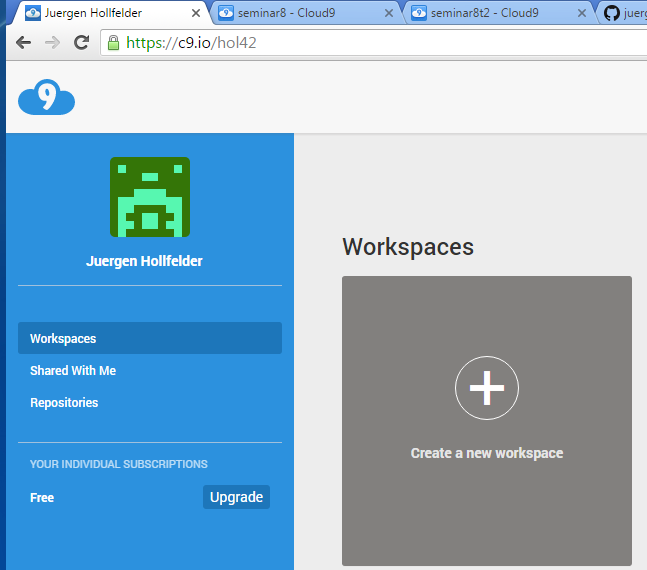
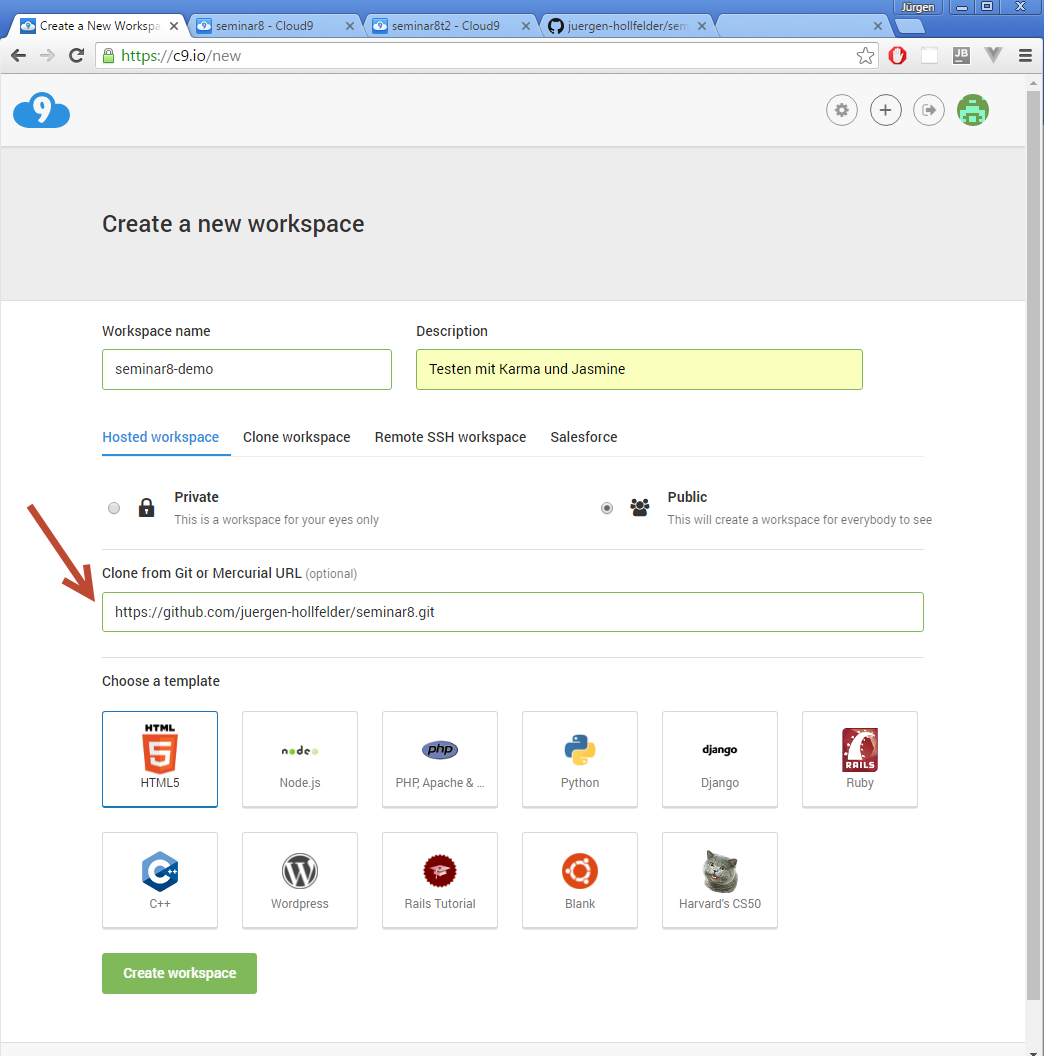
Seminar 8

