北京联合大学毕业设计程序清单

题目：　　　　　　团队协同办公系统

专业：　计算机科学与技术　　　　指导教师：　孙悦

学院：　信息学院　　　　学号：　2013080332029

班级：　　计科 1303B　　　　姓名：　　王庭旭

一、程序清单

文件名：.gitignore

```
*.csv
*.dat
*.iml
*.log
*.out
*.pid
*.seed
*.sublime-*
*.swo
*.swp
*.tgz
*.xml
.DS_Store
.idea
.project
.strong-pm
coverage
node_modules
npm-debug.log
dist/
yarn-error.log
```

文件名：.yo-rc.json

```
{
  "generator-loopback": {}
}
```

文件名：README.md

~~~
# worktile_vue
----
- 简述

  基于 vue.js 实现的 worktile 基础版

  本人的本科毕业设计

- 使用方法
  - 首先确认安装了 node 环境后开始下列步骤

  - 开启一个终端，在项目根目录下安装服务端依赖：
  ```shell
    $ npm install
  ```
  - 安装依赖后，在当前目录（根目录）下启动后台服务：
  ```shell
    $ npm run start
  ```
  - 新开启一个终端进入 client 文件夹：
  ```shell
    $ cd client
  ```
  - 安装客户端依赖
  ```shell
    $ npm install
  ```
  - 安装好后启动前端服务：
  ```shell
    $ npm run dev
  ```
  - 在浏览器访问 http://127.0.0.1:8080
~~~

文件名：package.json

```
{
  "name": "worktile_vue",
  "version": "1.0.0",
  "main": "server/server.js",
  "scripts": {
    "lint": "eslint .",
    "start": "node .",
    "posttest": "npm run lint && nsp check",
    "dev": "pm2-dev start ."
  },
  "dependencies": {
    "compression": "^1.0.3",
    "cors": "^2.5.2",
    "helmet": "^1.3.0",
    "loopback-boot": "^2.6.5",
    "serve-favicon": "^2.0.1",
    "strong-error-handler": "^1.0.1",
    "loopback-component-explorer": "^4.0.0",
    "loopback": "^3.0.0"
  },
  "devDependencies": {
    "eslint": "^2.13.1",
    "eslint-config-loopback": "^4.0.0",
    "nsp": "^2.1.0"
  },
  "repository": {
    "type": "",
    "url": ""
  },
  "license": "UNLICENSED",
  "description": "worktile_vue"
}
```

文件名：client/.babelrc

```
{
  "presets": [
    ["env", { "modules": false }],
    "stage-2"
  ],
  "plugins": ["transform-runtime"],
  "comments": false,
  "env": {
    "test": {
      "presets": ["env", "stage-2"],
      "plugins": [ "istanbul" ]
    }
  }
}
```

文件名：client/.editorconfig

```
root = true

[*]
charset = utf-8
indent_style = space
indent_size = 2
end_of_line = lf
insert_final_newline = true
trim_trailing_whitespace = true
```

文件名：client/.eslintignore

```
build/*.js
config/*.js
```

文件名：client/.eslintrc.js

```
// http://eslint.org/docs/user-guide/configuring
```

```
module.exports = {
  root: true,
  parser: 'babel-eslint',
  parserOptions: {
    sourceType: 'module'
  },
  env: {
    browser: true,
  },
  // https://github.com/feross/standard/blob/master/RULES.md#javascript-standard-style
  extends: 'standard',
  // required to lint *.vue files
  plugins: [
    'html'
  ],
  // add your custom rules here
  'rules': {
    // allow paren-less arrow functions
    'arrow-parens': 0,
    // allow async-await
    'generator-star-spacing': 0,
    // allow debugger during development
    'no-debugger': process.env.NODE_ENV === 'production' ? 2 : 0
  }
}
```

文件名：client/.postcssrc.js

```
// https://github.com/michael-ciniawsky/postcss-load-config

module.exports = {
  "plugins": {
    // to edit target browsers: use "browserlist" field in package.json
    "autoprefixer": {}
  }
}
```

文件名：client/:qu

文件名：client/README.md

~~~
# worktile_vue

> A Vue.js project

## Build Setup

``` bash
# install dependencies
npm install

# serve with hot reload at localhost:8080
npm run dev

# build for production with minification
npm run build

# build for production and view the bundle analyzer report
npm run build --report
```

For detailed explanation on how things work, checkout the [guide](http://vuejs-templates.github.io/webpack/) and [docs for vue-loader](http://vuejs.github.io/vue-loader).
~~~

文件名：client/index.html

```
<!DOCTYPE html>
<html>
  <head>
```

```
    <meta charset="utf-8">
    <title>worktile_vue</title>
    <link href="//cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">
    <script src="//cdn.bootcss.com/jquery/1.10.2/jquery.min.js"></script>
    <script src="//cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
  </head>
  <body>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

文件名：client/package.json

```
{
  "name": "worktile_vue",
  "version": "1.0.0",
  "description": "A Vue.js project",
  "author": "tingxu.wang <768165932@qq.com>",
  "private": true,
  "scripts": {
    "dev": "node build/dev-server.js",
    "build": "node build/build.js",
    "lint": "eslint --ext .js,.vue src"
  },
  "dependencies": {
    "bootstrap": "^3.3.7",
    "iview": "^2.0.0-rc.2",
    "jQuery": "^1.7.4",
    "marked": "^0.3.6",
    "vue": "^2.2.1",
    "vue-resource": "^1.2.1",
    "vue-router": "^2.2.0"
  },
  "devDependencies": {
    "autoprefixer": "^6.7.2",
    "babel-core": "^6.22.1",
    "babel-eslint": "^7.1.1",
    "babel-loader": "^6.2.10",
    "babel-plugin-transform-runtime": "^6.22.0",
    "babel-preset-env": "^1.2.1",
    "babel-preset-stage-2": "^6.22.0",
    "babel-register": "^6.22.0",
    "chalk": "^1.1.3",
    "connect-history-api-fallback": "^1.3.0",
    "copy-webpack-plugin": "^4.0.1",
    "css-loader": "^0.26.1",
    "eslint": "^3.14.1",
    "eslint-config-standard": "^6.2.1",
    "eslint-friendly-formatter": "^2.0.7",
    "eslint-loader": "^1.6.1",
    "eslint-plugin-html": "^2.0.0",
    "eslint-plugin-promise": "^3.4.0",
    "eslint-plugin-standard": "^2.0.1",
    "eventsource-polyfill": "^0.9.6",
    "express": "^4.14.1",
    "extract-text-webpack-plugin": "^2.0.0",
    "file-loader": "^0.10.0",
    "friendly-errors-webpack-plugin": "^1.1.3",
    "function-bind": "^1.1.0",
    "html-webpack-plugin": "^2.28.0",
    "http-proxy-middleware": "^0.17.3",
    "node-sass": "^4.5.0",
    "opn": "^4.0.2",
    "optimize-css-assets-webpack-plugin": "^1.3.0",
    "ora": "^1.1.0",
    "rimraf": "^2.6.0",
    "sass-loader": "^6.0.3",
    "semver": "^5.3.0",
    "url-loader": "^0.5.7",
    "vue-loader": "^11.1.4",
    "vue-style-loader": "^2.0.0",
    "vue-template-compiler": "^2.2.1",
    "webpack": "^2.2.1",
    "webpack-bundle-analyzer": "^2.2.1",
    "webpack-dev-middleware": "^1.10.0",
    "webpack-hot-middleware": "^2.16.1",
    "webpack-merge": "^2.6.1"
  },
  "engines": {
    "node": ">= 4.0.0",
    "npm": ">= 3.0.0"
  },
  "browserslist": [
    "> 1%",
    "last 2 versions",
    "not ie <= 8"
  ]
}
```

文件名：memory/db.json

```
{
  "ids": {
    "User": 1,
    "AccessToken": 1,
    "ACL": 1,
    "RoleMapping": 1,
    "Role": 1,
    "Person": 3,
    "Group": 4,
    "Person_has_group": 7,
    "Project": 6,
    "Task_group": 4,
    "Task": 10,
    "Document": 2
  },
  "models": {
    "User": {},
    "AccessToken": {},
    "ACL": {},
    "RoleMapping": {},
    "Role": {},
    "Person": {
      "1": "{\"firstName\":\"王\",\"secondName\":\"庭旭\",\"name\":\"王庭旭\",\"headName\":\"庭旭\",\"phone\":\"15801183454\",\"password\":\"qwaa1234\",\"id\":1}",
      "2": "{\"firstName\":\"蒋\",\"secondName\":\"圆\",\"name\":\"蒋圆\",\"headName\":\"蒋圆\",\"phone\":\"13901311150\",\"password\":\"qwaa1234\",\"id\":2}"
    },
    "Group": {
        "1":
"{\"leader_id\":1,\"leader_phone\":\"15801183454\",\"create_time\":\"1493647004005\",\"user_ids\":[2,
1],\"name\":\"金堤科技研发部\",\"id\":1}",
        "2":
"{\"leader_id\":1,\"leader_phone\":\"15801183454\",\"create_time\":\"1493653020221\",\"user_ids\":[2,
1],\"name\":\"金堤科技销售部\",\"id\":2}",
        "3":
"{\"leader_id\":1,\"leader_phone\":\"15801183454\",\"create_time\":\"1494067748868\",\"user_ids\":[2,
1],\"name\":\"金堤科技人力部\",\"id\":3}"
      },
      "Person_has_group": {
        "1": "{\"person_id\":2,\"group_id\":1,\"id\":1}",
        "2": "{\"person_id\":1,\"group_id\":1,\"id\":2}",
        "3": "{\"person_id\":2,\"group_id\":2,\"id\":3}",
        "4": "{\"person_id\":1,\"group_id\":2,\"id\":4}",
        "5": "{\"person_id\":2,\"group_id\":3,\"id\":5}",
        "6": "{\"person_id\":1,\"group_id\":3,\"id\":6}"
      },
      "Project": {
        "1": "{\"group_id\":1,\"name\":\"web标准版\",\"description\":\"付费标准版项目\",\"id\":1}",
        "2": "{\"group_id\":1,\"name\":\"web免费版\",\"description\":\"所有用户都可放哪个问的开放
式web免费版\",\"id\":2}",
        "4": "{\"group_id\":1,\"name\":\"手机移动版\",\"description\":\"web手机端项目，安卓及ios
都可使用\",\"id\":4}",
        "5": "{\"group_id\":1,\"name\":\"桌面版\",\"description\":\"用于windows系统的桌面客户端
\",\"id\":5}"
      },
      "Task_group": {
        "1": "{\"project_id\":1,\"name\":\"站点主入口页\",\"index\":0,\"project_name\":\"web标准版
\",\"id\":1}",
        "2": "{\"project_id\":1,\"name\":\"公司详情页\",\"index\":1,\"project_name\":\"web标准版
\",\"id\":2}",
        "3": "{\"project_id\":1,\"name\":\"关系图谱\",\"index\":2,\"project_name\":\"web标准版
\",\"id\":3}"
      },
      "Task": {
        "1": "{\"task_group_id\":1,\"task_group_name\":\"站点主入口页
\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":1,\"headName\":\"庭旭
\"}],\"dead_time\":\"\",\"tags\":[],\"name\":\"主搜索功能实现
\",\"create_time\":\"1493648472908\",\"create_person_id\":\"1\",\"is_solved\":false,\"project_id\":\"1\",\
"project_name\":\"web标准版\",\"id\":1}",
        "2": "{\"task_group_id\":1,\"task_group_name\":\"站点主入口页
```

\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":1,\"headName\":\"庭旭\"}],\"dead_time\":\"\",\"tags\":[],\"name\":\"调研通过自然人名称搜索相关公司的可行性\",\"create_time\":\"1493648526758\",\"create_person_id\":\"1\",\"is_solved\":true,\"project_id\":\"1\",\"project_name\":\"web标准版\",\"id\":2}",

    "3": "{\"task_group_id\":1,\"task_group_name\":\"站点主入口页\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":2,\"headName\":\"蒋圆\"},{\"firstName\":\"王\",\"secondName\":\"庭旭\",\"name\":\"王庭旭\",\"headName\":\"庭旭\",\"phone\":\"15801183454\",\"id\":1}],\"dead_time\":\"\",\"tags\":[],\"name\":\"通过企业id搜索公司名称实现\",\"create_time\":\"1493648583457\",\"create_person_id\":\"2\",\"is_solved\":true,\"project_id\":\"1\",\"project_name\":\"web标准版\",\"id\":3}",

    "4": "{\"task_group_id\":2,\"task_group_name\":\"公司详情页\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":1,\"headName\":\"庭旭\"}],\"dead_time\":\"\",\"tags\":[],\"name\":\"搜索结果页面跳转到公司详情页实现\",\"create_time\":\"1493648635863\",\"create_person_id\":\"1\",\"is_solved\":true,\"project_id\":\"1\",\"project_name\":\"web标准版\",\"id\":4}",

    "5": "{\"task_group_id\":2,\"task_group_name\":\"公司详情页\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":2,\"headName\":\"蒋圆\"}],\"dead_time\":\"\",\"tags\":[],\"name\":\"详情页公司报表功能实现\",\"create_time\":\"1493648667232\",\"create_person_id\":\"2\",\"is_solved\":true,\"project_id\":\"1\",\"project_name\":\"web标准版\",\"id\":5}",

    "6": "{\"task_group_id\":2,\"task_group_name\":\"公司详情页\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":2,\"headName\":\"蒋圆\"}],\"dead_time\":\"\",\"tags\":[],\"name\":\"公司经营信息抓取以及展示实现\",\"create_time\":\"1493648689738\",\"create_person_id\":\"2\",\"is_solved\":true,\"project_id\":\"1\",\"project_name\":\"web标准版\",\"id\":6}",

    "7": "{\"task_group_id\":2,\"task_group_name\":\"公司详情页\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":1,\"headName\":\"庭旭\"}],\"dead_time\":\"\",\"tags\":[],\"name\":\"企业报表功能实现\",\"create_time\":\"1493648706720\",\"create_person_id\":\"1\",\"is_solved\":false,\"project_id\":\"1\",\"project_name\":\"web标准版\",\"id\":7}",

    "8": "{\"task_group_id\":2,\"task_group_name\":\"公司详情页\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":1,\"headName\":\"庭旭\"}],\"dead_time\":\"\",\"tags\":[],\"name\":\"公司风险信息抓取以及展示实现\",\"create_time\":\"1493648922934\",\"create_person_id\":\"1\",\"is_solved\":false,\"project_id\":\"1\",\"project_name\":\"web标准版\",\"id\":8}",

    "9": "{\"task_group_id\":2,\"task_group_name\":\"公司详情页\",\"description\":\"\",\"solve_person_info\":[],\"focus_person_info\":[{\"id\":1,\"headName\":\"庭旭\"}],\"dead_time\":\"\",\"tags\":[],\"name\":\"公司评价功能实现\",\"create_time\":\"1493653281450\",\"create_person_id\":\"1\",\"is_solved\":false,\"project_id\":\"1\",\"project_name\":\"web标准版\",\"id\":9}"

  },

