|  |
| --- |
| Elmer Lukas, Heidt Christina, Steiner Diego, Treichler Delia, Waltenspül Remo  26. Mai 2011 |

|  |
| --- |
| SE2 Projekt MRT |
| Test Dokumentation |
|  |

****

# Dokumentinformationen

## Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 19.05.2011 | 1.0 | Erste Version des Dokuments | TD |

## Inhaltsverzeichnis

[1 Dokumentinformationen 1](#_Toc294100118)

[1.1 Änderungsgeschichte 1](#_Toc294100119)

[1.2 Inhaltsverzeichnis 1](#_Toc294100120)

[1.3 Abbildungsverzeichnis 2](#_Toc294100121)

[2 Systemtests 3](#_Toc294100122)

[2.1 Bemerkungen 3](#_Toc294100123)

[2.2 Voraussetzungen 3](#_Toc294100124)

[2.3 Vorbereitungen 3](#_Toc294100125)

[2.3.1 Tests auf dem Client und dem Server 3](#_Toc294100126)

[2.3.2 Tests auf dem Server 3](#_Toc294100127)

[2.4 UC1 Stundeneintrag erfassen 4](#_Toc294100128)

[2.4.1 Testprotokoll 4](#_Toc294100129)

[2.5 UC2 CRUD Stundeneintrag 4](#_Toc294100130)

[2.5.1 Testprotokoll 4](#_Toc294100131)

[2.6 UC4 CRUD Benutzer 5](#_Toc294100132)

[2.6.1 Testprotokoll 5](#_Toc294100133)

[2.7 UC5 Benutzer authentifizieren 5](#_Toc294100134)

[2.7.1 Testprotokoll 5](#_Toc294100135)

[2.8 UC6 CRUD Kunde 5](#_Toc294100136)

[2.8.1 Testprotokoll 6](#_Toc294100137)

[2.9 UC7 CRUD Material 6](#_Toc294100138)

[2.9.1 Testprotokoll 6](#_Toc294100139)

[2.10 UC8 CRUD StundeneintragsTyp 6](#_Toc294100140)

[2.10.1 Testprotokoll 6](#_Toc294100141)

[2.11 UC9 CRUD Auftrag 7](#_Toc294100142)

[2.11.1 Testprotokoll 7](#_Toc294100143)

[3 Unit Tests 8](#_Toc294100144)

[3.1 Client 8](#_Toc294100145)

[3.1.1 Gesamtübersicht Tests 8](#_Toc294100146)

[3.1.2 Testklasse MRTApplicationTest 8](#_Toc294100147)

[3.1.3 Testklasse HttpTestCase 9](#_Toc294100148)

[3.1.4 Testklasse CustomerTest 9](#_Toc294100149)

[3.1.5 Testklasse GpsPositionTest 9](#_Toc294100150)

[3.1.6 Testklasse UserTest 9](#_Toc294100151)

[3.1.7 Testklasse TimeEntryTypeTest 10](#_Toc294100152)

[3.1.8 Testklasse UserHelperTest 10](#_Toc294100153)

[3.1.9 Testklasse TimeEntryActivityTest 10](#_Toc294100154)

[3.1.10 Testklasse TimeEntryTest 11](#_Toc294100155)

[3.1.11 Testklasse LoginActivityTest 11](#_Toc294100156)

[3.1.12 Testklasse CustomerHelperTest 12](#_Toc294100157)

[3.1.13 Testklasse TimeEntryHelperTest 12](#_Toc294100158)

[3.1.14 Testklasse DatabaseHelperTest 13](#_Toc294100159)

[3.2 Server 14](#_Toc294100160)

[3.2.1 Gesamtübersicht Tests & Abdeckung 14](#_Toc294100161)

[3.2.2 TestklasseTimeEntriesControllerTest 14](#_Toc294100162)

[3.2.3 TestklasseTimeEntryTest 15](#_Toc294100163)

[4 Performance Test 15](#_Toc294100164)

## Abbildungsverzeichnis

[Abbildung 1 - Auflistung, Abdeckungsrate und Ergebnis der durchgeführten Tests für die Client Applikation 8](#_Toc294100206)

[Abbildung 2 - Testklasse MRTApplicationTest 8](#_Toc294100207)

[Abbildung 3 - Testklasse HttpTestCase 9](#_Toc294100208)

[Abbildung 4 - Testklasse CustomerTest 9](#_Toc294100209)

