

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
5.3	API Schnittstelle	8
5.4	Database	8
6	Abläufe	11
7	Datenbank	12
7.1	Datenbankaufbau	12

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

5.3 API Schnittstelle

5.4 Database

5.4.1 WorkflowData

Diese Klasse ist die Schnittstelle für alle Klassen die auf den Datenbankeinträgen für Workflows arbeiten wollen. Sie erstellt für jeden Zugriff den passenden Datenbankbefehl und konvertiert die Ergebnisse in das gewünschte Format des Aufrufers. Sie wirft bei fehlerhaften Werten eine **MatFlowException**.

createWorkflowInstanceFromTemplate(tName: String, wfName: String, confFold: Folder):void Create a new instance of a workflow by using the dag-file of a Template, with the Version set to 1. Search confDirect for every .conf-file and add them into the Database. Set this Version as active Version.

- **tName** name of Template of which the dag should be taken from
- **wfName** name of the new Workflow
- **confFold** the folder where all data of the Workflow is saved into

getNamesOfWorkflowsAndConfigFiles():String[][] Return all Workflow names and the names of their corresponding config files. The returning value is a list of lists of Strings where an inner list has the form [<Workflow Name>, <config File1 Name>, <config File2 Name>, ...]

createNewVersionOfWorkflowInstance(wfName:String, newVersion:DatabaseVersion, oldVersionNr:String):void Create a new Version of an existing Workflow with changed config Files. Search which config Files changed and set those in the new Version.

- **wfName** name of the Workflow
- **newVersion** new Version identifier
- **oldVersionNr** identifier of Version the new one is based on

getDatabaseVersionsOfWorkflowInstance(wfName:String):DatabaseVersion[] Return all Versions of a Workflow Instance as DatabaseVersion Objects in a list. Asks the Database for the data and constructs every Object.

- **wfName** name of the Workflow

setActiveVersionThroughNumber(wfName:String, version:String):void Set the previous Version as inactive and a new Version as active

- **wfName** name of the Workflow
- **version** version to be set active

getVersionNumbersOfWorkflowInstance(wfName:String):String[] Return a sorted list of Strings of all Versions of a Workflow.

- **wfName** name of Workflow which versions should be listed

5.4.2 TemplateData

Diese Klasse ist die Schnittstelle für alle Klassen die auf den Datenbankeinträgen für Workflow_Template arbeiten wollen. Sie erstellt für jeden Zugriff den passenden Datenbankbefehl und konvertiert die Ergebnisse in das gewünschte Format des Aufrufers. Sie wirft bei fehlerhaften Werten eine **MatFlowException**.

createTemplate(template:Template):void Read Values from template and convert them into a sql query. Throw error if name already exists.

- **template** Template object to convert

getTemplateNames():String[] Return a list of all existing Template names. List is empty when no Template exists.

getTemplateByNames(name:String):Template Return Template that is asked for. Throw error if Template doesn't exist.

- **name** name/identifier of the wanted Template

5.4.3 DatabaseTable

DatabaseTable ist die einzige Klasse die direkt mit der Datenbank kommuniziert. Ihre Funktion ist hauptsächlich MySQL Befehle entgegennimmt, diese an die Datenbank weiterzugeben und die dadurch erhaltenen Antworten zurückgibt. Im Fall eines Fehlers, wie einem leeren Ergebnis oder eines ungültigen Befehls wird eine **MatFlowException** geworfen.

set(create: String):String Create new entry on Database. Try to execute create on Database and return value.

- **create** a sql query to create an entry

delete(del: String):String Delete an existing entry. Try to execute del on Database and return value.

- **del** a sql query to delete an entry

modify(change: String):String Modify an existing entry. Try to execute change on Database and return value.

- **change** a sql query to change values

6 Abläufe

7 Datenbank

7.1 Datenbankaufbau

Die Datenbank beinhaltet folgende Tabellen:

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>
note	String	

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>

Notiz Die dag Datei hätte als Schlüssel zwischen Workflow_Template und Workflow fungieren können, ist aber redundant für alle Operationen auf einem Workflow. Deshalb wurde entschieden die Datei bei einer Workflowerstellung direkt zu kopieren.