

首页 / 开发指南 / 前端手册 Vue 3.x

芋道源码 2022-04-17

# 字典数据

本小节，讲解前端如何使用 [系统管理 -> 字典管理] 菜单的字典数据，例如说字典数据的下拉框、单选 / 多选按钮、高亮展示等等。

字典名称	字典类型	状态	创建时间	操作
交易订单项的售后...	trade_order_item_a...	开启	2022-12-10 20:5...	[编辑] [删除]
交易订单的状态	trade_order_status	开启	2022-12-10 16:4...	[编辑] [删除]
交易订单的类型	trade_order_type	开启	2022-12-10 16:3...	[编辑] [删除]
终端	terminal	开启	2022-12-10 10:5...	[编辑] [删除]
交易售后的方式	trade_after_sale_w...	开启	2022-11-19 21:3...	[编辑] [删除]
交易售后的类型	trade_after_sale_ty...	开启	2022-11-19 21:0...	[编辑] [删除]
交易售后状态	trade_after_sale_st...	开启	2022-11-19 20:5...	[编辑] [删除]
营销的条件类型	promotion_conditi...	开启	2022-11-04 22:5...	[编辑] [删除]
促销活动的状态	promotion_activity...	开启	2022-11-04 22:5...	[编辑] [删除]
优惠券的领取方式	promotion_coupon_w...	开启	2022-11-04 19:1...	[编辑] [删除]

  

数据标签	数据键值	状态	操作
退货退款	20	开启	[编辑] [删除]
仅退款	10	开启	[编辑] [删除]

## 1. 全局缓存

用户登录成功后，前端会从后端获取到全量的字典数据，缓存在 store 中。如下图所示：

```

    router.beforeEach((to: RouteLocationNormalized, from: RouteLocationNormalized, next: NavigationGuardNext) => {
      start()
      loadStart()
      if (getAccessToken()) {
        if (to.path === '/login') {
          next({ path: '/' })
        } else {
          // 获取所有字典
          const dictStore = useDictStoreWithOut()
          const userStore = useUserStoreWithoutOut()
          const permissionStore = usePermissionStoreWithoutOut()
          const dictMap = wsCache.get(CACHE_KEY.DICT_CACHE)
          if (!dictMap) {
            const res = await listSimpleDictDataApi()
            dictStore.setDictMap(res)
          }
        }
      }
      callbackForRouter(beforeEach)
    })
  
```

① 从后端获取字典数据

  

```

  actions: {
    setDictMap(dictMap: Recordable) {
      // 设置数据
      const dictDataMap = new Map<string, any>()
      dictMap.forEach((dictData: DictDataVO) => {
        // 获得 dictType 层级
        const enumValueObj = dictDataMap[dictData.dictType]
        if (!enumValueObj) {
          dictDataMap[dictData.dictType] = []
        }
        // 处理 dictValue 层级
        dictDataMap[dictData.dictType].push({
          value: dictData.value,
          label: dictData.label,
          colorType: dictData.colorType,
          cssClass: dictData.cssClass
        })
      })
      useDictStore.actions.setDictMap()
    }
  }
  
```

② 设置字典数据，到 store 中

这样，前端在使用到字典数据时，无需重复请求后端，提升用户体验。

不过，缓存暂时未提供刷新，所以在字典数据发生变化时，需要用户刷新浏览器，进行重新加载。

## 2. DICT\_TYPE

在 `dict.ts` 文件中，使用 `DICT_TYPE` 枚举了字典的 KEY。如下图所示：

```

  export enum DICT_TYPE {
    USER_TYPE = 'user_type',
    COMMON_STATUS = 'common_status',
    SYSTEM_TENANT_PACKAGE_ID = 'system_tenant_package_id',

    // ===== SYSTEM 模块 =====
    SYSTEM_USER_SEX = 'system_user_sex',
    SYSTEM_MENU_TYPE = 'system_menu_type',
    SYSTEM_ROLE_TYPE = 'system_role_type',
    SYSTEM_DATA_SCOPE = 'system_data_scope',
    SYSTEM_NOTICE_TYPE = 'system_notice_type',
    SYSTEM_OPERATE_TYPE = 'system_operate_type',
    SYSTEM_LOGIN_TYPE = 'system_login_type',
    SYSTEM_LOGIN_RESULT = 'system_login_result',
    SYSTEM_SMS_CHANNEL_CODE = 'system_sms_channel_code',
    SYSTEM_SMS_TEMPLATE_TYPE = 'system_sms_template_type',
    SYSTEM_SMS_SEND_STATUS = 'system_sms_send_status',
    SYSTEM_SMS_RECEIVE_STATUS = 'system_sms_receive_status',
    SYSTEM_ERROR_CODE_TYPE = 'system_error_code_type',
    SYSTEM_OAUTH2_GRANT_TYPE = 'system_oauth2_grant_type',

    // ===== INFRA 模块 =====
    INFRA_BOOLEAN_STRING = 'infra_boolean_string',
    INFRA_REDIS_TIMEOUT_TYPE = 'infra_redis_timeout_type',
  }
  
```

后续如果有新的字典 KEY，需要你自己进行添加。

## 3. DictTag 字典标签

`<dict-tag />` 组件，翻译字段对应的字典展示文本，并根据 `colorType`、`cssClass` 进行高亮。使用示例如下：

```
<!--
  type: 字典 KEY
  value: 字典值
-->
<dict-tag :type="DICT_TYPE.SYSTEM_LOGIN_TYPE" :value="row.logType" />
```

日志编号	日志类型	用户名	登录地址	浏览器	登陆结果	登录日期	操作
591	账号登录	admin	127.0.0.1	Mozilla/5.0 (Macintosh; Intel M...	成功	2022-05-13 19:3...	查看详情
592	账号登录	admin	127.0.0.1	Mozilla/5.0 (Macintosh; Intel M...	成功	2022-05-13 19:3...	查看详情
593	账号登录	admin	127.0.0.1	Mozilla/5.0 (Macintosh; Intel M...	验证码不正确	2022-05-13 19:3...	查看详情
594	账号登录	admin	127.0.0.1	Mozilla/5.0 (Macintosh; Intel M...	验证码不正确	2022-05-13 19:3...	查看详情
595	账号登录	admin	127.0.0.1	Mozilla/5.0 (Macintosh; Intel M...	成功	2022-05-13 19:3...	查看详情
596	账号登录	admin	127.0.0.1	Mozilla/5.0 (Macintosh; Intel M...	成功	2022-05-13 19:3...	查看详情
597	账号登录	admin	127.0.0.1	Mozilla/5.0 (Macintosh; Intel M...	验证码不正确	2022-05-13 19:3...	查看详情

【推荐】注意，一般情况下使用 CRUD schemas 方式，不需要直接使用 `<dict-tag />`，而是通过 `columns` 的 `dictType` 和 `dictClass` 属性即可。如下图所示：

```

import { reactive } from 'vue'
import { DICT_TYPE } from '@/utils/dict'
import { VxeCrudSchema, useVxeCrudSchemas } from '@/hooks/web/useVxeCrudSchemas'

// CrudSchema
const crudSchemas = reactive<VxeCrudSchema>(
  target: {
    primaryKey: 'id',
    primaryType: 'seq',
    primaryTitle: '日志编号',
    action: true,
    actionWidth: '100px',
    columns: [
      {
        title: '日志类型',
        field: 'logType',
        dictType: DICT_TYPE.SYSTEM_LOGIN_TYPE,
        dictClass: 'number'
      },
      {
        title: '用户名',
        field: 'username',
        dictType: DICT_TYPE.SYSTEM_LOGIN_TYPE,
        dictClass: 'string'
      },
      {
        title: '登录地址',
        field: 'loginAddress',
        dictType: DICT_TYPE.SYSTEM_LOGIN_TYPE,
        dictClass: 'string'
      },
      {
        title: '浏览器',
        field: 'browser',
        dictType: DICT_TYPE.SYSTEM_LOGIN_TYPE,
        dictClass: 'string'
      },
      {
        title: '登陆结果',
        field: 'result',
        dictType: DICT_TYPE.SYSTEM_LOGIN_RESULT,
        dictClass: 'number'
      },
      {
        title: '登录日期',
        field: 'loginDate',
        dictType: DICT_TYPE.SYSTEM_LOGIN_DATE,
        dictClass: 'date'
      }
    ]
  }
)
export const { allSchemas } = useVxeCrudSchemas(crudSchemas)

```

## 4. 字典工具类

在 `dict.ts` 文件中，提供了字典工具类，方法如下：

```
// 获取 dictType 对应的数据字典数组【object】
export const getDictOptions = (dictType: string) => { /* 省略代码 */ }

// 获取 dictType 对应的数据字典数组【int】
export const getIntDictOptions = (dictType: string) => { /* 省略代码 */ }

// 获取 dictType 对应的数据字典数组【string】
export const getStrDictOptions = (dictType: string) => { /* 省略代码 */ }

// 获取 dictType 对应的数据字典数组【boolean】
export const getBoolDictOptions = (dictType: string) => { /* 省略代码 */ }

// 获取 dictType 对应的数据字典数组【object】
export const getDictObj = (dictType: string, value: any) => { /* 省略代码 */ }
```



结合 Element Plus 的表单组件，使用示例如下：

```
<template>
  <!-- radio 单选框 -->
  <el-radio
    v-for="dict in getIntDictOptions(DICT_TYPE.COMMON_STATUS)"
    :key="dict.value"
    :label="parseInt(dict.value)"
  >
    {{dict.label}}
  </el-radio>

  <!-- select 下拉框 -->
  <el-select v-model="form.code" placeholder="请选择渠道编码" clearable>
    <el-option
      v-for="dict in getStrDictOptions(DICT_TYPE.SYSTEM_SMS_CHANNEL_CODE)"
      :key="dict.value"
      :label="dict.label"
      :value="dict.value"
    />
  </el-select>
</template>
```

```
<script setup lang="ts">
import { DICT_TYPE, getIntDictOptions, getStrDictOptions } from '@/utils/dict'
</script>
```

← [Icon 图标](#)

[系统组件](#) →



Theme by [Vdoing](#) | Copyright © 2019-2024 芸道源码 | MIT License