[Abbildung 5 - Testklasse GpsPostitionTest 9](#_Toc294100210)

[Abbildung 6 - Testklasse UserTest 10](#_Toc294100211)

[Abbildung 7 - Testklasse TimeEntryTypeTest 10](#_Toc294100212)

[Abbildung 8 - Testklasse UserHelperTest 10](#_Toc294100213)

[Abbildung 9 - Testklasse TimeEntryActivityTest 11](#_Toc294100214)

[Abbildung 10 - Testklasse TimeEntryTest 11](#_Toc294100215)

[Abbildung 11 - Testklasse LoginActivityTest 12](#_Toc294100216)

[Abbildung 12 - Testklasse CustomerHelperTest 12](#_Toc294100217)

[Abbildung 13 - Testklase TimeEntryHelperTest 13](#_Toc294100218)

[Abbildung 14 - Testklasse DatabaseHelperTest 13](#_Toc294100219)

[Abbildung 15 - Auflistung, Abdeckungsrate und Ergebnis der durchgeführten Tests für die Server Applikation 14](#_Toc294100220)

# Systemtests

## Bemerkungen

Der Use Case 3 Rapport Generieren wurde nicht umgesetzt. Daher wird er auch nicht getestet.

## Voraussetzungen

Die IP Adresse des Hosts wird für das korrekte funktionieren der Tests benötigt. Zudem ist eine funktionierende Netzwerkverbindung essentiell.

## Vorbereitungen

### Tests auf dem Client und dem Server

Eine Person meldet sich auf einem Computer über den Webbrowser an den Server an. Auf einem anderen Computer wird mit Hilfe eines Emulators der Client gestartet.

Dazu gehört folgender Test:

* 2.4 UC1 Stundeneintrag erfassen

### Tests auf dem Server

Eine Person meldet sich auf einem Computer über den Webbrowser an den Server an. Damit alle Tests erfolgreich ausgeführt werden können, muss zwingend ein Login vom Typ Sekretärin verwendet werden, da nur dann über alle Rechte (inkl. Bearbeiten, Löschen) verfügt wird.

Dazu gehören folgende Tests:

* 2.5 UC2 CRUD Stundeneintrag
* 2.6 UC4 CRUD Benutzer
* 2.7 UC5 Benutzer authentifizieren
* 2.8 UC6 CRUD Kunde
* 2.9 UC7 CRUD Material
* 2.10 UC8 CRUD StundeneintragsTyp
* 2.11 UC9 CRUD Auftrag

## UC1 Stundeneintrag erfassen

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Erwartetes Ergebnis |
| 1 | Der Benutzer kann einen Stundeneintrag auf dem Client erfassen. | Die Daten werden auf dem Client zwischengespeichert. |
| 2 | Der Benutzer kann die zwischengespeicherten Stundeneinträge bei bestehender Verbindung übertragen. | Die Daten werden auf dem Server korrekt abgelegt und auf dem Client gelöscht. |
| 3 | Der Benutzer versucht die erfassten Stundeneinträge zu übertragen. Während der Übertagung wird die Verbindung jedoch unterbrochen. | Die nicht übermittelten Stundeneinträge bleiben in der lokalen Datenbank bestehen und es wird eine entsprechende Meldung angezeigt (z.B. 3 von 5 wurden übertragen). |
| 4 | Der Benutzer versucht die zwischengespeicherten Stundeneinträge bei wiederhergestellter Verbindung erneut zu übertragen. | Die Daten werden auf dem Server korrekt abgelegt und auf dem Client gelöscht. |
| 5 | Der Benutzer versucht die zwischengespeicherten Stundeneinträge trotz unterbrochener Verbindung zu übertragen. | Die Daten werden auf dem Client nicht gelöscht und es wird eine Fehlermeldung angezeigt. |

### Testprotokoll

|  |  |  |
| --- | --- | --- |
| # | Erhaltenes Ergebnis | Resultat |
| 1 | Stundeneintrag mit lokaler ID wurde erstellt | Ok |
| 2 | Stundeneintrag wurde erfolgreich übermittelt | Ok |
| 3 | Meldung wurde angezeigt, nicht übertragene Stundeneinträge bleiben lokal gespeichert | Ok |
| 4 | Die verbliebenen Stundeneinträge konnten erfolgreich übertragen werden | Ok |
| 5 | Fehlermeldung wird angezeigt, Stundeneinträge bleiben in Datenbank erhalten | Ok |