```
      "Document": {
        "1": "{\"project_id\":1,\"title\":\"主搜索框需求详细说明\",\"content\":\"<ul>\\n<li>客户端发送的 ajax 请求中搜索关键字需要转码</li>\\n<li>客户端发送请求前应当进行信息校验，对于特殊字符加以屏蔽</li>\\n</ul>\\n\\n\",\"person_id\":1,\"person_name\":\"王庭旭\",\"create_time\":\"1493691748978\",\"id\":1}"
      }
  }
}
```

文件名：out-load/index.js

```
const fs=require('fs');
const path=require('path')

const ignoreNames=['node_modules','.idea','.git','yarn-error.log','npm-debug.log','assets','common','yarn.lock']//忽略的文件或文件夹名称
const dirPath='../'

function deleteIgnore(target){
  target.forEach((fileName,index)=>{
    ignoreNames.forEach((ignoreName)=>{
      if(fileName===ignoreName){
        target.splice(index,1)
      }
    })
  })
}

function getOutput(dirPath){
  fs.readdir(dirPath,(err,data)=>{
    if(err){console.log(err)}
    let files=data
    let fileArr=[]//{pathFileName:",content:"}

    deleteIgnore(files)
    files.forEach((fileName)=>{
      const filePath=path.join(dirPath,fileName)
      fs.stat(filePath,(err,data)=>{
        if(err){console.log(err)}
```

```
            if(data.isDirectory()){//是目录
               getOutput(filePath+'/')
            }else{//是文件
               fs.readFile(filePath,(err,data)=>{
                  if(err){console.log(err)}
                  const resultFilePath=filePath.replace('../',")
                  const result=`\n\n\n 文件名：${resultFilePath}\n\n\n${data}`
                  fs.appendFile('./output.txt',result.toString('utf-8'))
               })
            }
         })
      })
   })
}
getOutput(dirPath)
```

文件名：out-load/output.txt

文件名：server/component-config.json

```
{
   "loopback-component-explorer": {
      "mountPath": "/explorer"
   }
}
```

文件名：server/config.json

```
{
   "restApiRoot": "/api",
   "host": "0.0.0.0",
   "port": 3000,
  "remoting": {
    "context": false,
    "rest": {
      "handleErrors": false,
      "normalizeHttpPath": false,
      "xml": false
    },
    "json": {
      "strict": false,
      "limit": "100kb"
    },
    "urlencoded": {
      "extended": true,
      "limit": "100kb"
    },
    "cors": false
  }
}
```

文件名：server/datasources.json

```
{
  "db": {
    "name": "db",
    "connector": "memory",
    "file": "memory/db.json"
  }
}
```

文件名：server/middleware.development.json

```
{
  "final:after": {
    "strong-error-handler": {
      "params": {
        "debug": true,
        "log": true
      }
    }
  }
}
```

文件名：server/middleware.json

```
{
  "initial:before": {
    "loopback#favicon": {}
  },
  "initial": {
    "compression": {},
    "cors": {
      "params": {
        "origin": true,
        "credentials": true,
        "maxAge": 86400
      }
    },
    "helmet#xssFilter": {},
    "helmet#frameguard": {
      "params": [
        "deny"
      ]
    },
    "helmet#hsts": {
      "params": {
        "maxAge": 0,
        "includeSubdomains": true
      }
    },
    "helmet#hidePoweredBy": {},
    "helmet#ieNoOpen": {},
    "helmet#noSniff": {},
    "helmet#noCache": {
      "enabled": false
    }
  },
  "session": {},
  "auth": {},
  "parse": {},
  "routes": {
    "loopback#rest": {
      "paths": [
        "${restApiRoot}"
      ]
    }
  },
  "files": {},
  "final": {
    "loopback#urlNotFound": {}
  },
  "final:after": {
    "strong-error-handler": {}
  }
}
```

文件名：server/server.js

```
'use strict';

var loopback = require('loopback');
var boot = require('loopback-boot');

var app = module.exports = loopback();

app.start = function() {
  // start the web server
  return app.listen(function() {
    app.emit('started');
    var baseUrl = app.get('url').replace(/\/$/, '');
    console.log('Web server listening at: %s', baseUrl);
    if (app.get('loopback-component-explorer')) {
      var explorerPath = app.get('loopback-component-explorer').mountPath;
      console.log('Browse your REST API at %s%s', baseUrl, explorerPath);
    }
  });
};

// Bootstrap the application, configure models, datasources and middleware.
// Sub-apps like REST API are mounted via boot scripts.
boot(app, __dirname, function(err) {
  if (err) throw err;

  // start the server if `$ node server.js`
  if (require.main === module)
    app.start();
});
```

文件名：server/model-config.json

```
{
  "_meta": {
    "sources": [
      "loopback/common/models",
      "loopback/server/models",
      "../common/models",
      "./models"
    ],
    "mixins": [
      "loopback/common/mixins",
      "loopback/server/mixins",
      "../common/mixins",
      "./mixins"
    ]
  },
  "User": {
    "dataSource": "db"
  },
  "AccessToken": {
    "dataSource": "db",
    "public": false
  },
  "ACL": {
    "dataSource": "db",
    "public": false
  },
  "RoleMapping": {
    "dataSource": "db",
    "public": false,
    "options": {
      "strictObjectIDCoercion": true
    }
  },
  "Role": {
    "dataSource": "db",
    "public": false
  },
  "Person": {
    "dataSource": "db",
    "public": true
  },
  "Person_has_group": {
    "dataSource": "db",
    "public": true
  },
  "Group": {
    "dataSource": "db",
    "public": true
  },
  "Project": {
    "dataSource": "db",
    "public": true
  },
  "Moment": {
    "dataSource": "db",
    "public": true
  },
  "Task_group": {
    "dataSource": "db",
    "public": true
  },
  "Document": {
    "dataSource": "db",
    "public": true
  },
  "Task": {
    "dataSource": "db",
    "public": true
  },
  "Comment": {
    "dataSource": "db",
    "public": true
  }
}
```

文件名：client/config/dev.env.js

```
var merge = require('webpack-merge')
var prodEnv = require('./prod.env')

module.exports = merge(prodEnv, {
  NODE_ENV: '"development"'
})
```

文件名：client/config/index.js

```
// see http://vuejs-templates.github.io/webpack for documentation.
var path = require('path')

module.exports = {
  build: {
    env: require('./prod.env'),
    index: path.resolve(__dirname, '../dist/index.html'),
    assetsRoot: path.resolve(__dirname, '../dist'),
    assetsSubDirectory: 'static',
    assetsPublicPath: '/',
    productionSourceMap: true,
    // Gzip off by default as many popular static hosts such as
    // Surge or Netlify already gzip all static assets for you.
    // Before setting to `true`, make sure to:
    // npm install --save-dev compression-webpack-plugin
    productionGzip: false,
    productionGzipExtensions: ['js', 'css'],
    // Run the build command with an extra argument to
    // View the bundle analyzer report after build finishes:
    // `npm run build --report`
    // Set to `true` or `false` to always turn it on or off
    bundleAnalyzerReport: process.env.npm_config_report
  },
  dev: {
    env: require('./dev.env'),
    port: 8080,
    autoOpenBrowser: true,
    assetsSubDirectory: 'static',
    assetsPublicPath: '/',
    proxyTable: {
      '/api':{
        target: 'http://127.0.0.1:3000',
        changeOrigin: true,
        pathRewrite: {
          // '^/api': ''
        }
      }
    },
    // CSS Sourcemaps off by default because relative paths are "buggy"
    // with this option, according to the CSS-Loader README
    // (https://github.com/webpack/css-loader#sourcemaps)
    // In our experience, they generally work as expected,
    // just be aware of this issue when enabling this option.
    cssSourceMap: false
  }
}
```

文件名：client/config/prod.env.js

```
module.exports = {
  NODE_ENV: '"production"'
}
```

文件名：client/build/build.js

```
require('./check-versions')()

process.env.NODE_ENV = 'production'

var ora = require('ora')
var rm = require('rimraf')
var path = require('path')
var chalk = require('chalk')
var webpack = require('webpack')
var config = require('../config')
var webpackConfig = require('./webpack.prod.conf')

var spinner = ora('building for production...')
spinner.start()

rm(path.join(config.build.assetsRoot, config.build.assetsSubDirectory), err => {
  if (err) throw err
  webpack(webpackConfig, function (err, stats) {
    spinner.stop()
    if (err) throw err
    process.stdout.write(stats.toString({
      colors: true,
      modules: false,
      children: false,
      chunks: false,
      chunkModules: false
    }) + '\n\n')

    console.log(chalk.cyan('  Build complete.\n'))
    console.log(chalk.yellow(
      '  Tip: built files are meant to be served over an HTTP server.\n' +
      '  Opening index.html over file:// won\'t work.\n'
    ))
  })
})
```

文件名：client/build/check-versions.js

```
var chalk = require('chalk')
var semver = require('semver')
var packageConfig = require('../package.json')

function exec (cmd) {
  return require('child_process').execSync(cmd).toString().trim()
}

var versionRequirements = [
  {
    name: 'node',
    currentVersion: semver.clean(process.version),
    versionRequirement: packageConfig.engines.node
  },
  {
    name: 'npm',
    currentVersion: exec('npm --version'),
    versionRequirement: packageConfig.engines.npm
  }
]

module.exports = function () {
  var warnings = []
  for (var i = 0; i < versionRequirements.length; i++) {
    var mod = versionRequirements[i]
    if (!semver.satisfies(mod.currentVersion, mod.versionRequirement)) {
      warnings.push(mod.name + ': ' +
        chalk.red(mod.currentVersion) + ' should be ' +
        chalk.green(mod.versionRequirement)
      )
    }
  }

  if (warnings.length) {
    console.log('')
    console.log(chalk.yellow('To use this template, you must update following to modules:'))
    console.log()
    for (var i = 0; i < warnings.length; i++) {
      var warning = warnings[i]
      console.log('    ' + warning)
    }
    console.log()
    process.exit(1)
  }
}
```

文件名：client/build/dev-client.js

```
/* eslint-disable */
require('eventsource-polyfill')
var hotClient = require('webpack-hot-middleware/client?noInfo=true&reload=true')

hotClient.subscribe(function (event) {
  if (event.action === 'reload') {
    window.location.reload()
  }
})
```

文件名：client/build/dev-server.js

```
require('./check-versions')()

var config = require('../config')
if (!process.env.NODE_ENV) {
  process.env.NODE_ENV = JSON.parse(config.dev.env.NODE_ENV)
}

var opn = require('opn')
var path = require('path')
var express = require('express')
var webpack = require('webpack')
var proxyMiddleware = require('http-proxy-middleware')
var webpackConfig = require('./webpack.dev.conf')

// default port where dev server listens for incoming traffic
var port = process.env.PORT || config.dev.port
```

```
// automatically open browser, if not set will be false
var autoOpenBrowser = !!config.dev.autoOpenBrowser
// Define HTTP proxies to your custom API backend
// https://github.com/chimurai/http-proxy-middleware
var proxyTable = config.dev.proxyTable

var app = express()
var compiler = webpack(webpackConfig)

var devMiddleware = require('webpack-dev-middleware')(compiler, {
  publicPath: webpackConfig.output.publicPath,
  quiet: true
})

var hotMiddleware = require('webpack-hot-middleware')(compiler, {
  log: () => {}
})
// force page reload when html-webpack-plugin template changes
compiler.plugin('compilation', function (compilation) {
  compilation.plugin('html-webpack-plugin-after-emit', function (data, cb) {
    hotMiddleware.publish({ action: 'reload' })
    cb()
  })
})

// proxy api requests
Object.keys(proxyTable).forEach(function (context) {
  var options = proxyTable[context]
  if (typeof options === 'string') {
    options = { target: options }
  }
  app.use(proxyMiddleware(options.filter || context, options))
})

// handle fallback for HTML5 history API
app.use(require('connect-history-api-fallback')())

// serve webpack bundle output
app.use(devMiddleware)

// enable hot-reload and state-preserving
// compilation error display
```

```
app.use(hotMiddleware)

// serve pure static assets
var staticPath = path.posix.join(config.dev.assetsPublicPath, config.dev.assetsSubDirectory)
app.use(staticPath, express.static('./static'))

var uri = 'http://localhost:' + port

devMiddleware.waitUntilValid(function () {
  console.log('> Listening at ' + uri + '\n')
})

module.exports = app.listen(port, function (err) {
  if (err) {
    console.log(err)
    return
  }

  // when env is testing, don't need open it
  if (autoOpenBrowser && process.env.NODE_ENV !== 'testing') {
    opn(uri)
  }
})
```

文件名：client/build/utils.js

