

Test-Plan

Einführung

Dieser Testplan ist lediglich eine Übersicht über die zum aktuellen Zeitpunkt geplanten Tests mit *JUnit*. Er enthält noch keine konkreten Daten, sondern plant nur Fallunterscheidungen für die Integrationstests. Der Testplan ist damit nur als ein Leitfaden der Entwickler für die späteren Test gedacht. Überprüfung der Browser-Darstellung ist nicht geplant.

Aufbau der Testbezeichner

[U|I]-[Klasse]-[Methode]-[Nr.]

U

Unit Test

I

Integrationstest

Nr.

Von 0 startende Nummer des Tests für diese Kombination aus Klasse und Methode.

Bsp.

U-OrderController-list-2

Test Gegenstände

Zu testen ist der gesamte Java-Code, der für unsere Software geschrieben wurde.

Zudem muss das Zusammenspiel zwischen unserem Code und dem Client getestet werden. Auch das Zusammenarbeiten der einzelnen Pakete und der Pakete mit anderen Frameworks ist zu testen.

Zu testende Funktionalitäten

- Getter, Setter und Konstruktoren
- interne Methoden unserer Software
- Controller unserer Software unter Nutzung der **Model**-Klasse
- Zusammenspiel der eigenen Methoden mit Bibliotheken

Nicht zu testende Funktionalitäten

- clientseitige Darstellung des HTML und CSS

- JavaScript
- clientseitige Seiteninteraktion (Formulare etc.)
- Tests, die nur Bibliotheken von Drittanbietern miteinbeziehen

Herangehensweise

Die einzelnen Methoden einer Klasse werden mit *JUnit* in Unit-Tests getestet.

Die Nutzer-Interaktion wird über die Integrationstests der Controller überprüft. Hierzu wird nur das Model, das an das Template übergeben wird, und der Template-Name bzw. Redirect überprüft.

Überprüfung der Browser-Darstellung ist nicht Bestandteil der Tests.

Umgebungsvoraussetzungen

Hardware

- Computer mit einer Architektur und einem Betriebssystem, unter dem Java 11 lauffähig ist.
- Zum Starten des Tests bzw. zum einrichten von automatischen Tests sind Tastatur, Maus und Bildschirm (ggf. über eine Netzwerkverbindung) nötig.

Bereitzustellende Daten

- Die gesamten Projektdaten der Miss-Mint-Filialmanagement-Software müssen vorhanden sein. Diese können mit Git aus dem Repository geholt werden.
- Weitere Daten (z. B. Datenbanken) sind nicht nötig.

Zusätzliche Software

- Neben den Projektdaten ist auch Java 11 vonnöten.
- Zusätzlich muss Maven installiert sein, um Tests auszuführen.
- Abhängigkeiten des Projektes, wie Spring, Salespoint etc. können mit Maven heruntergeladen werden.

Kommunikation

- Die Software benötigt während der Tests keine Netzwerkverbindung. Alle Tests in Spring laufen intern ab.
- Eine Internet-Verbindung ist demzufolge auch nicht notwendig, da insbesondere die Miss-Mint-Filialmanagement-Software keine Kommunikation nach außen tätigt.

Testfälle und Verantwortlichkeit

Aufträge

Viktor Reusch

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
U-MissMintOrder-MissMintOrder-0	UC0040	-	ein UserAccount, Customer, Eingangs-Datum, erwartetes Fertigstellungsdatum, Dienstleistung, Beschreibung	Zustand wird auf WAITING gesetzt und Quantity der Dienstleistung ist 1
U-MissMintOrder-getLatestFinished-0	UC0041-UC0043	Fertigstellungsdatum liegt vor dem erwartetem Fertigstellungsdatum	-	erwartetes Fertigstellungsdatum
U-MissMintOrder-getLatestFinished-1	UC0041-UC0043	Fertigstellungsdatum ist gleich dem erwarteten Fertigstellungsdatum	-	eines der beiden
U-MissMintOrder-getLatestFinished-2	UC0041-UC0043	Fertigstellungsdatum liegt nach dem erwartetem Fertigstellungsdatum	-	Fertigstellungsdatum
U-MissMintOrder-canPickUp-0	UC0041-UC0043	Zustand ist FINISHED .	-	true → Gegenstand kann abgeholt werden
U-MissMintOrder-canPickUp-1	UC0041-UC0043	Zustand ist STORED .	-	true → Gegenstand kann abgeholt werden
U-MissMintOrder-canPickUp-2	UC0041-UC0043	Zustand ist weder FINISHED noch STORED .	-	false → Gegenstand kann nicht abgeholt werden
U-OrderService-calculateCharge-0	UC0041-UC0043	Auftrag kann nicht abgeholt werden	-	<i>Exception</i> → Gebühr sollte jetzt nicht berechnet werden
U-OrderService-calculateCharge-1	UC0041-UC0043	Auftrag, der 10€ kostet, wurde 3 Tage zu spät fertig. Er wird pünktlich abgeholt.	-	-3€ → 3€ bekommt der Kunde Entschädigung

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
U-OrderService-calculateCharge-2	UC0041-UC0043	Auftrag, der 10€ kostet, wurde pünktlich fertig. Er wird pünktlich abgeholt.	-	0€ → keine Transaktion
U-OrderService-calculateCharge-3	UC0041-UC0043	Auftrag, der 10€ kostet, wurde 2 Tage zu spät fertig. Er wird 4 Wochen zu spät abgeholt abgeholt.	-	0€ → keine Transaktion
U-OrderService-calculateCharge-4	UC0041-UC0043	Auftrag, der 10€ kostet, wurde 3 Tage zu spät fertig. Er wird 4 Wochen zu spät abgeholt abgeholt.	-	-1€ → 1€ bekommt der Kunde Entschädigung
U-OrderService-calculateCharge-5	UC0041-UC0043	Auftrag, der 10€ kostet, wurde pünktlich fertig. Er wird 4 Wochen zu spät abgeholt abgeholt.	-	2€ muss der Kunde nachzahlen
U-OrderService-updateOrders-0	UC0042, UC0043, UC0050	Auftrag, der jetzt seinen Raum-Slot zur Bearbeitung hat.	-	Der Auftrag wird auf IN_PROGRESS gesetzt.
U-OrderService-updateOrders-1	UC0042, UC0043, UC0050	Auftrag, dessen Bearbeitung jetzt abgeschlossen wird.	-	Der Auftrag wird auf FINISHED gesetzt und das Fertigstellungsdatum gespeichert.
U-OrderService-updateOrders-2	UC0042, UC0043, UC0050	Auftrag, der seit einer Woche fertig ist.	-	Der Auftrag wird auf STORED gesetzt.
U-OrderService-updateOrders-3	UC0042, UC0043, UC0050	Auftrag, der seit einer Woche und drei Monaten fertig ist.	-	Der Auftrag wird auf CHARITABLE_USED gesetzt und der Gegenstand aus dem Inventar gelöscht.

