



Hochschule für Technik,
Wirtschaft und Kultur Leipzig

FAKULTÄT INGENIEURWISSENSCHAFTEN

6010 - PRAXISPROJEKT

**Implementierung einer
OCPP-Backend-Testumgebung für
automatische Integrationstests einer DC
High Power Ladestationen der Plattform
Sicharge D**

Author Ivan Agibalov
Betreuer Prof. Dr.-Ing. Andreas Pretschner

6. Juni 2022

Inhaltsverzeichnis

Abbildungsverzeichnis	i
Tabellenverzeichnis	i
1 Introduction	1
2 Example Section	1
3 Gewünschtes Interfaces	2
4 Implementierung	3
4.1 Achitecture des Frameworks	3
4.1.1 Ports	3
4.1.2 Adapters	3
4.1.3 Controlllers	3
4.1.4 Dispatcher	4
4.1.5 UseCases	4
4.1.6 Interactors	4
4.1.7 Domain	4
4.2 Zugriff vom Testframework	4
4.3 Testbeispiel	5
5 Conclusion	5
Bibliography	6
Appendix	7
A Hello World Example	7
B Flow Chart Example	7
C Sub-figures Example	8

Abbildungsverzeichnis

1 Caption used in list of tables	1
2 Streamline results	8

Tabellenverzeichnis

1 Introduction

Hi! Welcome to this L^AT_EX-template. I will here aim to introduce you to, as well as motivate you to learn more about the features available in LaTeX through Overleaf. Many of the features you will come across in this template are not necessarily relevant to you at this point in time, and some will most likely seem way too advanced. However, keep in mind that you are not expected to understand everything at once either.

I hope that you, with the assistance of what I provide you with here, are able to make your own LaTeX-templates containing your personal preferences. You may do it by directly changing variables in this template, or you may create a brand new containing only carefully selected features of your own.

As a final note, I want to wish you the best of luck learning LaTeX, but do keep in mind that this template is only scratching the surface.

2 Example Section

In physics, the Navier–Stokes equations, named after Claude-Louis Navier and George Gabriel Stokes, describe the motion of viscous fluid substances.

(1) shows the incompressible Navier-Stokes equations using tensor notation.

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} \quad (1)$$



Abbildung 1: Caption written below figure.

Source: Insert image source here

3 Gewünschtes Interfaces

- 1. Erstellen des Servers mit gewünschten Netzwerkeinstellungen
- 2. Parametrieren/Festlegen des gewünschten Verhaltens des Servers
- 3. Server starten
- 4. Festlegen die Bedingungen für den erfolgreichen Test
- 5. Warten bis der Test abgeschlossen wird
- 6. Ergebnisse validieren
- 7. Alle Instanzen löschen

4 Implementierung

Es handelt sich um einen Integrationstests. Bei diesen Tests wird die Software komplett getestet, so wie sie dann im Betrieb ist. Aus diesem Grund, jedes Setup der Software wird Zeit in Anspruch nehmen und man soll das lieber soweit es möglich vermeiden. Es gibt mehrere Möglichkeiten dies zu vermeiden:

- mehrere Tests pro Setup zu definieren und auszuführen.
- die Tests die in unterschiedlichen Setups ablaufen (d.h. nicht miteinander verbunden) parallel laufen zu lassen.

Um die Einarbeitungszeit in das Testframework klein zu halten, wurde entschieden das OCPPServer Testframework basierend auf ein anderes Framework, das von dem Team bereits genutzt wird. Dieses Framework wird um die neuen Funktionalitäten erweitert und das restliche Verhalten soll genauso bleiben.

Um die Lesbarkeit des Tests zu verbessern wäre es vom Vorteil, wenn die erstellte OCPPServer Testinstanz bereits ein vordefiniertes Verhalten besitzt, das man ändern kann.

