|  |
| --- |
| 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](#_Toc294218208)

[1.1 Änderungsgeschichte 1](#_Toc294218209)

[1.2 Inhaltsverzeichnis 1](#_Toc294218210)

[1.3 Abbildungsverzeichnis 3](#_Toc294218211)

[2 Systemtests 4](#_Toc294218212)

[2.1 Bemerkungen 4](#_Toc294218213)

[2.2 Voraussetzungen 4](#_Toc294218214)

[2.3 Vorbereitungen 4](#_Toc294218215)

[2.3.1 Test auf dem Client und dem Server 4](#_Toc294218216)

[2.3.2 Tests auf dem Server 4](#_Toc294218217)

[2.4 UC1 Stundeneintrag erfassen 5](#_Toc294218218)

[2.4.1 Testprotokoll 5](#_Toc294218219)

[2.5 UC2 CRUD Stundeneintrag 5](#_Toc294218220)

[2.5.1 Testprotokoll 5](#_Toc294218221)

[2.6 UC4 CRUD Benutzer 6](#_Toc294218222)

[2.6.1 Testprotokoll 6](#_Toc294218223)

[2.7 UC5 Benutzer authentifizieren 6](#_Toc294218224)

[2.7.1 Testprotokoll 6](#_Toc294218225)

[2.8 UC6 CRUD Kunde 6](#_Toc294218226)

[2.8.1 Testprotokoll 7](#_Toc294218227)

[2.9 UC7 CRUD Material 7](#_Toc294218228)

[2.9.1 Testprotokoll 7](#_Toc294218229)

[2.10 UC8 CRUD StundeneintragsTyp 7](#_Toc294218230)

[2.10.1 Testprotokoll 7](#_Toc294218231)

[2.11 UC9 CRUD Auftrag 8](#_Toc294218232)

[2.11.1 Testprotokoll 8](#_Toc294218233)

[3 Unit Tests 9](#_Toc294218234)

[3.1 Client 9](#_Toc294218235)

[3.1.1 Gesamtübersicht Tests 9](#_Toc294218236)

[3.1.2 Testklasse MRTApplicationTest 9](#_Toc294218237)

[3.1.3 Testklasse HttpTestCase 10](#_Toc294218238)

[3.1.4 Testklasse CustomerTest 10](#_Toc294218239)

[3.1.5 Testklasse GpsPositionTest 11](#_Toc294218240)

[3.1.6 Testklasse UserTest 11](#_Toc294218241)

[3.1.7 Testklasse TimeEntryTypeTest 11](#_Toc294218242)

[3.1.8 Testklasse UserHelperTest 12](#_Toc294218243)

[3.1.9 Testklasse TimeEntryActivityTest 12](#_Toc294218244)

[3.1.10 Testklasse TimeEntryTest 13](#_Toc294218245)

[3.1.11 Testklasse LoginActivityTest 13](#_Toc294218246)

[3.1.12 Testklasse CustomerHelperTest 14](#_Toc294218247)

[3.1.13 Testklasse TimeEntryHelperTest 15](#_Toc294218248)

[3.1.14 Testklasse DatabaseHelperTest 15](#_Toc294218249)

[3.2 Server 16](#_Toc294218250)

[3.2.1 Gesamtübersicht getestete Klassen 16](#_Toc294218251)

[3.2.2 Testklasse AddressesControllerTest 17](#_Toc294218252)

[3.2.3 Testklasse AppliedMaterialsControllerTest 17](#_Toc294218253)

[3.2.4 Testklasse CustomersControllerTest 18](#_Toc294218254)

[3.2.5 Testklasse MaterialsControllerTest 19](#_Toc294218255)

[3.2.6 Testklasse OrdersControllerTest 19](#_Toc294218256)

[3.2.7 Testklasse SessionsControllerTest 20](#_Toc294218257)

[3.2.8 Testklasse TimeEntriesControllerTest 21](#_Toc294218258)

[3.2.9 Testklasse TimeEntryTypeMaterialsControllerTest 22](#_Toc294218259)

[3.2.10 Testklasse TimeEntryTypesControllerTest 22](#_Toc294218260)

[3.2.11 Testklasse UsersControllerTest 23](#_Toc294218261)

[3.2.12 Testklasse AddressTest 24](#_Toc294218262)

[3.2.13 Testklasse AppliedMaterialTest 24](#_Toc294218263)

[3.2.14 Testklasse CustomerTest 24](#_Toc294218264)

[3.2.15 Testklasse GeocoderTest 24](#_Toc294218265)

[3.2.16 Testklasse LocationTest 24](#_Toc294218266)

[3.2.17 Testklasse MaterialTest 25](#_Toc294218267)