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
U-ServiceDataInitiali- zer-initialize-0	-	Es existieren noch keine Dienstleistungen in der Datenbank.	-	Die 15 Dienstleistungen werden angelegt.
U-ServiceDataInitiali- zer-initialize-1	-	Es existieren schon alle Dienstleistungen in der Datenbank.	-	Keine neuen Dienstleistungen werden angelegt.
U-ServiceDataInitiali- zer-initialize-2	-	Es existieren schon alle Dienstleistungen in der Datenbank.	-	Keine neuen Dienstleistungen werden angelegt.
I-OrdersController- listOrders-0	UC0010	Der Nutzer ist nicht angemeldet.	-	Anfrage wird wegen @PreAuthorize abgelehnt.
I-OrdersController- listOrders-1	UC0041-UC0043	Der Mitarbeiter ist angemeldet.	-	Alle Aufträge aus dem System werden zusammen mit der zugehörigen Dienstleistung dem Nutzer über das Template angezeigt.
I-OrdersController- listOrders-2	UC0041-UC0043	Der Mitarbeiter ist angemeldet. Es gibt keine Aufträge im System	-	Die Auflistung der Aufträge bleibt leer.
I-PickUpController- pickupPage-0	UC0010	Der Nutzer ist nicht angemeldet.	-	Anfrage wird wegen @PreAuthorize abgelehnt.
I-PickUpController- pickupPage-1	UC0041-UC0043	Der Mitarbeiter ist angemeldet.	Ein leeres Optional, da der Auftrag nicht existiert.	Anfrage abgelehnt mit BAD_REQUEST.
I-PickUpController- pickupPage-2	UC0041	Der Mitarbeiter ist angemeldet.	Ein Auftrag, der pünktlich fertig wurde und nicht zu spät abgeholt wird.	Ein Template zur Bestätigung der Abgabe ohne die Aufforderung einer Transaktion.

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
I-PickUpController-pickupPage-3	UC0042	Der Mitarbeiter ist angemeldet.	Ein Auftrag, der pünktlich fertig wurde und 3 Wochen zu spät abgeholt wird.	Ein Template zur Bestätigung der Abgabe und die Aufforderung 1,50€ zu kassieren.
I-PickUpController-pickupPage-4	UC0042	Der Mitarbeiter ist angemeldet. Es existiert ein Auftrag, der 10€ gekostet hat.	Dieser Auftrag, der 3 Tage zu spät fertig wurde und pünktlich abgeholt wird.	Ein Template zur Bestätigung der Abgabe und die Aufforderung 3€ Entschädigung zu zahlen.
I-PickUpController-pickupPage-5	UC0042, UC0042	Der Mitarbeiter ist angemeldet. Es existiert ein Auftrag, der 10€ gekostet hat.	Dieser Auftrag, der 3 Tage zu spät fertig wurde und 6 Wochen zu spät abgeholt wird.	Ein Template zur Bestätigung der Abgabe ohne die Aufforderung einer Transaktion.
I-PickUpController-pickupPage-6	UC0041-UC0043	Der Mitarbeiter ist angemeldet.	Ein Auftrag, der noch in Bearbeitung ist.	Anfrage abgelehnt mit BAD_REQUEST .
I-PickUpController-pickupForm-0	UC0010	Der Nutzer ist nicht angemeldet.	-	Anfrage wird wegen @PreAuthorize abgelehnt.
I-PickUpController-pickupForm-1	UC0041-UC0043	Der Mitarbeiter ist angemeldet.	Ein Auftrag, der noch in Bearbeitung ist.	Anfrage abgelehnt mit BAD_REQUEST .
I-PickUpController-pickupPage-2	UC0041-UC0043	Der Mitarbeiter ist angemeldet.	Ein leeres Optional , da der Auftrag nicht existiert.	Anfrage abgelehnt mit BAD_REQUEST .
I-PickUpController-pickupPage-3	UC0041-UC0043	Der Mitarbeiter ist angemeldet.	Ein Auftrag.	Der Auftragszustand wird auf PICKED_UP gesetzt. Eine Umleitung wird zurückgegeben.
I-ReceivingController-receiving-0	UC0010	Der Nutzer ist nicht angemeldet.	-	Anfrage wird wegen @PreAuthorize abgelehnt.