Es soll auch möglich sein das vordefinierte Verhalten zu parametrieren. Dies erfordert einen Interfaces, die bestimmte Parameter der Instanz ändern kann.

Da nur das Verhalten von dem Charging Point getestet werden soll, sollen nur die Ereignisse, die den Zustand des Charging Points abbilden, abrufbar sein. Zum Beispiel: geschickte Nachrichten von dem Charging Point zu dem Server, Reihenfolge der Nachrichten.

4.1 Achitecture des Frameworks

Es wurde entschieden das Framework in 7 Abstraktionsschichten aufzuteilen.

4.1.1 Ports

Ports haben die Aufgabe die Schnittstelle nach Außen aufzubauen und die Verbindungen zu (z.B. WebSocket Server, Datenbank)

In dem Framework wird nur ein Port gebraucht - WebSocket Server

4.1.2 Adapters

Adapters sollen die ankommenden Events/Messages vom Port an das dazugehörige Controller zu übersetzen.

In dem Framework wird nur einen Adapter gebraucht - OCPP16 Adapter

4.1.3 Controllers

Controllers besitzen alle Informationen die den Zustand des jeweiligen Components (Controller + Adapter + Port) abbilden.

In dem Framework werden mehrere Controllers gebraucht:

- OCPP Controller(übernimmt die Verantwortung über die Verbindungen zu den Charging Points)

-
- User Controller(übernimmt die Verantwortung über die Nutzer der Charging Points und ihrer Berechtigungen)
 - Transaction Controller(übernimmt die Verantwortung über die Kontroller über die Ladevorgängen)
 - Charger Controller(übernimmt die Verantwortung über die Charging Points, die von dem Server bekannt sind und ihrer Zuständen)
 - Payment Controller(übernimmt die Verantwortung über die Bezahlvorgang nach dem Ladevorgang)

Die Contoller können von dem Nutzer des Frameworks parametrisiert werden, um das Verhalten des Servers zu ändern.

4.1.4 Dispatcher

Dispatcher informiert alle abonnierten UseCases über das geschehene Event

In dem Framework sind nur OCPP Events wichtig (Nachrichten und Verbindungsevent)

4.1.5 UseCases

UseCases beschreiben den Vorgang beim geschehen eines Events, das von denen abonniert wurde. Die vordefinierten UseCases dürfen nur die Interactor benutzen um das Verhalten zu definieren.

In dem Framework sind die UseCases, die das vordefinierte Verhalten definieren und sie können auch entsprechend umbeschrieben werden.

4.1.6 Interactors

Eine atomare Operation im Programm(die Operation lässt sich nicht mehr sinnvoll im Rahmen der Anwendung aufteilen). Benutzt mittels "Dependency Injection" die Controller.

In dem Framework wird nur als "Wrapper" für alle Funktionen von Controllern implementiert.

4.1.7 Domain

Definiert alle Types und Interfaces der Applications. Definiert die Verbindungen zwischen den Interfaces und Types, die dann in den anderen 6 Layers umgesetzt werden.

4.2 Zugriff vom Testframework

- OCPPPort soll nur bei der Initialisierung der Instanz parametrierbar sein (Netzwerkeinstellung)
- Adapters sollen nicht von der Seite des Frameworks aufrufbar sein
- Contollers sollen nicht von der Seite des Frameworks aufrufbar sein
- Dispatcher darf nur zum Abonnieren/Disabonnieren benutzt werden um das Verhalten des Charging Points beobachten zu können
- UseCases sollen überschreibbar und erweiterbar sein, falls man bestimmtes Verhalten hinzufügen möchte.

-
- Interactors sollen von der Seite des Frameworks aufrufbar sein, um die Serverinstanz parametrieren zu können.
 - Domain beinhaltet alle Typen die in den anderen Layers benutzt werden. Aus diesem Grund sollen die Typen von der Seite des Frameworks benutzbar sein.