[3.2.18 Testklasse OrderTest 25](#_Toc294218268)

[3.2.19 Testklasse TimeEntryTest 25](#_Toc294218269)

[3.2.20 Testklasse TimeEntryTypeMaterialTest 26](#_Toc294218270)

[3.2.21 Testklasse TimeEntryTypeTest 26](#_Toc294218271)

[3.2.22 Testklasse UserTest 26](#_Toc294218272)

[3.2.23 Testklasse TestCase 26](#_Toc294218273)

[4 Performance Test 27](#_Toc294218274)

## Abbildungsverzeichnis

[Abbildung 1 - Auflistung, Abdeckungsrate und Ergebnis der durchgeführten Tests für die Client Applikation 9](#_Toc294218275)

[Abbildung 2 - Testklasse MRTApplicationTest 9](#_Toc294218276)

[Abbildung 3 - Testklasse HttpTestCase 10](#_Toc294218277)

[Abbildung 4 - Testklasse CustomerTest 10](#_Toc294218278)

[Abbildung 5 - Testklasse GpsPostitionTest 11](#_Toc294218279)

[Abbildung 6 - Testklasse UserTest 11](#_Toc294218280)

[Abbildung 7 - Testklasse TimeEntryTypeTest 11](#_Toc294218281)

[Abbildung 8 - Testklasse UserHelperTest 12](#_Toc294218282)

[Abbildung 9 - Testklasse TimeEntryActivityTest 12](#_Toc294218283)

[Abbildung 10 - Testklasse TimeEntryTest 13](#_Toc294218284)

[Abbildung 11 - Testklasse LoginActivityTest 13](#_Toc294218285)

[Abbildung 12 - Testklasse CustomerHelperTest 14](#_Toc294218286)

[Abbildung 13 - Testklasse TimeEntryHelperTest 15](#_Toc294218287)

[Abbildung 14 - Testklasse DatabaseHelperTest 15](#_Toc294218288)

[Abbildung 15 - Auflistung, Abdeckungsrate und Ergebnis der durchgeführten Tests für die Server Applikation 16](#_Toc294218289)

[Abbildung 16 - Testklasse AddressesControllerTest 17](#_Toc294218290)

[Abbildung 17 - Testklasse AppliedMaterialsControllerTest 17](#_Toc294218291)

[Abbildung 18 - Testklasse CustomersControllerTest 18](#_Toc294218292)

[Abbildung 19 - Testklasse MaterialsControllerTest 19](#_Toc294218293)

[Abbildung 20 - Testklasse OrdersControllerTest 19](#_Toc294218294)

[Abbildung 21 - Testklasse SessionsControllerTest 20](#_Toc294218295)

[Abbildung 22 - Testklasse TimeEntriesControllerTest 21](#_Toc294218296)

[Abbildung 23 - Testklasse TimeEntryTypeMaterialsControllerTest 22](#_Toc294218297)

[Abbildung 24 - Testklasse TimeEntryTypesControllerTest 22](#_Toc294218298)

[Abbildung 25 - Testklasse UsersControllerTest 23](#_Toc294218299)

[Abbildung 26 - Testklasse AddressTest 24](#_Toc294218300)

[Abbildung 27 - Testklasse AppliedMaterialTest 24](#_Toc294218301)

[Abbildung 28 - Testklasse CustomerTest 24](#_Toc294218302)

[Abbildung 29 - Testklasse GeocoderTest 24](#_Toc294218303)

[Abbildung 30 - Testklasse LocationTest 25](#_Toc294218304)

[Abbildung 31 - Testklasse MaterialTest 25](#_Toc294218305)

[Abbildung 32 - Testklasse OrderTest 25](#_Toc294218306)

[Abbildung 33 - Testklasse TimeEntryTest 25](#_Toc294218307)

[Abbildung 34 - Testklasse TimeEntryTypeMaterialTest 26](#_Toc294218308)

[Abbildung 35 - Testklasse TimeEntryTypeTest 26](#_Toc294218309)

[Abbildung 36 - Testklasse UserTest 26](#_Toc294218310)

[Abbildung 37 - Testklasse TestCase 26](#_Toc294218311)

# 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

### Test 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

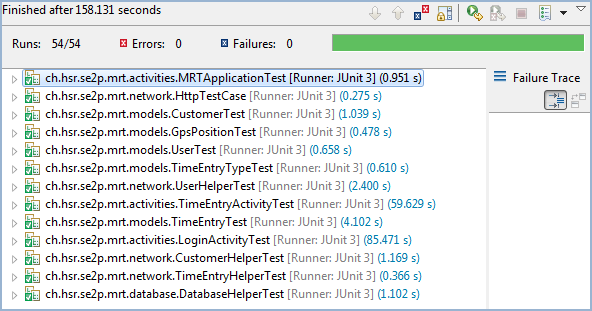


Abbildung 1 - Auflistung, Abdeckungsrate und Ergebnis der durchgeführten Tests für die Client Applikation