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
I- ReceivingControlle r-receiving-1	UC0040	Der Mitarbeiter ist angemeldet.	-	Ein Formular zum Erstellen eines neuen Auftrags.
I- ReceivingControlle r-cost-0	UC0010	Der Nutzer ist nicht angemeldet.	-	Anfrage wird wegen @PreAuthorize abgelehnt.
I- ReceivingControlle r-cost-1	UC0040	Der Mitarbeiter ist angemeldet.	Ein ungültiges Erstellformular.	Die Seite mit dem Formular mit einer Fehler-Meldung. Das Formular ist immer noch ausgefüllt.
I- ReceivingControlle r-cost-2	UC0040	Der Mitarbeiter ist angemeldet.	Ein Formular mit einer nicht existierenden Dienstleistung.	Anfrage abgelehnt mit BAD_REQUEST.
I- ReceivingControlle r-cost-3	UC0040	Der Mitarbeiter ist angemeldet.	Ein valides Formular.	Ein entsprechender Auftrag wird erstellt aber nicht gespeichert. Eine Seite, die das Kosten-Formular darstellt, wird angezeigt.
I- ReceivingControlle r-cost-4	UC0040	Der Mitarbeiter ist angemeldet. Es gibt keine Räume oder keinen Mitarbeiter für die Dienstleistung.	Ein valides Formular.	Ein Fehler, der über den Missstand informiert wird angezeigt.
I- ReceivingControlle r-ticket-0	UC0010	Der Nutzer ist nicht angemeldet.	-	Anfrage wird wegen @PreAuthorize abgelehnt.

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
I- ReceivingController-ticket-1	UC0040	Der Mitarbeiter ist angemeldet.	Eine Sitzung mit einem Auftrag.	Der Auftrag wird mit berechnetem Fertigstellungsdatum gespeichert, ein Raum gebucht und ein Mitarbeiter verplant. Das Ticket wird angezeigt.
I- ReceivingController-ticket-2	UC0040	Der Mitarbeiter ist angemeldet.	Eine Sitzung ohne einem Auftrag.	Anfrage abgelehnt mit BAD_REQUEST .

Finanzen

Anna Lopatkina

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
I- FinanceController-showFinanceForm-01	UC0030	Ein autorisierter Mitarbeiter, der als Admin angemeldet ist.	Admin wählt "Bilanzen anzeigen" in der Bilanzübersicht aus.	Admin bekommt die Bilanzen angezeigt.
I- FinanceController-showFinanceForm-02	UC0030	Ein autorisierter Mitarbeiter, der als Admin angemeldet ist.	Admin wählt "Gewinn anzeigen" in der Bilanzübersicht aus.	Admin bekommt den Gewinn angezeigt und kann ihn abführen.
I- FinanceController-AddNewFinanceItemInForm-01	UC0030	Ein autorisierter Mitarbeiter hat neue Finanz Eintrag noch nicht hingefügt.	Mitarbeiter wählt "Finance item hinzufügen" in der Bilanzübersicht aus.	Finance Eintrag wird hingefügt
U- FinanceController-AddNewFinanceItemInForm-02	UC0030	Ein autorisierter Mitarbeiter hat neue Finanz Eintrag noch nicht hingefügt.	Mitarbeiter wählt "Finance item hinzufügen" in der Bilanzübersicht aus. Finance Eintrag ist schon in System.	Fehler Meldung. Finance Eintrag wird nicht noch mal hingefügt.

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
U-FinanceController-removeFinanceItem-01	UC0030	Finanz Eintrag ist in Bilanzübersicht scheinbar.	Authorisierte Mitarbeiter wählt "Finance item löschen" in der Bilanzübersicht aus.	Finance Eintrag wird gelöscht.

Mitarbeiter

Kien Dang Tran

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
U-Staff-salary-01	-	Mitarbeiter bekommt Lohn.	-	Korrektter Lohn.
U-Staff-salary-02	-	Mitarbeiter bekommt kein Lohn.	-	0
U-Staff-creation-01	UC0020	Mitarbeiter existiert noch nicht im System.	UserAccount, nichtleerer Vor- und Nachname	Staff Objekt wird erstellt.
U-Staff-creation-02	UC0020	Mitarbeiter existiert noch nicht im System.	UserAccount, leerer Vor- oder Nachname	Assert.hasText Fehlermeldung
I-UserManagement-createUser-01	UC0020	Mitarbeiter ist noch nicht registriert.	Valides RegistrationForm	Staff mit UserAccount wird angelegt.
I-UserManagement-createUser-02	UC0020	Mitarbeiter ist noch nicht registriert.	RegistrationForm ist null	Assert.notNull Fehlermeldung
I-UserManagement-deleteUser-01	-	Mitarbeiter ist im System.	korrekter userName und id	Mitarbeiter wird aus StaffRepository und UserAccountManager entfernt.
I-UserManagement-deleteUser-02	-	Mitarbeiter ist nicht im System.	userName und id	Nichts
I-UserManagement-findByUsername-01	-	Mitarbeiter ist im System.	richtiger userName	korrekter UserAccount