```
var path = require('path')
var config = require('../config')
var ExtractTextPlugin = require('extract-text-webpack-plugin')

exports.assetsPath = function (_path) {
  var assetsSubDirectory = process.env.NODE_ENV === 'production'
    ? config.build.assetsSubDirectory
    : config.dev.assetsSubDirectory
  return path.posix.join(assetsSubDirectory, _path)
}

exports.cssLoaders = function (options) {
  options = options || {}
```

```
var cssLoader = {
  loader: 'css-loader',
  options: {
    minimize: process.env.NODE_ENV === 'production',
    sourceMap: options.sourceMap
  }
}

// generate loader string to be used with extract text plugin
function generateLoaders (loader, loaderOptions) {
  var loaders = [cssLoader]
  if (loader) {
    loaders.push({
      loader: loader + '-loader',
      options: Object.assign({}, loaderOptions, {
        sourceMap: options.sourceMap
      })
    })
  }

  // Extract CSS when that option is specified
  // (which is the case during production build)
  if (options.extract) {
    return ExtractTextPlugin.extract({
      use: loaders,
      fallback: 'vue-style-loader'
    })
  } else {
    return ['vue-style-loader'].concat(loaders)
  }
}

// http://vuejs.github.io/vue-loader/en/configurations/extract-css.html
return {
  css: generateLoaders(),
  postcss: generateLoaders(),
  less: generateLoaders('less'),
  sass: generateLoaders('sass', { indentedSyntax: true }),
  scss: generateLoaders('sass'),
  stylus: generateLoaders('stylus'),
  styl: generateLoaders('stylus')
  }
}

// Generate loaders for standalone style files (outside of .vue)
exports.styleLoaders = function (options) {
  var output = []
  var loaders = exports.cssLoaders(options)
  for (var extension in loaders) {
    var loader = loaders[extension]
    output.push({
      test: new RegExp('\\.' + extension + '$'),
      use: loader
    })
  }
  return output
}
```

文件名：client/build/vue-loader.conf.js

```
var utils = require('./utils')
var config = require('../config')
var isProduction = process.env.NODE_ENV === 'production'

module.exports = {
  loaders: utils.cssLoaders({
    sourceMap: isProduction
      ? config.build.productionSourceMap
      : config.dev.cssSourceMap,
    extract: isProduction
  })
}
```

文件名：client/build/webpack.base.conf.js

```
var path = require('path')
var utils = require('./utils')
var config = require('../config')
var vueLoaderConfig = require('./vue-loader.conf')

function resolve (dir) {
  return path.join(__dirname, '..', dir)
}

module.exports = {
  entry: {
    app: './src/main.js'
  },
  output: {
    path: config.build.assetsRoot,
    filename: '[name].js',
    publicPath: process.env.NODE_ENV === 'production'
      ? config.build.assetsPublicPath
      : config.dev.assetsPublicPath
  },
  resolve: {
    extensions: ['.js', '.vue', '.json'],
    alias: {
      'vue$': 'vue/dist/vue.esm.js',
      '@': resolve('src'),
      '@panelContent': path.join(__dirname,'..','src/components/center-panel-content'),
      '@pub': path.join(__dirname,'..','src/components/pub')
    }
  },
  module: {
    rules: [
      /*{
        test: /\.(js|vue)$/,
        loader: 'eslint-loader',
        enforce: "pre",
        include: [resolve('src'), resolve('test')],
        options: {
          formatter: require('eslint-friendly-formatter')
        }
      },*/
      {
        test: /\.vue$/,
        loader: 'vue-loader',
        options: vueLoaderConfig
      },
      {
        test: /\.js$/,
        loader: 'babel-loader',
        include: [resolve('src'), resolve('test')]
      },
      {
        test: /\.(png|jpe?g|gif|svg)(\?.*)?$/,
        loader: 'url-loader',
        query: {
          limit: 10000,
          name: utils.assetsPath('img/[name].[hash:7].[ext]')
        }
      },
      {
        test: /\.(woff2?|eot|ttf|otf)(\?.*)?$/,
        loader: 'url-loader',
        query: {
          limit: 10000,
          name: utils.assetsPath('fonts/[name].[hash:7].[ext]')
        }
      }
    ]
  }
}
```

文件名：client/build/webpack.dev.conf.js

```
var utils = require('./utils')
var webpack = require('webpack')
var config = require('../config')
var merge = require('webpack-merge')
var baseWebpackConfig = require('./webpack.base.conf')
var HtmlWebpackPlugin = require('html-webpack-plugin')
var FriendlyErrorsPlugin = require('friendly-errors-webpack-plugin')

// add hot-reload related code to entry chunks
Object.keys(baseWebpackConfig.entry).forEach(function (name) {
  baseWebpackConfig.entry[name] = ['./build/dev-client'].concat(baseWebpackConfig.entry[name])
})

module.exports = merge(baseWebpackConfig, {
  module: {
    rules: utils.styleLoaders({ sourceMap: config.dev.cssSourceMap })
  },
  // cheap-module-eval-source-map is faster for development
  devtool: '#cheap-module-eval-source-map',
  plugins: [
    new webpack.DefinePlugin({
      'process.env': config.dev.env
    }),
    // https://github.com/glenjamin/webpack-hot-middleware#installation--usage
    new webpack.HotModuleReplacementPlugin(),
    new webpack.NoEmitOnErrorsPlugin(),
    // https://github.com/ampedandwired/html-webpack-plugin
    new HtmlWebpackPlugin({
      filename: 'index.html',
      template: 'index.html',
      inject: true
    }),
    new FriendlyErrorsPlugin()
  ]
})
```

文件名：client/build/webpack.prod.conf.js

```
var path = require('path')
var utils = require('./utils')
var webpack = require('webpack')
var config = require('../config')
var merge = require('webpack-merge')
var baseWebpackConfig = require('./webpack.base.conf')
var CopyWebpackPlugin = require('copy-webpack-plugin')
var HtmlWebpackPlugin = require('html-webpack-plugin')
var ExtractTextPlugin = require('extract-text-webpack-plugin')
var OptimizeCSSPlugin = require('optimize-css-assets-webpack-plugin')

var env = config.build.env
var webpackConfig = merge(baseWebpackConfig, {
  module: {
    rules: utils.styleLoaders({
      sourceMap: config.build.productionSourceMap,
      extract: true
    })
  },
  devtool: config.build.productionSourceMap ? '#source-map' : false,
  output: {
    path: config.build.assetsRoot,
    filename: utils.assetsPath('js/[name].[chunkhash].js'),
    chunkFilename: utils.assetsPath('js/[id].[chunkhash].js')
  },
  plugins: [
    // http://vuejs.github.io/vue-loader/en/workflow/production.html
    new webpack.DefinePlugin({
      'process.env': env
    }),
    new webpack.optimize.UglifyJsPlugin({
      compress: {
        warnings: false
      },
      sourceMap: true
    }),
    // extract css into its own file
    new ExtractTextPlugin({
      filename: utils.assetsPath('css/[name].[contenthash].css')
    }),
    // Compress extracted CSS. We are using this plugin so that possible
    // duplicated CSS from different components can be deduped.
    new OptimizeCSSPlugin(),
    // generate dist index.html with correct asset hash for caching.
    // you can customize output by editing /index.html
    // see https://github.com/ampedandwired/html-webpack-plugin
    new HtmlWebpackPlugin({
      filename: config.build.index,
      template: 'index.html',
      inject: true,
      minify: {
        removeComments: true,
        collapseWhitespace: true,
        removeAttributeQuotes: true
        // more options:
        // https://github.com/kangax/html-minifier#options-quick-reference
      },
      // necessary to consistently work with multiple chunks via CommonsChunkPlugin
      chunksSortMode: 'dependency'
    }),
    // split vendor js into its own file
    new webpack.optimize.CommonsChunkPlugin({
      name: 'vendor',
      minChunks: function (module, count) {
        // any required modules inside node_modules are extracted to vendor
        return (
          module.resource &&
          /\.js$/.test(module.resource) &&
          module.resource.indexOf(
            path.join(__dirname, '../node_modules')
          ) === 0
        )
      }
    }),
    // extract webpack runtime and module manifest to its own file in order to
    // prevent vendor hash from being updated whenever app bundle is updated
    new webpack.optimize.CommonsChunkPlugin({
      name: 'manifest',
      chunks: ['vendor']
    }),
    // copy custom static assets
    new CopyWebpackPlugin([
      {
        from: path.resolve(__dirname, '../static'),
        to: config.build.assetsSubDirectory,
        ignore: ['.*']
      }
    ])
  ]
})

if (config.build.productionGzip) {
  var CompressionWebpackPlugin = require('compression-webpack-plugin')

  webpackConfig.plugins.push(
    new CompressionWebpackPlugin({
      asset: '[path].gz[query]',
      algorithm: 'gzip',
      test: new RegExp(
        '\\.(' +
        config.build.productionGzipExtensions.join('|') +
        ')$'
      ),
      threshold: 10240,
      minRatio: 0.8
    })
  )
}

if (config.build.bundleAnalyzerReport) {
  var BundleAnalyzerPlugin = require('webpack-bundle-analyzer').BundleAnalyzerPlugin
  webpackConfig.plugins.push(new BundleAnalyzerPlugin())
}

module.exports = webpackConfig
```

文件名：client/src/App.vue

```
<template>
  <div id="app">
    <router-view></router-view>
  </div>
</template>

<script>
export default {
  name: 'app'
}
</script>

<style lang="scss">
html,body,#app{
  height:100%;
}
```

```
#app{
   background: #F8F6F2;
}
.g-user-head{
   cursor:pointer;
   border-radius: 30px;
   background: #64b1ca;
   color:#fff;
   line-height: 40px;
   text-align: center;
}
.g-user-head-default{
   @extend .g-user-head;
   width:40px;
   height:40px;
   font-size: 14px;
}
.mt_30{
   margin-top:30px;
}
.mt_20{
   margin-top:20px;
}
.item{
   height: 70px;
   border-bottom: solid 1px #f2f1f7;
   cursor:pointer;
   .icon-container{
      display: inline-block;
      line-height: 70px;
      width: 50px;
      .icon{
         color: rgb(60, 140, 173);
         font-size: 25px;
      }
   }
   .text-container{
      display: inline-block;
      position: relative;
      top:5px;
      .description{
         color:lighten(#000,50);
```

```
            height:20px;
        }
        &:hover{
            color:#da4f4a;
        }
    }
    .delete-container{
        display: none;
        float: right;
        margin:5px 3px;
        &:hover{
            color:#da4f4a;
        }
    }
    &:hover{
        .delete-container{
            display: inline-block;
        }
    }
}
.content-main-container{
    background: #fff;
    min-height:900px;
    margin:7px 14px;
    padding:21px;
    .main-title{
        user-select: none;
        border-bottom: solid 1px #f8f6f2;
        padding: 0 15px 14px 2px;
        font-weight: 400;
        color: #777;
        font-size: 17px;
        line-height: 100%;
        display: inline-block;
    }
}
</style>
```

文件名：client/src/main.js

```
// The Vue build version to load with the `import` command
// (runtime-only or standalone) has been set in webpack.base.conf with an alias.
import Vue from 'vue'
import App from './App'
import router from './router'
import resource from 'vue-resource'

import {store} from './store/store.js'

import {MyPlugin} from './plugin/index.js'

Vue.config.productionTip = false

Vue.use(resource)
Vue.use(MyPlugin)

router.beforeEach((to,from,next)=>{
  store.dispatch('refreshRouterLevel',to.path)
  store.dispatch('refreshRouterName')
  next()
})

router.beforeEach((to,from,next)=>{//检测是否登录
  const isLogin=sessionStorage.getItem('isLogin'),
    firstRouterName=store.state.firstRouterName

  if(!isLogin && firstRouterName!=='login' && firstRouterName!=='regist'){
    next({name:'login'})
  }else{
    next()
  }
})

router.beforeEach((to,from,next)=>{//处理当初没想到的点击 head-nav 跳转三级路由的问题
  if(to.name==='project' && store.state.routerLevel[3]==='document'){
    router.push({name:'document'})
  }

  next()
})
/* eslint-disable no-new */
```

```
new Vue({
  el: '#app',
  router,
  store,
  template: '<App/>',
  components: { App },

  created (){
    console.log('created')
  },
  beforeMount (){
    console.log('beforeMount')
  },
  mounted (){
    console.log('mounted')
  },
  beforeDestroy (){
    console.log('beforeDestroy')
  },
  destroyed (){
    console.log('destroyed')
  },
  updated (){
    console.log('updated')
  },
  beforeUpdate (){
    console.log('beforeUpdate')
  }
})
```

文件名：client/static/.gitkeep

文件名：server/boot/authentication.js

```
'use strict';
```

```
module.exports = function enableAuthentication(server) {
  // enable authentication
  server.enableAuth();
};
```

文件名：server/boot/root.js

```
'use strict';

module.exports = function(server) {
  // Install a `/` route that returns server status
  var router = server.loopback.Router();
  router.get('/', server.loopback.status());
  server.use(router);
};
```

文件名：client/src/components/Hello.vue

```
<template>
  <div class="hello">
    <h1>{{ msg }}</h1>
    <h2>Essential Links</h2>
    <ul>
      <li><a href="https://vuejs.org" target="_blank">Core Docs</a></li>
      <li><a href="https://forum.vuejs.org" target="_blank">Forum</a></li>
      <li><a href="https://gitter.im/vuejs/vue" target="_blank">Gitter Chat</a></li>
      <li><a href="https://twitter.com/vuejs" target="_blank">Twitter</a></li>
      <br>
      <li><a href="http://vuejs-templates.github.io/webpack/" target="_blank">Docs for This Template</a></li>
    </ul>
    <h2>Ecosystem</h2>
    <ul>
      <li><a href="http://router.vuejs.org/" target="_blank">vue-router</a></li>
      <li><a href="http://vuex.vuejs.org/" target="_blank">vuex</a></li>
      <li><a href="http://vue-loader.vuejs.org/" target="_blank">vue-loader</a></li>
      <li><a href="https://github.com/vuejs/awesome-vue" target="_blank">awesome-vue</a></li>
    </ul>

    <button type="button" name="button" @click="test()">test</button>
  </div>
</template>

