项目管理系统 web-api 代码走查

REVIEER: 彭玲 TIME: 2021/11/30

package 包过时

"koa-proxy": "^0.9.0"

`koa-proxy` 版本过低，可以升级到最新版本，基于 Promise 对象。

"koa-convert": "^1.2.0"

`koa-proxy` 升级后，不再需要 `koa-convert` 包。

命名规范性

`t_rpm_customer.stause` 数据表字段 `stause` 及对应的 api 字段代表什么?

代码规范

魔数

`state: 0` 和 `stause: 2` 中的数值代表什么?

```
//新增客户记录
const customerData = {
    customerName,
    customerCode,
    industryId: Number(industryId),
    typeId: Number(typeId),
    //categoryId: categoryId && Number(categoryId) || null,
    //levelId: levelId && Number(levelId) || null,
    customerSource: Number(customerSource) || null,
```

```
    regionPath,
    address: address || null,
    longitude: longitude || null,
    latitude: latitude || null,
    companySize: companySize || null,
    companyProfile: companyProfile || null,
    createBy: userId,
    createTime: moment().toISOString(),
    state: 0,
    stause: 2
}
```

对象多次使用

`ctx.request.body` 对象属性没必要分多次获取。

```
let {
    customerName,
    industryId, typeId, customerSource,
    regionPath, address, longitude, latitude, companySize, companyProfile,
} = ctx.request.body;
let { customerCode } = ctx.request.body;
```

语法错误

app/lib/controllers/common/index.js

```
async function getDataDictionary(ctx) {
    try {
        const models = ctx.fs.dc.models;
        const { model } = ctx.params;
        const { where, attributes } = ctx.query;
        let findObj = {};
        if(attributes) {
            attributes = attributes.split(','); // 修改 const 变量，且该变量为查询参数
```

app/lib/controllers/company/index.js

```
resources.map(s => { // 改用 forEach
    if (!s.parentResource) {
        let children = []
        resources.map(l => {
            l.parentResource == s.code ?
                children.push({
                "id": l.code,
                "name": l.name,
                "available": roleResource.some(role => role.resource == l.code),
                "expanded": true,
                "children": []
            }) : ''
```

app/lib/controllers/customer/index.js

```
obj[next.saler] ? "" : obj[next.saler] = true && cur.push(next); // "" 和 true 的意义在哪儿?
```

app/lib/controllers/rpm-project/index.js

```
stateText: stateText || stateText, // 相同变量进行 或 操作
```

app/lib/models/inventory.js

```
code: { // 数据库表 code 字段重复定义，可使用 sequelize-automate 自动化生成，避免该类问题
      type: DataTypes.STRING,
      allowNull: false,
      defaultValue: null,
      comment: "清单编号",
      primaryKey: false,
      field: "code",
      autoIncrement: false
},
```

代码逻辑不合理

app/lib/controllers/construction-log/index.js

```
if (judgeRslt) {
    where.roleId == -1; // 若返回结果为[], 直接返回即可，不需要走数据库查询
} else {
    where.roleId == { $in: roles };
}
let rslt = { rows: [], count: 0 };
rslt = await models.MonthStatistics.findAndCountAll({
    where: where,
});
```

app/lib/controllers/rpm-project/detail.js

```
await models.FollowupRecord.create({
    itemId: Number(id),
    createTime: moment(),
    cause,
    competitor, fundsSufficient, planAgreed, stateText, problemAnalysis, nextTodo, remarks, // remarks 已引用
    stateId, groundTime, amountMoney, remarks, createUserId, bidding // remarks 重复引用，太多参数可以采用反向过滤 {..., rest}，避免此类问题
}, { transaction });
```

app/lib/controllers/rpm-project/index.js

```
module.exports = {
    getProjectIndustry,
    getProjectType,
    getProjectLevel,
```

```
    getProjectSucess,
    postRpmProject,
    putRmpProject,
    delRmpProject,
    getRmpProject,
    changeSale,
    getRmpProjectEstablishment,
    rmpProjectEstablishment,
    rmpProjectEstablishmentCancel,
    getRmpProjectAchievement,
    rmpProjectAchievement,
    rmpProjectAchievementCancel,
    getRmpRealationList,
    postRmpRealation,
    putRmpRealation,
    delRmpRealation,
    getRmpToRealationList,
    getRmpRealationCheckName,
    checkRmpAchievementCode, // checkRmpAchievementCode 已导出
    addSubProject,
    modifySubproject,
    delSubproject,
    checkRmpAchievementCode, // checkRmpAchievementCode 重复导出，改用在函数定义时直接导出，避免此类问题
```

app/lib/controllers/rpm-project/report-achievement.js

```
let actualAchievement = 0;
if (actualAchievementRes.length > 0) {
    actualAchievementRes.map(v => { // 典型的 reduce 用法
        actualAchievement += number(v.actualAchievement);
    })
})

// 以下，代码可整合复用

let rpmPerformance = 0;
let rpmPerformanceRes = groundTimeItems.filter(v => v.salespersonId == val.id);
if (rpmPerformanceRes.length > 0) {
    rpmPerformanceRes.map(v => {
        rpmPerformance += number(v.amountMoney) * 10000;
    })
}

let contractMoney = 0;
if (contractMoneyRes.length > 0) {
    contractMoneyRes.map(v => {
        contractMoney += number(v.contractMoney);
    });
}

let actualAchievement = 0;
if (actualAchievementRes.length > 0) {
    actualAchievementRes.map(v => {
        actualAchievement += number(v.actualAchievement);
    })
```

```
}

let actualCollectionRes = accountsReceiptDateItems.filter(v =>
v.salespersonId == val.id);
let actualCollection = 0;
if (actualCollectionRes.length > 0) {
    actualCollectionRes.map(v => {
        actualCollection += number(v.receipts);
    })
}
```