## UC2 CRUD Stundeneintrag

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Erwartetes Ergebnis |
| 1 | Der Benutzer kann einen Stundeneintrag über den Browser erfassen. | Die Daten werden korrekt abgelegt. |
| 2 | Der Benutzer kann einen Stundeneintrag über den Browser bearbeiten. | Die Daten werden korrekt abgelegt. |
| 3 | Der Benutzer kann einen Stundeneintrag über den Browser löschen. | Die Daten werden korrekt entfernt. |
| 4 | Der Benutzer kann sich die Liste aller Stundeneinträge über den Browser anzeigen lassen. | Die Daten werden korrekt angezeigt. |

### Testprotokoll

|  |  |  |
| --- | --- | --- |
| # | Erhaltenes Ergebnis | Resultat |
| 1 | Stundeneintrag wurde mit korrekten Daten erstellt | Ok |
| 2 | Daten wurden im Stundeneintrag korrekt angepasst | Ok |
| 3 | Stundeneintrag wurde erfolgreich gelöscht | Ok |
| 4 | Liste aller Stundeneinträge wurde erfolgreich angezeigt | Ok |

## UC4 CRUD Benutzer

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Erwartetes Ergebnis |
| 1 | Der aktuell angemeldete Benutzer kann einen Benutzer über den Browser erfassen. | Die Daten werden korrekt abgelegt. |
| 2 | Der aktuell angemeldete Benutzer kann einen Benutzer über den Browser bearbeiten. | Die Daten werden korrekt abgelegt. |
| 3 | Der aktuell angemeldete Benutzer kann einen Benutzer über den Browser löschen. | Die Daten werden korrekt entfernt. |
| 4 | Der aktuell angemeldete Benutzer kann sich die Liste aller Benutzer über den Browser anzeigen lassen. | Die Daten werden korrekt angezeigt. |

### Testprotokoll

|  |  |  |
| --- | --- | --- |
| # | Erhaltenes Ergebnis | Resultat |
| 1 | Benutzer wurde mit korrekten Daten erstellt | Ok |
| 2 | Benutzerdaten wurden korrekt angepasst | Ok |
| 3 | Benutzer wurde erfolgreich gelöscht | Ok |
| 4 | Liste aller Benutzer wurde erfolgreich angezeigt | Ok |

## UC5 Benutzer authentifizieren

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Erwartetes Ergebnis |
| 1 | Der Benutzer kann sich mit Email und Passwort über den Browser anmelden. | Der Benutzer wird angemeldet. |
| 2 | Der Benutzer kann sich nicht mit fehlerhaften Daten über den Browser anmelden. | Der Benutzer wird nicht angemeldet. |

### Testprotokoll

|  |  |  |
| --- | --- | --- |
| # | Erhaltenes Ergebnis | Resultat |
| 1 | Benutzer wurde am System angemeldet | Ok |
| 2 | Benutzer wurde am System nicht angemeldet | Ok |

## UC6 CRUD Kunde

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Erwartetes Ergebnis |
| 1 | Der Benutzer kann einen Kunden über den Browser erfassen. | Die Daten werden korrekt abgelegt. |
| 2 | Der Benutzer kann einen Kunden über den Browser bearbeiten. | Die Daten werden korrekt abgelegt. |
| 3 | Der Benutzer kann einen Kunden über den Browser löschen. | Die Daten werden auf inaktiv gesetzt. |
| 4 | Der Benutzer kann sich die Liste aller Kunden über den Browser anzeigen lassen. | Die Daten werden korrekt angezeigt. |
| 5 | Der Benutzer kann einen Kunden, welcher gelöscht wurde, immer noch auf einem Stundeneintrag anzeigen, welchem dieser Kunde vor dessen Löschung zugewiesen wurde. | Die Daten werden korrekt angezeigt. |

### Testprotokoll

|  |  |  |
| --- | --- | --- |
| # | Erhaltenes Ergebnis | Resultat |
| 1 | Kunde wurde mit korrekten Daten erstellt | Ok |
| 2 | Daten des Kunden wurden korrekt angepasst | Ok |
| 3 | Kunde wurde erfolgreich auf inaktiv gesetzt | Ok |
| 4 | Liste aller Kunden wurde erfolgreich angezeigt | Ok |