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
I-UserManagement-findByUsername-02	-	Mitarbeiter ist im System.	falscher userName	kein UserAccount
I-UserManagement-getAllUsers-01	-	Mitarbeiter existieren im System.	-	Alle Staff Objekte im System
I-UserController-login-01	UC0010	Benutzer ist nicht authentifiziert.	richtiger Benutzername und richtiges Passwort	Benutzer wird auf die Homepage verlinkt und ist angemeldet.
I-UserController-login-02	UC0010	Benutzer ist nicht authentifiziert.	Benutzername oder Passwort ist falsch	Es erfolgt eine Fehlermeldung.
I-UserController-logout-01	UC0010	Benutzer authentifiziert.	Logout in der Navigationsleiste	Benutzer wird abgemeldet und wird auf Login verlinkt.
I-UserController-registration-01	UC0020	Neuer Mitarbeiter ist noch nicht im System registriert.	Admin gibt neue Benutzerdaten bei der Registrierung ein. Benutzername ist noch nicht im System.	Ein neues Benutzerkonto wird angelegt
I-UserController-registration-02	UC0020	Neuer Mitarbeiter ist noch nicht im System registriert.	Admin gibt neue Benutzerdaten bei der Registrierung ein. Benutzername ist schon im System.	Es erfolgt eine Fehlermeldung und es wird kein neues Benutzerkonto angelegt.
I-UserController-users-01	-	Ein autorisierter Mitarbeiter, der aber kein Admin ist.	Anfrage zur Mitarbeiterübersicht	Anfrage wird wegen @PreAuthorize abgelehnt.
I-UserController-users-02	-	Ein autorisierter Mitarbeiter, der als Admin angemeldet ist.	Anfrage zur Mitarbeiterübersicht	Mitarbeiterübersicht wird angezeigt.

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
I-UserController-delete-01	-	Ein autorisierter Mitarbeiter, der als Admin angemeldet ist.	Entfernen eines Mitarbeiters	Mitarbeiter wird aus dem System entfernt.

Räume

Till Große

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
U-RoomController-addRoom-01	UC0010	Es wurden noch keine Räume hinzugefügt.	Valider Name, valide Dienstleistung	Raum wird hinzugefügt
U-RoomController-addRoom-02	UC0010	Es wurden noch keine Räume hinzugefügt.	Ein Eingabefeld leer	IllegalArgumentException, Error Meldung wird angezeigt
U-RoomController-addRoom-02	UC0010	Es wurden Räume hinzugefügt.	Valider Name, valide Dienstleistung	Raum wird hinzugefügt, Slots werden automatisch
U-RoomController-deleteRoom-01	UC0020	Es wurden Räume hinzugefügt, Slots erstellt.	Raum löschen	Raum wird gelöscht, Slots werden gelöscht
U-RoomController-deleteRoom-02	UC0020	Es existieren keine Räume.	keine Eingabe möglich	Es können keine Räume gelöscht werden.
U-RoomController-listFreeSlots-01	UC0030	Es existieren Räume und Slots.	Freie Slots anzeigen	Alle ungebuchten Slots werden angezeigt.
U-RoomController-listFreeSlots-02	UC0030	Es existieren Räume aber alle Slots sind gebucht.	Freie Slots anzeigen	Es gibt keine Slots zum anzeigen.
U-RoomController-bookFreeSlot-01	UC0040	Es existieren Räume mit freien Slots.	Freien Slot Buchen	Der nächste freie Slot bekommt einen Eintrag von einem Raum und wird auf gebucht gesetzt.
U-RoomController-bookFreeSlot-02	UC0040	Es existieren Räume, aber alle Slots sind gebucht.	Freien Slot Buchen	Kein Slots verändert sich.