<script>
export default {
  name: 'hello',
  data () {
    return {
      msg: 'Welcome to Your Vue.js App'
    }
  },
  methods:{
    test (){
      //this.$store.commit('increment',5)
      this.$store.dispatch('increment',10)
      //console.log(this.$store.state.moduleA.count)
      // this.$store.dispatch('incrementB',10)
    }
  }
}
</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>
h1, h2 {
  font-weight: normal;
}

ul {
  list-style-type: none;
  padding: 0;
}

li {
  display: inline-block;
  margin: 0 10px;
}
```

```
a {
  color: #42b983;
}
</style>
```

文件名：client/src/components/Login.vue

```
<template lang="html">
  <log :err-msg="errMsg">
    <div class="form-group">
      <input type="text" class="form-control" v-model="formInfo.phone" placeholder="电话号码">
    </div>
    <div class="form-group">
      <input type="password" class="form-control" v-model="formInfo.password" placeholder="密码" @keydown.enter="submit()">
    </div>
    <div class="form-group">
      <button class="btn btn-block btn-lg btn-success" @click="submit()">登录</button>
    </div>
    <div class="panel-footer">
      <router-link :to="{name:'regist'}">去注册</router-link>
    </div>
  </log>
</template>

<script>
import log from './log'
export default {
  data (){
    return {
      errMsg:",
      formInfo:{
        phone:",
        password:"
      }
    }
  },
  methods:{
        isEmpty (){
            const result=this.$isFormEmpty(this.formInfo)
            if(!result){
                this.errMsg='请填写全所有的字段'
            }
            return result
        },
        submit (){
            if(this.isEmpty()){
                let filter={where:this.formInfo}
                this.$http.get('/api/People',{params:{filter}})
                    .then(({status,body})=>{
                        if(status===200){
                            if(body.length===1){//已注册
                                // this.$store.dispatch('login',body[0])
                                let infoObj=body[0]
                                this.$saveLoginInfo(infoObj)
                                this.$router.push({name:'main'})
                            }else{
                                this.errMsg='用户名或密码错误'
                            }
                        }else{
                            this.errMsg='服务器出错鸟！'
                        }
                    })
                    .catch(err=>{
                        console.error(err)
                    })
            }
        }
    },
    components:{log}
}
</script>

<style lang="scss" scoped>
</style>
```

文件名：client/src/components/Main.vue

```
<template lang="html">
    <div class="main">
        <panel></panel>
        <div class="center-panel">
            <headNavbar></headNavbar>
            <div class="center-panel-content">
                <router-view></router-view>
            </div>
        </div>
    </div>
</template>

<script>
import panel from './left-panel/left-panel.vue'
import headNavbar from './head-navbar.vue'

export default {
    data (){
        return {

        }
    },
    components:{
        panel,
        headNavbar
    }
}
</script>

<style lang="scss" scoped>
.main{
    height:100%;
}
.layout_content{
    position: relative;
    height:100%;
}
.center-panel{
    height: 100%;
    margin-right: 0;
    margin-left: 60px;
    position: relative;
}
.center-panel-content{
    padding-top: 7px;
    overflow-x: auto;
    white-space: nowrap;
    position:absolute;
    top: 56px;
    bottom: 0;
    left: 0;
    right: 0;
    z-index: 0;
}
</style>
```

文件名：client/src/components/Regist.vue

```
<template lang="html">
    <log :err-msg="errMsg">
        <div class="form-group">
            <input type="text" class="form-control" v-model="formInfo.firstName" placeholder="姓">
        </div>
        <div class="form-group">
            <input type="text" class="form-control" v-model="formInfo.secondName" placeholder="名">
        </div>
        <div class="form-group">
            <input type="phone" class="form-control" v-model="formInfo.phone" placeholder="电话号码">
        </div>
        <div class="form-group">
            <input type="password" class="form-control" v-model="formInfo.password" placeholder="密码">
        </div>
        <div class="form-group">
            <input type="password" class="form-control" v-model="formInfo.confirmPassword" placeholder="请再输入密码一次">
        </div>
        <div class="form-group">
            <button class="btn btn-block btn-lg btn-success" @click="submit()">注册</button>
    </div>
    <div class="panel-footer">
      <router-link :to="{name:'login'}">去登录</router-link>
    </div>
  </log>
</template>

<script>
import log from './log'
export default {
  data (){
    return {
      errMsg:'',
      formInfo:{
        firstName:'',
        secondName:'',
        phone:'',
        password:'',
        confirmPassword:''
      }
    }
  },
  methods:{
    isEmpty (){
      const result=this.$isFormEmpty(this.formInfo)
      if(!result){
        this.errMsg='请填写全所有的字段'
      }
      return result
    },
    isPasswordSame (){
      const result= this.formInfo.password===this.formInfo.confirmPassword
      if(!result){
        this.errMsg='两次密码输入不相同'
      }
      return result
    },
    isRepeat (){
      let filter={where:{name:this.formInfo.name}}
      return this.$http.get('/api/People',
        {params:{filter}})
      .then((res)=>{
```

```
    return res
  })
  .catch((err)=>{
    console.error(err)
  })
},
postInfo (){
  let postInfo={
    firstName:this.formInfo.firstName,
    secondName:this.formInfo.secondName,
    name:this.formInfo.firstName+this.formInfo.secondName,
    phone:this.formInfo.phone,password:this.formInfo.password,
    headName:''
  }
  if(postInfo.secondName.length>1){
    postInfo.headName=postInfo.secondName
  }else{
    postInfo.headName=postInfo.name
  }
  this.$http.post('/api/People',postInfo)
    .then(({status,body})=>{
      if(status===200){
        this.$saveLoginInfo(body)
        this.$router.push({name:'main'})
      }
    })
    .catch((err)=>{
      console.error(err)
    })
},
submit (){
  this.errMsg=''
  if(this.isEmpty() && this.isPasswordSame()){
    this.isRepeat()
      .then(({status,body})=>{
        if(status===200){
          let personInfo=body[0]
          if(personInfo && personInfo.phone===this.formInfo.phone){
            this.errMsg='该手机号已注册！'
          }else{
            this.postInfo()
          }
```

```
                    }else{
                        this.errMsg='服务器出错鸟！'
                    }
                })
            }
        }
    },
    components:{log}
}
</script>

<style lang="scss" scoped>
</style>
```

文件名：client/src/components/head-navbar.vue

```
<template lang="html">
    <div class="mod-navbar">
        <div class="main">
            <div class="title">
                <h3>
                    <i class="glyphicon navbar-title" :class="navbarInfo.iconName"></i>
                    {{ navbarInfo.title }}
                </h3>
            </div>
            <div class="center-menu">
                <ul>
                    <li v-for="menuItem in navbarInfo.menuItems">

<router-link :class="{active:isActive(menuItem.routerName)}" :to="menuItem.routerName">{{ menuIt
em.name }}</router-link>
                    </li>
                </ul>
            </div>
        </div>
    </div>
</template>

<script>
```

```
export default {
  computed:{
    navbarInfo (){
      switch(this.$store.state.secondRouterName){
        case 'dashboard':
          return {
            iconName:'glyphicon-dashboard',
            title:'我的工作台',
            menuItems:[
              {name:'任务',routerName:'mission'},
              {name:'动态',routerName:'moments'}
            ]
          }
          break;
        case 'search':
          return {
            iconName:'glyphicon-search',
            title:'搜索',
            menuItems:[
            ]
          }
          break;
        case 'messages':
          return {
            iconName:'glyphicon-envelope',
            title:'消息中心',
            menuItems:[
              {name:'未读',routerName:'unread'},
              {name:'待处理',routerName:'pending'},
              {name:'已读',routerName:'readed'}
            ]
          }
          break;
        case 'projectReview':
          return {
            iconName:'glyphicon-folder-open',
            title:'项目一览',
            menuItems:[
              {name:'任务',routerName:'mission'},
              {name:'文档',routerName:'document'},
              {name:'话题',routerName:'topic'}
            ]
          }
          break;
        case 'group':
          const groupId=this.$route.params.groupId,
            groupName=this.$route.params.groupName
          return {
            iconName:'glyphicon-th-large',
            title:groupName,
            menuItems:[
            ]
          }
          break;
        case 'project':
          return {
            iconName:'glyphicon-folder-open',
            title:this.$route.params.projectName,
            menuItems:[
              {name:'文档',routerName:'document'}
            ]
          }
          break;
      }
    }
  },
  methods:{
    isActive (itemRouterName){
      let routerLevel=this.$store.state.routerLevel
      if(routerLevel[routerLevel.length-1]===itemRouterName){
        return true
      }
      return false
    }
  }
}
</script>

<style lang="scss" scoped>
.mod-navbar{
  user-select: none;
  padding: 0 7px;
  position: relative;
  background-color: #fff;
```

```
border-bottom: solid 1px #eeece8;
min-height: 55px;
// z-index: 3;

.main{
   line-height: 55px;
   min-height: 55px;

   .title{
      float: left;
      white-space: nowrap;
      line-height: 100%;
      max-width: 22%;

      h3{
         min-width: 100px;
         margin: 0;
         vertical-align: middle;
         line-height: 54px;
         font-size: 21px;
         display: inline-block;
         color: #333;
         margin-right: 7px;
         font-weight: 400;
         max-width: 100%;
         overflow: hidden;
         white-space: nowrap;
         text-overflow: ellipsis;

         .navbar-title{
            padding: 0 7px;
            width: 44px;
            display: inline-block;
            text-align: center;
            color: #acaba7;
            line-height: 30px;
            font-size: 25px;
            text-shadow: 0 2px 0 #fff, 0 -2px 0 #f2f2ea;
         }
      }
   }
```

```
.center-menu{
    line-height: 55px;
    min-height: 55px;
    margin-left: 28%;
    margin-right: 28%;
    width: 43%;
    padding-left: 14px;
    white-space: nowrap;
    text-align: center;

    ul{
        margin: 0;

        li{
            position: relative;
            display: inline-block;
            margin: 0 5px 0 6px;
            padding: 0;

            a{
                display: block;
                font-size: 16px;
                line-height: 26px;
                padding: 14px 8px 14px 8px;
                border-bottom: solid 2px transparent;
                color: #777;
                text-decoration: none;

                &.active{
                    $activeColor:#da4f4a;

                    color: $activeColor;
                    border-bottom-color: #da4f4a;

                    &:hover{
                        color: $activeColor;
                    }
                }
                &:hover{
                    color: #484744;
                }
            }
```

```
                }
            }
        }
    }
}

</style>
```

文件名：client/src/components/log.vue

```
<template lang="html">
    <div class="main-container">
        <alert-danger :message="errMsg" type="danger"></alert-danger>
        <div class="main">
            <div class="panel-header">
                团队协同办公系统
            </div>
            <div class="panel-body">
                <div class="form">
                    <slot></slot>
                </div>
            </div>
        </div>
        <div class="footer">
            <span>作者：王庭旭</span>
            <span>本项目<a  href="https://github.com/tingxu-wang/worktile_vue"  target="_blank">github
地址</a></span>
        </div>
    </div>
</template>

<script>
import alertDanger from './pub/alert'
export default {
    components:{alertDanger},
    props:['errMsg']
}
</script>
```

```
<style lang="scss" scoped>
$hoverColor:#da4f4a;

.main-container{
    .footer{
        text-align: center;

        span{
            margin:0 3px 5px;
        }
    }
    .main{
        width: 400px;
        margin: 0 auto;
        padding: 8% 0 2%;
        text-align: center;


        .panel-header{
            border: 0;
            text-align: center;
            padding-bottom: 10px;
            position: relative;
            margin:0 auto;
            color: #6f6e6b;
            outline: 0 none;
            vertical-align: middle;
            overflow: hidden;
            display: inline-block;
            font-size: 30px;
            cursor:default;
            transition: color 0.2s linear;

            &:hover{
                color: $hoverColor;
            }
        }

        .panel-body{
            margin-top: 5px;
            padding-bottom: 20px;
            padding-top: 20px;
            .panel-footer{
                border-top: 1px #e8edf3 solid;
                padding: 20px;
                text-align: center;
                line-height: 14px;

                & > a{
                    color: #777;

                    &:hover{
                        color: $hoverColor;
                        text-decoration: none;
                    }
                }
            }
        }
    }
}

</style>
```

文件名：client/src/plugin/README.md

```
# 个人编写的 vue 插件，全局公用方法
```

文件名：client/src/plugin/index.js

```
let MyPlugin={}
MyPlugin.install=function(Vue){
    const _this=Vue.prototype

    _this.$isFormEmpty=function(formInfo){
        const keys=Object.keys(formInfo)
        const result=keys.every((key)=>{
            return formInfo[key]!=="
```

```
  })
  return result
}

_this.$getFromSession=function(key,infoKey=false){
  if(infoKey){
    const sessionObj=JSON.parse(sessionStorage.getItem(key))
    return sessionObj[infoKey]
  }else{
    return sessionStorage.getItem(key)
  }
}

_this.$saveToSession=function(key,item){
  sessionStorage.setItem(key,JSON.stringify(item))
}

_this.$saveLoginInfo=function(infoObj){
  sessionStorage.setItem('isLogin',true)
  _this.$saveToSession('userInfo',infoObj)
}

_this.$post=function(url,obj){
  return this.$http.post(url,obj)
    .then(({status,body,code})=>{
      if(status===200){
        return body
      }else{
        throw new Error(`服务器出现错误！status:${status},code:${code},url:${url}`)
      }
    })
    .catch(err=>{
      throw new Error(err)
    })
}

_this.$paramsPost=function(url,obj,params){
  return this.$http.post(url,obj,{params})
    .then(({status,body,code})=>{
      if(status===200){
        return body
      }else{
        throw new Error(`服务器出现错误！status:${status},code:${code},url:${url}`)
      }
    })
    .catch(err=>{
      throw new Error(err)
    })
}

_this.$filterPost=function(url,obj,filter){
  return this.$http.post(url,obj,{params:{where:filter}})
    .then(({status,body,code})=>{
      if(status===200){
        return body
      }else{
        throw new Error(`服务器出现错误！status:${status},code:${code},url:${url}`)
      }
    })
    .catch(err=>{
      throw new Error(err)
    })
}