## UC7 CRUD Material

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Erwartetes Ergebnis |
| 1 | Der Benutzer kann Material über den Browser erfassen. | Die Daten werden korrekt abgelegt. |
| 2 | Der Benutzer kann ein eingetragenes Material über den Browser bearbeiten. | Der alte Materialeintrag wird historisiert und die neuen Daten werden korrekt abgelegt. |
| 3 | Der Benutzer kann ein eingetragenes Material über den Browser löschen. | Die Daten werden als gelöscht markiert. |
| 4 | Der Benutzer kann sich die Liste aller Kunden über den Browser anzeigen lassen. | Die Daten werden korrekt angezeigt. |
| 5 | Der Benutzer kann ein Material, welches geändert oder gelöscht wurde, immer noch auf einem Auftrag anzeigen, welchem dieses Material vor dessen Mutation zugewiesen wurde. | Die Daten werden korrekt angezeigt. |

### Testprotokoll

|  |  |  |
| --- | --- | --- |
| # | Erhaltenes Ergebnis | Resultat |
| 1 | Material wurde mit korrekten Daten erstellt | Ok |
| 2 | Daten des Materialeintrages wurden korrekt angepasst | Ok |
| 3 | Materialeintrag wurde erfolgreich als gelöscht markiert | Ok |
| 4 | Liste aller Materialien wurde erfolgreich angezeigt | Ok |

## UC8 CRUD StundeneintragsTyp

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Erwartetes Ergebnis |
| 1 | Der Benutzer kann einen Stundeneintragstyp über den Browser erfassen. | Die Daten werden korrekt abgelegt. |
| 2 | Der Benutzer kann einen Stundeneintragstyp über den Browser bearbeiten. | Die Daten werden korrekt abgelegt. |
| 3 | Der Benutzer kann einen Stundeneintragstyp über den Browser löschen. | Die Daten werden korrekt entfernt. |
| 4 | Der Benutzer kann sich die Liste aller Stundeneintragstypen über den Browser anzeigen lassen. | Die Daten werden korrekt angezeigt. |

### Testprotokoll

|  |  |  |
| --- | --- | --- |
| # | Erhaltenes Ergebnis | Resultat |
| 1 | Stundeneintragstyp wurde mit korrekten Daten erstellt | Ok |
| 2 | Daten des Stundeneintragstyps wurden korrekt angepasst | Ok |
| 3 | Stundeneintragstyp wurde erfolgreich gelöscht | Ok |
| 4 | Liste aller Stundeneintragstypen wurde erfolgreich angezeigt | Ok |

## UC9 CRUD Auftrag

|  |  |  |
| --- | --- | --- |
| # | Beschreibung | Erwartetes Ergebnis |
| 1 | Der Benutzer kann einen Auftrag über den Browser erfassen. | Die Daten werden korrekt abgelegt. |
| 2 | Der Benutzer kann einen Auftrag über den Browser bearbeiten. | Die Daten werden korrekt abgelegt. |
| 3 | Der Benutzer kann einen Stundeneintragstyp über den Browser löschen. | Die Daten werden korrekt entfernt. |
| 4 | Der Benutzer kann sich die Liste aller Aufträge über den Browser anzeigen lassen. | Die Daten werden korrekt angezeigt. |

### Testprotokoll

|  |  |  |
| --- | --- | --- |
| # | Erhaltenes Ergebnis | Resultat |
| 1 | Auftrag wurde mit korrekten Daten erstellt | Ok |
| 2 | Daten des Auftrags wurden korrekt angepasst | Ok |
| 3 | Auftrag wurde erfolgreich gelöscht | Ok |
| 4 | Liste aller Aufträge wurde erfolgreich angezeigt | Ok |

# Unit Tests

Die Unit Tests der Client- und Serverapplikation werden in den weiteren Unterkapiteln beschrieben. Dabei wird als erstes die Gesamtübersicht der Tests aufgezeigt und danach die Testklassen genauer beschrieben.

## Client

Der Test testAndroidTestCaseSetupProperly() wird in allen Testklassen, die von der AndroidTestCase-Klasse erben, ausgeführt. Er stellt sicher, dass das Setup ordnungsgemäss funktioniert hat.

Alle in diesem Unterkapitel 3.1 Client beschriebenen Tests leiten von der AndroidTestCase-Klasse ab – ausgenommen davon sind die Testklasse TimeEntryActivityTest (Unterkapitel 3.1.9) und die Testklasse LoginActivityTest (Unterkapitel 3.1.11), welche von der Klasse ActivityInstrumentationTestCase2 ableiten.