1) Cloud9: Neuen Blank Ubuntu Workspace anlegen (HTML5 würde auch gehen)



<https://github.com/juergen-hollfelder/seminar8.git>

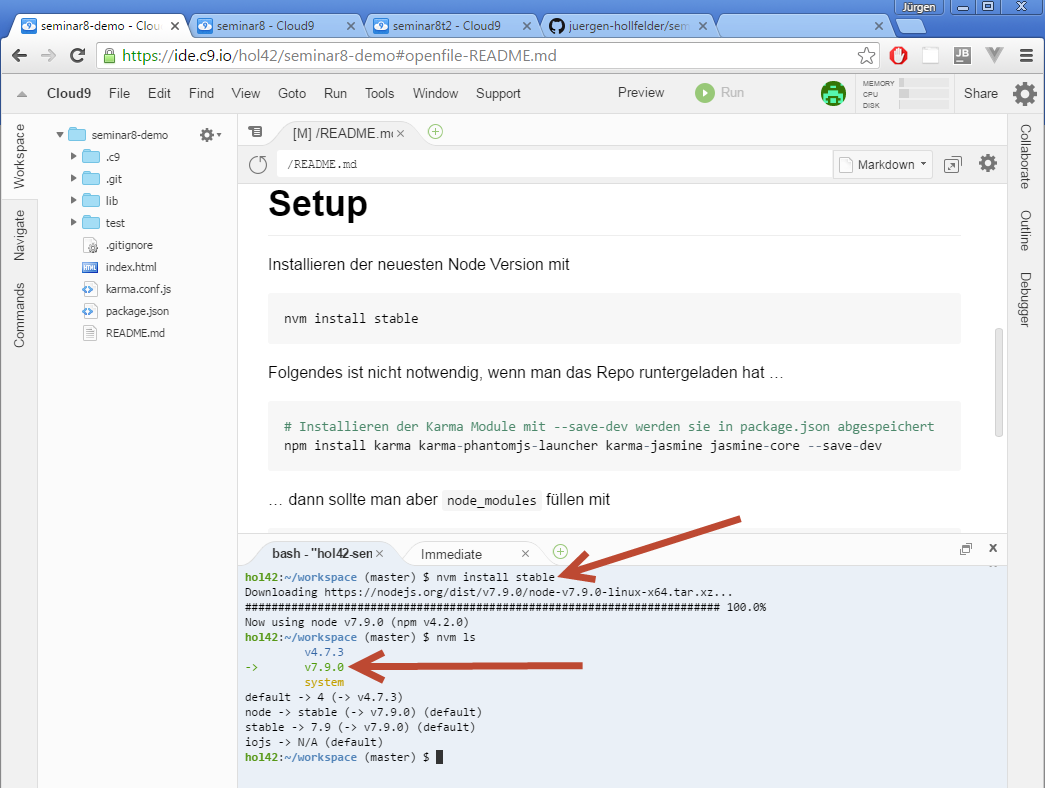


Nachträglich

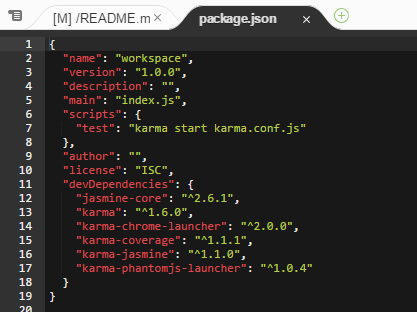
git clone https://github.com/juergen-hollfelder/seminar8.git

würde auch gehen.

2) Node auf neueste Version updaten



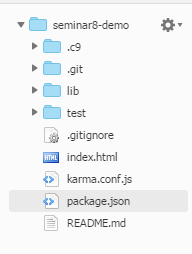
3) package.json kontrollieren, Module installieren



Hier sollte karma, karma-jasmine und jasmine-core drin sein

Das ^ Zeichen heisst Caret und bedeutet, dass es automatisch die neueste Unterversion bei einem npm update genommen wird. Also bei ^1.0.0 und es gibt eine 1.9.9, wird diese genommen. Aber wenn es eine 2.0.0 gibt nicht.

