

06.04.2021



WARENEINGANS APPLIKATION

ENTWICKLER DOKUMENTATION

VORGELEGT VON: MARTIN RANFT
BRANDSCOMMERCE GMBH
CHRISTINSTR 3
75177 PFORZHEIM
GERMANY

WARENEINGANGS APPLIKATION

INHALTSVERZEICHNISS

INHALT

Wareneingangs Applikation.....	1
Inhaltsverzeichnis.....	1
GUI Aufteilung Login	3
Verwendete Funktion->Login->LoginCom	3
Konstruktor.....	3
Authcheck Methode	3
Verwendete Funktion->Communication->JsonConnect.....	4
ComToApi.....	4
ComToApiLogin	4
Verwendete Funktion->Communication->Sendmail	4
send.....	4
GUI Belegsearch.....	5
GUI belegWorksheet.....	6
Verwendete Funktion->order	7
GUI dialogBox.....	7
Verwendete Funktion->SettingBelegdata.....	7
ChangeLagerPlatz.....	7
SendBelegData.....	7
SetBelegData.....	7
SetCountGescannt	7
Der Data Ordner	8

GUI AUFTEILUNG LOGIN

button_login_Click -> auslösen der Login Methode

combobox_firma_SelectionChanged -> Ändern des Loginlayout anhand des Verben und Sichtbarmachen des jeweiligen Grid's, mit Farbabänderung des Login Buttons.

passwordbox_passwort_KeyDown -> Auslösen des Key Events beim Betätigen der Enter Taste, dies löst den button_login_Check aus

Window_MouseDown -> lässt das GUI Fenster sich per Drag und Drop bei gedrückter linken Maustaste verschieben.

VERWENDETE FUNKTION->LOGIN->LOGINCOM

Beschreibung einiger Login Methoden

KONSTRUKTOR

Dieser greift über die übergebene Variable window die vom User getätigten Eingaben ab. übergibt die benötigten Daten (Username u. Password) dann an die Authcheck Methode, sollte der Login fehlgeschlagen sein oder keine Firma ausgewählt worden sein wird dem User dies per MessageBox angegeben. Ansonsten wird die Methode Belegselection entsprechend aufgerufen.

AUTHCHECK METHODE

Per try wird der Username, Password sowie company an die Funktion->Communication->JsonConnect übergeben. Im catch Fall wird über die Mail Class eine E-Mail an die IT-Abteilung gesendet mit entsprechenden Fehler Code.

VERWENDETE FUNKTION->COMMUNICATION->JSONCONNECT

Beschreibung der Json Kommunikation

COMTOAPI

Die function übernimmt die zu benutzende end URL und sendet entsprechend ein in JSON Codiertes Dictionary in <string, string> Format an die entsprechende API.

ACHTUNG: Zur Verwendung im Livebetrieb muss die bool Variable TestModus auf false gesetzt werden, ansonsten gehen alle Daten an das Testsystem!!!!

Bei einem Fehler bekommt der User eine für Ihn verständliche Fehleransage das die Übertragung nicht geklappt hat, die IT Abteilung wird im automatisch von der Applikation über den Fehler mit der Exception informiert.

COMTOAPILOGIN

Die übergebenen daten werden zusammen mit dem festgelegten apiKey an die API-Schnittstelle übergeben. Der JSON response wird als string wiedergegeben. Die Exception wird wie in der ComToApi methode gehandelt.

VERWENDETE FUNKTION->COMMUNICATION->SENDMAIL

Beschreibung der E-Mail-Send Funktion.

SEND

Die übergebenen daten werden hier zusammengefasst und per smtpClient versendet über eine bestehende E-Mail Adresse.

Im Subject wird immer das Modul in dem der Fehler aufgetreten ist benannt gefolgt von der Exception als message.

GUI BELEGSEARCH

Mit dem aufruf des Belegsearch GUI wird der MauveAuthApi _user beschrieben und der First Name und last Name zusammengesetzt und in das label label_username.Content geschrieben.