_this.$get=function(url){
  return this.$http.get(url)
    .then(({status,body,code})=>{
      if(status===200){
        return body
      }else{
        throw new Error(`服务器出现错误！status:${status},code:${code},url:${url}`)
      }
    })
    .catch(err=>{
      throw new Error(err)
    })
}

_this.$delete=function(url){
  return this.$http.delete(url)
  .then(({status,body,code})=>{
    if(status===200){
      return body
    }else{
        throw new Error(`服务器出现错误！status:${status},code:${code},url:${url}`)
      }
    })
    .catch(err=>{
      throw new Error(err)
    })
  }

  _this.$paramsGet=function(url,params){
    return this.$http.get(url,{params})
      .then(({status,body,code})=>{
        if(status===200){
          return body
        }else{
          throw new Error(`服务器出现错误！status:${status},code:${code},url:${url}`)
        }
      })
      .catch(err=>{
        throw new Error(err)
      })
  }

  _this.$filterGet=function(url,filter){
    return this.$http.get(url,{params:{filter:{where:filter}}})
      .then(({status,body,code})=>{
        if(status===200){
          return body
        }else{
          throw new Error(`服务器出现错误！status:${status},code:${code},url:${url}`)
        }
      })
      .catch(err=>{
        throw new Error(err)
      })
  }

  _this.$hideModal=function(){
    $('.modal').modal('hide')
  }
}

export {MyPlugin}
```

文件名：client/src/router/index.js

```
import Vue from 'vue'
import Router from 'vue-router'

import Login from '@/components/Login'
import Main from '@/components/Main'
import Regist from '@/components/Regist'

import mission from '@panelContent/mission/mission'
import messages from '@panelContent/messages'
import projectReview from '@panelContent/projectReview'
import project from '@panelContent/project/project'
import search from '@panelContent/search'
import moments from '@panelContent/moments'
import group from '@panelContent/group'

import document from '@panelContent/project/document'

Vue.use(Router)

export default new Router({
  routes:[
    {
      path:'/',
      redirect:{name:'main'}
    },
    {
      path:'/login',
      name:'login',
      component: Login
    },
    {
      path:'/regist',
      name:'regist',
      component: Regist
    },
    {
```

```
path:'/main',
name:'main',
component: Main,
redirect:{name:'search'},
children:[
  {
    path:'dashboard',
    name:'dashboard',
    redirect:{name:'mission'}
  },
  {
    path:'dashboard/mission',
    name:'mission',
    component:mission
  },
  {
    path:'dashboard/moments',
    name:'moments',
    component:moments
  },
  {
    path:'messages',
    name:'messages',
    component:messages
  },
  {
    path:'projectReview',
    name:'projectReview',
    component:projectReview
  },
  {
    path:'project/:projectId/:projectName',
    name:'project',
    component:project
  },
  {
    path:'project/:projectId/:projectName/document',
    name:'document',
    component:document
  },
  {
    path:'search',
                    name:'search',
                    component:search
                },
                {
                    path:'group/:groupId/:groupName',
                    name:'group',
                    component:group
                }
            ]
        }
    ]

})
```

文件名：client/src/store/store.js

```
import Vue from 'vue'
import Vuex from 'vuex'

Vue.use(Vuex)

const store=new Vuex.Store({
    state:{
        routerLevel:[],
        firstRouterName:",
        secondRouterName:",
        thirdRouterName:",
        isLogin:false
    },
    getters:{
    },
    mutations:{
        refreshRouterLevel (state,path){
            state.routerLevel=path.split('/').splice(1)
        },
        refreshRouterName (state){
            let routerLevel=state.routerLevel
            const leventLength=routerLevel.length
```

```
      if(leventLength>0) {
        state.firstRouterName=routerLevel[0]
      }
      if(leventLength>1){
        state.secondRouterName=routerLevel[1]
      }
      if(leventLength>2){
        state.thirdRouterName=routerLevel[2]
      }
    }
  },
  actions:{
    increment ({commit},n){
      commit('increment',n)
    },
    refreshRouterLevel ({commit},path){
      commit('refreshRouterLevel',path)
    },
    refreshRouterName ({commit}){
      commit('refreshRouterName')
    }
  }
})

export {store}
```

文件名：client/src/components/center-panel-content/group.vue

```
<template lang="html">
  <div class="main-container">
    <span class="main-title">项&nbsp;&nbsp;目</span>
    <button class="btn btn-primary btn-sm" @click="createProject()"><span class="glyphicon glyphicon-plus"></span></button>
    <project-modal :hook="projectHook">
      <template slot="title">
        新建项目
      </template>
      <template>
        <danger-alert type="danger" :message="errMsg"></danger-alert>
        <div class="form">
          <div class="form-group">
            <input type="text" class="form-control" v-model="projectName" placeholder="请输入项目名称">
          </div>
          <div class="form-group">
            <textarea class="form-control" v-model="projectDescription" placeholder="项目描述（可选）"></textarea>
          </div>
        </div>
      </template>
      <template slot="footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">取消</button>
        <button type="button" class="btn btn-primary" @click="postProjectInfo">创建</button>
      </template>
    </project-modal>
    <project-item :project-arr="projectArr"></project-item>
  </div>
</template>

<script>
import modal from '@pub/modal'
import alert from '@pub/alert'
import projectItem from './project/project-item'
export default {
  created (){
    this.getProjectInfo()
  },
  watch:{
    '$route':'getProjectInfo'
  },
  data (){
    return {
      projectHook:false,
      errMsg:'',
      projectName:'',
      projectDescription:'',
      projectArr:[]
    }
  },
  methods:{
    createProject (){
      this.projectHook=!this.projectHook
    },
    postProjectInfo (){
      this.errMsg=''
      let infoObj={
        name:this.projectName,
        description:this.projectDescription,
        group_id:this.$route.params.groupId
      }
      if(this.projectName===''){
        this.errMsg='请填写项目名称！'
      }else{
        this.$post('/api/Projects',infoObj)
          .then(body=>{
            this.projectArr.push(body)
            this.$hideModal()
            this.projectName=''
            this.projectDescription=''
          })
      }
    },
    getProjectInfo (){
      const groupId=this.$route.params.groupId
      this.$get(`/api/Groups/${groupId}/projects`)
        .then(body=>{
          this.projectArr=body
        })
    }
  },
  components:{projectItem,projectModal:modal,dangerAlert:alert}
}
</script>

<style lang="scss" scoped>
.main-container{
  background: #fff;
  min-height:900px;
  margin:7px 14px;
  padding:21px;
  .main-title{
    user-select: none;
    border-bottom: solid 1px #f8f6f2;
```

```
        padding: 0 15px 14px 2px;
        font-weight: 400;
        color: #777;
        font-size: 17px;
        line-height: 100%;
        display: inline-block;
    }
}
</style>
```

文件名：client/src/components/center-panel-content/messages.vue

```
<template lang="html">
    <div>
        messages
    </div>
</template>

<script>
export default {
}
</script>

<style lang="scss" scoped>
</style>
```

文件名：client/src/components/center-panel-content/moments.vue

```
<template lang="html">
    <div>
        moments
    </div>
</template>

<script>
export default {
}
</script>

<style lang="css">
</style>
```

文件名：client/src/components/center-panel-content/search.vue

```
<template lang="html">
  <div class="content-main-container">
    <div class="form-horizontal">
      <div class="input-group">
        <input type="text" class="form-control" placeholder="输入查询关键词" v-model="keyWord" @keyup="search()">
        <span class="input-group-addon"><span class="glyphicon glyphicon-search" @click="search()"></span></span>
      </div>
    </div>
    <div class="searched-container">
      <div class="item" v-for="(doc,$index) in filteredDocArr" @click="showDoc($index)">
        <div class="icon-container">
          <i class="glyphicon glyphicon-file icon"></i>
        </div>
        <div class="text-container">
          <div>
            {{ doc.title }}
          </div>
          <div class="description">
            <br/>
          </div>
        </div>
      </div>
      <documentModal :docHook="docHook" :docModalInfo="docModalInfo"></documentModal>

      <projectItem :projectArr="filteredProjectArr"></projectItem>

      <div class="item" v-for="task in filteredTaskArr" @click="showSlide(task.id,task.name)">
        <div class="icon-container">
          <i class="glyphicon glyphicon-edit icon"></i>
        </div>
        <div class="text-container">
          <div>
            {{ task.name }}
          </div>
          <div class="description">
            关注的人：<span v-for="item in task.focus_person_info">{{ item.headName }} </span>
          </div>
        </div>
      </div>

<task-slide :task-id="taskslideInfo.taskId" :task-name="taskslideInfo.name" :is-show="taskslideInfo.isShow" v-on:slideClose="closeSlide"></task-slide>

      <div class="item" v-for="group in filteredGroupArr" @click="toGroup(group.id,group.name)">
        <div class="icon-container">
          <i class="glyphicon glyphicon-user icon"></i>
        </div>
        <div class="text-container">
          <div>
            {{ group.name }}
          </div>
          <div class="description">
            {{group.user_ids.length}}人
          </div>
        </div>
      </div>

    </div>
  </div>
</template>

<script>
import documentModal from './project/document-modal'
import projectItem from './project/project-item'
import taskSlide from './mission/task-slide'

export default {
  created (){//获取所有的检索范围信息
    this.$get('/api/Documents').then((res)=>{
      this.docArr=res
    })
    this.$get('/api/Projects').then((res)=>{
        this.projectArr=res
    })
    this.$get('/api/Groups').then((res)=>{
        this.groupArr=res
    })
    this.$get('/api/Tasks').then((res)=>{
        this.taskArr=res
    })
},
data (){
    return {
        keyWord:'',
        docHook:false,

        docArr:[],
        filteredDocArr:[],
        docModalInfo:{},

        projectArr:[],
        filteredProjectArr:[],

        groupArr:[],
        filteredGroupArr:[],

        taskArr:[],
        filteredTaskArr:[],
        taskslideInfo:{
            isShow:false,
            taskId:-1,
            taskName:''
        }
    }
},
methods:{
    getFiltered (arr,mathProp){
        return arr.filter((item)=>{
            if(this.keyWord){
                let reg=new RegExp(this.keyWord)
                return item[mathProp].match(reg)
            }else{
                return []
                }
            })
        },
        search (){
            this.filteredDocArr=this.getFiltered(this.docArr,'title')
            this.filteredProjectArr=this.getFiltered(this.projectArr,'name')
            this.filteredGroupArr=this.getFiltered(this.groupArr,'name')
            this.filteredTaskArr=this.getFiltered(this.taskArr,'name')
        },
        showDoc (index){
            const docInfo=this.docArr[index]
            this.docModalInfo=docInfo
            this.docHook=!this.docHook
        },
        toGroup (groupId,groupName){
            this.$router.push({name:'group',params:{groupId,groupName}})
        },
        closeSlide (){
            this.taskslideInfo.isShow=false
        },
        showSlide (taskId,taskName){
            console.log('run')
            let target=this.taskslideInfo
            target.isShow=true
            target.taskId=taskId
            target.taskName=taskName
        }
    },
    components:{documentModal,projectItem,taskSlide}
}
</script>

<style lang="scss" scoped>
.searched-container{
    margin-top: 15px;
}
</style>
```

文件名：client/src/components/center-panel-content/projectReview.vue

```
<template lang="html">
  <div>

  </div>
</template>

<script>
export default {
}
</script>

<style lang="scss" scoped>
</style>
```

文件名：client/src/components/left-panel/create-group.vue

```
<template lang="html">
  <modal :hook="hook">
    <template slot="title">新建团队</template>
    <template>
      <ul class="modal-ul">
        <li :class="{step_cursor:stepIndex==index}" v-for="(modelRowInfo,index) in
modelRowInfoArr">
          {{ modelRowInfo }}
        </li>
      </ul>
      <alert-danger :message="errMsg" type="danger"></alert-danger>
      <div class="form modal-form-container">
        <div v-if="stepIndex===0">
          <div class="form-group">
            <label>团队名称</label>
            <input type="text" class="form-control" v-model="groupName" placeholder="输入团队
名称">
          </div>
          <div class="form-group">
            <label>队长</label>
            <input type="text" class="form-control" v-model="leaderName" placeholder="输入队长
姓名" disabled>
```

```
          </div>
          <div class="form-group">
            <label>队长手机号</label>
            <input type="text" class="form-control" v-model="leaderPhone" placeholder="输入队长手机号" disabled>
          </div>
        </div>
        <div v-else-if="stepIndex===1">
          <label>成员手机号</label>
          <div class="form-group" v-for="(phone,index) in phoneGroup" :class="{'has-error':phone.isInValid}">
            <input type="phone" class="form-control" v-model="phoneGroup[index].phoneNum" >
          </div>
          <div class="form-group row">
            <button class="btn btn-link" @click="addPhoneGroup()">+ 添加</button>
            <button class="btn btn-link" @click="reducePhoneGroup()">- 减少</button>
          </div>
        </div>
        <div v-else-if="stepIndex===2">
          <div class="team_create_info_show">
            <div class="info">
              请确认以下信息
            </div>
            <div class="mt_30 group-name">
              {{ groupName }}
            </div>
            <div class="mt_20">
              <span class="g-user-head-default group-member" v-for="groupMember in groupMembers">{{ groupMember }}</span>
            </div>
          </div>
        </div>
      </div>
    </template>
    <template slot="footer">
      <button type="button" class="btn btn-default" data-dismiss="modal" @click="cancelCreate()">取消</button>
      <button type="button" class="btn btn-default" v-if="stepIndex>0" @click="prvStep()">上一步</button>
      <button type="button" class="btn btn-primary" @click="nextStep()" v-if="stepIndex!==2">下一步</button>
      <button type="button" class="btn btn-primary" @click="postGroupInfo()"
```

