗时',

'TCP链接耗时'

],

key: 'type',

value: 'ms'

})

**const** data = dv.rows

**this**.lineData = data

**const** scale = [{

dataKey: 'ms',

sync: true,

alias: 'ms',

formatter: (value) => value + ' ms'

}, {

dataKey: 'index\_timestamp\_ms',

type: 'time',

tickCount: 10,

mask: 'MM-DD HH:mm'

}]

**this**.lineScale = scale

**if** (**this**.lineData && **this**.lineScale) {

**this**.isSpinShowDetail = false

}

},

**export** **const** fetchTimeDetail = (params) => {

**return** axios.request({

url: `project/${getProjectId()}/api/performance/url/line\_chart`,

method: 'get',

params: {

...params

}

})

}

http://localhost:3000/project/1/api/performance/url/line\_chart?st=1591977600000&et=1592053122452&url=localhost:9000%2F&summaryBy=hour

server\src\routes\api\performance\index.js

**let** lineChartData = RouterConfigBuilder.routerConfigBuilder('/api/performance/url/line\_chart', RouterConfigBuilder.METHOD\_TYPE\_GET, **async** (req, res) => {

**let** projectId = \_.get(req, ['fee', 'project', 'projectId'], 0)

**let** request = \_.get(req, ['query'], {})

res.send(API\_RES.showResult(resultList))

}

)

server\src\model\parse\performance.js

**async** **function** **getDistinctUrlListInRange** (projectId, indicatorList, startAt, endAt, countType = DATE\_FORMAT.UNIT.MINUTE) {

#### **7.4.3 urlOverview**

* 在获取 URL之后，可以根据此接口提供的时间范围指定的URL的各项指标平均值

client\src\view\performance\index.vue

**async** getTimeLine (params = {}) {

**const** {st,et} = params

**const** res = **await** fetchTimeLine({

st: st || +**this**.dateRange[0],

et: et || +**this**.dateRange[1],

url: **this**.url,

summaryBy: 'minute'

})

client\src\api\performance\index.js

**export** **const** fetchTimeLine = (params) => {

**return** axios.request({

url: `project/${getProjectId()}/api/performance/url/overview`,

method: 'get',

params: {

...params

}

})

}

http://localhost:3000/project/1/api/performance/url/overview?st=1591977600000&et=1592054332424&url=localhost:9000%2F&summaryBy=minute

server\src\routes\api\performance\index.js

**let** urlOverview = RouterConfigBuilder.routerConfigBuilder('/api/performance/url/overview', RouterConfigBuilder.METHOD\_TYPE\_GET, **async** (req, res) => {

**let** projectId = \_.get(req, ['fee', 'project', 'projectId'], 0)

**let** request = \_.get(req, ['query'], {});

}

**async** **function** **getUrlOverviewInSameMonth** (projectId, urlList, startAt, endAt, countType) {

}

* [front-monitor](https://gitee.com/zhufengpeixun/front-monitor)
* node dist/fee.js Utils:TemplateSQL

Powered by [idoc](https://github.com/jaywcjlove/idoc" \t "https://zhufeng-document.vercel.app/html/_blank). Dependence [Node.js](https://nodejs.org/) run.