ts+vue项目应用

UI库等第三方库引入

- 声明文件

```
// 常用库都是有声明文件的，如果没有就手动安装
// 安装地址形式通常是@types/xxx
// 这里是：npm i -D @types/xxx
// 假设lodash：npm i -D @types/lodash
```

```
// 以moment为例
import moment from 'moment'

@Component
export default class App extends Vue {
    msg = moment().format('YYYY-MM-DD')
}
```

- 组件实例引用ref

```
<el-form ref="loginForm">
```

```
import { Component, Prop, Vue, Ref } from "vue-property-decorator";
import { Form } from "element-ui";
@Component
export default class Hello extends Vue {
    @Ref() loginForm!: Form;
}
```

vue-router

- 路由配置

```
import { RouteConfig, Route } from "vue-router";

const routes: RouteConfig[] = [...]
//...

router.beforeEach((to: Route, from: Route, next) => {
  console.log("beforeEach");
  next();
});
```

- 路由守卫：只有组件级守卫需要额外操作

```
// App.vue
Component.registerHooks([
  'beforeRouteEnter',
  'beforeRouteLeave',
  'beforeRouteUpdate',
]);

@Component
export default class App extends Vue {}
```

> 注意放在App声明前面，让Component先注册

```
// Home.vue
import { Route } from "vue-router";

export default class Home extends Vue {
  beforeRouteEnter(to: Route, from: Route, next) {
    console.log("beforeRouteEnter");
    next(vm => {
      console.log(vm);
    });
  }
}
```

vuex

- store定义

```
import { RootState } from "@/types";

export default new Vuex.Store<RootState>({...})
```

定义RootState，@/types/index.ts

```
export interface RootState {
  counter: number;
}
```

- 模块化

子模块声明，store/user.ts

```
import { Module } from "vuex";
import { RootState, UserState } from "@/types";

export const user: Module<UserState, RootState> = {
  namespaced: true,
  state: {
    name: "",
    token: "",
  },
  mutations: {
    setUser(state, { name, token }) {
      state.name = name;
      state.token = token;
    },
  },
  actions: {
    login({ commit }, user): Promise<void> {
      return new Promise(resolve => {
        setTimeout(() => {
          resolve();
          commit(user);
        }, 1000);
      });
    },
  },
};
```

使用，Hello.vue

```
import { namespace } from "vuex-class";

const userModule = namespace("user");

@Component
export default class Hello extends Vue {
  @userModule.State("name") username!: string;
  @userModule.Action login!: (userInfo) => Promise<void>;
}
```

Mixins

mixins本意是扩展、继承，利用vue-class-component提供的Mixin函数可以将多个mixin合成一个基类，从而巧妙利用继承实现扩展。

编写minxin

```
@Component
class Mixin extends Vue {
  created(){
    console.log('created in mixin');
  }
}
```

使用

```
@Component
export default class Hello extends Mixins(Mixin) {}
```