4.3 Testbeispiel

5 Conclusion

But the fact that some geniuses were laughed at does not imply that all who are laughed at are geniuses. They laughed at Columbus, they laughed at Fulton, they laughed at the Wright Brothers. But they also laughed at Bozo the Clown - Sagan (1993).

Bibliography

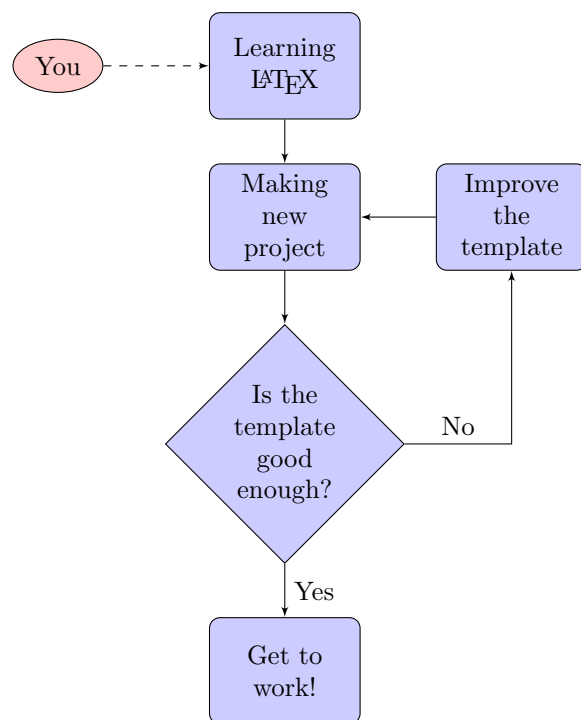
- Ghia, U., K. N. Ghia und C. T. Shin (1982). „High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method“. In: *Journal of Computational Physics* 48, S. 387–411.
- NTNU, Department of Marine Technology (2020). *IMT Software Wiki - LaTeX*. URL: <https://www.ntnu.no/wiki/display/imtsoftware/LaTeX> (besucht am 15. Sep. 2020).
- Sagan, Carl (1993). *Brocas brain: reflections on the romance of science*. Presidio Press.