### Testklasse MRTApplicationTest

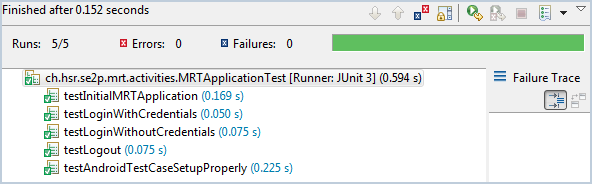


Abbildung 2 - Testklasse MRTApplicationTest

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

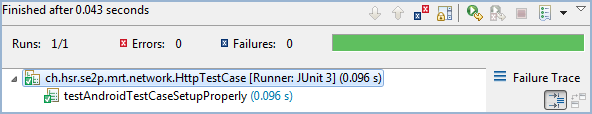


Abbildung 3 - Testklasse HttpTestCase

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

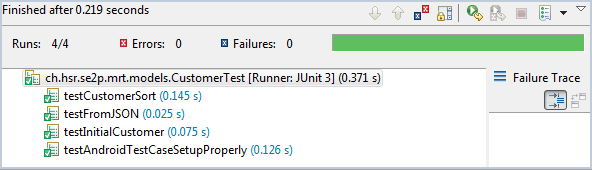


Abbildung 4 - Testklasse CustomerTest

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

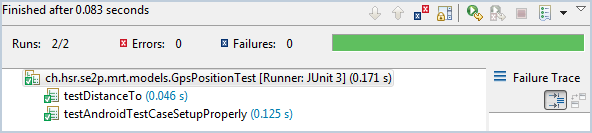


Abbildung 5 - Testklasse GpsPostitionTest

Enthaltene Tests:

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

### Testklasse UserTest

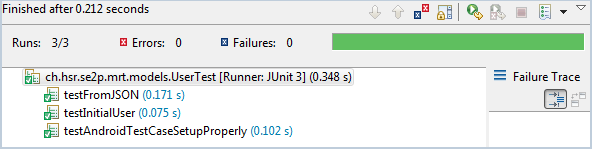


Abbildung 6 - Testklasse UserTest

Enthaltene Tests:

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

### Testklasse TimeEntryTypeTest

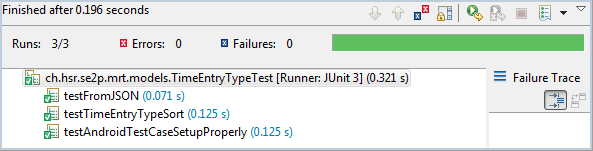


Abbildung 7 - Testklasse TimeEntryTypeTest

Enthaltene Tests:

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

### Testklasse UserHelperTest

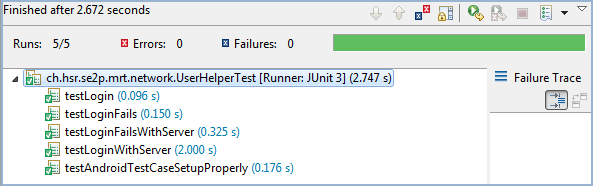


Abbildung 8 - Testklasse UserHelperTest

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

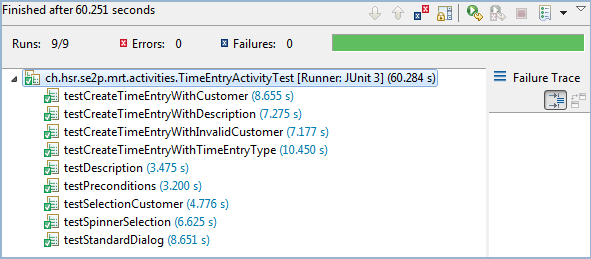


Abbildung 9 - Testklasse TimeEntryActivityTest

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

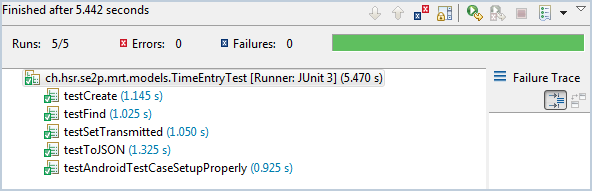


Abbildung 10 - Testklasse TimeEntryTest

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

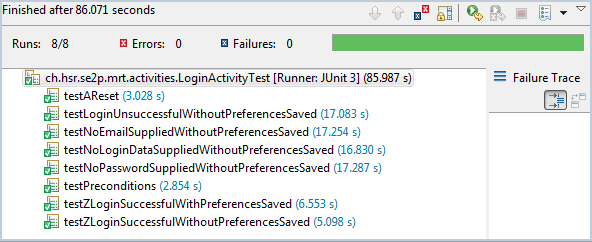


Abbildung 11 - Testklasse LoginActivityTest

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

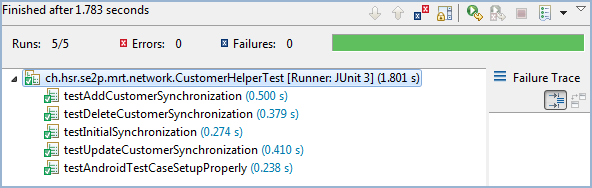


Abbildung 12 - Testklasse CustomerHelperTest

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

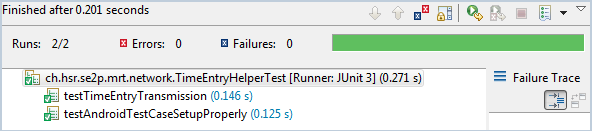


Abbildung 13 - Testklasse TimeEntryHelperTest

Enthaltene Tests:

1. testTimeEntryTransmission()  
   Testet die Übertragung eines Stundeneintrags

### Testklasse DatabaseHelperTest

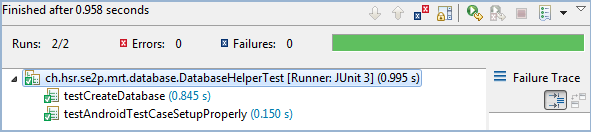


Abbildung 14 - Testklasse DatabaseHelperTest

Enthaltene Tests:

1. testCreateDatabase()  
   Testet das Erstellen der Datenbank

## Server

### Gesamtübersicht getestete Klassen

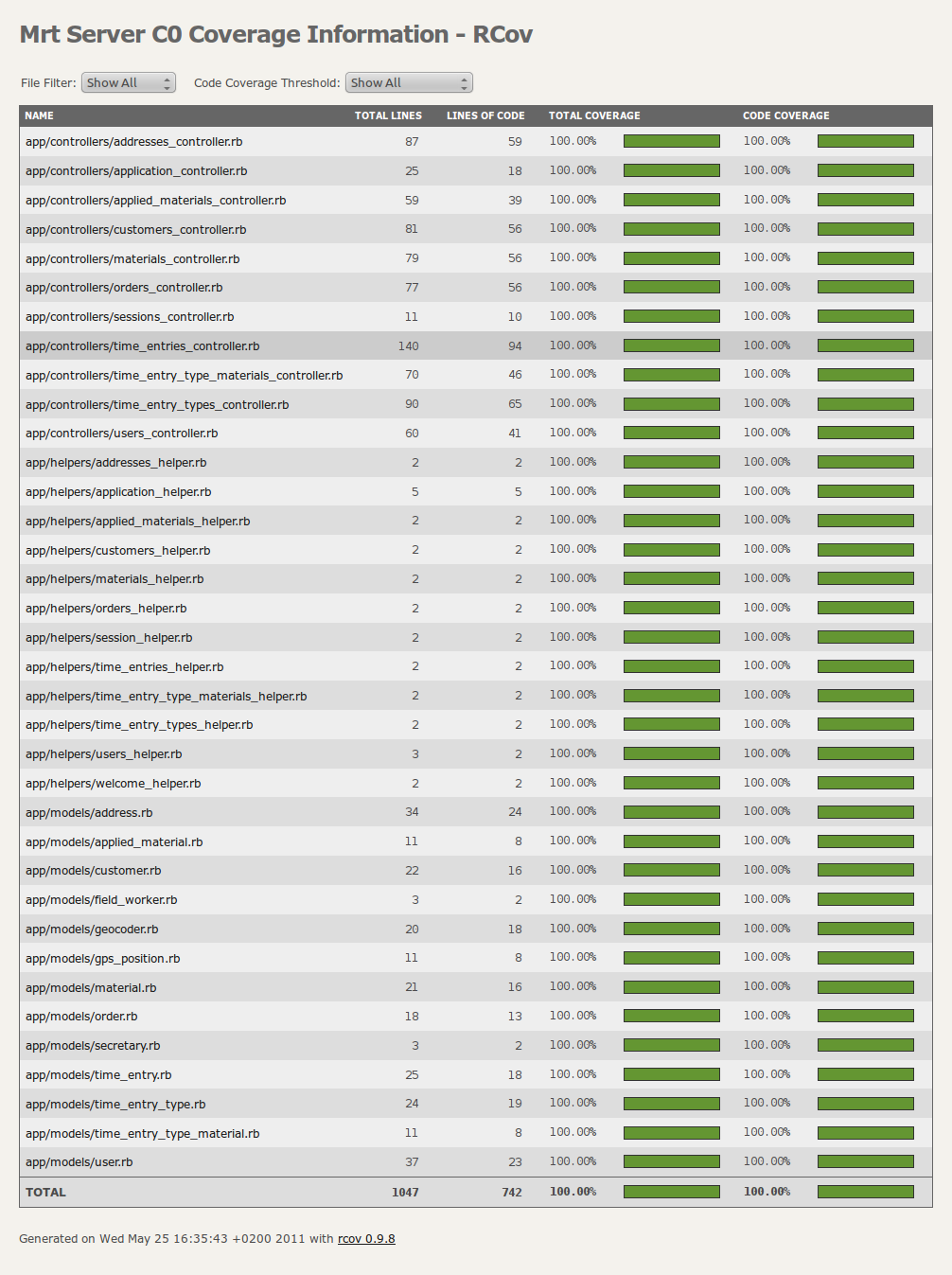


Abbildung 15 - Auflistung, Abdeckungsrate und Ergebnis der durchgeführten Tests für die Server Applikation

### Testklasse AddressesControllerTest

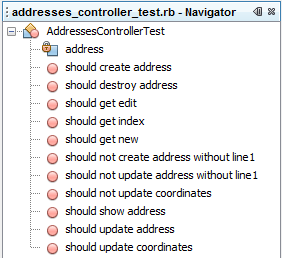


Abbildung 16 - Testklasse AddressesControllerTest

#### Ausgabe

>>> addresses\_controller\_test.rb

Loaded suite test/functional/addresses\_controller\_test

Started

test\_should\_create\_address(AddressesControllerTest): .

test\_should\_destroy\_address(AddressesControllerTest): .

test\_should\_get\_edit(AddressesControllerTest): .

test\_should\_get\_index(AddressesControllerTest): .

test\_should\_get\_new(AddressesControllerTest): .

test\_should\_not\_create\_address\_without\_line1(AddressesControllerTest): .

test\_should\_not\_update\_address\_without\_line1(AddressesControllerTest): .

test\_should\_not\_update\_coordinates(AddressesControllerTest): .

test\_should\_show\_address(AddressesControllerTest): .

test\_should\_update\_address(AddressesControllerTest): .

test\_should\_update\_coordinates(AddressesControllerTest): .

Finished in 1.241412 seconds.

11 tests, 21 assertions, 0 failures, 0 errors

### Testklasse AppliedMaterialsControllerTest

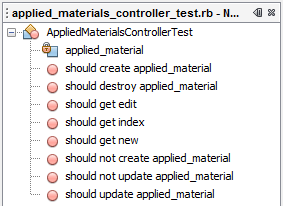


Abbildung 17 - Testklasse AppliedMaterialsControllerTest

#### Ausgabe

>>> applied\_materials\_controller\_test.rb

Loaded suite test/functional/applied\_materials\_controller\_test

Started

test\_should\_create\_applied\_material(AppliedMaterialsControllerTest): .

test\_should\_destroy\_applied\_material(AppliedMaterialsControllerTest): .

test\_should\_get\_edit(AppliedMaterialsControllerTest): .

test\_should\_get\_index(AppliedMaterialsControllerTest): .

test\_should\_get\_new(AppliedMaterialsControllerTest): .

test\_should\_not\_create\_applied\_material(AppliedMaterialsControllerTest): .

test\_should\_not\_update\_applied\_material(AppliedMaterialsControllerTest): .

test\_should\_update\_applied\_material(AppliedMaterialsControllerTest): .

Finished in 0.468409 seconds.

8 tests, 11 assertions, 0 failures, 0 errors

### Testklasse CustomersControllerTest

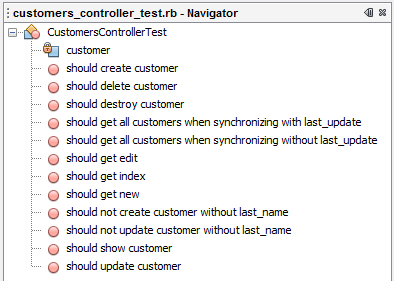


Abbildung 18 - Testklasse CustomersControllerTest

#### Ausgabe

>>> customers\_controller\_test.rb

Loaded suite test/functional/customers\_controller\_test

Started

test\_should\_create\_customer(CustomersControllerTest): .

test\_should\_delete\_customer(CustomersControllerTest): .

test\_should\_destroy\_customer(CustomersControllerTest): .

test\_should\_get\_all\_customers\_when\_synchronizing\_with\_last\_update(CustomersControllerTest): .

test\_should\_get\_all\_customers\_when\_synchronizing\_without\_last\_update(CustomersControllerTest): .

test\_should\_get\_edit(CustomersControllerTest): .

test\_should\_get\_index(CustomersControllerTest): .

test\_should\_get\_new(CustomersControllerTest): .

test\_should\_not\_create\_customer\_without\_last\_name(CustomersControllerTest): .

test\_should\_not\_update\_customer\_without\_last\_name(CustomersControllerTest): .

test\_should\_show\_customer(CustomersControllerTest): .

test\_should\_update\_customer(CustomersControllerTest): .

