

ESDK BEST PRACTICE WORKSHOP

Übungsaufgabe

abas  ERP

1 ÜBERSICHT

Wir wollen einen Ersatzteilkatalog als Zusatzdatenbank in abas anlegen. Es soll im Artikelstamm möglich sein mittels Klick auf einen neuen Button „Ersatzteil anlegen“ einen neuen Datensatz in der ZDB anzulegen. Wird dieser Button gedrückt, wird im Hintergrund ein neues Ersatzteil angelegt und im Modus bearbeiten geöffnet. Wenn bisher noch kein Ersatzteil zu dem Artikel angelegt wurde, wird der Kenner „Ersatzteil angelegt“ im Artikel gesetzt. Hinzukommend soll es ein Infosystem geben, welches die Ersatzteile selektiert und mittels dem der Verbrauch der Ersatzteile erfasst wird. Zum Erfassen des Verbrauchs gibt es in der Infosystemtabelle die beschreibbaren Felder „Datum“(GD2), „Grund“ (AZL) und „Verbrauch erfassen“(BU3). Wird der Button Verbrauch erfassen gedrückt, werden in der Tabelle des Ersatzteils die beiden Felder gespeichert.

2 EINRICHTUNG

Ziele:

- GIT
- Issue-Management
- Einrichtung eines ESDK Projekts

1. Gemeinsam: Lest euch das Konzept (Detailbeschreibung bei 5.) durch und unterteilt die Aufgaben in kleinere Issues (die bestenfalls so unterteilt sind, dass es immer drei Issues gibt die parallel bearbeitet werden können und nicht aufeinander warten müssen). Legt diese Issues in eurem GitHub/Bitbucket Board an und weist sie Bearbeitern zu.
2. Jeder einzeln: Legt einen Ordner „myFirstApp“ an, in diesem die Ordner „workspace“ und „sparePartApp“. Öffnet euer abasTools und wählt den workspace Ordner aus.
3. Gemeinsam/1PC: Erstellt ein ESDK Projekt. Das Projekt soll in dem Ordner „sparePartApp“ liegen:

Appid = spart

Vendor = ae

versionsRange: ["2018r1n00-2018r4n16"]

language: „D“

Package Name: de.abas.custom.owsp.spart

4. Gemeinsam/1PC: Erstellt ein Git Repository in Github, öffnet die Gitbash in eurem esdk Projektordner und fügt eurer Repository hinzu. Anschließend das esdk Projekt mit commit und push in eurer Remote Repository schieben.

5. Gemeinsam/1PC: Gitignore anpassen + commit & push

```
.gradle/
build/
out/
bin/
.project
.classpath
.settings/
gradle.properties
.gradle/
```

6. Gemeinsam/1PC: Powershell im Projektordner öffnen & initGradleProperties.sh ausführen.
7. Gemeinsam/1PC: Importiert das esdk Projekt in euren Workspace (new Gradle Project).
8. Jeder einzeln: Die Teammitglieder, die nicht die Projekteinrichtung auf ihrem Laptop gemacht haben, clonen sich jetzt das Repository und importieren es als gradle Projekt.

3 ABAS INFRASTRUKTUR

Ziele:

- Build.gradle pflegen.
 - Überblick und Bedeutung der Dateien im Resources Ordner
1. Gemeinsam/1PC: Legt auf eurem Docker Mandanten den Arbeitsbereich owspart an und fügt diesen in die build.gradle ein.
 2. Gemeinsam/1PC: Legt eine Aufzählung SPUREASON (inkl. Wertemengenbezeichner) an. Exportiert eure angelegten Daten in das esdk Projekt.
 3. Gemeinsam/1PC: Legt die Zusatzdatenbank ErsatzteileApp an
 4. Gemeinsam/1PC: Artikel Datenbank erweitern
Legt die Variable „Ersatzteil anlegen“ – BU3 und „Ersatzteil vorhanden“ – boolean.
Anschließend Export in euer Projekt.
 5. Gemeinsam/1PC: Masken
Erweitert die Artikelmaske um die Felder. Fügt die angelegten Felder der ZDB in die jeweilige Gruppenmaske ein. Anschließend Export in euer Projekt.
 6. Gemeinsam/1PC: Legt einen Datensatz mit Suchwort „KONFIG“, Identnummer „1“ und Bezeichnung „Konfiguration für ErsatzteileApp“ in eurer Konfigurationsgruppe an, exportiert den Datensatz mit diesen drei Feldern in euer Projekt.

