

Design Documentation

Architecture

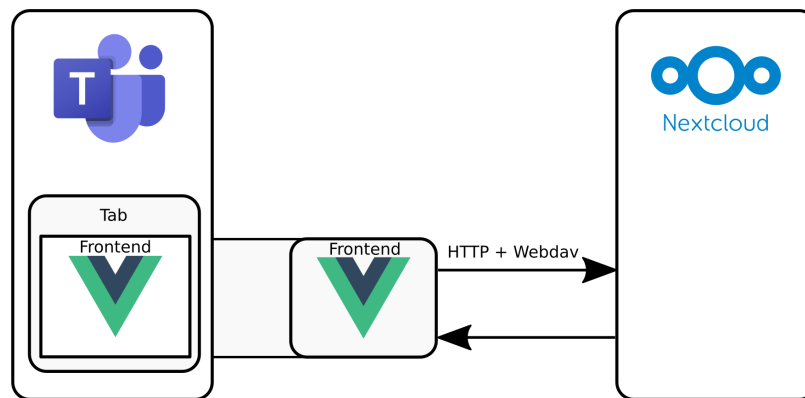


Figure 1: Architecture

Our Architecture currently consists of three parts. 1. The Teams Tab that displays our frontend within an iFrame. 2. Our Vuejs Frontend being served by an online host (<https://tms2nc.de/>). 3. A Nextcloud instance setup to work with SSO and Azure Active Directory

When a user is using our Microsoft Teams Plugin then it will work the following way:

They will add a new Tab to their Teams channel and choose our plugin which is listed on the current Teams App store.

The first page they'll see is a configuration page for which they can setup our plugin for their channel.

After that every user in the channel should see a new Tab on the top of the channel that they can click on.

When clicking on that Tab our frontend should load and show the user a landing page at first. This page only allows to login via Nextcloud, after clicking the

Login button the user should see a login flow within their browser that asks them to allow the user to handle their account.

After successful login the page will display the root directory of the user's nextcloud account.

From here on the user can edit, delete, upload and download files directly from our frontend which communicates with Nextcloud via HTTP Requests using the Webdav protocol.

Code

The repository contains two major parts.

The public folder contains the `manifest.json` file which hosts information about our Teams Addon, the url of our frontend, the icons and other important information to embed our frontend within Teams.

The other part is our frontend code, written in vuejs (<https://vuejs.org/>).

The structure of the vue app is the following

```
frontend/src/  
  main.js  
  App.vue  
  actions  
  components  
  file-wrapper  
  router  
  store  
    index.js  
    modules  
  views
```

`main.js` is the entry point of our application, here we load the Vue instance and top level interfaces for other modules.

`App.vue` is the entry point for the Vue instance itself, it contains the top level view of our application that will get mounted onto the `#app` div.

`actions` contains the actions for the top Navbar which contains the action buttons for working with the file list.

`components` contains our vue components <https://v3.vuejs.org/guide/component-basics.html>

`file-wrapper` contains our basic file wrapper object which is the representation of a Nextcloud file as a Javascript Object

`router` contains our routing code, this part will display different views from the `views` folder depending on the url in the browser.

store contains the vuex part of our code, vuex is a state management library for vue that allows us to access certain state and objects from all parts of our code without having to rely on handing down state and props through different dependency chains.

views contains the different pages that are to be displayed depending on the current url.