### Gesamtübersicht Tests

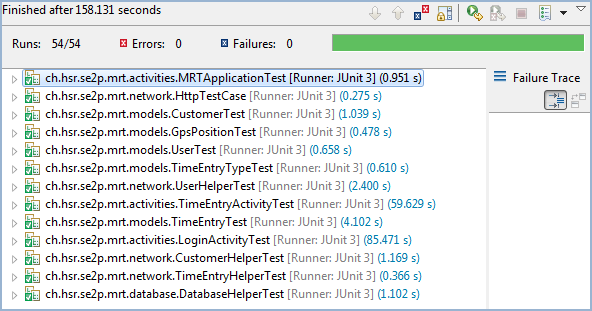


Abbildung 1 - Auflistung, Abdeckungsrate und Ergebnis der durchgeführten Tests für die Client Applikation

### Testklasse MRTApplicationTest

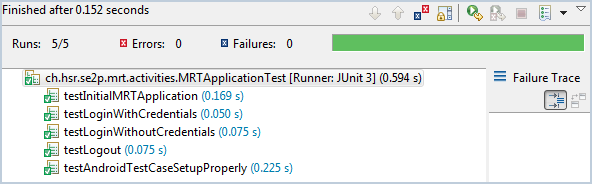


Abbildung - Testklasse MRTApplicationTest

Enthaltene Tests:

1. testInitialMRTApplication()  
   Testet die initiale Applikation
2. testLoginWithCredentials()  
   Testet das Einloggen mit gespeicherten Daten
3. testLoginWithoutCredentials()  
   Testet das Einloggen mit Loginangaben
4. testLogout()  
   Testet das Ausloggen

### Testklasse HttpTestCase

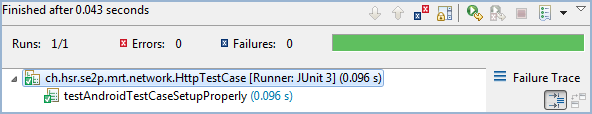


Abbildung 3 - Testklasse HttpTestCase

Dieser TestCase erstellt ein Mock-Objekt, welches für die Tests 3.1.8 Testklasse UserHelperTest, 3.1.12 Testklasse CustomerHelperTest und 3.1.13 Testklasse TimeEntryHelperTest verwendet wird.

### Testklasse CustomerTest

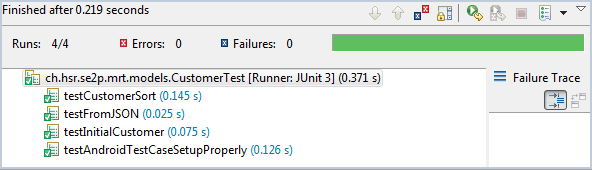


Abbildung - Testklasse CustomerTest

Enthaltene Tests:

1. testCustomerSort()  
   Testet die Sortierung der Kunden
2. testFromJSON()  
   Testet den vom Server erhaltenen Kunden
3. testInitialCustomer()  
   Testet, dass anfänglich kein Kunde existiert

### Testklasse GpsPositionTest

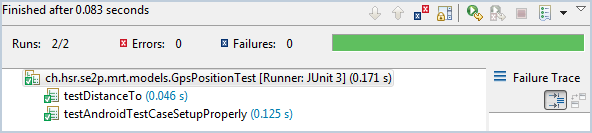


Abbildung - Testklasse GpsPostitionTest

Enthaltene Tests:

1. testDistanceTo()  
   Testet die Distanz zwischen zwei GPS-Postitionen

### Testklasse UserTest

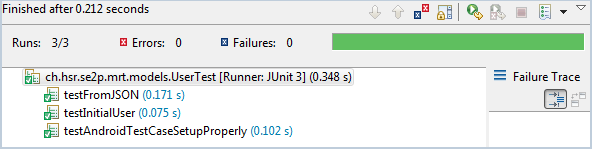


Abbildung - Testklasse UserTest

Enthaltene Tests:

1. testFromJSON()  
   Testet den vom Server erhaltenen Benutzer
2. testInitialUser()  
   Testet, dass anfänglich kein Benutzer gespeichert ist

### Testklasse TimeEntryTypeTest

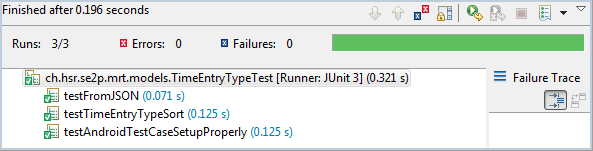


Abbildung - Testklasse TimeEntryTypeTest

Enthaltene Tests:

1. testFromJSON()  
   Testet den vom Server erhaltenen Stundeneintrag
2. testTimeEntryTypeSort()  
   Testet die Sortierung der Stundeneinträge

### Testklasse UserHelperTest

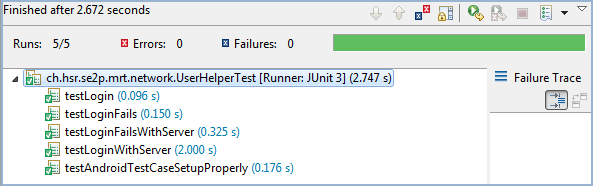


Abbildung - Testklasse UserHelperTest

Enthaltene Tests:

1. testLogin()  
   Testet das Einloggen mit einem Mock-Objekt
2. testLoginFails()  
   Testet das Fehlschlagen des Einloggens mit einem Mock-Objekt
3. testLoginFailsWithServer()  
   Testet das Fehlschlagen des Einloggens mit einer Verbindung zum Server
4. testLoginWithServer()  
   Testet das Einloggens mit einer Verbindung zum Server

### Testklasse TimeEntryActivityTest

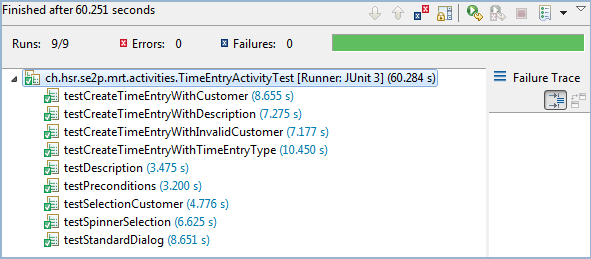


Abbildung - Testklasse TimeEntryActivityTest

Damit dieser Test funktioniert, muss sichergestellt werden, dass die Kundendaten und die Stundeneintragstypen bereits synchronisiert wurden.

Enthaltene Tests:

1. testCreateTimeEntryWithCustomer()  
   Testet das Erstellen eines Stundeneintrags mit einem Kunden
2. testCreateTimeEntryWithDescription()  
   Testet das Erstellen eines Stundeneintrags mit einer Beschreibung
3. testCreateTimeEntryWithInvalidCustomer()  
   Testet das Erstellen eines Stundeneintrags mit einem nicht existierenden Kunden
4. testCreateTimeEntryWithTimeEntryType()  
   Testet das Erstellen eines Stundeneintrags mit einem Stundeneintragstyp
5. testDescription()  
   Testet die Beschreibung
6. testPreconditions()  
   Testet die Voreinstellungen
7. testSelectionCustomer()  
   Testet die Auswahl eines Kunden
8. testSpinnerSelection()  
   Testet die Auswahl eines Stundeneintragstyps
9. testStandardDialog()  
   Testet den Dialog, der beim Erfassen eines Stundeneintrags abläuft

### Testklasse TimeEntryTest

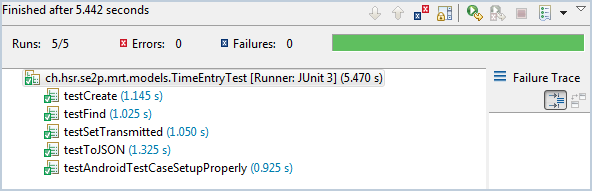


Abbildung - Testklasse TimeEntryTest

Enthaltene Tests:

1. testCreate()  
   Testet die Erstellung eines Stundeneintrags
2. testFInd()  
   Testet, ob der erstellte Stundeneintrag in der Datenbank gefunden werden kann
3. testSetTransmitted()  
   Testet, ob ein Stundeneintrag an den Server übertragen wurde
4. testToJSON()  
   Testet die Konvertierung in ein JSON-Objekt

### Testklasse LoginActivityTest

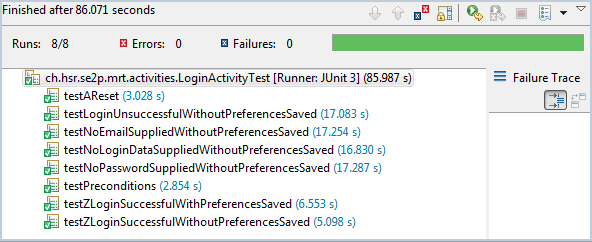


Abbildung - Testklasse LoginActivityTest

Damit dieser Test funktioniert, muss sichergestellt werden, dass sich der Benutzer auf dem Client abgemeldet hat und somit keine Daten in den Preferences gespeichert sind.

