If you haven't noticed: I already **published a plugin for the IntelliJ platform** of IDEs by **Jetbrains**. Developing for the IntelliJ platform is sometimes a bit hard, because I'm used to have a API-documentation for reference, but somehow Jetbrains didn't provide one for the platform.

Looking at their **sources over at github**, I found, that their code is actually quite well commented using **Javadoc** comments in most parts, so generating an API documentation shouldn't be *that* hard.

That's what I thought at least.

But, well, Javadoc's a bitch. It comes along with the JDK, so there's the first problem: What JDK? Oracle? OpenJDK? 8? 9? I started writing on a Mac with a recent Oracle JDK 8 and got everything running quite quickly. To automatically generate the doc every night, I thought that using **Travis** would be a great idea (hint: it is!). Travis, however, uses the same JDK, but on a linux agent, which somehow produced a different result.

To fix problems with different platforms, I wrapped all up in a docker container.

But a good I/O performance for a fairly large group of files using Docker for Mac? Yeah, sure.

Docker for Mac, which is using **HyperKit** for its Linux-Docker Host, is extremely bad when it comes to I/O performance (Docker really works best in an all-Linux environment). However, I wrapped something up using **docker-sync**, but found out, that it's really a niche solution for macOS. So, I ended writing a macOS-docker solution and a linux solution.

But an up-to-date Oracle JDK as a docker container? Yeah, sure.

Automating the installation of an Oracle JDK is *hell* and there's no official or even up-to-date Docker image available. So instead, I used the nice **OpenJDK image**. I used the Javadoc from that one, but couldn't get it to produce an output, when there's a library missing (the other javadoc simply skipped that missing library). So, finally, I ended up with telling the javadoc of OpenJDK 9 to use the old javadoc implementation using the -Xold

It works now and we have a **daily generated API doc** (see the **generator code over at github**), although I'm not quite sure how stable the system is. So, keep your fingers crossed.