```
v-if="stepIndex===2">创建</button>
    </template>
  </modal>
</template>

<script>
import modal from '@pub/modal'
import alert from '@pub/alert'
export default {
  data(){
    return {
      modelRowInfoArr:['1  基本信息','2  邀请成员','3  完成'],
      stepIndex:0,
      groupName:'',
      errMsg:'',
      phoneGroup:[{phoneNum:'',isInValid:false,id:'',headName:''}]
    }
  },
  computed:{
    leaderId (){
      return this.$getFromSession('userInfo','id')
    },
    leaderHead (){
      return this.$getFromSession('userInfo','headName')
    },
    leaderName (){
      return this.$getFromSession('userInfo','name')
    },
    leaderPhone (){
      return this.$getFromSession('userInfo','phone')
    },
    groupMembers (){
      return this.phoneGroup.map(item=>{
        return item.headName
      })
      .filter(headName=>{
        return headName!==this.leaderHead
      })
      .concat(this.leaderHead)
    },
    user_ids (){
      return this.phoneGroup.map(item=>{
```

```
      return item.id
    })
    .filter(id=>{
      return id!==this.leaderId
    })
    .concat(this.leaderId)
  }
},
methods:{
  cancelCreate (){
    this.stepIndex=0
  },
  prvStep (){
    this.stepIndex-=1
  },
  addStep (){
    this.stepIndex+=1
  },
  nextStep (){
    this.errMsg="
    switch(this.stepIndex){
      case 0:
        if(!this.groupName){
          this.errMsg='请填写团队名称'
        }else{
          this.addStep()
        }
        break;
      case 1:
        let result=true
        let promiseArr=[]
        for(let i=0;i<this.phoneGroup.length;i++){
          const phone=this.phoneGroup[i].phoneNum
          const filter={phone}
          if(phone==="){
            this.errMsg='请填写全所有手机号'
            return
          }
          let p=this.$filterGet('/api/People',filter)//检测用户是否存在
          promiseArr.push(p)
        }
        Promise.all(promiseArr)
          .then(bodys=>{
            let result=true
            bodys.forEach((body,index)=>{
              if(body.length===0){
                this.errMsg='用户不存在'
                this.phoneGroup[index].isInValid=true
                result=false
              }else{
                this.phoneGroup[index].id=body[0].id
                this.phoneGroup[index].headName=body[0].headName
                this.phoneGroup[index].isInValid=false
              }
            })
            if(result){
              this.addStep()
            }
          })
        break;
    }
  },
  postGroupInfo (){
    let groupInfo={
      leader_id:this.$getFromSession('userInfo','id'),
      leader_phone:this.$getFromSession('userInfo','phone'),
      create_time:String(Date.now()),
      user_ids:this.user_ids,
      name:this.groupName
    }
    this.$post('/api/Groups',groupInfo)
      .then(({id})=>{
        let promiseArr=[]
        this.user_ids.forEach(userId=>{
          let postInfo={
            person_id:userId,
            group_id:id
          }
          promiseArr.push(this.$post('/api/Person_has_groups',postInfo))
        })
        Promise.all(promiseArr)
          .then(()=>{
            this.$emit('createdGroup',groupInfo)
            this.$hideModal()
```

```
              this.stepIndex=0
              this.groupName="
              this.phoneGroup=[{phoneNum:",isInValid:false,id:",headName:"}]
            })
        })
    },
    addPhoneGroup (){
      this.phoneGroup.push({phoneNum:",isInValid:false})
    },
    reducePhoneGroup (){
      if(this.phoneGroup.length>1){
        this.phoneGroup.pop()
      }
    }
  },
  components:{modal,alertDanger:alert},
  props:['hook']
}
</script>

<style lang="scss" scoped>

ul.modal-ul{
  display: flex;
  padding-left: 0px;
  li{
    line-height: 35px;
    font-size: 14px;
    text-align: center;
    color: #b9b9b7;
    border-top: solid 5px #f8f6f2;
    flex:1;
    &.step_cursor{
      color: #2e2d2b;
      border-top: solid 5px #54d09f;
    }
  }
}
.modal-form-container{
  padding:0 15px;
  .team_create_info_show{
    padding: 0 20px 20px 20px;
    margin: 0 auto;
    min-height: 200px;
    text-align: center;
    border: solid 1px #e8edf3;
    .info{
      margin: -10px auto 0 auto;
      width: 130px;
      color: #cb1a44;
      background: #fff;
    }
    .group-name{
      color: #54d09f;
      font-size: 21px;
    }
    .group-member{
      display: inline-block;
      cursor: default;
      margin-right: 5px;
    }
  }
}
</style>
```

文件名：client/src/components/left-panel/group-review.vue

```
<template lang="html">
    <div class="popbox group-pop" v-show="isShow">
        <div class="popbox-body">
            <ul class="popbox-menu">
                <li class="pop-title">
                    <span>我的团队</span>
                </li>
                <li class="divider"></li>
                <li class="popbox-li" v-for="group in groupList" @click="toGroup(group.id,group.name)">
                    <div>{{ group.name }}</div>
                    <small class="group-info-container">
                        <span><i class="glyphicon glyphicon-user"></i>{{ group.user_ids.length }}人</span>
                    </small>
        </li>
        <li class="divider"></li>
        <li class="create-group-btn-container">
          <button class="btn btn-block btn-success" @click="createGroup()">创建新团队</button>
        </li>
      </ul>
    </div>
    <create-group :hook="modalHook" v-on:createdGroup="insertGroup"></create-group>
  </div>
</template>

<script>
import createGroup from './create-group'
export default {
  created (){
    this.getGroupList()
  },
  data (){
    return {
      modalHook:false,
      groupList:[]
    }
  },
  methods:{
    createGroup (){
      this.modalHook=!this.modalHook
    },
    getGroupList (){
      const id=this.$getFromSession('userInfo','id')
      this.$get(`api/People/${id}/person_has_groups`)
        .then(body=>{
          let promiseArr=[]
          body.forEach(({group_id})=>{
            promiseArr.push(this.$filterGet(`api/Groups/findOne`,{id:group_id}))
          })
          return promiseArr
        })
        .then(promiseArr=>{
          Promise.all(promiseArr)
            .then(groupList=>{
              groupList.forEach(group=>{
                this.groupList.push(group)
```

```
                    })
                })
            })
        },
        insertGroup (group){
            this.groupList.push(group)
        },
        toGroup (groupId,groupName){
            this.$router.push({name:'group',params:{groupId,groupName}})
            this.$emit('groupClose')
        }
    },
    components:{createGroup},
    props:['isShow']
}
</script>

<style lang="scss" scoped>
$liPadding:8px 30px;
.group-pop{
    top:0px;

    .create-group-btn-container{
        padding:$liPadding;
    }
    .pop-title{
        padding:$liPadding;
        font-size: 16px;
    }
    .popbox-li{
        // padding:0 $paddingSize;
        .group-info-container{
            i{
                margin-right: 5px;
                color:#777;
            }
        }
    }
}
</style>
```

文件名：client/src/components/left-panel/left-panel.vue

```
<template lang="html">
  <div class="left-panel">
    <div class="left-menu">
      <div class="left-menu-tab">
        <div class="left-menu-top">
          <div class="left-panel-top-wrap">
            <div class="left-panel-top" @click="toggleGroupActive">
              <i class="glyphicon glyphicon-th-large"></i>
            </div>
            <group-review :is-show="groupIsShow"
v-on:groupClose="closeGroup"></group-review>
          </div>
          <div class="left-panel-tab-wrap">
            <ul class="left-panel-tab">
              <tab-item v-for="(tabProp,index) in
tabProps" :key="index" :icon-name="tabProp.iconName" :tab-name="tabProp.tabName" :is-active="tab
Prop.isActive" :router-name="tabProp.routerName" :index="index"></tab-item>
            </ul>
          </div>
        </div>
        <div class="left-panel-footer">
          <div class="quick-add" @click="quickAdd">
            <i class="glyphicon glyphicon-plus"></i>
          </div>
          <div class="user-setting g-user-head" @click="openUserSetting">
            {{ headName }}
          </div>
          <user-setting :user-box-is-show="userBoxIsShow"></user-setting>
        </div>

<pop-box :pop-box-is-show="popBoxIsShow" :user-setting-is-show="userBoxIsShow"></pop-box>
      </div>
    </div>
  </div>
</template>

<script>
import tabItem from './tab-item'
import popBox from './pop-box'
import userSetting from './user-setting'
import groupReview from './group-review'

export default {
  data (){
    return {
      tabProps:[
        // {iconName:'glyphicon-dashboard',tabName:'我的',routerName:'dashboard'},
        {iconName:'glyphicon-search',tabName:'搜索',routerName:'search'},
        // {iconName:'glyphicon-envelope',tabName:'消息',routerName:'messages'},
        {iconName:'glyphicon-folder-open',tabName:'项目总览',routerName:'projectReview'},
      ],
      popBoxIsShow:false,
      userBoxIsShow:false,
      groupIsShow:false
    }
  },
  computed:{
    headName (){
      return this.$getFromSession('userInfo','headName')
    }
  },
  methods:{
    togglePopBoxActive (){
      this.popBoxIsShow=!this.popBoxIsShow
    },
    toggleUserBoxActive (){
      this.userBoxIsShow=!this.userBoxIsShow
    },
    toggleGroupActive(){
      this.groupIsShow=!this.groupIsShow
    },
    quickAdd (){
      if(this.userBoxIsShow){
        this.toggleUserBoxActive()
      }
      this.togglePopBoxActive()
    },
    openUserSetting (){
      if(this.popBoxIsShow){//弹窗打开时关闭
        this.togglePopBoxActive()
```

```
            }
            this.toggleUserBoxActive()
        },
        closeGroup (){
            this.groupIsShow=false
        }
    },
    components:{
        tabItem,
        popBox,
        userSetting,
        groupReview
    },
    props:['projectId']
}
</script>
<style lang="scss">
.left-panel{
    width:60px;
    height:100%;
    position:absolute;
    top:0;
    left:0;
}
.left-menu{
    user-select: none;
    color: #92908d;
    height: 100%;
    position: relative;

    .left-menu-tab{
        background-color: #2e2d2b;
        border-right: solid 1px #1a2224;
        position: absolute;
        left: 0;
        top: 0;
        bottom: 0;
        width: 60px;
        z-index: 4;
    }
    .left-menu-top{
        text-align: center;
        .left-panel-top-wrap{
            text-align: center;

            .left-panel-top{
                font-size: 25px;
                color: #fff;
                padding: 15px 0 8px;
                cursor: pointer;
            }
        }
    }
    .left-panel-tab-wrap{
        position: absolute;
        top: 56px;
        bottom: 113px;
        left: 0;
        right: 0;
    }
    .left-panel-tab{
        padding: 10px 0 0 0;
    }
    .left-panel-footer{
        position: absolute;
        bottom: 0;
        right: 0;
        left: 0;
        text-align: center;
        .user-setting{
            margin:0 auto 15px;
        }
    }
    .quick-add{
        width:60px;
        height:30px;
        display: block;
        padding: 6px;
        text-align: center;
        color: #92908d;
        background-color: #2e2d2b;
        border-left: solid 3px transparent;
        cursor:pointer;
```

```
        margin-bottom: 15px;

        &:hover{
            color:#fff;
        }
    }
    .user-setting-container{

    }
    .user-setting{
        width:40px;
        height:40px;
    }
}

$boxBgColor:#fff;
.popbox{
    position:absolute;
    display: block;
    left: 60px;
    min-width: 252px;
    max-width:400px;
    text-align: left;
    background-color: $boxBgColor;
    border: 1px solid #eeece8;
    box-shadow: 0 2px 8px rgba(0,0,0,.2);
    color:#484744;
}
.popbox-menu{
    padding: 5px;
    margin-bottom: 0px;
}
li{
    list-style: none;
}
.popbox-li{
    padding:8px 30px;
    cursor: pointer;
    border-radius: 5px;

    &:hover{
        background: #f1ede5;
    }
}
.divider{
    border: 0;
    border-top: solid 1px #e8edf3;
    margin: 0;
    padding: 0;
    cursor: default;
    height: 1px;

    &:hover{
        background: $boxBgColor;
    }
}
</style>
```

文件名：client/src/components/left-panel/pop-box.vue

```
<template lang="html">
    <div class="popbox mission-box" v-show="popBoxIsShow">
        <div class="popbox-body">
            <ul class="popbox-menu">
                <li v-for="popboxItem in popboxItems" :class="{divider:popboxItem.isLine}"
class="popbox-li">
                    <i class="glyphicon" :class="popboxItem.iconName"></i>
                    <span>{{ popboxItem.name }}</span>
                </li>
            </ul>
        </div>
    </div>
</template>

<script>
export default {
    data (){
        return {
            popboxItems:[
                {iconName:'glyphicon-user',name:'团队'},
                {iconName:'glyphicon-inbox',name:'项目'},
                {iconName:'',name:'',isLine:true},
                {iconName:'glyphicon-tasks',name:'任务'},
                {iconName:'glyphicon-list-alt',name:'日程'},
                {iconName:'glyphicon-file',name:'文件'},
            ]
        }
    },
    methods:{
        empty (){
            consoel.log('empty')
        }
    },
    props:['popBoxIsShow']
}
</script>

<style lang="scss" scoped>
.mission-box{
    bottom: 60px;
}
</style>
```

文件名：client/src/components/left-panel/tab-item.vue

```
<template lang="html">
<li>
    <span class="panel-tab text-center" :class="{panelTabActive:isActive}"
@click="changeRouterPath()">
        <i :class="iconName" class="glyphicon panel-icon"></i>
        <div>{{ tabName }}</div>
    </span>
</li>
</template>

<script>
export default {
    computed:{
        isActive (){
            if(this.$store.state.secondRouterName===this.routerName){
                return true
            }
            return false
        }
    },
    methods:{
        changeRouterPath (){
            this.$router.push({name:this.routerName})
        }
    },
    props:['iconName','tabName','index','routerName']
}
</script>

<style lang="scss" scoped>
$tabColor:#fff;