Ein Timer für den Autologout wird auch gestartet der den User nach 30 min Automatisch ausloggt.

SetListView -> lädt die von der API-Schnittstelle übergebenen und übergibt diese an den listview_beleg_auswahl.ItemsSource der per Pattern Matching von XAML (Bsp.: DisplayMemberBinding="{Binding belegnr}") die Einträge erstellt. Diese werden gleich durch die Funktion->Filter->SortAdorner absteigenden der „belegnr“ Sortiert.

ausgabenColumnHeader_Click sorgt das beim klick auf einen Header im ListView die Sortierung entsprechend geändert wird.

BelegFilter -> filtert bei Eingabe von Zahlen die ListView Ansicht.

button_belegauswahl_Click -> öffnet das Beleg Fenster mit den entsprechenden Beleg daten. Außerdem stoppt die Funktion den Autologout und setzt das Belegsearch Fenster auf IsEnable false. im Fall eines Problems mit dem Beleg wird nur ein Fehler dem Benutzer angezeigt aber nicht die IT Informiert.

icon_aktualisieren_MouseLeftButtonDown -> ruft die SetListView Methode auf zur Aktualisierung.

image_exit_MouseLeftButtonDown -> ruft die Logout Methode zum ausloggen des User auf.

listview_beleg_auswahl_MouseDoubleClick -> Löst die button_belegauswahl Event aus

t
listview_beleg_auswahl_SelectionChanged -> Aktivieret oder Deaktiviert die Funktion zum Beleg öffnen bei mehrfach auswahl von Belegen.

LogoutTimerEvent -> Prüft ob das Belegsearch Fenster Active und geöffnet ist bevor die Logout Methode ausgelöst wird. Auch wird überprüft, ob dieses auf IsEnable = true gesetzt ist.

textbox_beleg_userinput_KeyUp -> löst die Methode button_belegauswahl aus sobald die Entertaste in der textbox_beleg_userinput losgelassen wird.

textbox_beleg_userinput_TextChanged -> sorgt für die Aktualisierung der ListView Ansicht bei usereingaben.

Logout -> Öffnet das Login Fenster und schließt das Belegsearch Fenster, also wird der Garbage Collector aufgerufen zum Säubern.

GUI BELEGWORKSHEET

Detailfenster mit Beleg Details aus der Datenbank.

Hier werden Ersteller und Datum sowie Bemerkungen und Lieferant angezeigt in der oberen Hälfte. Der „Beleg ist komplett“ Button steht nur Usern mit dem flag Admin zur Verfügung (das flag kommt vom User Login).

Die textbox_sku wird gleich nach dem Aufruf fokussiert zum Schnellen Abarbeiten der posten im listview_BelegDaten Feld. Die befüllen wird durch ein Binding in XAML erreicht.

Button_admin_beleg_complete_OnClick -> hier werden allen posten im ListView in der Column „Gescannt“ auf die listitem.anzahl gesetzt die der „Anzahl Bestellt“ entspricht.

button_close_save_beleg_Click -> Händelt das Übertragen des Beleges so wie das Schließen und Löschen der InitSessionData. Der Benutzer wird vorher per MessageBox gefragt, ob er den Vorgang wirklich machen möchte mit Warnhinweis das der Beleg danach nicht mehr in der Applikation geändert werden kann.

button_close_without_saving_Click -> schließt das Beleg Fenster und löscht die InitSessionData. Der Benutzer wird hier daraufhin nochmals gefragt, ob er den Beleg ohne Änderungen verlassen möchte.

button_set_lagerplatz_Click -> ruft die ChangeLagerPlatz Methode in Funktion->SettingBelegdata->Beleg auf, mit Verweis auf den Hauptlagerplatz.

button_set_reservelagerplatz_Click -> ruft die Methode ChangeLagerPlatz mit Verweis auf den Reserveplatz auf.