Finished in 0.571441 seconds.

12 tests, 19 assertions, 0 failures, 0 errors

### Testklasse MaterialsControllerTest

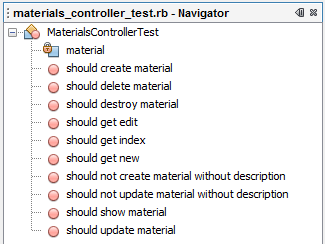


Abbildung 19 - Testklasse MaterialsControllerTest

#### Ausgabe

>>> materials\_controller\_test.rb

Loaded suite test/functional/materials\_controller\_test

Started

test\_should\_create\_material(MaterialsControllerTest): .

test\_should\_delete\_material(MaterialsControllerTest): .

test\_should\_destroy\_material(MaterialsControllerTest): .

test\_should\_get\_edit(MaterialsControllerTest): .

test\_should\_get\_index(MaterialsControllerTest): .

test\_should\_get\_new(MaterialsControllerTest): .

test\_should\_not\_create\_material\_without\_description(MaterialsControllerTest): .

test\_should\_not\_update\_material\_without\_description(MaterialsControllerTest): .

test\_should\_show\_material(MaterialsControllerTest): .

test\_should\_update\_material(MaterialsControllerTest): .

Finished in 0.739342 seconds.

10 tests, 15 assertions, 0 failures, 0 errors

### Testklasse OrdersControllerTest

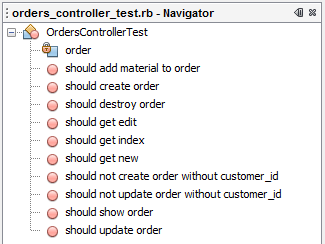


Abbildung 20 - Testklasse OrdersControllerTest

#### Ausgabe

>>> orders\_controller\_test.rb

Loaded suite test/functional/orders\_controller\_test

Started

test\_should\_add\_material\_to\_order(OrdersControllerTest): .

test\_should\_create\_order(OrdersControllerTest): .

test\_should\_destroy\_order(OrdersControllerTest): .

test\_should\_get\_edit(OrdersControllerTest): .

test\_should\_get\_index(OrdersControllerTest): .

test\_should\_get\_new(OrdersControllerTest): .

test\_should\_not\_create\_order\_without\_customer\_id(OrdersControllerTest): .

test\_should\_not\_update\_order\_without\_customer\_id(OrdersControllerTest): .

test\_should\_show\_order(OrdersControllerTest): .

test\_should\_update\_order(OrdersControllerTest): .

Finished in 0.586307 seconds.

10 tests, 15 assertions, 0 failures, 0 errors

### Testklasse SessionsControllerTest

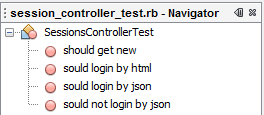


Abbildung 21 - Testklasse SessionsControllerTest

#### Ausgabe

>>> session\_controller\_test.rb

Loaded suite test/functional/session\_controller\_test

Started

test\_should\_get\_new(SessionsControllerTest): .

test\_sould\_login\_by\_html(SessionsControllerTest): .

test\_sould\_login\_by\_json(SessionsControllerTest): .

test\_sould\_not\_login\_by\_json(SessionsControllerTest): .

Finished in 0.352735 seconds.

4 tests, 2 assertions, 0 failures, 0 errors

### Testklasse TimeEntriesControllerTest

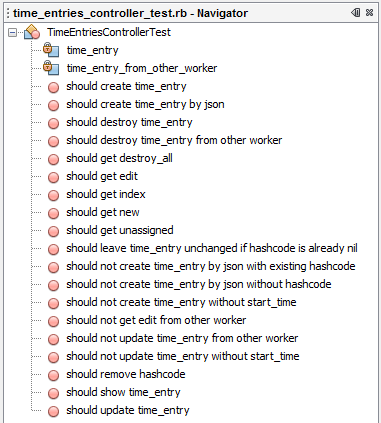


Abbildung 22 - Testklasse TimeEntriesControllerTest

#### Ausgabe

>>> time\_entries\_controller\_test.rb

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

Started

test\_should\_create\_time\_entry(TimeEntriesControllerTest): .