7. *Gemeinsam/1PC*: Infosystem SPUSAGE

Verwendung von Ersatzteilen in Arbeitsbereich „owsp“ anlegen und Maske anpassen. Export in das Projekt.

4 ENTWICKLUNG

Ziele:

- Java-Programmierung
- Clean Code
- Automatisierte Tests

Setzt die Business Logik, gemäß eurer definierten Issues um. Hierbei soll jede Issue in einem eigenen Branch entwickelt werden, damit die Aufgaben parallel bearbeitet werden können. Nach Fertigstellung einer Issue, soll der entsprechende Branch als Pull-Request angelegt und von den Teammitgliedern gereviewt werden. Sind alle Qualitätskriterien erfüllt, kann der Branch des Pull-Request gemerged werden.

Abnahmekriterien für Reviewer:

- Aufgabenbeschreibung der Issue erfüllt.
- Jeder Reviewer hat den Code verstanden und kann diesen nachvollziehen. Falls nicht, muss entweder die unklare Stelle refactored werden, damit diese verständlich ist oder die Stelle mit einem erklärenden Kommentar erläutert werden. (Ersteres wird bevorzugt).
- Funktionalität durch automatische Tests abgesichert.
- Bei Merge Konflikten: Vorher im Branch lösen

5 DETAILBESCHREIBUNG

5.1 ABAS RESSOURCEN

5.1.1 AUFZÄHLUNGEN

„SPUREASON“ – „Grund der Verwendung“

Aufzählungselement: Verschleiß, Defekt, regulärer Tausch, Rückruf(alle Suchwörter mit SP beginnend).

Leerzeile soll in der Aufzählung wählbar sein.

5.1.2 DATENKBANKEN

Zusatzdatenbank: ErsatzteileApp

Gruppe 0 = Konfiguration

Gruppe 1 = Ersatzteil

Konfiguration (nur Kopffelder):

Datenbankname für Kommandos	GL40	skip	zeigen änderbar
Datenbanknummer	GL3	skip	zeigen änderbar
Datenbankname	A177	skip	zeigen änderbar

Ersatzteil

Kopffelder

Artikel	Verweis Artikel
Lieferant	Verweis Lieferant
Lieferantenartikelnummer	GL14

Tabellenfelder

Verwendungsdatum	GD2
Verwendungsgrund	AZLxxxxxxxx

5.1.3 INFOSYTEME

Infosystem: SPUSAGE

Kopffelder

Programmnamen für individuelle Fops anzeigen B1 nicht in Maske name yspartshowfopname

Ersatzteil	Verweis zdb
Artikel	Verweis Artikel
Letzte Verwendung von	GD2

Letzte Verwendung bis	GD2
Verwendungsgrund	Aufzählung aus 2.

Tabellenfelder

Ersatzteil	Pzdb:1
Ersatzteil Bezeichnung	PTzdb:1 (alias zu Ersatzteil)
Artikel	Verweis Artikel
Letzte Verwendung am	GD2
Letzter Verwendungsgrund	AZLxxxxxxx
Neue Verwendung am	GD2 – beschreibbar
Grund der Verwendung	AZLxxxxxxx – beschreibbar
Verwendung erfassen	BU4

5.2 MASKE ARTIKEL

Der Button „Ersatzteil anlegen“ soll nur im Modus „bearbeiten“ gedrückt werden dürfen. Ansonsten erfolgt eine Hinweismeldung „Anlage nur im bearbeiten-Modus erlaubt“. Ist der User im Bearbeiten-Modus, wird im Hintergrund ein neuer Ersatzteildatensatz angelegt.

