

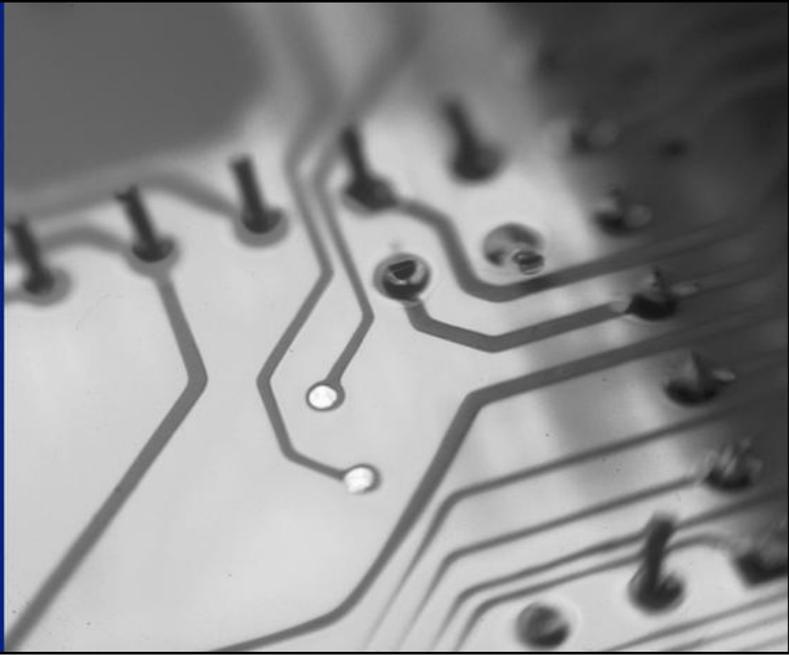
AppGyver

SAP Build Apps



Lagerbestand per App
abfragen (Barcode-Scan)

Online-Tutorial



Zielsetzung

The screenshot shows a mobile application interface for inventory management. The main screen is titled 'Lagerbestand' and features a 'Scannen' button (callout 1). Below the button, the 'Materialnummer' field is populated with 'E510-00' (callout 2). The 'Bezeichnung' field shows 'Druckerpatrone Carlo 123 rot'. The 'Materialart' is 'HAWA' and the 'Warengruppe' is 'L001'. A table titled 'Bestand' displays inventory data for material E510-00 across different plants and storage locations.

Werk	Lager	Bestand	Einh.
1020	101A	15	ST
1010	101B	20	ST
1010	101A	1963	ST

Two SAP data browser tables are shown on the right. The first table, 'Data Browser: Tabelle MARAV', displays material details for E510-00 (callout 3):

Material	MArt	Warengruppe	Materialkurztext
E510-00	HAWA	L001	Druckerpatrone Carlo 123 rot

The second table, 'Data Browser: Tabelle MARD', displays inventory data for material E510-00 across different plants and storage locations (callout 4):

Material	Werk	LOrt	Frei verwend...
E510-00	1010	101A	1 963,000
E510-00	1010	101B	20,000
E510-00	1020	101A	15,000

1. In der App wird durch Klicken auf „Scannen“ der Scanner des mobilen Endgerätes gestartet.
2. Der Barcode wird gescannt und die Materialnummer in das Feld „Materialnummer“ übernommen.
3. Per OData-Service werden aus dem SAP-System Materialgrunddaten gelesen.
4. Zum Material werden die Bestandsdaten ermittelt und angezeigt,



AppGyver (SAP Build Apps)



AppGyver

- No-Code-Plattform für die Entwicklung von mobilen Apps und Web-Apps
- 2021 Übernahme durch SAP, aus „AppGyver“ wird „SAP AppGyver“
- 2022 Weiterentwicklung zu „SAP Build Apps“
→ Anwendungsentwicklung für Unternehmen
- Community-Edition von SAP AppGyver verfügbar
- Abruf der App aus Cloud-Service oder Veröffentlichung in Stores wie App Store oder Google Play



SAP Build Apps ↔ Community Edition

	SAP Build Apps	Community
Build pixel perfect web & mobile apps	✓	✓
Integrate with REST APIs	✓	✓
Co-create with your team	✓	✗
Build cloud hosted business logic and data models with visual cloud functions	✓	✗
Integrate seamlessly with SAP solutions	✓	✗
Enterprise authentication & governance	✓	✗
	Explore	Join our community



Build professional apps **visually.**

Appgyver (SAP Build Apps)

Introducing full-stack visual development

SAP Build Apps

Build pixel-perfect web and mobile apps. Create backends with the new cloud functions. Integrate seamlessly with SAP solutions. Co-create with your team. All without writing code.

Explore SAP Build Apps Free Tier

Explore



Looking for the Community Edition?

The original, free-to-use no code application development platform for non-enterprise and learning projects.

Learn more →



App Gyver Community Edition
No-Code App-Entwicklung



EclipseIDE mit ADT
Entwicklung OData-Services



Mobiles Endgerät
AppGyver Preview App



SAP /4 HANA
Backend für Datenbereitstellung



Datenbereitstellung im SAP-System

```
1 @EndUserText.label: 'Materialgrunddaten'
2 @AccessControl.authorizationCheck: #NOT_REQUIRED
3
4 @odata.publish: true
5
6 define view entity ZBA_MAT_DATA
7   as select from mara
8   join makt on mara.matnr = makt.matnr
9             and makt.spras = $session.system_language
10 {
11   key mara.matnr as MaterialID,
12     makt.maktx as MaterialDescription,
13     mara.matkl as MaterialGroup,
14     mara.mtart as MaterialType
15 }
16
```

Hinweis: DDIC-Views wie z.B. MARAV sind für CDS-Views vom Typ „View Entity“ nicht zulässig!