test\_should\_create\_time\_entry\_by\_json(TimeEntriesControllerTest): .

test\_should\_destroy\_time\_entry(TimeEntriesControllerTest): .

test\_should\_destroy\_time\_entry\_from\_other\_worker(TimeEntriesControllerTest): .

test\_should\_get\_destroy\_all(TimeEntriesControllerTest): .

test\_should\_get\_edit(TimeEntriesControllerTest): .

test\_should\_get\_index(TimeEntriesControllerTest): .

test\_should\_get\_new(TimeEntriesControllerTest): .

test\_should\_get\_unassigned(TimeEntriesControllerTest): .

test\_should\_leave\_time\_entry\_unchanged\_if\_hashcode\_is\_already\_nil(TimeEntriesControllerTest): .

test\_should\_not\_create\_time\_entry\_by\_json\_with\_existing\_hashcode(TimeEntriesControllerTest): .

test\_should\_not\_create\_time\_entry\_by\_json\_without\_hashcode(TimeEntriesControllerTest): .

test\_should\_not\_create\_time\_entry\_without\_start\_time(TimeEntriesControllerTest): .

test\_should\_not\_get\_edit\_from\_other\_worker(TimeEntriesControllerTest): .

test\_should\_not\_update\_time\_entry\_from\_other\_worker(TimeEntriesControllerTest): .

test\_should\_not\_update\_time\_entry\_without\_start\_time(TimeEntriesControllerTest): .

test\_should\_remove\_hashcode(TimeEntriesControllerTest): .

test\_should\_show\_time\_entry(TimeEntriesControllerTest): .

test\_should\_update\_time\_entry(TimeEntriesControllerTest): .

Finished in 1.059362 seconds.

19 tests, 29 assertions, 0 failures, 0 errors

### Testklasse TimeEntryTypeMaterialsControllerTest

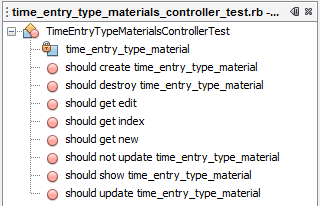


Abbildung 23 - Testklasse TimeEntryTypeMaterialsControllerTest

#### Ausgabe

>>> time\_entry\_type\_materials\_controller\_test.rb

Loaded suite test/functional/time\_entry\_type\_materials\_controller\_test

Started

test\_should\_create\_time\_entry\_type\_material(TimeEntryTypeMaterialsControllerTest): .

test\_should\_destroy\_time\_entry\_type\_material(TimeEntryTypeMaterialsControllerTest): .

test\_should\_get\_edit(TimeEntryTypeMaterialsControllerTest): .

test\_should\_get\_index(TimeEntryTypeMaterialsControllerTest): .

test\_should\_get\_new(TimeEntryTypeMaterialsControllerTest): .

test\_should\_not\_update\_time\_entry\_type\_material(TimeEntryTypeMaterialsControllerTest): .

test\_should\_show\_time\_entry\_type\_material(TimeEntryTypeMaterialsControllerTest): .

test\_should\_update\_time\_entry\_type\_material(TimeEntryTypeMaterialsControllerTest): .

Finished in 0.504579 seconds.

8 tests, 11 assertions, 0 failures, 0 errors

### Testklasse TimeEntryTypesControllerTest

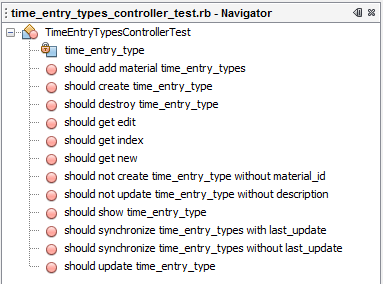


Abbildung 24 - Testklasse TimeEntryTypesControllerTest

#### Ausgabe

>>> time\_entry\_types\_controller\_test.rb

Loaded suite test/functional/time\_entry\_types\_controller\_test

Started

test\_should\_add\_material\_time\_entry\_types(TimeEntryTypesControllerTest): .

test\_should\_create\_time\_entry\_type(TimeEntryTypesControllerTest): .

test\_should\_delete\_time\_entry\_type(TimeEntryTypesControllerTest): .

test\_should\_destroy\_time\_entry\_type(TimeEntryTypesControllerTest): .

test\_should\_get\_edit(TimeEntryTypesControllerTest): .

test\_should\_get\_index(TimeEntryTypesControllerTest): .

test\_should\_get\_new(TimeEntryTypesControllerTest): .

test\_should\_not\_create\_time\_entry\_type\_without\_material\_id(TimeEntryTypesControllerTest): .

test\_should\_not\_update\_time\_entry\_type\_without\_description(TimeEntryTypesControllerTest): .

test\_should\_show\_time\_entry\_type(TimeEntryTypesControllerTest): .

test\_should\_synchronize\_time\_entry\_types\_with\_last\_update(TimeEntryTypesControllerTest): .

test\_should\_synchronize\_time\_entry\_types\_without\_last\_update(TimeEntryTypesControllerTest): .

test\_should\_update\_time\_entry\_type(TimeEntryTypesControllerTest): .

