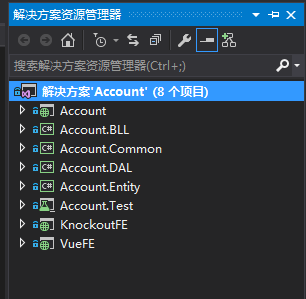
最近正在学习Vue2.0相关知识，正好近期饿了么桌面端组件Element-UI发布，便动手做了一款简易个人记账系统，以达到实践及巩固目的。

**1.开发环境**

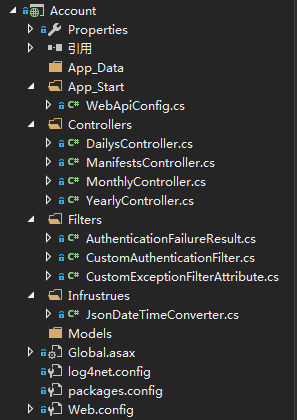
Win10 + VS2015 + Sqlserver2008R2 + WebAPI + Dapper + Vue2.0 + Element-UI

**2.项目解决方案概览**

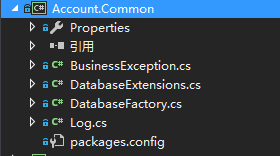


简单介绍下，Account是WebAPI项目，承载前端请求；Account.BLL、Account.DAL、Account.Entity不废话；Account.Common是对Dapper及Log4net等简单封装；KnockoutFE是较早时候弄的一个基于Bootstrap + Knockout的前端；VueFE是主角，待会儿重点介绍。

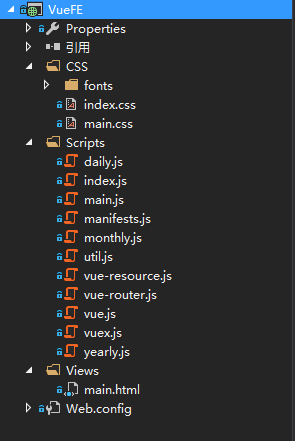
Account工程截图：



Account.Common工程截图：



VueFE工程截图：



**3.前端实现**

初始界面：



**1.左边导航部分**

使用el-menu + vue-router实现：

[复制代码](javascript:void(0);)

<div id="sideBar">

<!--<ul>

<router-link to="/manifests" tag="li">日消费明细</router-link>

<router-link to="/daily" tag="li">日消费清单</router-link>

<router-link to="/monthly" tag="li">月消费清单</router-link>

<router-link to="/yearly" tag="li">年消费清单</router-link>

</ul>-->

<el-menu default-active="manifests" theme="dark" v-bind:router="true">

<el-menu-item index="manifests">日消费明细</el-menu-item>

<el-menu-item index="daily">日消费清单</el-menu-item>

<el-menu-item index="monthly">月消费清单</el-menu-item>

<el-menu-item index="yearly">年消费清单</el-menu-item>

</el-menu>

</div>

[复制代码](javascript:void(0);)

　　注释掉部分是最开始时候没有采用el-menu组件导航，而是使用了vue-router自己的路由导航。

路由部分对应JS代码：

[复制代码](javascript:void(0);)

const router = new VueRouter({

routes: [

{ name: "manifests", path: "/manifests", component: Manifests },

{ name: "daily", path: "/daily", component: Daily },

{ name: "monthly", path: "/monthly", component: Monthly },

{ name: "yearly", path: "/yearly", component: Yearly }

]

});

[复制代码](javascript:void(0);)

　　其中，Manifests、Daily、Monthly、Yearly分别代表日消费明细、日消费清单、月消费清单、年消费清单自定义Vue组件。

**2.具体内容页组件实现**

这里以日消费明细组件为例来详细介绍，里边囊括了CRUD。

**组件完整定义：**

[复制代码](javascript:void(0);)

/// <reference path="index.js" />

/// <reference path="vue.js" />

/// <reference path="vue-resource.js" />

/// <reference path="util.js" />

const Manifests = {

template: "#manifests",

created: function () {

this.fetchData();

},

data: function () {

let currentDate = new Date();

let costValidator = (rule, value, callback) => {

if (!/^[0-9]+(.[0-9]{2})?$/.test(value)) {

callback(new Error("请输入合法金额"));

}

else {

callback();

}

};

return {

start: new Date(currentDate.getFullYear(), currentDate.getMonth() - 3, 1),

end: new Date(),

manifests: [],

title: "",

manifest: {},

showOperateManifest: false,

isAdd: false,

rules: {

Date: [

{ type: "date", required: true, message: "请选择消费日期", trigger: "change" }

],

Cost: [

{ required: true, message: "请填写消费金额", trigger: "blur" },

{ validator: costValidator, trigger: "change" }

],

Remark: [

{ required: true, message: "请填写消费明细", trigger: "blur" }

]

},

pageIndex: 0,

pageSize: 10,

total: 0,

pageSizes: [10, 20, 50, 100]

}

},

methods: {

fetchData: function () {

this.manifests = [];

this.$http.get("http://localhost:1500/api/Manifests/paged", {

params: {

start: this.start.format("yyyy-MM-dd"),

end: this.end.format("yyyy-MM-dd"),

pageIndex: this.pageIndex,

pageSize: this.pageSize

}

})

.then(response => {

this.total = response.body.count;

this.manifests = response.body.data;

})

.catch(response => this.$alert(response.body.Message, "日消费明细", { type: "error" }));

},

add: function () {

this.title = "添加消费明细";

this.manifest = {

ID: Guid.NewGuid().ToString("D"),

Date: new Date(),

Cost: "",

Remark: ""

};

this.isAdd = true;

this.showOperateManifest = true;

},

save: function () {

this.$refs.formManifest.validate(valid => {

if (valid) {

let operateManifest = JSON.parse(JSON.stringify(this.manifest));

operateManifest.Date = this.manifest.Date.format("yyyy-MM-dd");

if (this.isAdd) {

this.$http.post("http://localhost:1500/api/Manifests", operateManifest)

.then(() => {

this.manifests.push(operateManifest);

this.showOperateManifest = false;

bus.$emit("manifestChanged");

this.$message({

message: "添加成功",

type: "success"

});

})

.catch(err => {

//console.log(err);

this.$alert(err.body.Message, "添加日消费明细", { type: "error" });

});

}

else {

this.$http.put("http://localhost:1500/api/Manifests", operateManifest)

.then(response => {

let updatedManifest = this.manifests.find(x => x.ID == this.manifest.ID);

updatedManifest.Date = operateManifest.Date;

updatedManifest.Cost = operateManifest.Cost;

updatedManifest.Remark = operateManifest.Remark;

this.showOperateManifest = false;

bus.$emit("manifestChanged");

this.$message({

message: "修改成功",

type: "success"

});

})

.catch(err => {

//console.log(err);

this.$alert(err.body.Message, "修改消费明细", { type: "error" });

});

}

}

else {

return false;

}

});

},

cancel: function () {

this.manifest = {};

this.showOperateManifest = false;

},

edit: function (ID) {

let currentManifest = this.manifests.find(x => x.ID == ID);

this.manifest = JSON.parse(JSON.stringify(currentManifest));

this.manifest.Date = new Date(this.manifest.Date);

this.title = "编辑消费明细";

this.isAdd = false;

this.showOperateManifest = true;

},

del: function (ID) {

this.$confirm("是否删除？", "警告", { type: "warning" })

.then(() => {

this.$http.delete("http://localhost:1500/api/Manifests/" + ID)

.then(response => {

let index = this.manifests.findIndex(x => x.ID == ID);

this.manifests.splice(index, 1);

bus.$emit("manifestChanged");

this.$message({

message: "删除成功",

type: "success"

});

})

.catch(err => {

this.$alert(err.body.Message, "删除消费明细", { type: "error" });

//console.log(err);

});

});

},

dialogClosed: function () {

this.$refs.formManifest.resetFields();

},

sizeChange: function (pageSize) {

this.pageSize = pageSize;

this.fetchData();

},

pageIndexChange: function (pageIndex) {

this.pageIndex = pageIndex;

this.fetchData();

}

}

}

[复制代码](javascript:void(0);)

**组件对应的模板定义：**

[复制代码](javascript:void(0);)

<script type="text/x-template" id="manifests">

<div>

<div>

开始日期：

<el-date-picker v-model="start" type="date" placeholder="选择日期"></el-date-picker>

结束日期：

<el-date-picker v-model="end" type="date" placeholder="选择日期"></el-date-picker>

<el-button type="primary" size="small" v-on:click="fetchData" icon="search">查 询</el-button>

