Who wouldn’t love cache, right? Let’s see how we can cache data with Svelte & Appwrite!

We're going to explore how we can create a Svelte store that caches our data and is dynamically updated.

Luckily for us, it’s pretty easy.

The TL;DR is:

* Create a store with set, subscribe, and load
* Set the data to the cache
* Update the cache by subscribing to the data changes
* Update the data dynamically by calling the load function in your component

## Let's set everything up 📦

We’ll start by creating an appwrite.js store to make our life easier.

You can check a complete example [here](https://github.com/appwrite/demo-todo-with-svelte).

This allows us to access Appwrite's sdk from wherever we want.

Remember to add all the endpoints in the .env file at your project's root.

## The fun begins! 🥳

Now let's initialize another store where we will implement our caching:

export const documentsList = createDocumentsListStore();

Great! Let's work a little bit on that function. We want to create a writable store that can fetch our cached data, and is easy to update.

So, with this function, we check if there is data in the session storage and set the response to it.

## Where's my cache? 😕

We have a small problem: we've yet to save anything in our cache!

Luckily for us it's pretty straightforward:

We simply subscribe to the store and save the data in the session storage each time it changes.

## ✅ set, ✅ subscribe, 🔧 load

Now we have another problem! The data is set, and our cache is always up-to-date... but the data never changes!

That's what "load" is for ;)

For example, in our component we can do something like this:

This way, each time one of the arguments passed to "load" changes, the data will be updated!

And now we have an Appwrite store with cached data!

## Let's put it all together 🙌

In the end our store looks like this: