

Centralizando o domínio da API

Transcrição

Hoje, usamos o endereço `localhost:3000` para consumir nossa API localmente. Sabemos que esse é um endereço de desenvolvimento e que futuramente precisamos alterá-lo para apontar para o endereço da API hospedada em algum servidor da web.

Talvez, você tenha que trocar ainda mesmo em ambiente de desenvolvimento, por exemplo, caso esteja acessando uma API na porta 8080. São todos motivos genuínos de mudança, mas do jeito que nossa aplicação esta teremos problemas de manutenção.

Acessamos a API nos componentes `Home` `Cadastro`. Se o endereço mudar, teremos que lembrar de alterar nos dois componentes. Em uma aplicação com escopo reduzido como a nossa não é tão difícil procurar por todos os locais que acessam a API, mas em uma aplicação maior isso pode ser tornar um pesadelo. Basta esquecermos de alterar em um local para que a aplicação deixe de funcionar corretamente.

A boa notícia é que o `VueRouter` nos permite centralizar o domínio da nossa API em um único local evitando assim que tenhamos de informá-lo toda vez que formos acessar nossa API.

Vamos abrir o arquivo `alurapic/src/main.js`, local onde registramos o `VueRouter`.

Com o arquivo aberto, vamos atribuir o endereço `http://localhost:3000` para a propriedade `Vue.http.options.root`:

```
// alurapic/src/main.js

import Vue from 'vue'
import App from './App.vue'
import VueResource from 'vue-resource';
import VueRouter from 'vue-router';

import { routes } from './routes';

import './directives/Transform';

Vue.use(VueRouter);

const router = new VueRouter({
  routes,
  mode: 'history'
});

Vue.use(VueResource);

// http usará sempre o endereço abaixo

Vue.http.options.root = 'http://localhost:3000';

new Vue({
  el: '#app',
  router,
```

```
render: h => h(App)
})
```

O valor passado para a propriedade `root` será a raiz de toda requisição que fizermos através de `VueRouter` e seus artefatos como `$http`.

Agora, em todo local que usamos `$http` podemos remover a parte do endereço `http://localhost:3000`.

Alterando nosso componente `Cadastro`.

```
// alurapic/src/components/cadastro/Cadastro.vue
// código anterior omitido

methods: {

  grava() {

    console.log(this.foto);

    // removido parte do endereço!

    this.$http
      .post('v1/fotos', this.foto)
      .then(() => this.foto = new Foto(), err => console.log(err));

  }
}

// código posterior omitido
```

Agora em nosso componente `Home`:

```
// alurapic/src/components/home/Home.vue
// código anterior omitido

methods: {

  remove(foto) {

    // removido parte do endereço aqui

    this.$http
      .delete(`v1/fotos/${foto._id}`)
      .then(
        () => {
          let indice = this.fotos.indexOf(foto);
          this.fotos.splice(indice, 1);
          this.mensagem = 'Foto removida com sucesso'
        },
        err => {
          this.mensagem = 'Não foi possível remover a foto';
          console.log(err);
        }
      )
  }
}
```

```
    )  
  }  
  
  },  
  
  created() {  
  
    // removido parte do endereço aqui  
  
    this.$http  
      .get('v1/fotos')  
      .then(res => res.json())  
      .then(fotos => this.fotos = fotos, err => console.log(err));  
  }  
}  
  
// código posterior omitido
```

Nossa aplicação deve continuar funcionando da mesma forma como antes, apenas centralizamos o domínio da nossa API em um único lugar.

Que tal se pudermos melhorar ainda mais nosso código? É isso que veremos no próximo vídeo!