Appendix

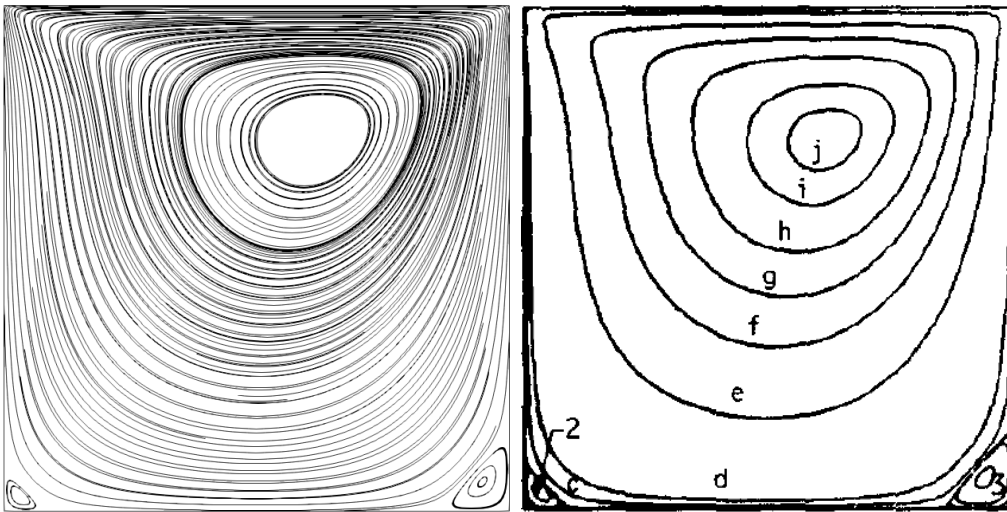
A Hello World Example

⋮
⋮
⋮

B Flow Chart Example

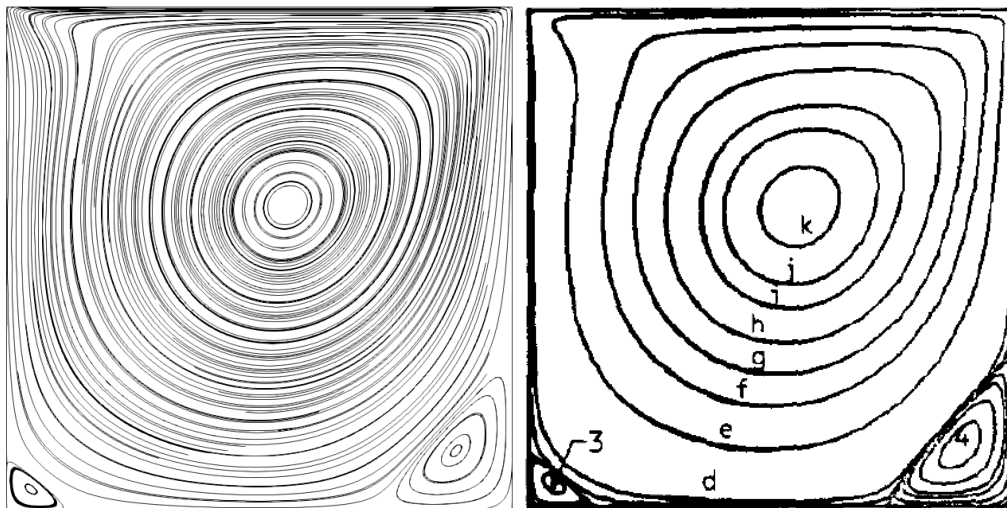


C Sub-figures Example



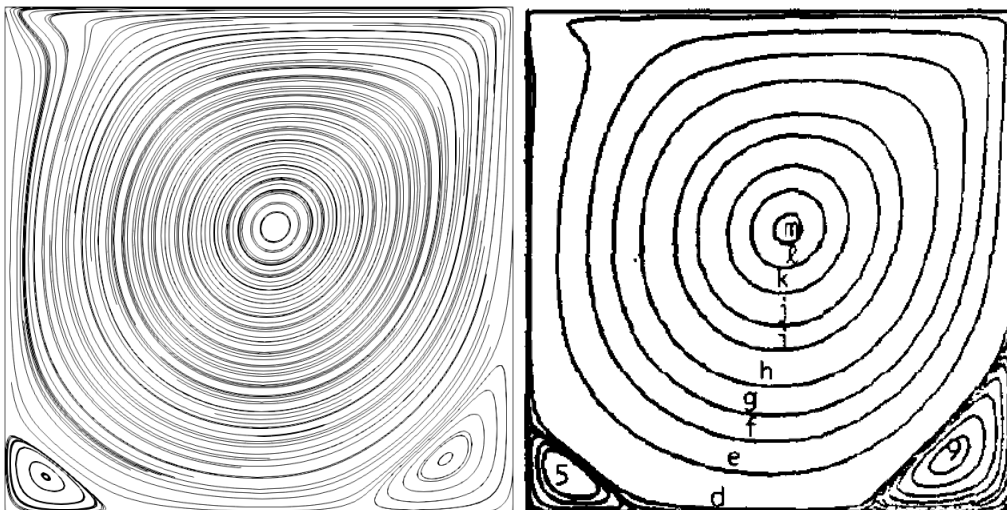
$Re = 100$ - Exercise

$Re = 100$ (Ghia u. a. 1982)



$Re = 400$ - Exercise

$Re = 400$ (Ghia u. a. 1982)



$Re = 1000$ - Exercise

$Re = 1000$ (Ghia u. a. 1982)

Abbildung 2: Streamlines for the problem of a lid-driven cavity.