Implementierung MatFlow Workflowanwendung für Machine Learning Experimente

Florian Küfner, Soeren Raymond, Alessandro Santospirito,
Lukas Wilhelm, Nils Wolters

Januar 2021

Inhaltsverzeichnis

1 Einleitung

Lorem ispum

2 Aufteilung

2.1 Geplante Aufteilung

Im Folgenden ist eine Grafik zu sehen, die einen Überblick über die Implementierung des MatFlow Systems gibt. Eine detailliertere Ansicht über die Aufgabenteile findet man im Anhang.

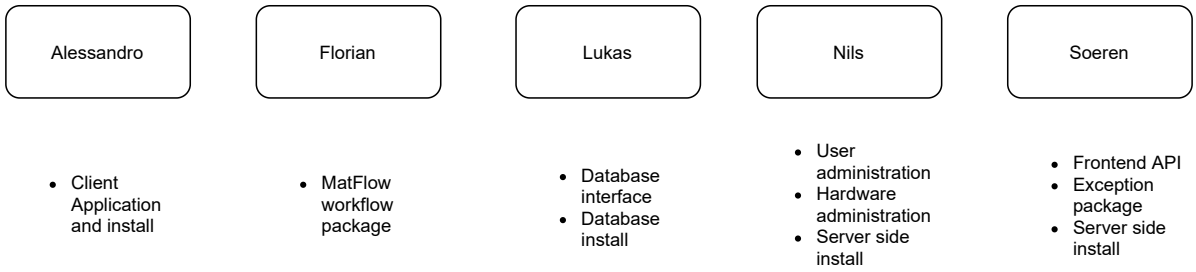

Abbildung 1: Aufteilung

2.2 Endgültiger Stand nach Implementierung

Frontend

- **Jede** der anzuwählenden Seiten ist implementiert
- Alle benötigten Datenklassen sind voll funktionsfähig implementiert

Frotend API

- **Jede** der zur Verfügung stehenden Methoden ist bis auf die Authentifizierung voll funktionsfähig implementiert
- Alle Unit Tests sind geschrieben

User Administration

- **Jede** der zur Verfügung stehenden Methoden ist voll funktionsfähig implementiert
- Alle Unit Tests sind geschrieben

Hardware Administration

- **Jede** der zur Verfügung stehenden Methoden ist voll funktionsfähig implementiert
- Alle Unit Tests sind geschrieben

Workflow Package

- **Jede** der zur Verfügung stehenden Methoden ist voll funktionsfähig implementiert
- Alle Unit Tests sind geschrieben

Exception package

- **Jede** der zur Verfügung stehenden Methoden ist voll funktionsfähig implementiert

Database package

- Der Großteil der zur Verfügung stehenden Methoden ist voll funktionsfähig implementiert
- Alle Unit Tests sind geschrieben

Zu implementieren bleibt noch

- Die request Klasse BackendCommunicator und Unit Tests im Frontend
- Die WorkflowData Schnittstelle im Database package
- Der alternierende Operator für airflow
- Die containerweise Ausführung der Serverapplikation im Backend (Nomad Job File)

3 Entwurfsumentscheidungen

3.1 JSON Converter

3.1.1 Frontend

bla

3.1.2 API

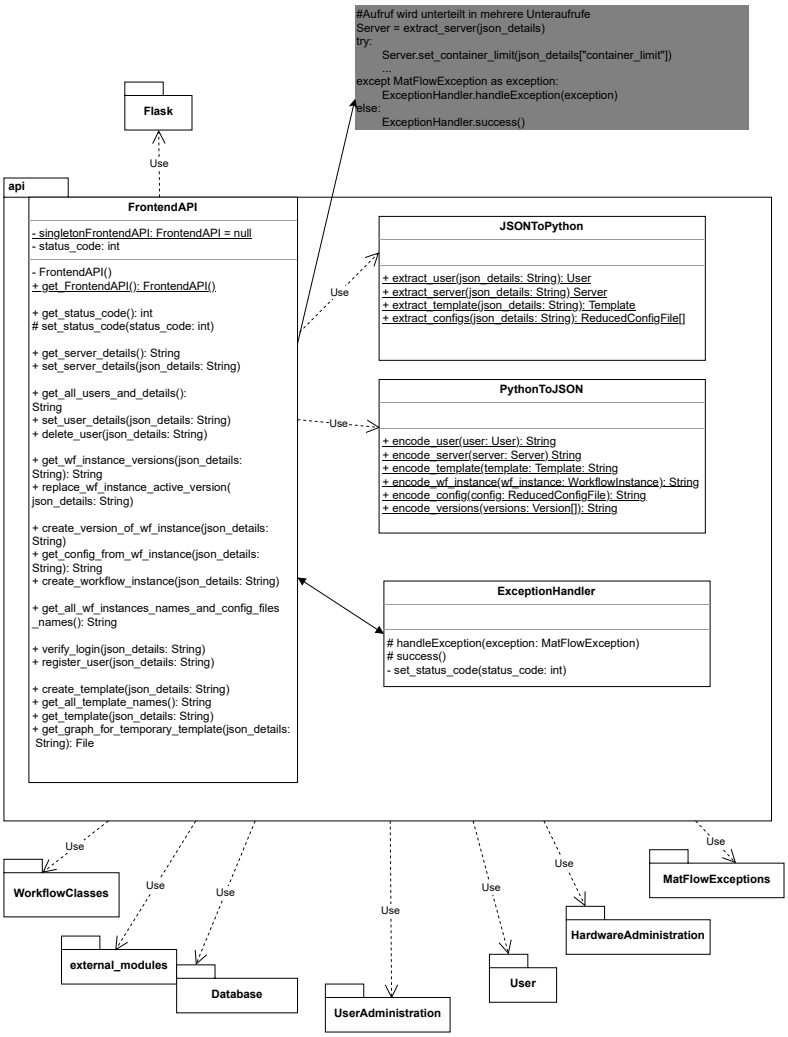

Abbildung 2: API package

Im Backend sind auch analog zum Frontend zwei JSON Converter(jewils von und zu) vorhanden. Diese Entwurfsentscheidung wurde aber verworfen, da sie das Kapselungsprinzip verletzt. Von nun an sind alle in den Converter vorhandenen Methoden direkt als Klassen- (von JSON) oder Objektmethoden (zu JSON) direkt in den jeweiligen Klassen implementiert. Als Beispiel: encode_template(template: Template) und extract_template(json_details: String): Template sind nun als encode_template(self): str und classmethod extract_template: Template in Template vorhanden.

3.2 JSON Status Code

Wie man auch in der API sieht, ist die Bereitstellung eines Status Codes bei einem request an die API nicht eindeutig dem Entwurf zu entnehmen, deswegen hier noch einmal eine saubere Erläuterung: Es wird bei einer Anfrage an die API diese Anfrage durchgeführt. Je nachdem ob sie nicht erfolgreich war (MatFlowException geworfen, spezieller Status Code vorhanden) oder doch erfolgreich war (success Status Code und eventuell Daten vorhanden) wird in der Klasse ExceptionHandler die Antwort in json gebaut und an den Client zurück geschickt. Der Client bekommt also **immer** eine Antwort mit einem json response body, indem sich der status code befindet. Es handelt sich hier also nur um eine Anlehnung an die bereits etablierten HTTP Status Codes.

3.3 Authentifizierung

Im Frontend kann man theoretisch unabhängig der User Privilegien mittels URL Manipulation auf alle Seiten zugreifen. Um dies zu verhindern, wird im Frontend ein einzigartiges Cookie mit der Anfrage an die Frontend API geschickt, die dann ??

4 Anhang

4.1 Detailliertere Aufteilung

Man sieht im Folgenden die exportierten Issues aus Gitlab:

Issue ID	URL	Title	State	Description	Author	Author Username	Assignee	Assignee Username	Confidential	Locked	Due Date	Created At (UTC)	Updated At (UTC)	Closed At (UTC)	Milestone	Weight	Labels	Time Estimate	Time Spent	Epic ID	Epic Title
1	https://git.scc.kit.e	Exceptions schreiben	Closed	Alle WaitFlowExceptions, die im System geworfen werden sollen	udfnb	udfnb	udfnb	udfnb	No	No		2022-01-20	2022-01-19 14:39	2022-01-20 09:39	2022-01-20 09:39			0	0		
2	https://git.scc.kit.e	FrontendAPI	Closed	Jegliche Kommunikation zwischen Client Application und Server	udfnb	udfnb	udfnb	udfnb	No	No		2022-01-20	2022-01-19 14:39	2022-01-20 17:32	2022-01-20 17:32			0	0		
3	https://git.scc.kit.e	Get FrontendAPI up and running	Open	API mit Flask auf Server laufen lassen	udfnb	udfnb	udfnb	udfnb	No	No		2022-01-20	2022-01-19 14:39	2022-01-20 17:32				0	0		
4	https://git.scc.kit.e	MySQL auf Server laufen lassen	Open		udfnb	udfnb	udfnb, ulfth	udfnb, ulfth	No	No		2022-01-23	2022-01-19 14:40	2022-01-20 17:31				0	0		
5	https://git.scc.kit.e	Backend App	Open	Backend Application verpackert (u.a. mit Docker, Nomad)	udfnb	udfnb	udfnb, ulfth	udfnb, ulfth	No	No		2022-02-01	2022-01-19 14:40	2022-01-20 17:32				0	0		
6	https://git.scc.kit.e	IMBO: getVersionsFromWorkflowInstance() - and setActiveVersionThroughNumber()-method in WorkflowManager	Open	- in both cases forward the request to the database	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-19 16:40	2022-01-20 17:31				0	0		
7	https://git.scc.kit.e	backend functionality for creating templates /M10/	Closed	Methods - createTemplate() - getDiagRepresentationFromTemplate() in class WorkflowManager - implementation of the class Template	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-20	2022-01-17 10:26	2022-01-20 17:34	2022-01-20 10:18			0	0		
8	https://git.scc.kit.e	IMBO: createWorkflowInstanceFromTemplate()-method in WorkflowManager	Open	- arrange the given input inside a WorkflowInstance object - 'forward' request to the database	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-17 10:32	2022-01-20 17:31				0	0		
9	https://git.scc.kit.e	IFAM3: operator for tasks alternating dependencies (used for /gds-models)	Open		uczrb	uczrb	uczrb	uczrb	No	No		2022-02-01	2022-01-17 10:42	2022-01-20 17:31				0	0		
10	https://git.scc.kit.e	Install vuje and dependencies	Closed	Installing: vuejs, vuetify, vuex	ulrjha	ulrjha	ulrjha, ulfth	ulrjha, ulfth	No	No		2022-01-12	2022-01-17 10:36	2022-01-20 17:32	2022-01-17 16:38			0	0		
11	https://git.scc.kit.e	Connection between MySQL and python Code	Closed	maybe with mysql connector	upqzk	upqzk	upqzk	upqzk	No	No		2022-01-20	2022-01-17 10:36	2022-01-20 17:32	2022-01-20 12:13			0	0		
12	https://git.scc.kit.e	Dummy-Data for frontend	Closed	In order to design the frontend features it's necessary to create dummy data that are placeholder for the backend-data	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-01-20	2022-01-17 10:36	2022-01-20 17:32	2022-01-17 18:26			0	0		
13	https://git.scc.kit.e	Frontend: connect BackendServerCommunicator with dummy-data	Closed	Connect the data from folder 'DummyData' with the corresponding functions in class "BackendServerCommunicator"	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-01-20	2022-01-17 10:36	2022-01-20 17:32	2022-01-17 18:38			0	0		
14	https://git.scc.kit.e	Frontend: integrate vues in the 'Model'-folder	Closed	Vues has to be integrated in every Class of 'Model'-folder	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-01-20	2022-01-17 10:36	2022-01-20 17:32	2022-01-18 10:2			0	0		
15	https://git.scc.kit.e	frontend: integrate Text editing	Open	The dag-definition file should be editable in the frontend-application	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-02-01	2022-01-19 12:46	2022-01-20 17:31				0	0		
16	https://git.scc.kit.e	Frontend: File transfer and integration	Open	Eventhough most classes are implemented and dummy-data are created by now, there is no integration of files. The dag-definition file has to be integrated for template creation.	ulrjha	ulrjha	ulrjha, ulfth	ulrjha, ulfth	No	No		2022-02-01	2022-01-19 12:46	2022-01-20 17:32				0	0		
17	https://git.scc.kit.e	frontend: Design frontend with vuetify	Open	The frontend application is designed with vuetify. The files containing vuetify are not implemented	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-02-01	2022-01-19 12:30	2022-01-20 17:31				0	0		
18	https://git.scc.kit.e	frontend: Messages	Open	For most requests done in the frontend-application an approval, error, connection loss message should appear	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-02-01	2022-01-19 12:31	2022-01-20 17:31				0	0		
19	https://git.scc.kit.e	frontend: Connect frontend to the frontend-api	Open	frontend-data is received from the frontend-api	ulrjha	ulrjha	ulrjha, ulfth	ulrjha, ulfth	No	No		2022-02-01	2022-01-19 12:32	2022-01-20 17:31				0	0		
20	https://git.scc.kit.e	frontend: JSON-typescript and typescript-JSON converter implementation	Open	Implementing the JSON-typescript, typescript-JSON converter	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-02-01	2022-01-19 12:33	2022-01-20 17:31				0	0		
21	https://git.scc.kit.e	frontend: Make vue with typescript compatible	Closed	Ts is not supported in vue3 by default, but a requirement from the university is to use typescript. The files in folder classes and model are all made with typescript	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-01-20	2022-01-19 16:54	2022-01-20 17:31	2022-01-19 17:3			0	0		
22	https://git.scc.kit.e	Frontend: Remove store component and disable-eslint0rError	Closed		ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-01-20	2022-01-19 16:53	2022-01-20 17:31	2022-01-19 16:5			0	0		
23	https://git.scc.kit.e	frontend: fix issue with 'Material Icons'	Closed	vuetify can't use 'Material Icons' at the moment	ulrjha	ulrjha	ulrjha	ulrjha	No	No		2022-01-20	2022-01-19 17:46	2022-01-20 17:31	2022-01-19 18:3			0	0		
24	https://git.scc.kit.e	FrontendAPI	Closed	Hardware calls	udfnb	udfnb	udfnb	udfnb	No	No		2022-01-20	2022-01-20 08:31	2022-01-20 17:32	2022-01-20 08:3			0	0		
25	https://git.scc.kit.e	FrontendAPI	Closed	User calls	udfnb	udfnb	udfnb	udfnb	No	No		2022-01-20	2022-01-20 09:31	2022-01-20 17:32	2022-01-20 09:3			0	0		
26	https://git.scc.kit.e	FrontendAPI	Closed	Workflow Calls	udfnb	udfnb			No	No		2022-01-20	2022-01-20 09:31	2022-01-20 17:32	2022-01-20 09:3			0	0		
27	https://git.scc.kit.e	IM10: Implementation of class Template	Open	- file management via paths - ensure that no templates are created from invalid dag-definition-files	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-20 08:52	2022-01-20 17:37				0	0		
28	https://git.scc.kit.e	IM10: createTemplate()-method in WorkflowManager	Open	- forwards request to the Airflow system by creating a temporary dag	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-20 08:52	2022-01-20 17:28				0	0		