Vorbelegungen:

- Suchwort: SP + Artikelsuchwort
- Bezeichnung: „Ersatzteil zu Artikel“ + Artikelnummer
- Artikel = Aktueller Artikel
- Lieferant und Lieferantenartikelnummer von Lieferant 1 der im Artikelstamm hinterlegt ist.

War die Anlage nicht erfolgreich soll eine Meldung „Ersatzteil konnte nicht angelegt werden“ erscheinen. Wenn die Anlage erfolgreich war, wird das Verweisfeld auf das Ersatzteil in dem Artikelstamm mit der neu angelegten ID befüllt und der neue Datensatz öffnet sich im Bearbeiten-Modus.

5.3 MASKE KONFIGURATION

Bei jedem Maskeneintritt der Konfiguration sollen die drei Skipfelder mit den Informationen zu der Zusatzdatenbank befüllt werden. Dazu müssen aus der aktuellen Vartab die Werte für „Datenbankname für Kommandos“, „Datenbanknummer“ und „Datenbankname“ ermittelt werden. Da esdk die Zusatzdatenbanken immer in die nächst freie Zusatzdatenbank schreibt, dient diese Info als Unterstützung für den Kunden und die Berater, um schnell zu sehen welche die aktuelle Datenbank ist.

5.4 MASKE ERSATZTEILE

Hier soll eine Neuanlage von der GUI aus verboten sein. Im Bearbeiten-Modus dürfen nur die Felder Lieferant und Lieferantenartikelnummer von Hand geändert werden.

5.5 INFOSYSTEM SPUSAGE

Beim Eintragen des „Von“ und „Bis“ Datums im Kopf soll eine Plausibilitätsprüfung stattfinden, dass das „Vor“ Datum nicht nach „Bis“ Datum liegt und umgekehrt. Bei Drücken auf den Startbutton sollen alle Ersatzteile selektiert und in die Tabelle geschrieben werden, auf welche die im Kopf angegebenen Selektionskriterien zutreffen.

Füllen der Tabellenfelder:

- **Ersatzteil / Bezeichnung:** Aktuelle ID des Selektionsergebnisses
- **Artikel:** Der im Ersatzteil hinterlegte Artikel
- **Letzte Verwendung am:** Das aktuellste Datum aus der Tabelle des Ersatzteils
- **Letzter Verwendungsgrund:** Der Grund der in der ermittelten „letzte Verwendung am“ Zeile steht
- **Neue Verwendung am:** Vorbelegt mit heute
- **Grund der Verwendung:** Leer

Drücken des „Verwendung erfassen“ Buttons

Der Button darf nur gedrückt werden wenn Datum und Grund in der Zeile angegeben sind. Ist die Bedingung erfüllt, werden im Hintergrund die beiden Werte am Ende der Tabelle des Ersatzteils eingefügt und gespeichert. Der User erhält eine Erfolgsmeldung. Konnten die Änderungen nicht gespeichert werden, wird der User auch darüber informiert.

5.6 ZUSATZ: UCM- FILE

Legt ein neues ucm-file in eurem Mandanten mit den Kommandos „Ersatzteil zeigen“ und „Ersatzteilverwendung erfassen“ an. Ersatzteil zeigen öffnet die Maske Ersatzteile ohne Eintrag und Ersatzteilverwendung erfassen das Infosystem ohne Eintrag.

Kopiert das ucm-file in euren Base-Ordner. Ersetzt den Aufruf der auf eure Zusatzdatenbank geht durch einen eindeutigen Kenner, z.B. <%OWSPDB%>. Erweitert eure Konfiguration um den Button „Konfiguration abschließen“. Wird der Button gedrückt, wird das ucm-File zeilenweise von eurem Programm eingelesen und der eindeutige Kenner durch den korrekten ucm-Aufruf der Zusatzdatenbank ersetzt. Hintergrund: Das ucm-File in eurem Projekt hat die Datenbanknummer, welche auf eurem Testmandanten vergeben wurde hinterlegt. Hier muss evtl. bei jeder Installation etwas anderes stehen.

6 LÖSUNG