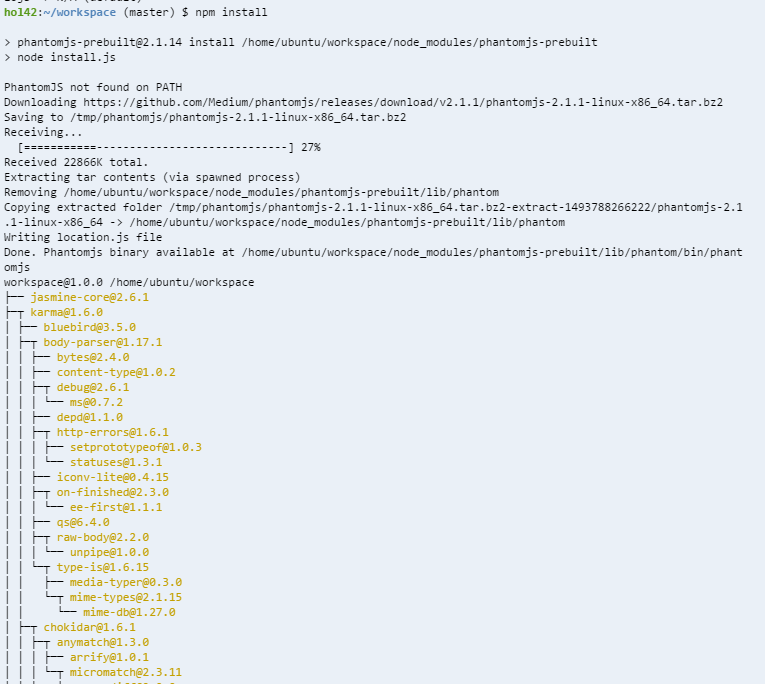
Im Clone fehlen diese node\_modules:



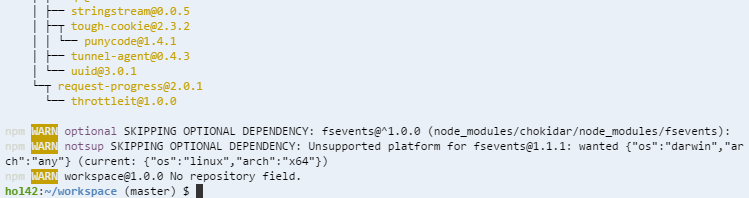
Diese müssen jetzt installiert werden:

hol42:~/workspace (master) $ npm install

Man erhält eine lange Liste von Modulen die geladen werden, inklusive Untermodule

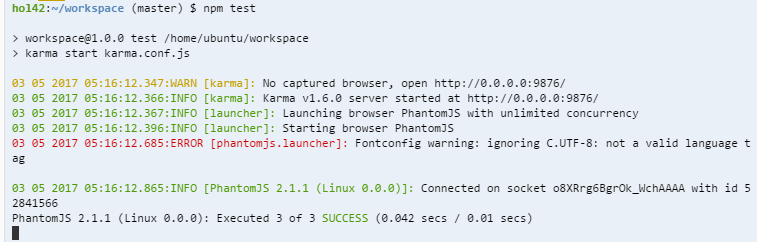


usw.



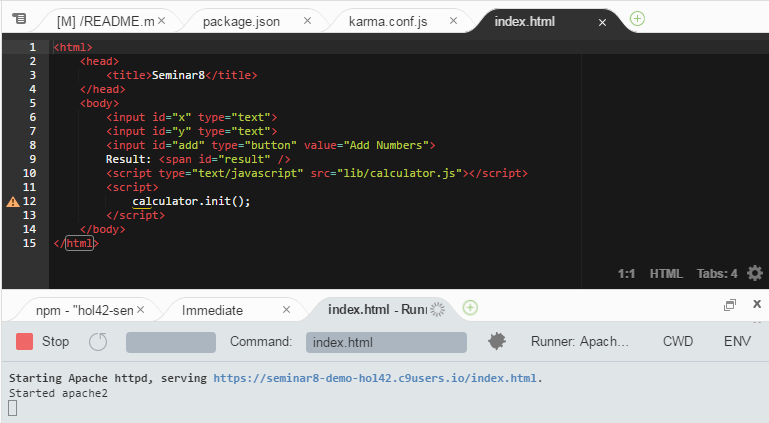
Die Warnungen zum Schluss können in unserem Fall ignoriert werden.

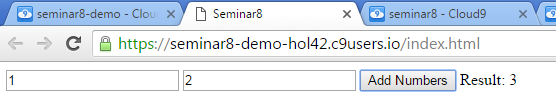
4) Test durchführen



5) Was macht das Programm?

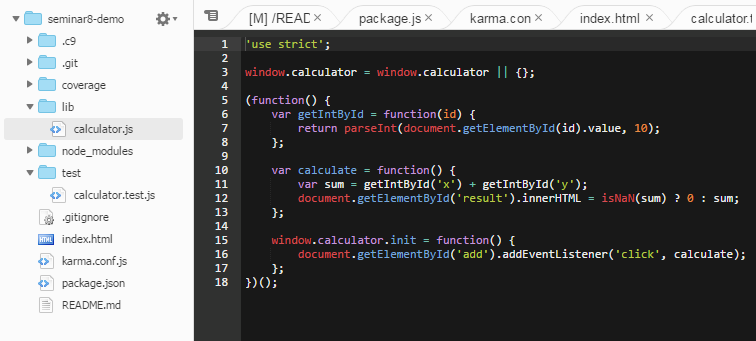
index.html mal ausführen:



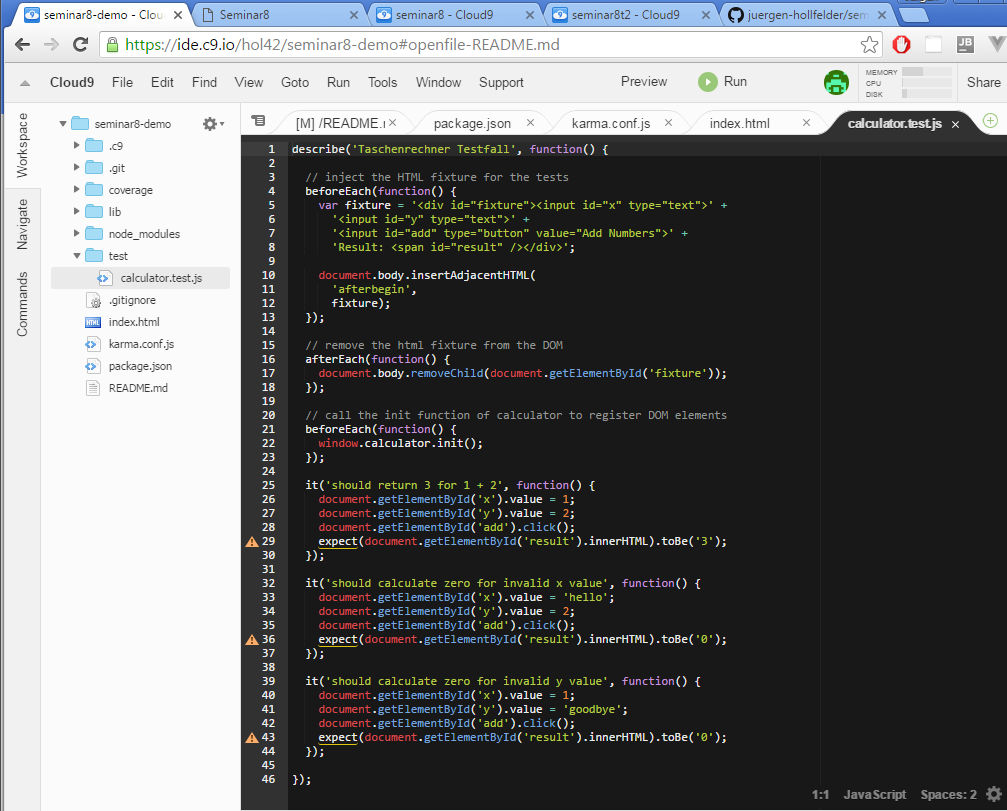


6) Wie und was wird getestet?

Getestet wird das Javascript im lib Verzeichnis

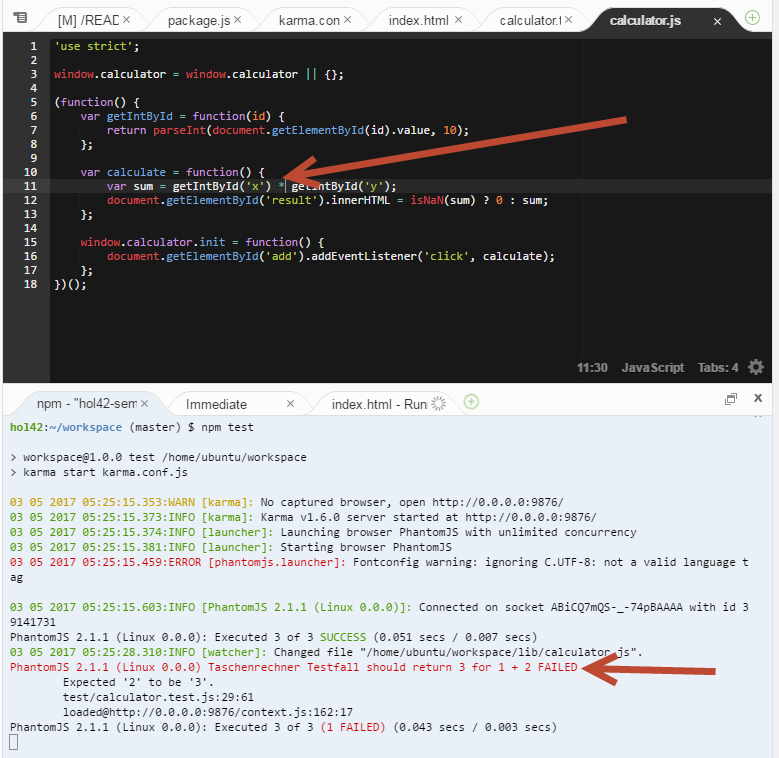
:

Das Testscript sitzt im test Unterverzeichnis

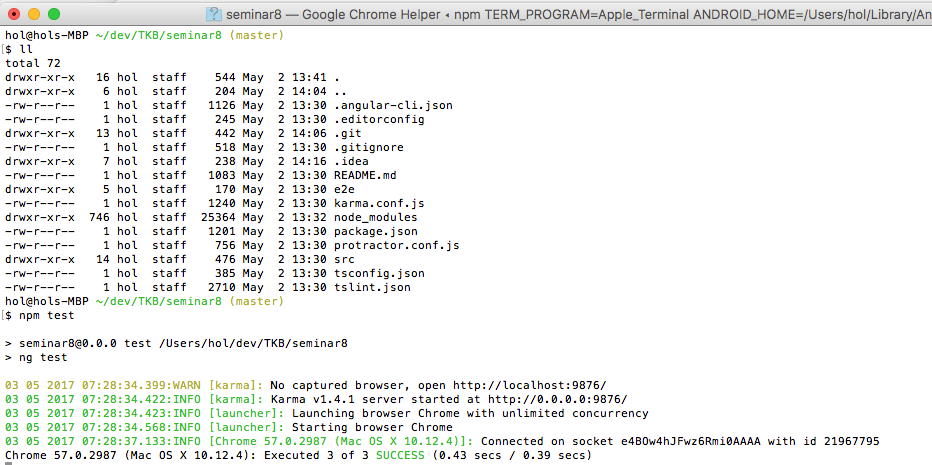


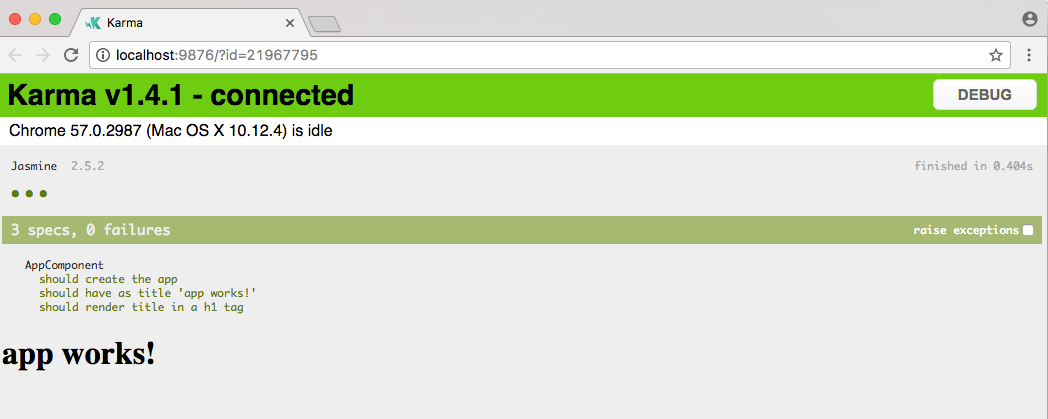
Das ist jetzt eher nur für Demozwecke. Es wiederholt das HTML, simuliert, die Eingabe 1 und 2 und prüft ob das Ergebnis dann 3 ist.

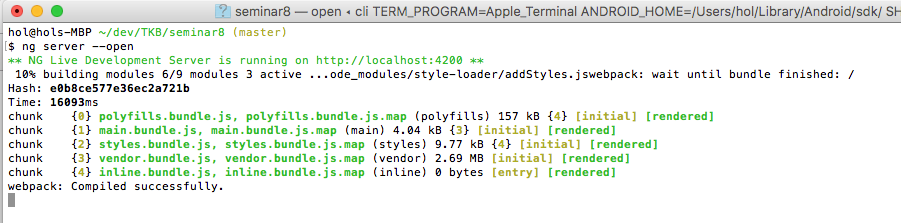
Sobald man etwas an den Testfällen oder am Script ändert, wird automatisch der Test durchgeführt

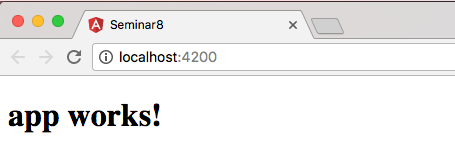


8) Eine Angular 2 App testen. Karma etc. wird gratis mitgeliefert und funktioniert problemlos Lokal. In Cloud 9 gibt es aber kein Browser. Hier die Bilder vom Mac:



