Ze względu na prace serwisowe na serwerze firmowym i wynikający z tego zablokowany dostęp do systemu, nie byłem w stanie zmienić swojej aplikacji. Z tego samego powodu nie posiadam również dokładnego kodu ABAP, który był użyty w aplikacji. Załączony kod jest kodem poglądowym.

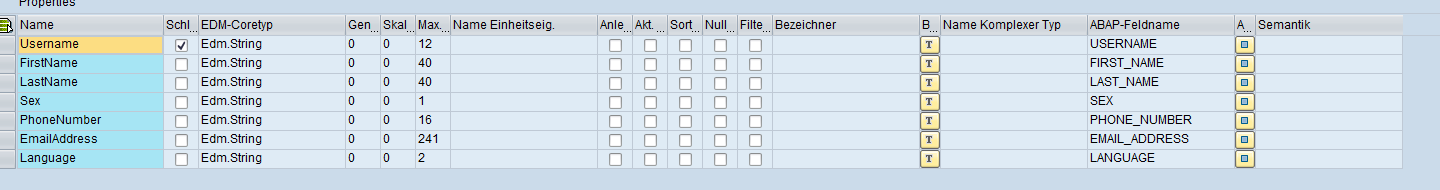
## Nr indeksu: 147728

## Technologie i języki programowania: ABAP, SAPUI5, XML, JS

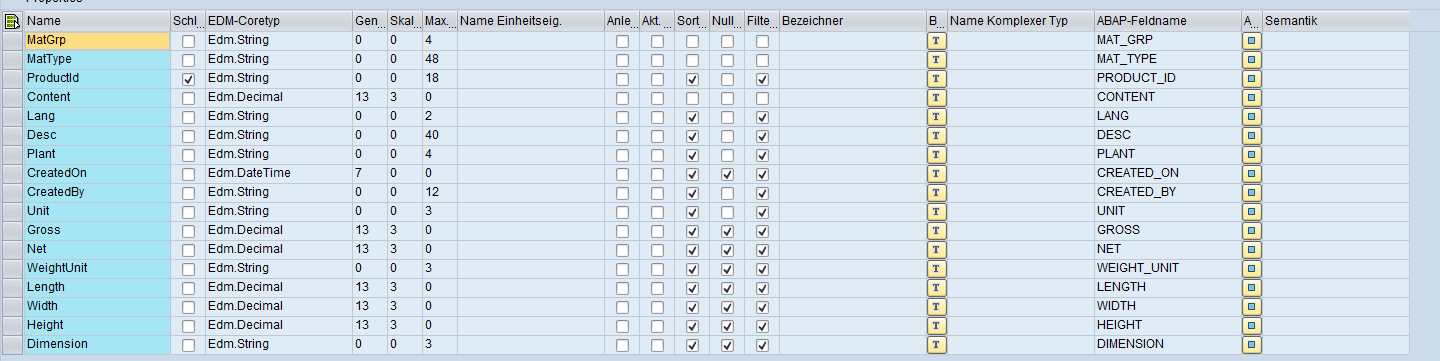
## Struktura encji.

Projekt składa się z trzech zestawów encji:

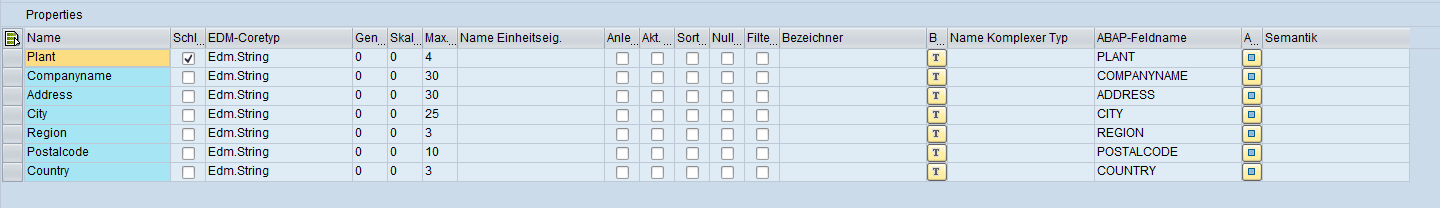
ContactSet



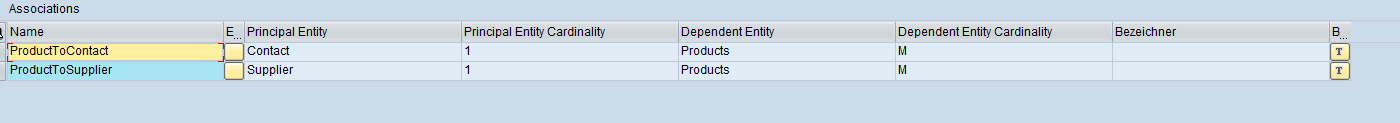
ProductSet



SupplierSet



Połączenia encji:



ContactSet (kolumna Username) z ProductSet (kolumna CreatedBy),

ProductSet (kolumna Plant) z SupplierSet (kolumna Plant).

## Ściąganie danych.

Żeby uzyskać dane, trzeba oprogramować odpowiednie metody, które są używane zarówno przez klienta REST jak i późniejszą aplikację Fiori.

Oprogramowuje się osobno metody GET, POST, MERGE czy DELETE. Jeżeli chodzi o metodę GET, należy oprogramować osobno „GetEntitySet” jak i „GetEntity”. Pierwsza zwraca zestaw danych, druga jest używana w momencie wyboru pojedynczego rekordu.

Poglądowe opracowanie „GetEntitySet” dla ProductSet.

Data:

Lt\_product TYPE STANDARD TABLE OF zcf\_product\_set.

REFRESH: lt\_product, et\_entityset.

SELECT a~matnr a~maktx b~ernam b~ersda b~matkl b~bstme

FROM makt as a

INTO CORRESPONDING FIELDS OF TABLE lt\_product

UP TO 20 ROWS

LEFT JOIN mara as b  
ON a~matnr = b~matnr.

IF sy-subrc = 0.

INSERT LINES OF lt\_product INTO TABLE et\_entityset.

ENDIF.

Powyższy kod wybiera nam pierwsze 20 rzędów z tabeli makt i dobiera nam odpowiadające im dane z tabeli mara

Objaśnienie kodu:

Lt\_product to tabela zawierająca pola ProductSet. Typ tabelaryczny zcf\_produt\_set został osobno stworzony. Zawiera on wszystkie wykorzystywane pola.

Polecenie REFRESH czyści tabele.

Następnie jest select, który wybiera dane.

IF sy-subrc = 0 – pole subrc ze struktury SY zawiera informacje o błędach. Jeżeli select przebiegł pomyślnie, ma ona wartość zero. W takim wypadku wybrane dane są przekazywane do et\_entityset, czyli do tabeli, która jest eksportowana z metody.

W analogiczny sposób zostały stworzone metody dla pozostałych zestawów danych.

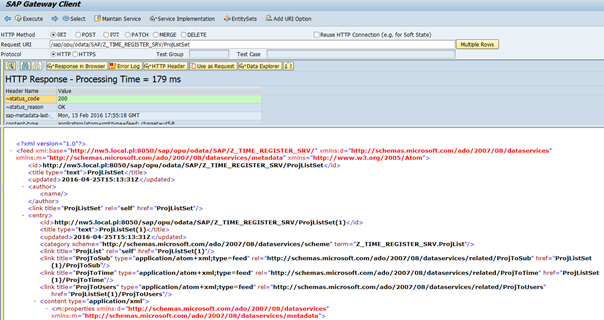
Metoda „GetEntity” różni się od „GetEntitySet” tym, że występuje tam „SELECT SINGLE”, który wybiera pojedynczy rekord w zależności od zadanych warunków. Przy wywoływaniu metody „GetEntity” podawane są klucze, po których przeprowadzony będzie SELECT. W praktyce takim kluczem jest np.: numer materiału na który nacisnął użytkownik aplikacji FIORI.

Dane wybrane przy SELECT SINGLE są przekazywane do struktury (W SAPie struktura to pojedyncza linia tabeli), która jest przekazywana na zewnątrz.

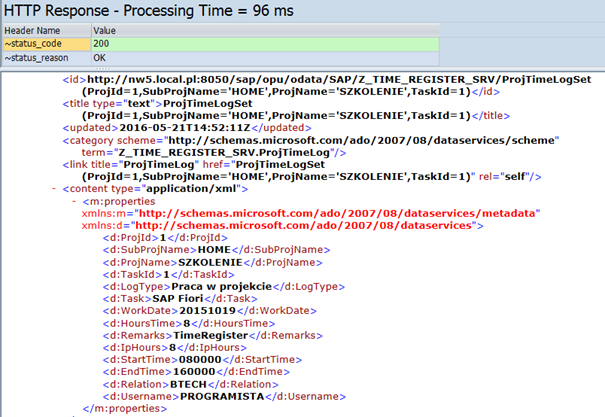
## Klient REST.

Z powodu braku możliwości dostępu do serwera, jestem w stanie dostarczyć jedynie screeny wyniku zapytania. Są to jednak jedynie screeny poglądowe. Nie prezentują one danych używanych w projekcie.

Wygląd klienta:



Wynik zapytania:



Informacją, że zapytanie przebiegło pomyślnie jest wartość „~status\_code” = 200.

Wynik jest zwrócony w formacie XML.

## HUB

Rolę HUBa ma aplikacja napisana we FIORI. Wykorzystuje ona XML oraz JavaScript.

Podstawą aplikacji jest tzw. MasterView, czyli lista wszystkich pobranych danych (wyświetla ona wybrane przez nas wartości z każdego wiersza tabeli. Korzysta ona głownie z metod ‘GetEntitySet”.

Widok szczegółowy, czyli Detail.View używa „GetEntity” do pobrania brakujących czy dodatkowych danych z innych zestawów danych.

Całość kodu aplikacji dostępna jest w repozytorium.

## Napotkane problemy:

Podczas sprawdzania napisanego kodu w kliencie REST często napotyka się problem „zer wiodących” czyli dopełnienia zerami do zadanej długości pola. Np.: Pole liczbowe ma zadeklarowaną długość 10 znaków. Numer, który wyświetla nam się podczas przeglądania tabeli to 5 znaków (12345). Jednak przy pisaniu zapytania w kliencie musimy napisać pełną wartość pola (0000012345). Jest to mylące, ponieważ później, w wyniku zapytania znowu mamy podaną wartość bez zer wiodących.

Zmusza nas to do dobrej znajomości pól, z których korzystamy lub do częstego odwiedzania tabel w celu uzyskania dokładnej informacji o polu, po którym wyszukujemy.

## Adres do repozytorium: https://github.com/Arafine/Fiori\_test