Vue.js Axios

WASA: Web and Software Architecture

Enrico Bassetti

AJAX

AJAX: Asynchronous JavaScript and XML

AJAX, Asynchronous JavaScript and XML, is a set of technologies for incremental updates in web pages. It contains:

- HTML/XHTML/CSS
- JavaScript
- DOM
- XML/XSLT
- **XMLHttpRequest** (a.k.a. XHR)

XMLHttpRequest

XMLHttpRequest is an object for issuing web requests and get the result. Despite its name, it can be used even for non-XML requests/responses.

It's supported by every browser! Even Internet Explorer supports* it!

If you use vanilla JavaScript and you want AJAX, chances are that you're using XMLHttpRequest.

*although Internet Explorer supported XMLHttpRequest via ~~the evil~~ ActiveX

XMLHttpRequest

Usage example

```javascript
let httpRequest;

function makeRequest() {
  httpRequest = new XMLHttpRequest();

  if (!httpRequest) {
    alert("Giving up :( Cannot create an XMLHTTP instance");
    return false;
  }
  httpRequest.onreadystatechange = alertContents;
  httpRequest.open("GET", "test.html");
  httpRequest.send();
}

function alertContents() {
  if (httpRequest.readyState === XMLHttpRequest.DONE) {
    if (httpRequest.status === 200) {
      alert(httpRequest.responseText);
    } else {
      alert("There was a problem with the request.");
    }
  }
}
```

XMLHttpRequest

We won't use XMLHttpRequest as it's too verbose. Its legacy interface is not up to date (e.g., missing support for asynchronous JavaScript).

What about JavaScript Fetch APIs?

Modern JavaScript applications support "Fetch API", a new way for fetching data from a remote server.

We won't use "Fetch" as it's still a work in progress. Example of missing features:

- **Timeout** (!!!)
- Upload/download progress
- Backward compatibility
- …

Axios

Axios

Axios is a library built on top of XMLHttpRequests (in browsers).

- Make XMLHttpRequests from the browser
- Supports the Promise API
- Intercept request and response
- Transform request and response data
- Cancel requests
- Automatic transforms for JSON data
- Client side support for protecting against XSRF

Bullet points from the Axios website

Axios is already available in Fantastic Coffee (decaffeinated).

Usage example using async (from webui/src/views/HomeView.vue):

```
async refresh() {
  try {
    let response = await this.$axios.get("/");
    // response.data contains the JSON in response body
  } catch (e) {
    console.error(e.toString());
  }
}
```

POST example:

```javascript
async sendData() {
  try {
    let response = await this.$axios.post("/some/api", {
      firstName: 'Fred',
      lastName: 'Flintstone',
    });
    // response.data contains the JSON in response body
  } catch (e) {
    console.error(e.toString());
  }
}
```

Response data

```
{
  data: {},
  status: 200,
  statusText: 'OK',
  headers: {},
  config: {},
  request: {}
}
```

Request Interceptor

```
axios.interceptors.request.use(function (config) {
    return config;
  }, function (error) {
    return Promise.reject(error);
  });
```

Response interceptor

```
axios.interceptors.response.use(function (response) {
    return response;
  }, function (error) {
    return Promise.reject(error);
  });
```

Cancellation

```javascript
const controller = new AbortController();

axios.get('/foo/bar', {
  signal: controller.signal
}).then(function(response) {
  // …
});

// cancel the request
controller.abort()
```