

Qualitätssicherung

4. April 2017

Inhaltsverzeichnis

1	Einleitung	2
2	Informationen über Tests	3
2.1	Unit-Tests	3
2.2	Integrations-Tests	4
3	Coverage	5
4	Testfälle	6
5	Gefundene und behobene Fehler	8
5.1	Probleme mit Parameterübergabe beim Starten einer Instanz	8
5.1.1	Symptom	8
5.1.2	Fehler	8
5.1.3	Behebung	8
5.2	Nicht-Berücksichtigung der Anzahl Ausführungen	8
5.2.1	Symptom	8
5.2.2	Fehler	8
5.2.3	Behebung	8
5.3	Falsche Erkennung des Tokens	8
5.3.1	Symptom	8
5.3.2	Fehler	9
5.3.3	Behebung	9
5.4	Fehlende Fehlermeldungen	9
5.4.1	Symptom	9
5.4.2	Fehler	9
5.4.3	Behebung	9
6	Zusammenfassung	10

Kapitel 1

Einleitung

In diesem Dokument wird die Qualitätssicherungsphase des Projekts Serverless Server for Industry 4.0 dokumentiert. Im folgenden werden die ausgeführten Tests vorgestellt, sowie die Testabdeckung und Vergleich mit dem geplanten Testablauf aus dem Pflichtenheft. Anschließend werden die entdeckten Fehler dokumentiert.

Kapitel 2

Informationen über Tests

2.1 Unit-Tests

Datei	Anzahl
RestApiControllerTest	9
AuthFacadeImplTest	3
AuthTest	6
ContentValidatorTest	2
SubtokenCreatorTest	1
TokenCreatorTest	5
RequestServerConverterTest	7
RuntimeTestNegative	12
RuntimeTestPositive	8
LambdaManagerFacadeImplTest	7
Gesamt	60

2.2 Integrations-Tests

Datei	Anzahl
testAuthNegative401	6
testAuthPositive	7
testDeleteLambdaPositive	2
testDeleteLambdaNegative	1
testExecuteLambdaPositive	8
testExecuteLambdaNegative	2
testGetLambdaPositive	1
testGetLambdaNegative	1
testUpdateLambdaPositive	13
testUpdateLambdaNegative	4
testUploadLambdaPositive	2
testUploadLambdaNegative	13
Gesamt	58

Kapitel 3

Coverage

Modul	Coverage
apicontroller	54%
auth	89%
model.converter	80%
model.exception	66%
model.lambda	73%
model.messages	79%
model.servicelayer.lambdaruntime	93%
model.servicelayer.service	35%
Gesamt	74%

Ungetestet blieben einige Getter- und Setter-Methoden sowie Konstruktoren. Dies betrifft vor allem die Module lambda und messages. Einige Codezeilen konnten durch automatische Tests nicht abgedeckt werden (zum Beispiel Verbindungsfehler zur Runtime-Umgebung). Diese Fälle wurden aber ausreichend manuell getestet. Auch gibt es nicht getestete ungenutzte Methoden, die für zukünftige Erweiterungen nicht gelöscht wurden.

Kapitel 4

Testfälle

Nr.	Beschreibung	Erfüllt
/T010/	Lambda-Entwickler lädt eine Lambda-Funktion hoch und bekommt ein Zugriffs-Token zurück.	JA
/T020/	Lambda-Entwickler stellt die Ausführungsumgebung für die Lambda-Funktion bereit.	JA
/T030/	Lambda-Entwickler gibt Token aus für Lambda-Verwender.	JA
/T040/	Lambda-Entwickler authentifiziert sich.	JA
/T050/	Lambda-Entwickler löscht seine Lambda-Funktion.	JA
/T051/	Lambda-Entwickler ändert seine Lambda-Funktion.	JA
/T052/	Lambda-Entwickler benennt seine Lambda-Funktion.	JA
/T053/	Lambda-Entwickler konfiguriert seine Lambda-Funktion.	JA
/T054/	Lambda-Entwickler bindet Daten für seine Lambda-Funktion ein.	JA
/T060/	Lambda-Verwender führt eine Lambda-Funktion mit einem Token aus.	JA
/T061/	Lambda-Verwender gibt an, wie viel mal die Lambda-Funktion ausgeführt werden muss.	JA
/T070/	Gleichzeitiger Zugriff auf eine hochgeladene Lambda-Funktion von mehreren Rechnern aus.	JA (manuell getestet)
/T080/	Terminierung von einer Lambda-Funktion.	JA