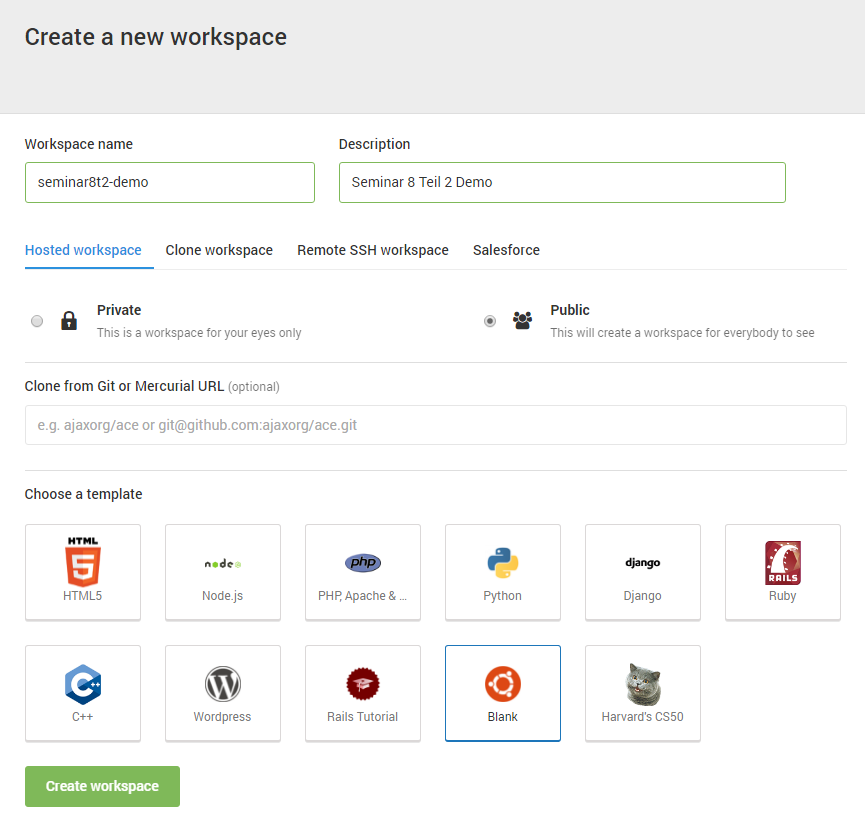






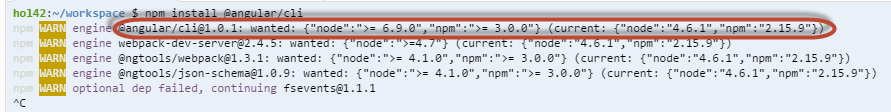
9) So in der Cloud 9:

Blank Ubuntu ohne Git Repository



10) Angular Command Line Tool installieren

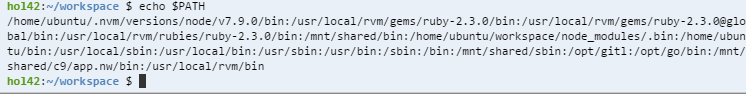
Zuerst upgrade node, weil sonst geht es nicht (Fehler wie dieser darf nicht kommen)



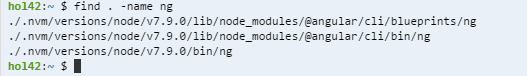
So geht es:



Mit der „-g“ Option kann man angeben, dass das Angular Command Line Tool global verfügbar ist. Normalerweise installier „npm install“ einfach nur im aktuellen Verzeichnis die „node\_modules“ usw. Aber mit –g wird ein Verzeichnis auf Betriebssystemebene angelegt und der Befehl ist dann von dort verfügbar, weil das ein zentrales Verzeichnis ist, von dem alle Befehle geholt werden. Im $PATH



Da ist es

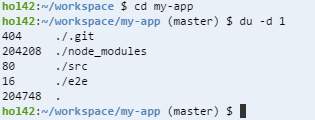


Also benutzen wir es:



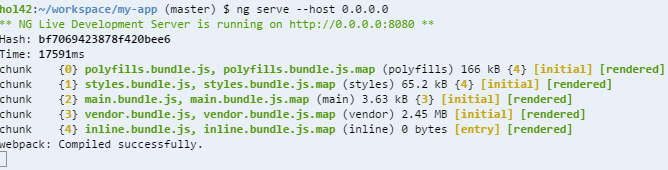
Das kann ein Moment lang gehen. Es wird auch hier beachtlich runtergeladen. Für NG z.B. wurden schon über 140000 Files runter geladen.

Um eine Basic Angular App anzulegen werden auch gleich mal jede Menge Files angelegt. Es ist eigentlich unglaublich…

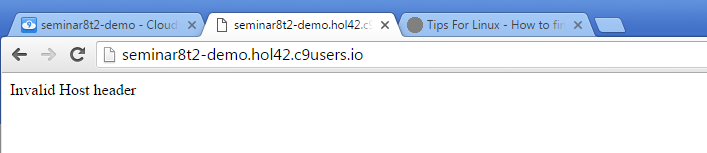


Nun gut, damit arbeiten wir jetzt.

11) Ausprobieren

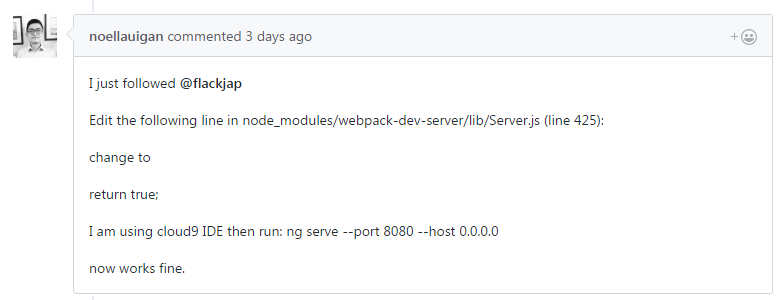


Uups:

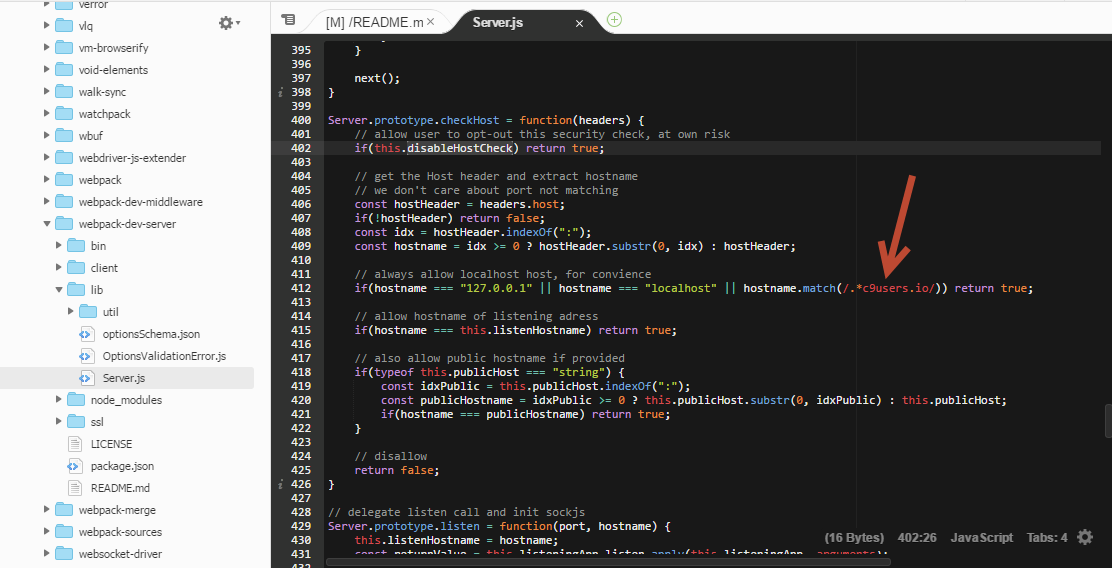


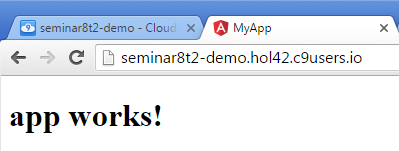
Ein Security Update von Node verhindert, dass mit einer anderen URL als localhost oder 0.0.0.0 auf eine Angular App im Developer Mode zugegriffen werden kann. Das ist erst seit April 2017 so. Ein kleiner Hack setzt das ausser Kraft.

<https://github.com/angular/angular-cli/issues/6070>



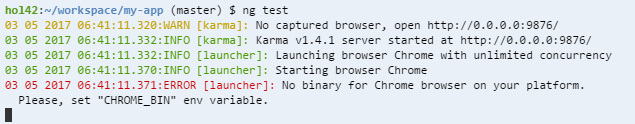
Man kann es auch so machen





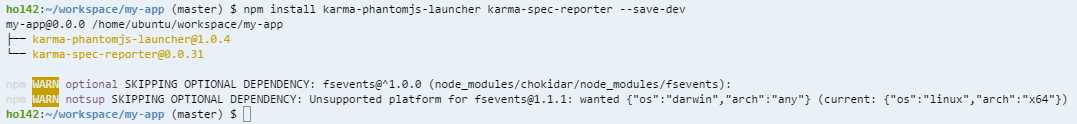
Eigentlich müsste man die Webpack Config ändern, dass lässt Angular CLI aber nicht zu. Man könnte „NG EJECT“ eingeben aber dann lässt sich NG garnicht mehr benutzen…

