链滴

Element-Ui 之 Upload 组件, 利用 axios+腾讯云 cos 上传

作者：flee-lether

原文链接：https://ld246.com/article/1578412996710

来源网站：链滴

利用腾讯云cos做文件服务器，配合element-ui的upload组实现。

腾讯cos上传主要分为三个步骤

1. 设置文件存储位置并返回上传文件地址
2. 将文件转化为二进制流
3. 通过二进制流进行文件上传

设置文件存储位置

通过生成guid保证上传的文件不会被覆盖

```
function guid() {
  function S4() {
    return (((1 + Math.random()) * 0x10000) | 0).toString(16).substring(1);
  }
  return S4() + S4() + S4() + S4() + S4() + S4() + S4() + S4();
}
```

设置并获取cos文件存储路径

```
_axios
    .get(url + "?path=" + guid())
    .then(data => {
      const uploadUrl = data;
      // TODO 上传文件
    });
```

将文件转化为二进制流

主要利用FileReader实现。

```
let fr = new FileReader();
    fr.readAsDataURL(file);
    fr.addEventListener(
      "load",
      () => {
        let arr = fr.result.split(",");
        let bstr = atob(arr[1]);
        let n = bstr.length;
        let u8arr = new Uint8Array(n);
        while (n--) {
          u8arr[n] = bstr.charCodeAt(n);
        }
        // TODO 设置文件存储路径
      },
      false
    );
```

原文链接：Element-Ui 之 Upload 组件, 利用 axios+ 腾讯云 cos 上传

上传文件至cos服务器

```
_axios
  .put(uploadUrl, binary, {
    timeout: 60 * 60 * 1000,
    onUploadProgress: e => {
      if (e.total > 0) {
        e.percent = (e.loaded / e.total) * 100;
      }
      const { onProgress } = option;
      if (
        Object.prototype.toString.call(onProgress) === "[object Function]"
      ) {
        onProgress(e);
      }
    },
    headers: {
      "Content-Type": fileType
    }
  })
  .then(() => {
    // 上传成功
  })
  .catch(error => {
    // 上传失败
  });
```

完整代码如下

```
import _axios from "@/plugins/axios";

/**
 * 生成GUID
 */
function guid() {
  function S4() {
    return (((1 + Math.random()) * 0x10000) | 0).toString(16).substring(1);
  }
  return S4() + S4() + S4() + S4() + S4() + S4() + S4() + S4();
}

/**
 * 腾讯云cos文件上传
 * @param {String} uploadUrl cos上传路径
 * @param {Blob} binary 二进制文件
 * @param {String} fileType 文件类型，默认为image/jpeg
 * @param {Object} option ElementUI上传组件配置
 */
function upload(uploadUrl, binary, fileType = "image/jpeg", option = {}) {
  return new Promise((resolve, reject) => {
    _axios
      .put(uploadUrl, binary, {
        timeout: 60 * 60 * 1000,
```

```
        onUploadProgress: e => {
          if (e.total > 0) {
            e.percent = (e.loaded / e.total) * 100;
          }
          const { onProgress } = option;
          if (
            Object.prototype.toString.call(onProgress) === "[object Function]"
          ) {
            onProgress(e);
          }
        },
        headers: {
          "Content-Type": fileType
        }
      })
      .then(() => {
        resolve(uploadUrl);
      })
      .catch(error => {
        reject(error);
      });
  });
}

/**
 * 腾讯云cos上传，获取文件上传路径，并上传文件
 * @param {String} url 文件上传路径
 * @param {File} file 上传文件
 * @param {Object} option 上传配置
 */
export function fileUpload(url, file, option = {}) {
  return new Promise((resolve, reject) => {
    let fr = new FileReader();
    fr.readAsDataURL(file);
    fr.addEventListener(
      "load",
      () => {
        let arr = fr.result.split(",");
        let bstr = atob(arr[1]);
        let n = bstr.length;
        let u8arr = new Uint8Array(n);
        while (n--) {
          u8arr[n] = bstr.charCodeAt(n);
        }
        _axios
          .get(url + "?path=" + guid())
          .then(data => {
            const uploadUrl = data;
            return upload(uploadUrl, u8arr, file.type, option);
          })
          .then(path => {
            const url = path.split("?")[0];
            resolve(url);
          })
```

```
          .catch(reason => {
            reject(reason);
          });
      },
      false
    );
  });
}

/**
 * 重写ElementUI上传时ajax方法
 * @param {Object} option ElementUI上传组件配置
 */
export function uploadOverrideElement(option) {
  fileUpload(option.action, option.file, option)
    .then(url => {
      option.onSuccess(url);
    })
    .catch(reason => {
      option.onError(reason);
    });
}
```

最后

在引用Upload时重写httpRequest,即可

```
import { Upload } from "element-ui";
import { uploadOverrideElement } from "@/utils/uploadUtils";

Upload.props.httpRequest = {
  type: Function,
  default: uploadOverrideElement
};

Vue.use(Upload);
```