.panel-tab{
    display: block;
    padding: 7px 12px 7px 13px;
    margin: 14px 0;
    text-align: center;
    line-height: 138%;
    // white-space: nowrap;
    font-size: 12px;
    cursor: pointer;
    transition: color 0.2s linear;

    &:hover{
        color:$tabColor;
    }
}
.panelTabActive{
    color:$tabColor;
    border-left: solid 3px #da4f4a;
}
.panel-icon{
    font-size: 22px;
}
</style>
```

文件名：client/src/components/left-panel/user-setting.vue

```
<template lang="html">
  <div class="popbox user-setting-box" v-show="userBoxIsShow">
    <div class="popbox-body">
      <ul class="popbox-menu">
        <li
class="popbox-li"        :class="{divider:userSettingItem.isLine,'text-danger':userSettingItem.isRed}"
@click="userSettingItem.method()" v-for="userSettingItem in userSettingItems">
          <span>{{ userSettingItem.name }}</span>
        </li>
      </ul>
    </div>
  </div>
</template>

<script>
export default {
  data (){
    return {
      userSettingItems:[
        {name:'个人偏好设置',},
        {name:'账号资料设置'},
        {name:'提交反馈'},
        {isLine:true,method:this.empty},
        {name:'退出登录',isRed:true,method:this.logout}
      ]
    }
  },
  methods:{
    logout (){
      sessionStorage.removeItem('userInfo')
      sessionStorage.removeItem('isLogin')
      this.$router.replace({name:'login'})
    },
    empty (){
      console.log('empty')
    }
  },
  props:['userBoxIsShow']
}
</script>

<style lang="scss" scoped>
.user-setting-box{
   bottom:5px;
}
</style>
```

文件名：client/src/components/pub/alert.vue

```
<template lang="html">
<div class="alert-container">
   <div class="alert" :class="'alert-'+type" v-show="message">
      {{ message }}
   </div>
</div>
</template>

<script>
export default {
   props:['message','type']
}
</script>

<style lang="scss" scoped>
.alert-container{
   width:100%;
   padding:10px;
   .alert{
      float: left;
      width:100%;
   }
}
</style>
```

文件名：client/src/components/pub/modal.vue

},

```
<template lang="html">
  <div class="modal fade" role="dialog" ref="modal">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
          <h4 class="modal-title">
            <slot name="title"></slot>
          </h4>
        </div>
        <div class="modal-body">
          <slot></slot>
        </div>
        <div class="modal-footer">
          <slot name="footer"></slot>
        </div>
      </div>
    </div>
  </div>
</template>

<script>
export default {
  watch:{
    hook (curVal){
      let modal=this.$refs.modal,
        $modal=$(modal)

      $modal.modal('show')
      $modal.after($('.modal-backdrop'))
      if(this.emitName){
        $modal.on('hidden.bs.modal',()=>{
          this.$emit(this.emitName)
          $modal.off()
        })
      }
    }
  },
  props:['hook','emitName']//hook:弹窗是否显示钩子  emitName:弹窗关闭时发送的事件名称
}
```

```
</script>

<style lang="scss" scoped>

</style>
```

文件名：client/src/components/pub/slide.vue

```
<template lang="html">
  <div class="slide-content" v-show="isShow">
    <div class="entity-well-header clearfix">
      <slot name="header"></slot>
      <div class="close-container">
        <i class="glyphicon glyphicon-remove pull-right" @click="close()"></i>
      </div>
    </div>
    <div class="slider-body">
      <slot name="body"></slot>
    </div>
  </div>
</template>

<script>
export default {
  methods:{
    close (){
      this.$emit('slideClose')
    }
  },
  props:['isShow']
}
</script>

<style lang="scss" scoped>
$hover-color:#da4f4a;
$default-color:#acaba7;
$border:solid 1px #eeece8;
.slide-content{
  border-left: solid 1px #e8edf3;
  box-shadow: 0 0 15px rgba(0,0,0,.1);
  background: #fff;
  position: fixed;
  right: 0;
  bottom: 0;
  top: 0;
  width: 700px;
  z-index: 100;
  .entity-well-header{
    padding: 15px 20px;
    border-bottom: $border;
    position: relative;
    color: $default-color;
    .close-container{
      float:right;
      width:40px;
      padding-top: 2px;
      border-left: $border;
      i{
        cursor: pointer;
        &:hover{
          color:$hover-color;
        }
      }
    }
  }
  .slider-body{
    padding:10px 10px 0px;
  }
}
</style>
```

文件名：client/src/components/center-panel-content/mission/create-mission-group.vue

```
<template lang="html">
  <div class="create-mission-container" :class="{'gray-bg':!showCreate}" v-if="!isDashboard">
    <div class="create-mission" @click="createMission" v-if="!showCreate">
      <i class="glyphicon glyphicon-plus-sign"></i>
      新建任务列表
    </div>
    <div v-else class="form">
      <div class="form-group" v-if="errMsg">
        <danger-alert type="danger" :message="errMsg"></danger-alert>
      </div>
      <div class="form-group">
        <input type="text" class="form-control" v-model="taskGroupName" placeholder="列表名">
      </div>
      <div class="form-group">
        <button class="btn btn-success" @click.stop="postCreate">保存</button>
        <button class="btn btn-default" @click.stop="cancelCreate">取消</button>
      </div>
    </div>
  </div>
</template>

<script>
import alert from '@pub/alert'
export default {
  data (){
    return {
      showCreate:false,
      taskGroupName:'',
      errMsg:''
    }
  },
  computed:{
    isDashboard (){
      return this.$store.state.secondRouterName==='dashboard'
    }
  },
  methods:{
    createMission (){
      this.showCreate=true
    },
    cancelCreate (){
      this.showCreate=false
    },
    postCreate (){
      this.errMsg=''
      if(this.taskGroupName){
        let infoObj={
          project_id:this.projectId,
          project_name:this.projectName,
          name:this.taskGroupName,
          index:this.taskGroupLength
        }
        this.$paramsGet('/api/Task_groups/count',{where:{name:infoObj.name}})
          .then(({count})=>{
            if(count===0){
              this.$post('/api/Task_groups',infoObj)
                .then(body=>{
                  this.$emit('taskGroupCreated',body)
                  this.showCreate=false
                  this.taskGroupName=''
                })
            }else{
              this.errMsg='该列表已存在！'
            }
          })
      }
    }
  },
  components:{dangerAlert:alert},
  props:['projectId','taskGroupLength','projectName']
}
</script>

<style lang="scss" scoped>
.gray-bg{
  background-color: #eeece8;
}
.create-mission-container{
  position: relative;
  min-height: 44px;
  -webkit-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
  user-select: none;
  vertical-align: top;
  display: inline-block;
  zoom: 1;
  min-width: 180px;
  margin: 7px 0 0 10px;
```

```
    padding-bottom: 6px;
    border-radius: 5px;
    margin: 7px 14px 0 10px;
    padding: 0;
    min-width: 279px;
    border-color: transparent;
    .create-mission{
        display: block;
        color: #b9b9b7;
        border-radius: 3px;
        line-height: 42px;
        width: 100%;
        font-size: 14px;
        padding-left: 17px;
        cursor: pointer;
        &:hover{
            color: #acaba7;
        }
    }
}
</style>
```

文件名：client/src/components/center-panel-content/mission/mission-group-item.vue

```
<template lang="html">
    <div class="mission-group-item">
        <div class="mission-group-header clearfix">
            <div class="mission-group-title-container pull-left">
                <div class="mission-group-title">
                    {{ taskGroup.name }}
                </div>
            </div>
            <div class="item-num-container pull-right">
                <div class="item-num">{{ tasks.length }}</div>
            </div>
        </div>
        <div class="task-main-container">
            <div class="task-container">
                <mission-task :tasks="tasks"></mission-task>
```

```
      </div>
    </div>
    <div class="task-footer-container">
      <div class="task-footer" @click="createTask" v-if="!iscreateTask">
        <i class="glyphicon glyphicon-plus-sign"></i>
        新建任务
      </div>
      <div class="form" v-else>
        <div class="form-group">
          <textarea class="form-control" rows="3" v-model="taskName" placeholder="请输入任务内容"></textarea>
        </div>
        <div class="form-group">
          <button class="btn btn-success" @click.stop="postTask">保存</button>
          <button class="btn btn-default" @click.stop="cancelCreateTask">取消</button>
        </div>
      </div>
    </div>
  </div>
</template>

<script>
import missionTask from './mission-task'
export default {
  created (){
    this.$filterGet('/api/Tasks',{task_group_id:this.taskGroup.id})
      .then(body=>{
        this.tasks=body
      })
  },
  data (){
    return {
      iscreateTask:false,
      taskName:'',
      tasks:[]
    }
  },
  methods:{
    createTask (){
      this.iscreateTask=true
    },
    cancelCreateTask (){
      this.iscreateTask=false
    },
    postTask (){
      if(this.taskName!==''){
        let infoObj={
          task_group_id:this.taskGroup.id,
          task_group_name:this.taskGroup.name,
          project_id:this.taskGroup.project_id,
          project_name:this.taskGroup.project_name,

focus_person_info:[{id:this.$getFromSession('userInfo','id'),headName:this.$getFromSession('userInfo',
'headName')}],
          name:this.taskName,
          create_time:Date.now(),
          create_person_id:this.$getFromSession('userInfo','id'),
          is_solved:false,
          solve_person_info:[],
          description:'',
          dead_time:'',
          tags:[]
        }
        this.$post('/api/Tasks',infoObj)
          .then(body=>{
            this.iscreateTask=false
            this.taskName=''
            this.tasks.push(body)
          })
      }
    }
  },
  components:{missionTask},
  props:['taskGroup']
}
</script>

<style lang="scss" scoped>
.mission-group-item{
  position: relative;
  min-height: 60px;
  width: 279px;
  -webkit-user-select: none;
  -moz-user-select: none;
```

```
-ms-user-select: none;
user-select: none;
vertical-align: top;
display: inline-block;
zoom: 1;
min-width: 180px;
margin: 7px 0 0 10px;
padding-bottom: 6px;
border-radius: 5px;
background-color: #eeece8;

.mission-group-header{
   cursor: default;
   z-index: 2;
   position: relative;

   .mission-group-title-container{
      margin-right: 50px;

      .mission-group-title{
         font-size: 16px;
         margin: 0;
         color: #6f6e6b;
         line-height: 30px;
         padding: 7px 9px;
      }
   }

   .item-num-container{
      .item-num{
         color: #bcb7af;
         font-size: 12px;
         width: 47px;
         text-align: center;
         font-weight: 400;
         line-height: 30px;
         padding: 7px 9px;
      }
   }
}

.task-main-container{
```

```
            overflow-x: hidden;
            overflow-y: auto;
            max-height: 822px;
            margin: 0 4px;
            padding: 0 4px;

            .task-container{
                padding: 0;
                min-height: 20px;



            }
        }

        .task-footer-container{
            position: relative;
            z-index: 2;
            padding: 5px 8px 0;
            line-height: 28px;

            .task-footer{
                display: block;
                color: #bcb7af;
                text-align: left;
                padding: 3px 0px;
                cursor: pointer;

                &:hover{
                    color: #acaba7;
                }
            }
        }
}
</style>
```

文件名：client/src/components/center-panel-content/mission/mission-group.vue

```
<template lang="html">
    <div>
    <mission-group-item :task-group="taskGroup" :key="index" v-for="(taskGroup,index) in taskGroups"></mission-group-item>

<create-mission-group :project-id="projectId" :project-name="projectName" :task-group-length="taskGroupLength" v-on:taskGroupCreated="addTaskGroup"></create-mission-group>
  </div>
</template>

<script>
import missionGroupItem from './mission-group-item'
import createMissionGroup from './create-mission-group'

export default {
  data (){
    return {
    }
  },
  computed:{
    taskGroupLength (){
      if(this.taskGroups){
        return this.taskGroups.length
      }else{
        return 0
      }
    }
  },
  methods:{
    addTaskGroup (taskGroup){
      // this.innerTaskGroups.push(taskGroup)
      this.$emit('taskGroupCreated',taskGroup)
    }
  },
  components:{missionGroupItem,createMissionGroup},
  props:['projectId','projectName','taskGroups']
}
</script>

<style lang="scss" scoped>
</style>
```

文件名：client/src/components/center-panel-content/mission/mission-task.vue

```
<template lang="html">
  <div>
    <div class="task clearfix" v-for="task in tasks" @click="showSlide(task.id,task.name)">
      <div class="task-title-container" :class="{completed:task.is_solved}">
        <div class="radio-box">
          <input type="checkbox" v-model="task.is_solved"
@click.stop="changeSolved(task.id,task.is_solved)">
        </div>
        <div class="task-title">{{ task.name }}</div>
        <div class="task-badges" v-if="!isProject">
          <div class="task-badge">
            <i class="glyphicon glyphicon-pushpin"></i>
            {{ task.project_name }}
          </div>
        </div>
        <div class="task-members">
          <div class="task-member-item" v-for="personInfo in task.focus_person_info">
            <div class="task-member-face">
              <div class="task-member-text">
                {{ personInfo.headName }}
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <task-slide :task-id="slideTaskId" :task-name="taskName" :is-show="isSlideShow"
v-on:slideClose="closeSlide"></task-slide>
  </div>
</template>

<script>
import taskSlide from './task-slide'
export default {
  data (){
    return {
      slideTaskId:-1,
      taskName:'',
      isSlideShow:false
```

```
        }
    },
    computed:{
        isProject (){
            return this.$store.state.secondRouterName==='project'
        }
    },
    methods:{
        changeSolved (id,isSolved){
            let infoObj={
                is_solved:isSolved
            }
            this.$paramsPost('/api/Tasks/update',infoObj,{where:{id}})
                .then(body=>{

                })
        },
        showSlide (taskId,taskName){
            this.isSlideShow=true
            this.slideTaskId=taskId
            this.taskName=taskName
        },
        closeSlide (){
            this.isSlideShow=false
        }
    },
    components:{taskSlide},
    props:['tasks']
}
</script>

