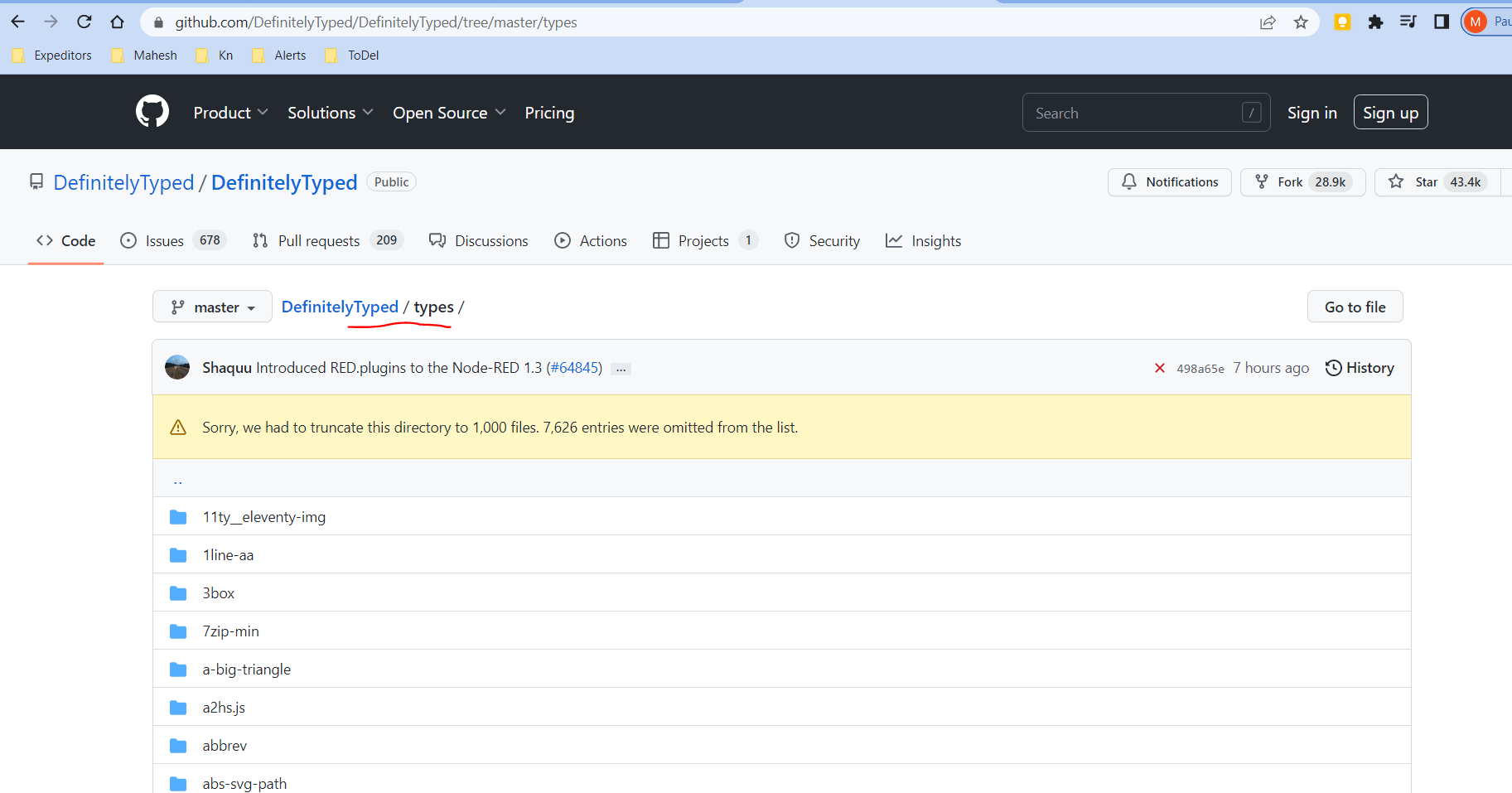
* - [Instructor] In the previous video, I showed you how to use TypeScript to parse JavaScript files without actually converting them to full fledged TypeScript files.
* I also showed you how TypeScript found a few errors and I fixed most of those errors, but I left one, the one in which TypeScript warns me that ***it doesn't know what the dollar sign means***, or rather, it's not aware of the jQuery library that I'm using here.
* Since TypeScript has been around for quite some time now, you'll probably find the majority of the JavaScript libraries you use will already have TypeScript support built in, especially if you're installing them from the NPM package repository.
* However, it is likely that at some point you will run across a library that doesn't offer TypeScript support, particularly if you are migrating an older application or one with a lot of custom code.
* In these cases, **you've got two options.**
* A) You can create the type definitions yourself, or B) in the case of an open source library, you can install type definitions that others have created online, hosted in the Definitely Typed repository, which you can find at github.
* com/definitelytyped/definitelytyped.

<https://github.com/definitelytyped/definitelytyped>

* Here, under the types folder, you'll find the type definitions for literally thousands of open source libraries.



* In fact, this list is so big, GitHub doesn't even show the whole thing at once.
* But don't worry, you don't have to search here because all of these folders are published as packages in the NPM package repository, npmjs.
* com. <https://www.npmjs.com/>
* And this site is much easier to search.
* Simply search for @types and see what you find.

Graphical user interface, text, application, email

Description automatically generated

* In my case, I'm looking for jQuery, so I'll search for the @types jQuery package.

Graphical user interface, application, email

Description automatically generated

* Once I find what I'm looking for, I'll simply copy the command from the site, and execute it in the root folder of my project.

Graphical user interface, application

Description automatically generated

* Once this is done, you'll see that this command has downloaded the package from the NPM repository and extracted it to the node modules folder.

Graphical user interface, text, application

Description automatically generated

* And inside of that folder, there will be a types folder with a jQuery folder inside of that.
* This folder contains all of the jQuery TypeScript definitions.

Graphical user interface, text, application

Description automatically generated

* The most important files will be the ones with the .
* d.
* ts extension, TypeScript declaration files.

Text

Description automatically generated

* If I open the one named JQuery.
* d.
* ts, I'll see a bunch of relatively advanced TypeScript code mixed with some pretty impressive documentation.
* More importantly, if I switch back to my JavaScript file where TypeScript was previously complaining about the missing jQuery definitions, I can now see that not only is TypeScript no longer complaining, but I actually have really good auto complete and inline documentation to help me use the library.

Graphical user interface, text, application

Description automatically generated

* **And that's how you import open source type definitions.**
* Now, what about your own internal JavaScript libraries? Well, you'll be using the same .
* d.
* ts files.
* You'll just need **to manually add them to your project yourself.**
* I'm not going to lie.
* Writing these definitions from scratch can be quite daunting.
* Heck, it could be an entirely separate course.
* However, you could probably get a long way by simply taking a look at some of these publicly available definitions and copying from them and modifying them for yourself.
* It'll be a lot of work, but if you have JavaScript libraries that you use in a lot of different projects, it might just be worth it.