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
U-RoomController-debookFreeSlot-01	UC0050	Es existieren Räume mit freien Slots.	Slot Buchen zurückbuchen	Der nächste freie Slot verliert seinen Eintrag von einem Raum und wird auf frei gesetzt.
U-RoomController-debookFreeSlot-02	UC0050	Es existieren Räume und kein Slot ist gebucht.	Slot Buchen zurückbuchen	Kein Slots verändert sich.
U-TimeTableEntry-Constructor-01	UC0060	Es existieren keine Einträge.	Raum erstellen, Startzeit = Endzeit	IllegalArgumentEx ception
U-TimeTableEntry-Constructor-02	UC0060	Es existieren keine Einträge.	Raum erstellen, Startzeit > Endzeit	IllegalArgumentEx ception
U-TimeTableEntry-Constructor-03	UC0060	Es existieren keine Einträge.	Raum erstellen, Startzeit < Endzeit	Eintrag wird erstellt.
U-TimeTableEntry-Constructor-04	UC0060	Es existieren keine Räume.	Raum erstellen, keine Eingabe	Kein Raum wird erstellt, Meldung im Frontend
U-TimeTableEntry-Constructor-05	UC0060	Es existieren Räume.	Raum erstellen, keine Eingabe	Kein Raum wird erstellt, Meldung im Frontend
U-Room-Constructor-01	UC0070	Es existieren keine Räume.	Raum erstellen, keine Eingabe	Kein Raum wird erstellt, keine Einträge werden erstellt, Meldung im Frontend
U-Room-Constructor-02	UC0070	Es existieren Räume.	Raum erstellen, keine Eingabe	Kein Raum wird erstellt, keine Einträge werden erstellt, Meldung im Frontend
U-Room-Constructor-03	UC0070	Es existieren keine Räume.	Raum erstellen, valide Eingabe	Räume werden, Einträge werden erstellt
U-Room-Constructor-04	UC0070	Es existieren Räume.	Raum erstellen, valide Eingabe	Räume werden, Einträge werden erstellt

Inventar

Tobias Köllner

ID	Anwendungsfall	Vorbedingungen	Eingabe	Ausgabe
I-Catalog<Material>findAll		Sicherstellung das alle nötigen Materialien im Catalog sind.	Catalog<Material>	Alle Materialien sind vorhanden.
I-Catalog<Material>correctCategory		Sicherstellung das alle Materialien der korrekten Kategorie zugeordnet sind.	Catalog<Material>	Alle Materialien haben die korrekte Kategorie.
I-CatalogController WebIntegration		Simulation von Requests	MockMvc, CatalogController	Nicht-leerer Catalog wird angezeigt.
I-Catalog sampleController		Überprüfung ob angezeigter Catalog stimmt mit tatsächlichem Catalog überein.	CatalogController	Korrektter Catalog wird angezeigt.
I-InventoryController publicAccess		Nutzer versucht auf Lager zuzugreifen.	Nutzer ohne Adminrechte	Zugriffsverweigerung aufgrund fehlender Rechte.
I-InventoryController adminAccess		Admin versucht auf Lager zuzugreifen.	Adminrechte	Zugriff wird zugelassen.
I-InventoryController restock		Sicherstellung das das korrekte InventoryItem mit der korrekten QuantityMetric nachgefüllt wird.	Adminrechte	Quantity wird nachgefüllt.
I-InventoryController consume		Sicherstellung das das korrekte InventoryItem mit der korrekten QuantityMetric verbraucht wird.	Adminrechte	Quantity wird verbraucht.
U-InventoryInitializer		Ein Inventar wird mit null initialisiert.	Adminrechte	IllegalArgumentException