<style lang="scss" scoped>
.task{
    margin-bottom: 4px;
    border: solid 1px #e8e8ea;
    position: relative;
    border-bottom: solid 1px #e1e1e5;
    background: #fff;
    min-height: 46px;
    padding-bottom: 2px;
    line-height: 20px;
    cursor: pointer;
```

```
&:hover{
  background: #f8f6f2;;
}
.completed{
  text-decoration: line-through;
}
.task-title-container{
  position: relative;
  padding: 11px 0 3px 0;

  .radio-box{
    display: block;
    position: absolute;
    left: 4px;
    top: 6px;
    width: 25px;
    height: 30px;
    line-height: 30px;
    text-align: center;
    color: #e1ddd5;
    font-size: 18px;

  }
  .task-title{
    display: block;
    word-wrap: break-word;
    white-space: normal;
    line-height: 21px;
    font-size: 14px;
    color: #484744;
    margin: 0 24px 0 32px;
  }
  .task-badges{
    float: left;
    margin: 0 8px 0 28px;
    white-space: normal;

    .task-badge{
      color: #acaba7;
      border-radius: 3px;
      -webkit-text-size-adjust: none;
      font-size: 11px;
```

```
        display: inline-block;
        line-height: 27px;
        padding: 0;
        margin-right: 7px;
        margin-left: 0;
    }
}
.task-members{
    float: right;
    padding: 0 8px;

    .task-member-item{
        margin-left: 5px;
        margin-bottom: 5px;
        float: right;
        position: relative;
        display: inline-block;
        zoom: 1;


        .task-member-face{
            display: inline-block;
            position: relative;
            zoom: 1;

            .task-member-text{
                width: 28px;
                height: 28px;
                line-height: 28px;
                font-size: 10px;
                text-align: center;
                overflow: hidden;
                display: inline-block;
                zoom: 1;
                vertical-align: middle;
                background: #64b1ca;
                border: solid 1px #64b1ca;
                font-size: 10px;
                border-radius: 50%;
                color: #fff;
            }
        }
```

```
            }
        }
    }
}
</style>
```

文件名：client/src/components/center-panel-content/mission/mission.vue

```
<template lang="html">
    <div>
        <mission-group></mission-group>
    </div>
</template>

<script>
import missionGroup from './mission-group.vue'

export default {
    components:{missionGroup}
}
</script>

<style lang="scss" scoped>
</style>
```

文件名：client/src/components/center-panel-content/mission/task-slide.vue

```
<template lang="html">
    <slide :is-show="isShow" v-on:slideClose="emitClose">
        <template slot="header">
            <div class="entity-project-name">
                {{ task.project_name }}
            </div>
        </template>
        <div slot="body" class="slide-body">
            <div class="body-title" :class="{'completed':task.is_solved}">
```

```
  {{ task.name }}
  <small>/{{ task.description || '项目暂无描述'}}</small>
  <input type="checkbox" class="tesk-status-btn" v-model="task.is_solved">
</div>
<div class="person-container clear-fix row">
  <div class="col-sm-6">
    <div class="title">
      分配任务给
    </div>
    <div class="person-header-container" v-for="person in task.solve_person_info">
      <span class="glyphicon glyphicon-remove"></span>
      <div class="g-user-head-default">
        {{ person.headName }}
      </div>
    </div>
    <div class="add-person-btn" @click="addSolve()" v-if="!isSolved">
      <span class="glyphicon glyphicon-plus"></span>
    </div>
  </div>
  <div class="col-sm-6">
    <div class="title">
      关注的人
    </div>
    <div class="person-header-container" v-for="person in task.focus_person_info">
      <span class="glyphicon glyphicon-remove"></span>
      <div class="g-user-head-default">
        {{ person.headName }}
      </div>
    </div>
    <div class="add-person-btn" @click="addFocus()" v-if="!isFocus">
      <span class="glyphicon glyphicon-plus"></span>
    </div>
  </div>
</div>
<div class="comment-container">
  评论
  <div class="comment clear-fix">
    <div class="g-user-head-default pull-left">
      庭旭
    </div>
    <div class="content-container">
      <div>
```

```
              <span>王庭旭</span> <span class="time">2017年4月12日 12:12</span>
              <span class="delete-btn">删除</span>
            </div>
            <div>
              首先实现信息抓取的部分。
            </div>
          </div>
        </div>
      </div>
      <div class="post-comment-container">
        <div class="form-group">
          <textarea name="name" rows="8" cols="80" class="form-control"></textarea>
        </div>
        <div class="form-group">
          <button class="btn btn-primary">发表评论</button>
        </div>
      </div>
    </div>
  </slide>
</template>

<script>
import slide from "@pub/slide"
export default {
  data (){
    return {
      task:{}
    }
  },
  watch:{
    taskId (newVal){
      this.$filterGet('/api/Tasks',{id:this.taskId})
      .then(([task])=>{
        this.task=task
      })
    }
  },
  computed:{
    isFocus (){
      if(this.task.focus_person_info){
        return this.task.focus_person_info.some((item)=>{
          return item.id===this.$getFromSession('userInfo','id')
```

```
            })
        }else{
            return false
        }
    },
    isSolved (){
        if(this.task.solve_person_info){
            return this.task.solve_person_info.some((item)=>{
                return item.id===this.$getFromSession('userInfo','id')
            })
        }else{
            return false
        }
    }
},
methods:{
    addFocus (){
        let userInfo=JSON.parse(this.$getFromSession('userInfo'))
        delete userInfo.password
        this.$filterPost('/api/Tasks/update',{
            focus_person_info:[...this.task.focus_person_info,userInfo]
        },{
            id:this.task.id
        }).then((res)=>{
            this.task.focus_person_info.push(userInfo)
        })
    },
    addSolve (){
        let userInfo=JSON.parse(this.$getFromSession('userInfo'))
        delete userInfo.password
        this.$filterPost('/api/Tasks/update',{
            solve_person_info:[...this.task.solve_person_info,userInfo]
        },{
            id:this.task.id
        }).then((res)=>{
            this.task.solve_person_info.push(userInfo)
        })
    },
    emitClose (){
        this.$emit('slideClose')
    },
    updateTask (){
            this.$emit('updateTask',{
                name:this.task.name,
                is_solved:this.task.is_solved
            })
        }
    },
    components:{
        slide,
        addPerson:{
            template:`
            <div class="add-person-btn" @click="clickFn()">
                <span class="glyphicon glyphicon-plus"></span>
            </div>
            `,
            props:['clickFn']
        }
    },
    props:['taskId','taskName','isShow']
}
</script>

<style lang="scss" scoped>
$hover-color:#da4f4a;
$default-color:#acaba7;
$border:solid 1px #eeece8;
.entity-project-name{
    float:left;
    font-size: 18px;
    text-overflow: ellipsis;
    overflow: hidden;
    white-space: nowrap;
    cursor:pointer;
    &:hover{
        color:$hover-color;
    }
}
.slide-body{
    padding:0 40px;
    .completed{
        text-decoration: line-through;
    }
    .body-title{
        font-size: 20px;
        font-weight: 500;
        position: relative;
        small{
            color:$default-color;
        }
        .tesk-status-btn{
            position: absolute;
            left:-30px;
            top:4px;
        }
    }
    .person-container{
        margin-top: 30px;
        margin-bottom: 20px;
        border-bottom: $border;
        padding-bottom: 30px;
        .title{
            color:$default-color;
            margin-bottom: 20px;
        }
        .person-header-container{
            position: relative;
            display: inline-block;
            float:left;
            margin-right: 10px;
            &:hover{
                span{
                    display: inline-block;
                }
            }
            span{
                font-size: 10px;
                position: absolute;
                right: -3px;
                top:-2px;
                display: none;
                cursor: pointer;
            }
        }
        .add-person-btn{
            width:40px;
            line-height:40px;
            border-radius: 30px;
            border:$border;
            float:left;
            text-align: center;
            color:$default-color;
            cursor: pointer;
            &:hover{
                border:1px solid $hover-color;
                color:$hover-color;
            }
        }
    }
    .comment-container{
        .comment{
            height:90px;
            padding:10px 0 5px;
            .content-container{
                margin-left: 50px;
                &:hover{
                    .delete-btn{
                        display: inline-block;
                    }
                }
                .time{
                    margin-left: 10px;
                    color:$default-color;
                }
                .delete-btn{
                    float: right;
                    display: none;
                    cursor: pointer;
                    &:hover{
                        color:$hover-color;
                    }
                }
            }
        }
    }
    .post-comment-container{

    }
```

```
}
</style>
```

文件名：client/src/components/center-panel-content/project/document-modal.vue

```
<template lang="html">
  <modal :hook="docHook">
    <template slot="title">
      文档信息详情
    </template>
    <template>
      <div class="form">
        <div class="form-group">
          文档名称：{{ docModalInfo.title }}
        </div>
        <div class="form-group" v-html="docModalInfo.content">
        </div>
      </div>
    </template>
    <template slot="footer">
      <button type="button" class="btn btn-primary" data-dismiss="modal">确定</button>
    </template>
  </modal>
</template>

<script>
import modal from '@pub/modal'
export default {
  components:{modal},
  props:['docModalInfo','docHook']
}
</script>

<style lang="css">
</style>
```

文件名：client/src/components/center-panel-content/project/project-item.vue

```
<template lang="html">
  <div>
    <div class="item" v-for="project in projectArr" @click="toMission(project.id,project.name)">
      <div class="icon-container">
        <i class="glyphicon glyphicon-inbox icon"></i>
      </div>
      <div class="text-container">
        <div>
          {{ project.name }}
        </div>
        <div class="description" v-html="project.description || '<br/>'">
        </div>
      </div>
    </div>
  </div>
</template>

<script>
export default {
  methods:{
    toMission (projectId,projectName){
      this.$router.push({name:'project',params:{projectId,projectName}})
    }
  },
  props:['projectArr']
}
</script>

<style lang="scss">

</style>
```

文件名：client/src/components/center-panel-content/project/document.vue

```
<template lang="html">
  <div class="content-main-container">
    <span class="main-title">添加文档</span>
```

```
<button class="btn btn-primary btn-sm" @click="createDocument()"><span class="glyphicon glyphicon-plus"></span></button>
<modal :hook="createHook">
  <template slot="title">
    新建文档
  </template>
  <template>
    <alert type="danger" :message="errMsg"></alert>
    <div class="form">
      <div class="form-group">
        <input type="text" class="form-control" v-model="documentName" placeholder="请输入文档名称">
      </div>
      <div class="form-group">
        <textarea class="form-control" v-model="documentContent" placeholder="文档正文" rows="8"></textarea>
      </div>
    </div>
  </template>
  <template slot="footer">
    <button type="button" class="btn btn-default" data-dismiss="modal">取消</button>
    <button type="button" class="btn btn-primary" @click="postDocumentInfo">创建</button>
  </template>
</modal>

<div class="project-main-container">
  <div class="item" v-for="(doc,$index) in docArr" @click="showDoc($index)">
    <div class="icon-container">
      <i class="glyphicon glyphicon-file icon"></i>
    </div>
    <div class="text-container">
      <div>
        {{ doc.title }}
      </div>
      <div class="description">
        <br/>
      </div>
    </div>
    <div class="delete-container" @click.stop="deleteDoc(doc.id,$index)">
      删除
    </div>
  </div>
```

```
      <documentModal :docHook="docHook" :docModalInfo="docModalInfo"></documentModal>
    </div>

  </div>
</template>

<script>
import modal from '@pub/modal'
import alert from '@pub/alert'
import marked from 'marked'
import documentModal from './document-modal'
export default {
  created (){
    this.getDocs()
  },
  data (){
    return {
      createHook:false,
      docHook:false,
      documentName:'',
      documentContent:'',
      docArr:[],
      docModalInfo:{
      },
      errMsg:''
    }
  },
  methods:{
    createDocument (){
      this.createHook=!this.createHook
    },
    postDocumentInfo (){
      this.errMsg=''
      const markedContent=marked(this.documentContent)
      const params=this.$route.params
      if(markedContent && this.documentName){
        let postInfo={
          project_id:Number(params.projectId),
          content:markedContent,
          person_id:this.$getFromSession('userInfo','id'),
          person_name:this.$getFromSession('userInfo','name'),
          create_time:Date.now().toString(),
          title:this.documentName
        }
        this.$post('/api/Documents',postInfo)
        .then((res)=>{
          this.documentName="
          this.documentContent="
          this.docArr.push(res)
          this.$hideModal()
        })
      }else{
        this.errMsg='请填写全所有信息'
      }
    },
    getDocs (){
      this.$filterGet('/api/Documents',{
        project_id:this.$route.params.projectId
      })
      .then((res)=>{
        this.docArr=res
      })
    },
    showDoc (docIndex){
      const docInfo=this.docArr[docIndex]
      this.docModalInfo=docInfo
      this.docHook=!this.docHook
    },
    deleteDoc (id,index){
      this.$delete(`api/Documents/${id}`)
      .then((res)=>{
        this.docArr.splice(index,1)
      })
    }
  },
  components:{
    modal,
    alert,
    documentModal
  }
}
</script>

<style lang="scss" scoped>
```

```
</style>
```

文件名：client/src/components/center-panel-content/project/project.vue

```
<template lang="html">

<mission-group :project-id="$route.params.projectId" :project-name="projectName" :task-groups="task
Groups" v-on:taskGroupCreated="taskGroupCreated"></mission-group>
</template>

<script>
import missionGroup from '../mission/mission-group'

export default {
  created (){
    this.$filterGet('/api/Task_groups',{project_id:this.projectId})
      .then(body=>{
        this.taskGroups=body
      })
  },
  data (){
    return {
      taskGroups:[]
    }
  },
  computed:{
    projectName (){
      return this.$route.params.projectName
    },
    projectId (){
      return this.$route.params.projectId
    }
  },
  methods:{
    taskGroupCreated (taskgroup){
      this.taskGroups.push(taskgroup)
    }
  },
  components:{missionGroup}
}
</script>

<style lang="scss" scoped>
</style>
```