Finished in 0.800472 seconds.

13 tests, 21 assertions, 0 failures, 0 errors

### Testklasse UsersControllerTest

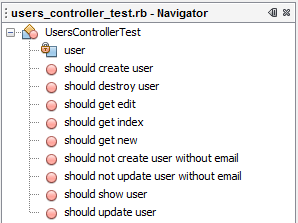


Abbildung 25 - Testklasse UsersControllerTest

#### Ausgabe

>>> users\_controller\_test.rb

Loaded suite test/functional/users\_controller\_test

Started

test\_should\_create\_user(UsersControllerTest): .

test\_should\_destroy\_user(UsersControllerTest): .

test\_should\_get\_edit(UsersControllerTest): .

test\_should\_get\_index(UsersControllerTest): .

test\_should\_get\_new(UsersControllerTest): .

test\_should\_not\_create\_user\_without\_email(UsersControllerTest): .

test\_should\_not\_update\_user\_without\_email(UsersControllerTest): .

test\_should\_show\_user(UsersControllerTest): .

test\_should\_update\_user(UsersControllerTest): .

Finished in 0.614425 seconds.

9 tests, 13 assertions, 0 failures, 0 errors

### Testklasse AddressTest

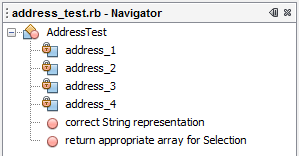


Abbildung 26 - Testklasse AddressTest

### Testklasse AppliedMaterialTest

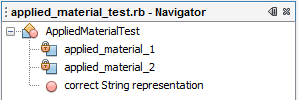


Abbildung 27 - Testklasse AppliedMaterialTest

### Testklasse CustomerTest

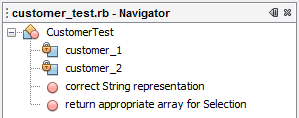


Abbildung 28 - Testklasse CustomerTest

### Testklasse GeocoderTest

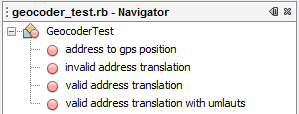


Abbildung 29 - Testklasse GeocoderTest

### Testklasse LocationTest

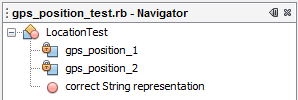


Abbildung 30 - Testklasse LocationTest

### Testklasse MaterialTest

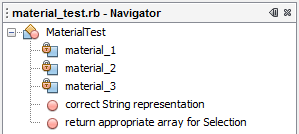


Abbildung 31 - Testklasse MaterialTest

### Testklasse OrderTest

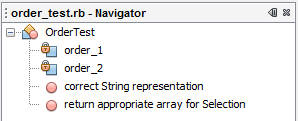


Abbildung 32 - Testklasse OrderTest

### Testklasse TimeEntryTest

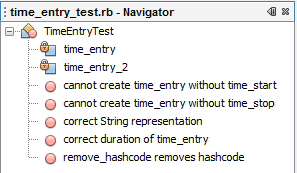


Abbildung 33 - Testklasse TimeEntryTest

#### 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

### Testklasse TimeEntryTypeMaterialTest

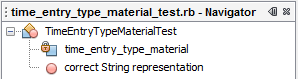


Abbildung 34 - Testklasse TimeEntryTypeMaterialTest

### Testklasse TimeEntryTypeTest

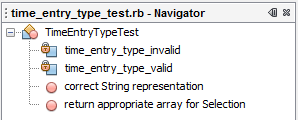


Abbildung 35 - Testklasse TimeEntryTypeTest

### Testklasse UserTest

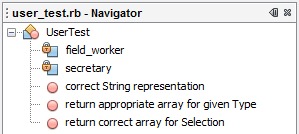


Abbildung 36 - Testklasse UserTest

### Testklasse TestCase

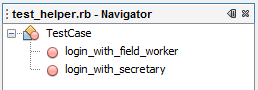


Abbildung 37 - Testklasse TestCase

Das Setup, damit mit der Authentisierung getestet werden kann.

# Performance Test

Einige Zeiten vorgeben, Testen ob in dieser Zeit geschafft

* Starten des Systems
* Übertragung eines Stundeneintrags