<el-button type="primary" size="small" v-on:click="add" icon="plus">添 加</el-button>

</div>

<div class="table">

<el-table v-bind:data="manifests" highlight-current-row border height="500">

<el-table-column prop="Date" label="日期"></el-table-column>

<el-table-column prop="Cost" label="金额"></el-table-column>

<el-table-column prop="Remark" label="备注"></el-table-column>

<el-table-column inline-template label="操作">

<span>

<el-button type="text" size="small" v-on:click="edit(row.ID)" icon="edit">编 辑</el-button>

<el-button type="text" size="small" v-on:click="del(row.ID)" icon="delete">删 除</el-button>

</span>

</el-table-column>

</el-table>

</div>

<div class="pager">

<el-pagination v-bind:current-Page="pageIndex" v-bind:page-size="pageSize" :total="total"

layout="total,sizes,prev,pager,next,jumper" v-bind:page-sizes="pageSizes"

v-on:size-change="sizeChange" v-on:current-change="pageIndexChange">

</el-pagination>

</div>

<div>

<el-dialog v-bind:title="title" v-bind:close-on-click-modal="false" v-model="showOperateManifest" v-on:close="dialogClosed">

<el-form v-bind:model="manifest" v-bind:rules="rules" ref="formManifest" label-position="left" label-width="80px">

<el-form-item label="日 期" prop="Date">

<el-date-picker v-model="manifest.Date"></el-date-picker>

</el-form-item>

<el-form-item label="金 额" prop="Cost">

<el-input v-model="manifest.Cost"></el-input>

</el-form-item>

<el-form-item label="备 注" prop="Remark">

<el-input v-model="manifest.Remark"></el-input>

</el-form-item>

<el-form-item>

<el-button type="primary" v-on:click="save">确 定</el-button>

<el-button type="primary" v-on:click="cancel">取 消</el-button>

</el-form-item>

</el-form>

</el-dialog>

</div>

</div>

</script>

[复制代码](javascript:void(0);)

查询部分　https://images2015.cnblogs.com/blog/481360/201611/481360-20161116102543435-560542487.png：

[复制代码](javascript:void(0);)

<div>

开始日期：

<el-date-picker v-model="start" type="date" placeholder="选择日期"></el-date-picker>

结束日期：

<el-date-picker v-model="end" type="date" placeholder="选择日期"></el-date-picker>

<el-button type="primary" size="small" v-on:click="fetchData" icon="search">查 询</el-button>

<el-button type="primary" size="small" v-on:click="add" icon="plus">添 加</el-button>

</div>

[复制代码](javascript:void(0);)

这里关于事件处理绑定，官网推荐简写的@click，但这里没有采用，而是使用了完整绑定V-on：click，因为考虑到以后可能会和Razor整合，@符可能会冲突

查询JS：

[复制代码](javascript:void(0);)

fetchData: function () {

this.manifests = [];

this.$http.get("http://localhost:1500/api/Manifests/paged", {

params: {

start: this.start.format("yyyy-MM-dd"),

end: this.end.format("yyyy-MM-dd"),

pageIndex: this.pageIndex,

pageSize: this.pageSize

}

})

.then(response => {

this.total = response.body.count;

this.manifests = response.body.data;

})

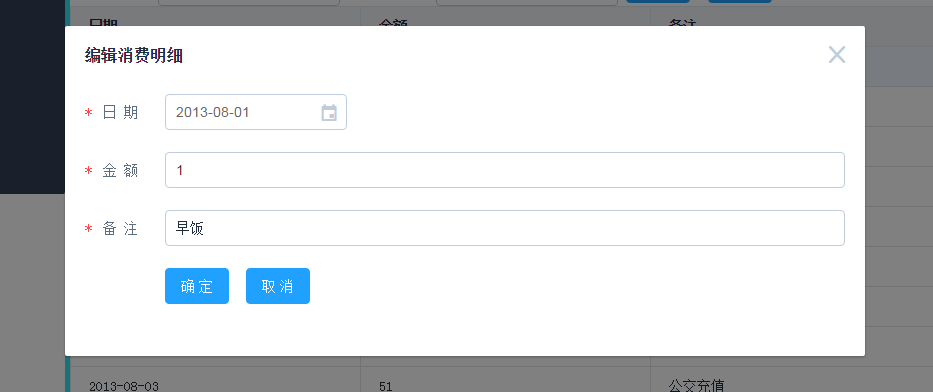
.catch(response => this.$alert(response.body.Message, "日消费明细", { type: "error" }));