29	https://git.scc.kit.e	IM10': getDagRepresentationFromTemplate()-method in WorkflowManager	Open	- extract an return the given image	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-20 08:52	2022-01-20 17:28				0	0		
30	https://git.scc.kit.e	IM20': getTemplateNames() - and getTemplateFromName()-methods in WorkflowManager	Open		uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-20 10:52	2022-01-20 17:28				0	0		
31	https://git.scc.kit.e	IM20': implementation of the class WorkflowInstance	Open	- mainly the functionality inside the constructor of the class	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-20 10:04	2022-01-20 17:28				0	0		
32	https://git.scc.kit.e	IM40': implementation of classes ReducedConfigFile and ConfigFile	Closed		uczrb	uczrb	uczrb	uczrb	No	No		2022-01-20	2022-01-20 10:12	2022-01-20 17:32	2022-01-20 10:2			0	0		
33	https://git.scc.kit.e	IM50': implementation of class Version and Subclasses	Open	subclasses are - FrontendVersion - DatabaseVersion - ...	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-20 10:12	2022-01-20 17:28				0	0		
34	https://git.scc.kit.e	IM60': getNamesOfWorkflowsAndConfigFiles()- and getKeyValuePairsFromConfigFile()-method in WorkflowManager	Open	- forwarding request to database - bring results in the right format	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-20 10:13	2022-01-20 17:28				0	0		
35	https://git.scc.kit.e	IM60': createNewVersionOfWorkflowInstance()-method in WorkflowManager	Open	- write the given changes into actual files - build version object that is accepted by the database interface	uczrb	uczrb	uczrb	uczrb	No	No		2022-01-24	2022-01-20 10:20	2022-01-20 17:28				0	0		
36	https://git.scc.kit.e	Set file saving method for database	Open		upqzk	upqzk	upqzk	upqzk	No	No		2022-01-25	2022-01-20 10:12	2022-01-20 17:28				0	0		
37	https://git.scc.kit.e	Database package implementation	Open	Functions of DatabaseTable first, other classes later	upqzk	upqzk	upqzk	upqzk	No	No		2022-01-25	2022-01-20 10:14	2022-01-20 17:28				0	0		
38	https://git.scc.kit.e	MySQL database table setups	Open	mysql queries for one-time setup upon starting without the established tables	upqzk	upqzk	upqzk	upqzk	No	No		2022-01-25	2022-01-20 12:11	2022-01-20 17:32				0	0		
39	https://git.scc.kit.e	Database package tests	Open	Unclear if TDD is best approach	upqzk	upqzk	upqzk	upqzk	No	No		2022-02-01	2022-01-21 08:12	2022-01-21 08:12				0	0		
40	https://git.scc.kit.e	creating Hardware- und User-Administration package	Open		ulfth	ulfth	ulfth	ulfth	No	No		2022-01-20	2022-01-20 10:20	2022-01-20 17:22				0	0		
41	https://git.scc.kit.e	implementing User and UserController class	Open		ulfth	ulfth	ulfth	ulfth	No	No		2022-02-01	2022-01-20 10:20	2022-01-20 17:22				0	0		
42	https://git.scc.kit.e	User-Administration airflow rest api calls implemented	Open		ulfth	ulfth	ulfth	ulfth	No	No		2022-02-01	2022-01-20 10:21	2022-01-20 17:19				0	0		
43	https://git.scc.kit.e	implementing User-Administration Unittests	Open		ulfth	ulfth	ulfth	ulfth	No	No		2022-02-01	2022-01-20 10:21	2022-01-20 17:11				0	0		
44	https://git.scc.kit.e	Server and ServerController class implementation	Open		ulfth	ulfth	ulfth	ulfth	No	No		2022-01-23	2022-01-20 10:21	2022-01-20 17:19				0	0		
45	https://git.scc.kit.e	implementing Hardware-Administration Unittests	Open		ulfth	ulfth	ulfth	ulfth	No	No		2022-02-01	2022-01-20 10:21	2022-01-20 17:11				0	0		

Abbildung 3: Aufteilung