

Entwurf MatFlow Workflowanwendung für Machine Learning Experimente

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

Dezember 2021

Inhaltsverzeichnis

1	Einleitung	3
2	Systemüberblick	4
3	Kommunikation	5
4	Komponentenbeschreibung	6
4.1	Client	6
4.2	Server	7
5	Paketbeschreibung	8
5.1	Client Application	8
5.2	Server Application	8
6	Abläufe	9
7	Datenbank	10
7.1	Datenbankaufbau	10
7.2	Database Package	11

1 Einleitung

2 Systemüberblick

3 Kommunikation

4 Komponentenbeschreibung

4.1 Client

4.2 Server

5 Paketbeschreibung

5.1 Client Application

5.2 Server Application

5.2.1 Workflow Package

Class: Version Kommentar zur Klasse TODO

Constructor

Version(versionNumber:VersionNumber, note:String):Version

Parameters

- versionNumber: Number that identifies the new Version
- note: TODO

5.2.2 API Schnittstelle

Class: FrontendAPI Diese Klasse ist der Dreh- und Angelpunkt der ganzen Anwendung. Alle Anfragen von der Client Applikation werden an diese API, entworfen im Bequemlichkeitsmuster der **Fassade**, (im JSON Format) geschickt. Die Entscheidung FrontendAPI als **Singleton** zu entwerfen besteht darin, dass die API eine global ansprechbare Instanz sein soll, die auf dem Server an einem bestimmten Port laufen soll. Wie dies geschieht sind Implementierungsdetails, bei denen das eingebundene Framework **Flask** hilft, muss aber in die Entwurfsentscheidung einbezogen werden.

Constructor

Version(versionNumber:VersionNumber, note:String):Version

Parameters

- versionNumber: Number that identifies the new Version
- note: TODO

6 Abläufe

7 Datenbank

7.1 Datenbankaufbau

Die Datenbank beinhaltet folgende Tabellen:

Server		
name	String	<i>key; notNull</i>
adress	String	<i>notNull</i>
containerlimit	int	<i>notNull</i>

Workflow_Template		
name	String	<i>key; notNull</i>
dag	.py -File	<i>notNull</i>

Workflow		
name	String	<i>key; notNull</i>
files	String	<i>notNull</i>
dag	.py -File	<i>notNull</i>

Version		
wfName	String	<i>key; notNull; name from Workflow</i>
version	String	<i>key; notNull</i>

ActiveVersion		
wfName	String	<i>key; notNull; name from Workflow</i>
version	String	<i>notNull; from Version</i>

VersionFile		
wfName	String	<i>key; notNull; name from Workflow</i>
version	String	<i>key; notNull; from Version</i>
confKey	String	<i>key; notNull; from ConfFiles</i>

ConfFile		
confKey	String	<i>key; notNull</i>
file	.conf -File	<i>notNull</i>

Note While the dag-file could be used as a key between Workflow_Template and Workflow, it is not efficient to traverse over. On creation of a new Workflow, the based Template dag-file will be copied.

7.2 Database Package

Hier Package png und Erklärung

7.2.1 Class.DatabaseTable

DatabaseTable ist die einzige Klasse die direkt mit der Datenbank kommuniziert. Ihre Funktion ist hauptsächlich MySQL Befehle entgegennimmt, diese an die Datenbank weitergibt und die dadurch erhaltenen Antworten zurückgibt.

set(create: String):String Erstelle neuen Eintrag. Versuche *create* auf MySQL auszuführen und gib die Antwort als String zurück.

delete(del: String):String Lösche einen existierenden Eintrag. Versuche *del* auf MySQL auszuführen und gib die Antwort als String zurück.

modify(change: String):String Ändere einen existierenden Eintrag. Versuche *change* auf MySQL auszuführen und gib die Antwort als String zurück.