}

[复制代码](javascript:void(0);)

API请求采用了Vue开源社区的vue-resource，简单轻便，再搭配ES6 Promise，写起来很顺手。

添加/编辑的实现：



这里使用了el-dialog嵌套el-form

[复制代码](javascript:void(0);)

<el-dialog v-bind:title="title" v-bind:close-on-click-modal="false" v-model="showOperateManifest" v-on:close="dialogClosed">

<el-form v-bind:model="manifest" v-bind:rules="rules" ref="formManifest" label-position="left" label-width="80px">

<el-form-item label="日 期" prop="Date">

<el-date-picker v-model="manifest.Date"></el-date-picker>

</el-form-item>

<el-form-item label="金 额" prop="Cost">

<el-input v-model="manifest.Cost"></el-input>

</el-form-item>

<el-form-item label="备 注" prop="Remark">

<el-input v-model="manifest.Remark"></el-input>

</el-form-item>

<el-form-item>

<el-button type="primary" v-on:click="save">确 定</el-button>

<el-button type="primary" v-on:click="cancel">取 消</el-button>

</el-form-item>

</el-form>

</el-dialog>

[复制代码](javascript:void(0);)

[复制代码](javascript:void(0);)

save: function () {

this.$refs.formManifest.validate(valid => {

if (valid) {

let operateManifest = JSON.parse(JSON.stringify(this.manifest));

operateManifest.Date = this.manifest.Date.format("yyyy-MM-dd");

if (this.isAdd) {

this.$http.post("http://localhost:1500/api/Manifests", operateManifest)

.then(() => {

this.manifests.push(operateManifest);

this.showOperateManifest = false;

bus.$emit("manifestChanged");

this.$message({

message: "添加成功",

type: "success"

});

})

.catch(err => {

//console.log(err);

this.$alert(err.body.Message, "添加日消费明细", { type: "error" });

});

}

else {

this.$http.put("http://localhost:1500/api/Manifests", operateManifest)

.then(response => {

let updatedManifest = this.manifests.find(x => x.ID == this.manifest.ID);

updatedManifest.Date = operateManifest.Date;

updatedManifest.Cost = operateManifest.Cost;

updatedManifest.Remark = operateManifest.Remark;

this.showOperateManifest = false;

bus.$emit("manifestChanged");

this.$message({

message: "修改成功",

type: "success"

});

})

.catch(err => {

//console.log(err);

this.$alert(err.body.Message, "修改消费明细", { type: "error" });

});

}

}

else {

return false;

}

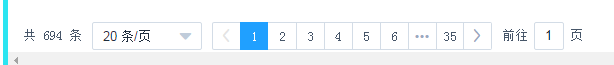
});

}

[复制代码](javascript:void(0);)

其中包括了表单验证部分，也是采用Element-UI文档中介绍的验证方式，目前有些验证支持不是很好，比如number，可能是哪里用的不对吧，所以上边对金额部分采取了正则自定义验证。

底部分页部分：



<el-pagination v-bind:current-Page="pageIndex" v-bind:page-size="pageSize" :total="total"

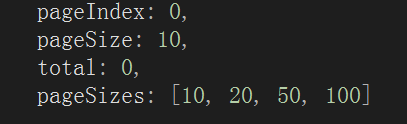
layout="total,sizes,prev,pager,next,jumper" v-bind:page-sizes="pageSizes"

v-on:size-change="sizeChange" v-on:current-change="pageIndexChange">

</el-pagination>

　　如上所示，直接使用了饿了么分页组件，设置几个属性， 再绑定几个JS属性、分页事件处理程序即可，十分方便

Vue Date对象中分页相关的属性：



页索引及分页大小变动事件处理：

[复制代码](javascript:void(0);)

sizeChange: function (pageSize) {

this.pageSize = pageSize;

this.fetchData();

},

pageIndexChange: function (pageIndex) {

this.pageIndex = pageIndex;

this.fetchData();

}

[复制代码](javascript:void(0);)

　　没什么特别， 根据最新页索引或页尺寸大小从新拉取一遍数据就行了。

**4.源码**

https://github.com/KINGGUOKUN/Account/tree/master/AccountWeb