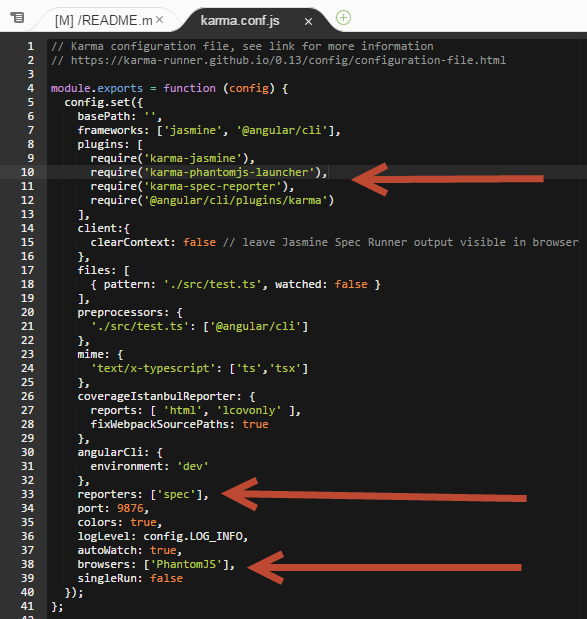
12) Jetzt zum Test



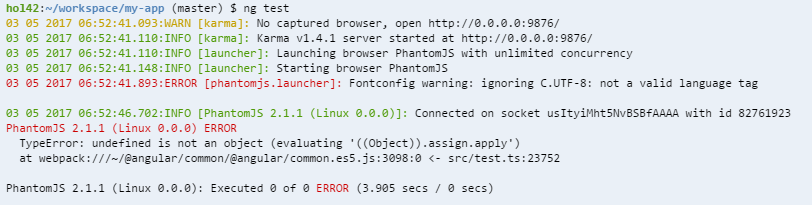
Oh nein! Wir sind in der Unix Box ohne GUI ohne Browser ohne alles…

So vorgehen

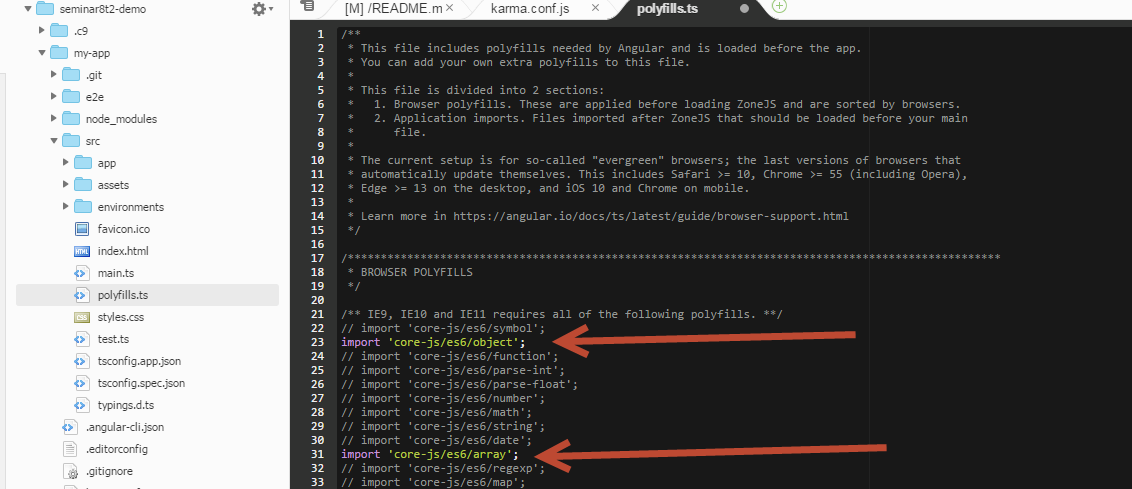


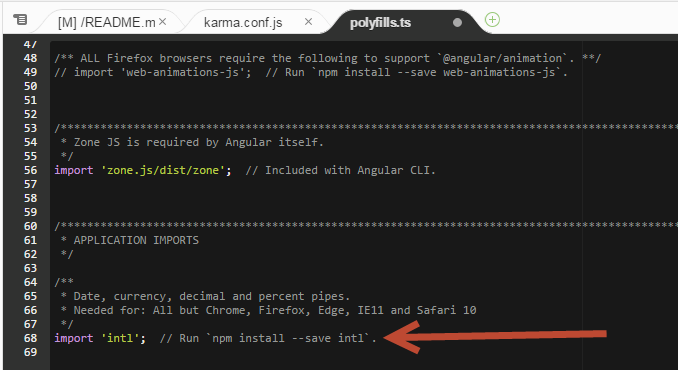


Leider geht es dann immer noch nicht, der PhantomJS Browser kann nicht alles was Angular tun will und Chrome zum Beispiel auch kann

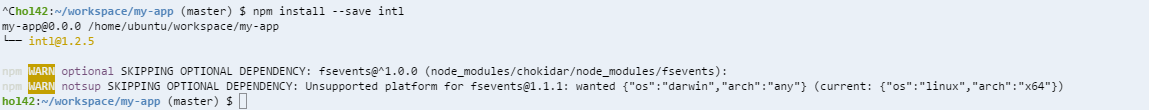


Editieren von pollyfill.ts

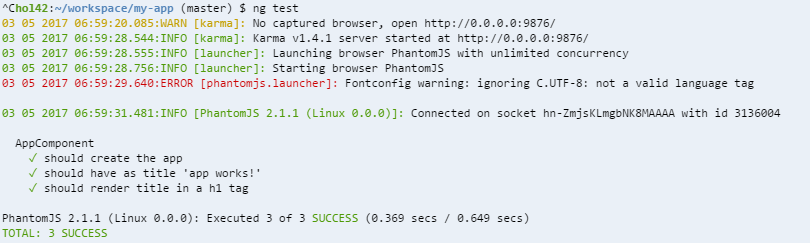




und



Jetzt funktioniert es:



13) Etwas ändern und beobachten

