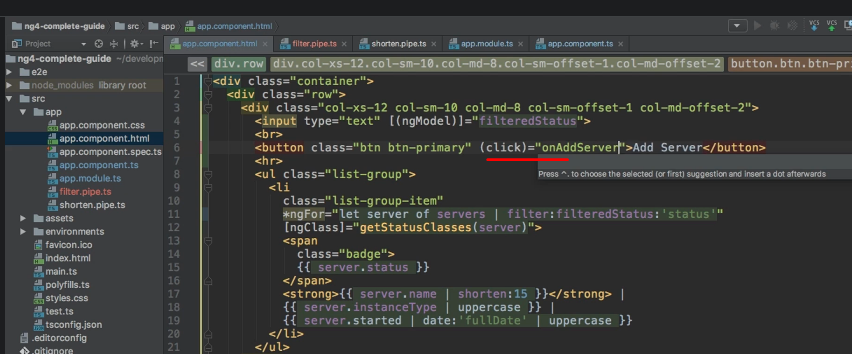
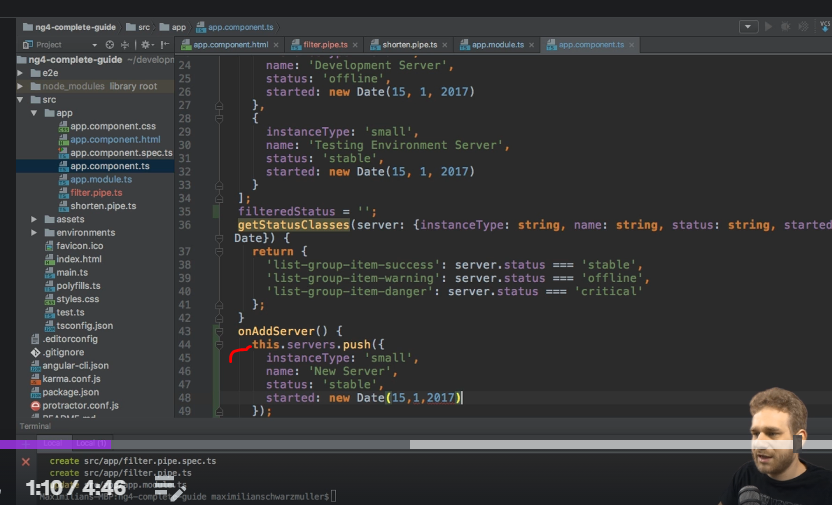
* -: In the last video, we created this filter pipe here, which allows us to filter our service for their state.
* And it seems to work great.
* We do have a certain issue with it though.
* If we allow the user to add a new server, and I will quickly add this functionality by going to my template and by adding a line break and then a button with the classes btn and btn-primary where I say Add Server, which will execute a new method onAddServer like this.



* And then in the app component, I will quickly add this method onAddServer.
* Here I will simply take my servers, push a new server on it and, of course, this will just be a generic server here which will have the instance type of, let's say, small, a name of New Server, let's keep it super simple here, a status of stable, and started could be new Date and then the same date as all the other servers here.



* This is not the recording date, as a side note, just like or how the way you want it.
* So with this, we are able to add a new server.

Graphical user interface, application

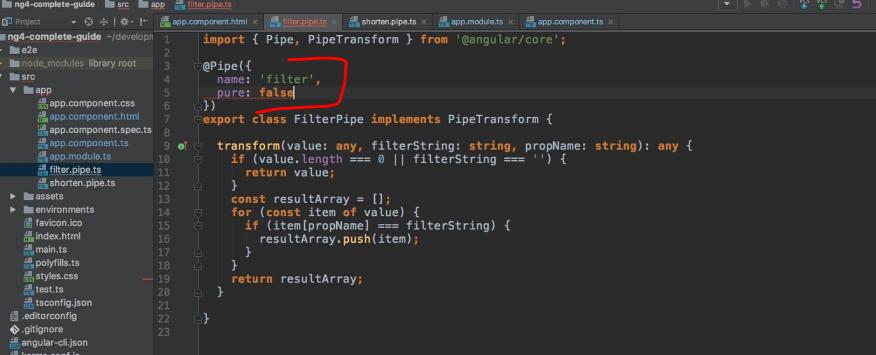
Description automatically generated

* And if I now click on Add Server, you see it's getting added here.
* If I filter for stable though, and I hit this button, you don't see it getting added here, but it is there.

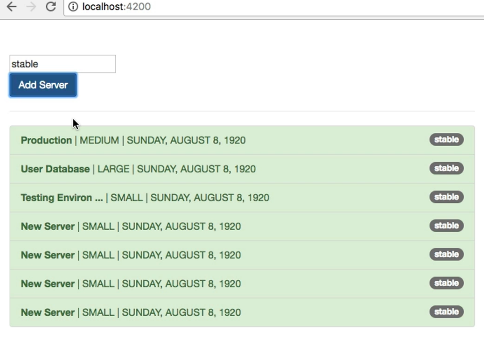
Graphical user interface, application

Description automatically generated

* You can see if I removed the filter altogether.
* So I added a couple of servers, and now we can filter for them again, but they weren't added whilst we had this list filtered.
* And this is actually not a bug.
* I'll quickly reload this and filter for stable again so that we can see this.
* If I hit Enter here, as mentioned, it's not getting added, but it's there.
* We can see as soon as we clear our filter.
* ***Now the reason for this behavior is that Angular is, thankfully, I should say, not rerunning our pipe on the data whenever this data changes.***
* **So as soon as we change what we enter here, and even if we would only add a blank space and then remove it, we would update our pipe again, as you can see.**
* **So adding the input, or excuse me, changing the input of the pipe will trigger a recalculation, will trigger the pipe being applied to the data again.**
* **But changing the data won't trigger this.**
* And this is a good behavior, because o***therwise Angular would have to run this pipe or rerun this pipe whenever any data on the page changes.***
* This would be really bad, because that would cost a lot of performance.
* ***And this is also the reason why no built-in filter pipe exists because filtering, you would say, is a pretty common task.***
* But the Angular team decided against adding such a pipe, because you typically have a high performance cost if you want to enforce it being updated, even if you are in filter mode.
* So, by default, it doesn't work here, but we can force it to work.
* But again, be aware that the *following change will make sure that whenever we change data on the page, our pipe is recalculated,* you could say.
* So this might lead to performance issues.
* So you should know what you're doing, which is why I'm really emphasizing this.
* **PURE:**
* ***You can force this pipe to be updated whenever the data changes by adding a second property to the pipe decorator.***
* ***It's called pure, and you can set it to false.***



* By default, this is true and doesn't need to be added.
* Now if you do this and the app reloads, let's filter for stable servers and add new server.
* And now you see they get added here too.



* The reason simply is that the pipe now gets recalculated whenever data changes.
* And as mentioned before, this can lead to performance issues, but might be what you're interested in.
* So it is fine to use, this is what this pure property is there for.
* You can set it to false, which basically means this is not, you could say, purely focusing on whether the pipe arguments changed, but now it will also be recalculated whenever anything changes on the page.
* And again, this might be a bad behavior.
* I'm just emphasizing this, because it's so important to understand.
* But if you know what you're doing, if you do need this behavior, definitely go ahead, set pure to false and you are good to go.
* And with that, we now get a working filter pipe with this little downside of potentially hitting performance issues if we have a very long list here.