

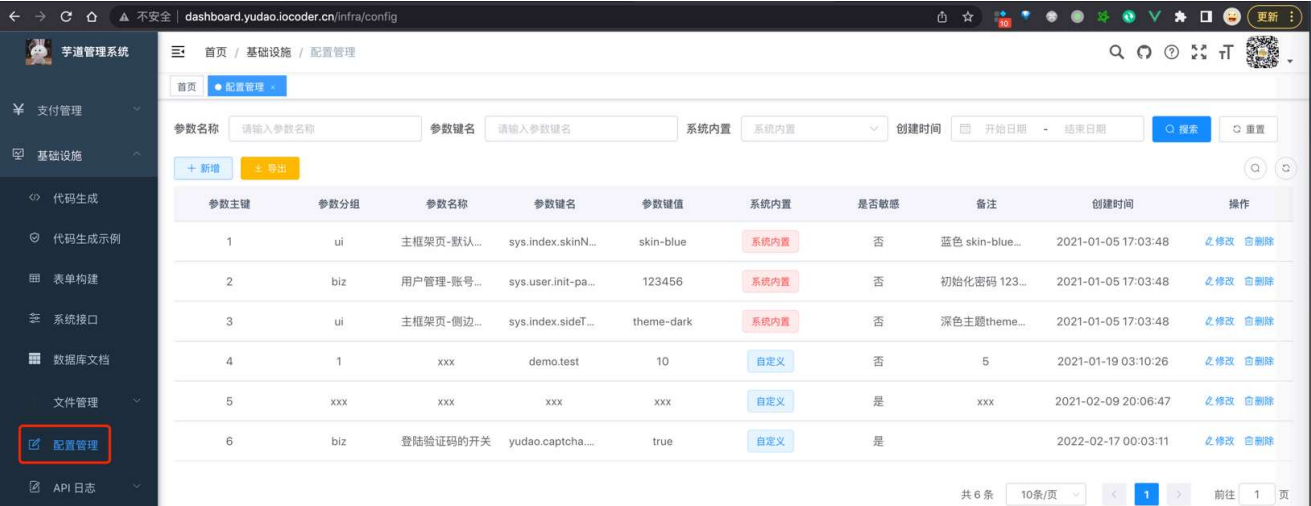
🏠 / 开发指南 / 后端手册

👤 芋道源码 📅 2022-04-04

配置管理

在 [基础设施 -> 配置管理] 菜单，可以查看和管理配置，适合业务上需要动态的管理某个配置。

例如说：创建用户时，需要配置用户的默认密码，这个密码是不会变的，但是有时候需要修改这个默认密码，这个时候就可以通过配置管理来修改。



对应的后端代码是 `yudao-module-infra` 的 `config` 业务模块。

1. 配置的表结构

`infra_config` 的表结构如下：

```
CREATE TABLE `infra_config` (  
  `id` int NOT NULL AUTO_INCREMENT COMMENT '参数主键',  
  `group` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL  
  `type` tinyint NOT NULL COMMENT '参数类型',  
  `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL  
  `key` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL  
  `value` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL  
  `sensitive` bit(1) NOT NULL COMMENT '是否敏感',  
  `remark` varchar(500) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `creator` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `updater` varchar(64) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT  
  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TI  
  `deleted` bit(1) NOT NULL DEFAULT b'0' COMMENT '是否删除',  
  PRIMARY KEY (`id`) USING BTREE
```

- key 字段，对应到 Spring Boot 配置文件的配置项，例如说 yudao.captcha.enable 、 sys.user.init-password 等等。

3. 后端案例

TODO 芋艿：待补充

4. 前端案例

后端提供了 /admin-api/infra/config/get-value-by-key RESTful API 接口，返回指定配置项的值。前端的使用示例如下图：

Index.vue

```
1 <template>
2 <div>
3   <doc-alert title="服务监控" url="https://doc.iocoder.cn/server-monitor/" />
4   <i-frame v-if="!loading" :src="url" />
5 </div>
6 </template>
7 <script>
8 > import ...
9
10 export default {
11   name: "Druid",
12   components: { iFrame },
13   data() {
14     return {
15       url: "http://skywalking.shop.iocoder.cn",
16       loading: true,
17     };
18   },
19   created() {
20     getConfByKey("url.skywalking").then(response => {
21       if (!response.data || response.data.length === 0) {
22         return;
23       }
24       this.url = response.data;
25     }).finally(() => {
26       this.loading = false;
27     });
28   }
29 }
```

ConfigController.java

```
41 public CommonResult<Long> createConfig(@Valid @RequestBody ConfigCreateReqVO reqVO) {...}
42
43 @{...}
44
45 public CommonResult<Boolean> updateConfig(@Valid @RequestBody ConfigUpdateReqVO reqVO) {...}
46
47 @{...}
48
49 public CommonResult<Boolean> deleteConfig(@RequestParam("id") Long id) {...}
50
51 @{...}
52
53 public CommonResult<ConfigRespVO> getConfig(@RequestParam("id") Long id) {...}
54
55 @Operation(summary = "根据参数键名查询参数值", description = "不可见的配置，不允许返回给前端")
56 @Parameter(name = "key", description = "参数键", required = true, example = "yunai.biz.username")
57 public CommonResult<String> getConfByKey(@RequestParam("key") String key) {
58   ConfigDO config = configService.getConfigByKey(key);
59   if (config == null) {
60     return success(data: null);
61   }
62   if (!config.getVisible()) {
63     throw exception(ErrorCodeConstants.CONFIG_GET_VALUE_ERROR_IF_VISIBLE);
64   }
65   return success(config.getValue());
66 }
```