Enthaltene Tests:

1. testAReset()  
   Testet das Zurücksetzen der gespeicherten Benutzerdaten
2. testLoginUnsuccessfulWithoutPreferencesSaved()  
   Testet das Einloggen mit falschen Benutzerdaten
3. testNoEmailSuppliedWithoutPreferencesSaved()  
   Testet das Einloggen ohne Emailangaben
4. testNoLoginDataSuppliedWithoutPreferencesSaved()  
   Testet das Einloggen ohne Benutzerdaten
5. testNoPasswordSuppliedWithoutPreferencesSaved()  
   Testet das Einloggen ohne Passwort
6. testPreconditions()  
   Testet die Voreinstellungen
7. testZLoginSuccessfulWithPreferencesSaved()  
   Testet das Einloggen mit gespeicherten Benutzerdaten
8. testZLoginSuccessfulWithoutPreferencesSaved()  
   Testet das Einloggen ohne das Speichern der Benutzerdaten

### Testklasse CustomerHelperTest

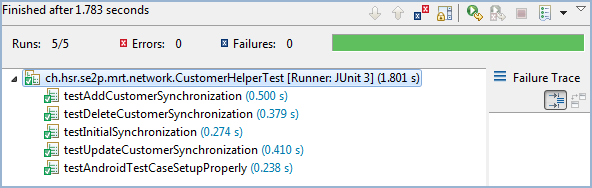


Abbildung - Testklasse CustomerHelperTest

Enthaltene Tests:

1. testAddCustomerSynchonization()  
   Testet die Synchronisierung nach dem Hinzufügen eines Kunden auf dem Server
2. testDeleteCustomerSynchonization()  
   Testet die Synchronisierung nach dem Löschen eines Kunden auf dem Server
3. testInitialSynchonization()  
   Testet die anfängliche Synchronisierung
4. testUpdateCustomerSynchonization()  
   Testet die Synchronisierung nach dem Mutieren eines Kunden auf dem Server

### Testklasse TimeEntryHelperTest

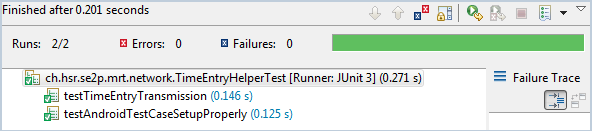


Abbildung - Testklase TimeEntryHelperTest

Enthaltene Tests:

1. testTimeEntryTransmission()  
   Testet die Übertragung eines Stundeneintrags

### Testklasse DatabaseHelperTest

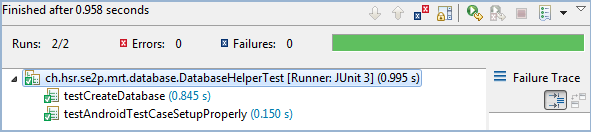


Abbildung - Testklasse DatabaseHelperTest

Enthaltene Tests:

1. testCreateDatabase()  
   Testet das Erstellen der Datenbank

## Server

### Gesamtübersicht Tests & Abdeckung

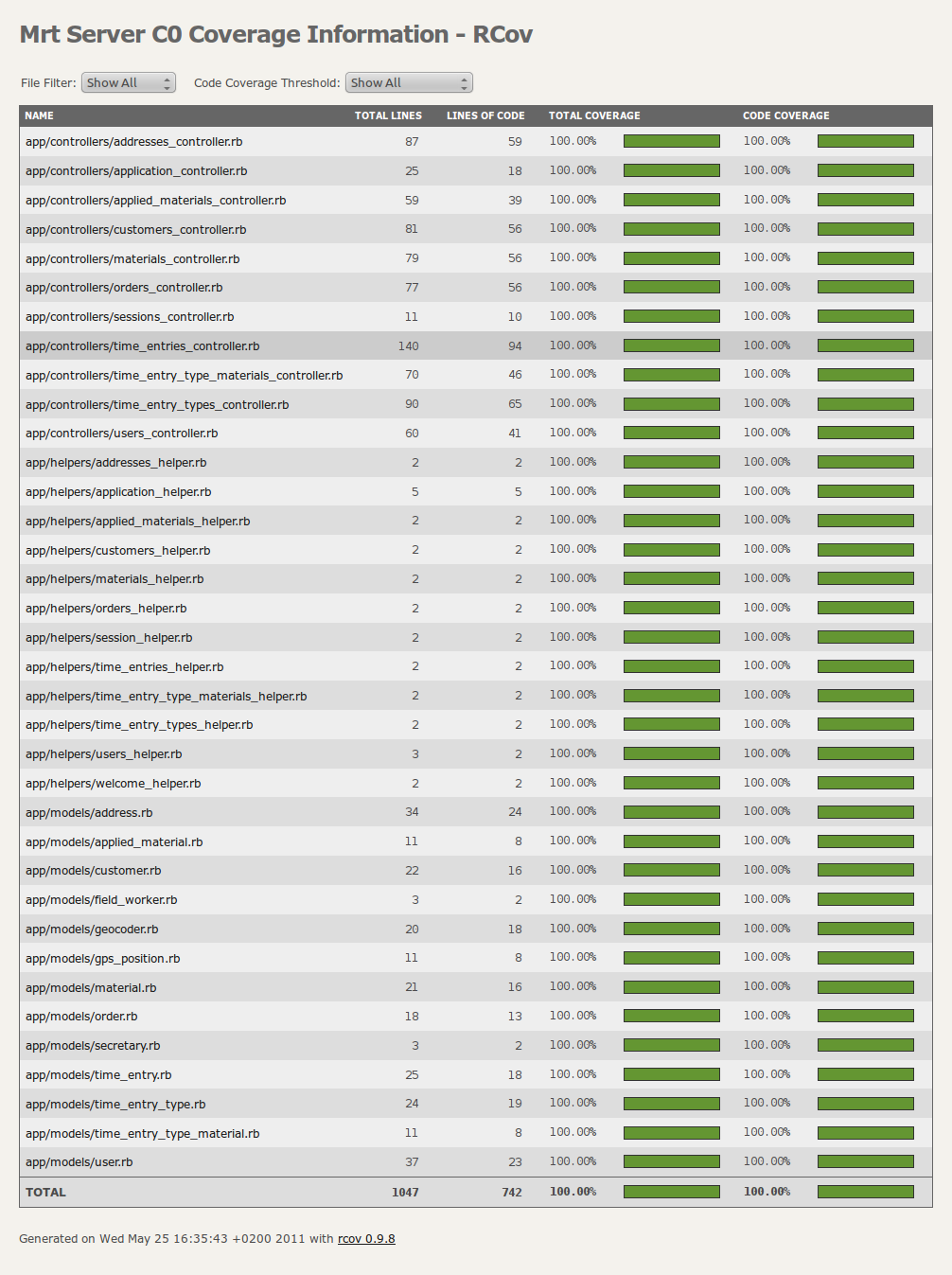


Abbildung 15 - Auflistung, Abdeckungsrate und Ergebnis der durchgeführten Tests für die Server Applikation

### TestklasseTimeEntriesControllerTest

Beinhaltete Tests:

1. "should get index"
2. "should get new"
3. "should create time\_entry"
4. "should create time\_entry by json"
5. "should not create time\_entry by json without hashcode"
6. "should remove hashcode"
7. "should leave time\_entry unchanged if hashcode is already nil"
8. "should not create time\_entry by json with existing hashcode"
9. "should show time\_entry"
10. "should get edit"
11. "should update time\_entry"
12. "should not update time\_entry without start\_time"
13. "should not create time\_entry without start\_time"
14. "should destroy time\_entry"

#### Ausgabe

$> ruby -I test test/functional/time\_entries\_controller\_test.rb

Loaded suite test/functional/time\_entries\_controller\_test

Started

..............

Finished in 0.24991 seconds.

14 tests, 22 assertions, **0 failures**, 0 errors

### TestklasseTimeEntryTest

Beinhaltete Tests:

1. "remove\_hashcode removes hashcode"
2. "cannot create time\_entry without time\_start"
3. "cannot create time\_entry without time\_stop"

#### Ausgabe

$> ruby -I test test/unit/time\_entry\_test.rb

Loaded suite test/unit/time\_entry\_test

Started

...

Finished in 0.118605 seconds.

3 tests, 4 assertions, **0 failures**, 0 errors

# Performance Test

Einige Zeiten vorgeben, Testen ob in dieser Zeit geschafft

* Starten des Systems
* Übertragung eines Stundeneintrags