GridViewColumnHeader_Click -> sortiert das ListView Element, dies wurde nachträglich eingebaut und noch nicht getestet. (Sortierung muss getestet und Fehler behoben werden sofern Zeit vorhanden.)

BelegDaten_SelectionChanged -> stellt sicher das auch ein Posten im Beleg ausgewählt wurde, bevor ein lager- oder Reservelagerplatz geändert werden kann.

ContextListView -> lässt das Ändern des Gescannt counter über eines ContextMenu zu. Hierzu wird SetCountGescannt Methode in Beleg verwendet.

textbox_sku_KeyDown -> Trigger Event wen in der textbox_sku der Enter Button gedrückt wird.

worksheet_MouseDown -> dient des verschieben des Fensters.

die interne class PostenView : INotifyPropertyChanged dient dazu den Hintergrund entsprechend dem Status vom Gescannt Counter anzupassen.

VERWENDETE FUNKTION->ORDER

Dient zur Kommunikation mit der API, um ein OrderList Objekt zu erhalten und zu verarbeiten.
sollten hier Fehler entstehen werden diese direkt an die IT Abteilung geleitet.

GUI DIALOGBOX

Die selbsterstellte dialogBox is der MessageBox nachempfunden mit dem Vorteil das über den booleschen wert „numbersonly“ eine Überprüfung veranlasst, ist der Wert true übertragen so wird anhand der RegEx Funktion die Benutzer Eingabe geprüft und nur Zahleneingaben zugelassen.
Über die Tooltip funktion wird der Benutzer bei falsch Eingabe informiert das nur zahlen zulässig sind.

VERWENDETE FUNKTION->SETTINGBELEGDATA

Auflistung der verschiedenen Methoden zur Beleg Bearbeitung.

CHANGELAGERPLATZ

Ändert den im BelegWorksheet ausgewählten ListView Item eintrag an der stelle des Lager- oder Reserverlagerplatzes je nach Angabe des Lagerplatztype's (1 Lagerplatz, 2 Reserverlagerplatz).
Der User nimmt die Änderung über die selbstgestaltete DialogBox Eingabe vor.
Antwortet die DialogBox mit einem true so werden die Lagerdaten gleich an die API gesendet.

SENDBELEGDATA

Bereitet die Daten zur Versendung über JSON an die API-Schnittstelle vor, per try-catch werden die daten dann Übertragen bei einem Fehler wird die IT sofort per E-Mail informiert.
Der User wird per MessageBox darüber informiert, ob die Daten gespeichert werden konnten oder ob ein Fehler aufgetreten ist.

SETBELEGDATA

Spricht per JSON die API-Schnittstelle an und lässt sich den gewünschten beleg übergeben.
Dieser wird dann nach einer null or Empty Prüfung in den BelegWorksheet entsprechenden ListView eingetragen und angezeigt

SETCOUNTGESCANNT

Ruft die eigenerstellte DialogBox auf mit numbersonly = true, um die Gescannt anzeige zu ändern.
Danach werden die SKU nummer und der Count an die Methode SkuScan übergeben.

SKUSCAN

Sucht den Artikel der gescannt wurde.
Über die Eingabe einer forceSKU und forceCount kann ein bestimmter Artikel nur geändert werden.

GETDATABYUSERINPUT

Prüft ob der eingegebene wert zu lang ist und kürzt diesen dann (Barcode benutzen gerne auffüllende nullen) wird dieser gekürzt.
Außerdem prüft die Methode ob es sich bei der Eingabe um eine ArtikelNr, Barcode oder einen barcodeSecond handelt und gibt dann den entsprechenden Eintrag wieder.

DER DATA ORDNER

Im Data Ordner sind sämtliche Klassen vorhanden die für die Kommunikation mit der API-Schnittstelle benötigt werden damit der JSON string in eine Verständliche und bearbeitete Klasse oder auch Dictionary gewandelt werden kann.