/T090/	Eine Lambda-Funktion, die länger läuft als vom Server angegeben, wird terminiert.	JA
/UT010/	Lambda-Verwender ruft eine nicht-existierende Lambda-Funktion auf.	JA
/UT020/	Lambda-Verwender ruft eine Lambda-Funktion mit einem falschen Token aus.	JA
/UT030/	Lambda-Entwickler gibt einen zu langen Zeitintervall für Funktionsausführung an.	Nicht implementiert
/WT070/	Lambda-Entwickler kauft die Token für seine Lambda-Funktion.	? (nur wenn bitcoins implementiert)

Kapitel 5

Gefundene und behobene Fehler

5.1 Probleme mit Parameterübergabe beim Starten einer Instanz

5.1.1 Symptom

Parameter wurden beim Start einer Lambda-Instanz nicht übergeben.

5.1.2 Fehler

Falsches Befehlsformat der Datei zur Imageerzeugung (Dockerfile)

5.1.3 Behebung

Nun wird das korrekte Befehlsformat verwendet.

5.2 Nicht-Berücksichtigung der Anzahl Ausführungen

5.2.1 Symptom

Der Parameter "times" bei Execute-Anfragen wurde ignoriert. Die Funktion wurde immer genau einmal ausgeführt.

5.2.2 Fehler

Der Parameter wurde im Modul Runtime nicht berücksichtigt.

5.2.3 Behebung

Der Parameter wird nun von der Runtime unterstützt und das Lambda die korrekte Anzahl oft ausgeführt.

5.3 Falsche Erkennung des Tokens

5.3.1 Symptom

Gültige Tokens wurden nicht als solche erkannt, und als Folge haben alle Methoden nicht funktioniert, die ein Token als Parameter entgegengenommen haben.

5.3.2 Fehler

In dem TokenCreator war der secret key falsch kodiert, und die Methode parseToken hat eine Invalid Signature Exception geworfen.

5.3.3 Behebung

Der secret key wird vor der Verwendung als UTF-8 dekodiert (`secret.getBytes("UTF-8")`).

5.4 Fehlende Fehlermeldungen

5.4.1 Symptom

Es wurden die Namen der Fehlerklassen statt einer Fehlermeldung bei falschen Eingaben an den Benutzer gesendet.

5.4.2 Fehler

Fehlermeldungen zu den betreffenden Fehlerklassen waren im System nicht vorhanden oder nicht mit den Fehlerklassen verbunden.

5.4.3 Behebung

Entsprechende Fehlermeldungen für die auftretenden Fehler wurden hinzugefügt. Der Benutzer des Systems erhält nun bei falschen Eingaben korrekte Fehlermeldungen.

Kapitel 6

Zusammenfassung

Alle Testfälle und Szenarios für die implementierten Funktionalitäten konnten erfolgreich durchgeführt werden. Modultests wurden mithilfe von JUnit und Mockito Framework durchgeführt, die Integrations- und Belastungstests wurden in Python geschrieben und teilweise manuell ausgeführt. Dabei konnten aufgetretene Fehler erfolgreich behoben werden. Die bei der Qualitätssicherung erreichte Testabdeckung weist auf ein robustes und stabiles Programmverhältnis hin. Dementsprechend kann die Qualitätssicherungsphase erfolgreich abgeschlossen werden.