1. Es wird die CDS-View-Entity ZBA_MAT_DATA definiert.
2. Die mit Hilfe der Transaktion SE11 definierten Views wie z.B. MARAV sind für eine CDS-View-Entity als Datenquelle nicht zulässig. Daher werden die Materialgrunddaten und die Materialbezeichnung per Join der Tabellen MARA und MAKT ermittelt.
3. Die ausgewählten View-Felder sind mit Aliasnamen versehen. Diese Namen sind für die spätere Verwendung der Felder z.B. in unserer App relevant.

The screenshot shows the SAP Studio interface. At the top, a code editor displays the following CDS view definition:

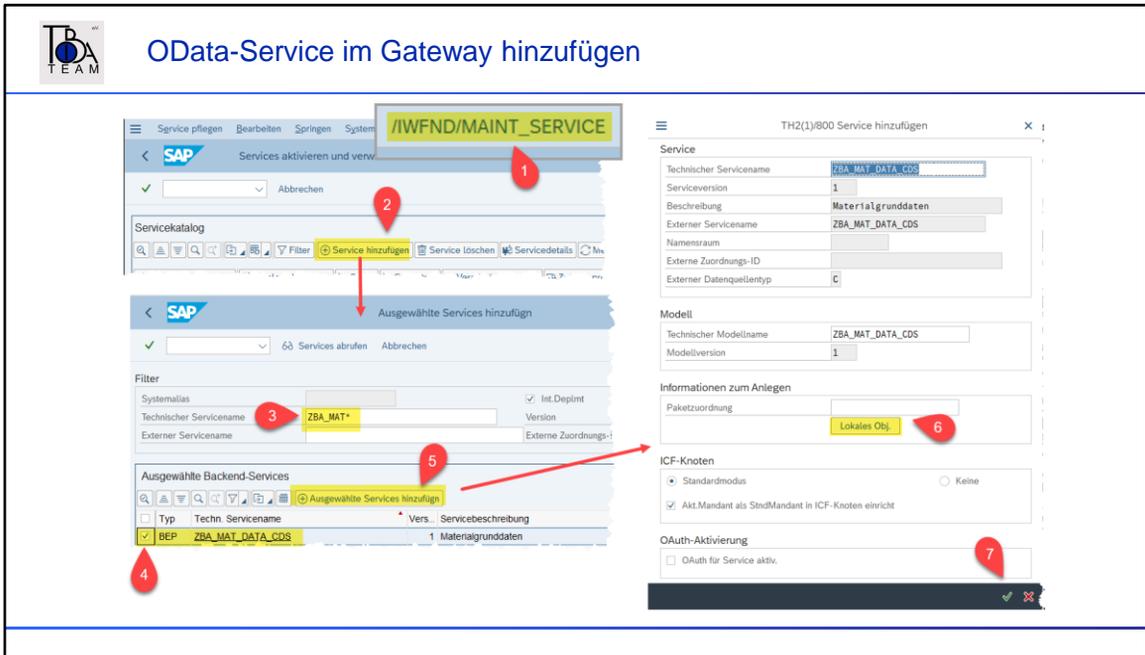
```
4 @ACCESSCONTROLLED.AUTHORITY.ZBA_MAT_DATA_CDS...
5
6 @odata.publish: true
7 define view entity ZBA_MAT_DATA
8 as select from mara
9
```

Red callouts 1 and 2 point to the annotation and the 'G' icon, respectively. Below the code, a 'Generated Objects' popup is visible, showing a tree structure with 'OData-Service' highlighted. Red callout 3 points to this link. Below the popup, a 'Server' tab is selected in a table with columns 'Text', 'Anfrage', 'Antwort', 'Server', 'URI', and 'Laufzeit'. The 'Server' column is highlighted, and red callout 4 points to it. Below the table, a 'Serverumgebung' (Server Environment) table is displayed with the following data:

Name	Wert
~request_line	GET /sap/opu/odata/sap/ZBA_MAT_DATA_CDS/?sap-ds-debug=true&sap-client=800 HTTP/1.0
~request_method	GET
~request_url	/sap/opu/odata/sap/ZBA_MAT_DATA_CDS/?sap-ds-debug=true&sap-client=800
~path	/sap/opu/odata/sap/ZBA_MAT_DATA_CDS/
~path_translated	/sap/opu/odata/sap/ZBA_MAT_DATA_CDS/
~query_string	sap-ds-debug=true&sap-client=800
~server_protocol	HTTP/1.0
~server_name	192.168.70.122

Red callout 5 points to the '~path' value in the table.

1. Die CDS-View wird durch die Annotation „@odata.publish: true“ als OData-Service publiziert.
2. Klicken Sie auf das Icon **G** in der Zeile mit „@odata“. Beachten Sie die Ausführungen auf der Folgeseite für den Fall, dass statt dieses Icons ein Warndreieck angezeigt wird.
3. Im angezeigten Popup klicken Sie auf den Link Odata Service.
4. Im Browser sehen Sie (nach der eventuell notwendigen Anmeldung am SAP-System) auf der Karteikarte „Server“ weitere Informationen zum generierten OData-service.
5. Relevant ist insbesondere die Pfad-Angabe. Diese wird verwendet, um von der App aus den OData-Service aufzurufen.



Die CDS-View muss als OData-Service im SAP-Gateway verfügbar sein. Sollte dies nicht der Fall sein, gehen Sie wie folgt vor:

1. Starten Sie die Transaktion /IWFND/MAINT_SERVICE.
2. Wählen Sie die Aktion „Service hinzufügen“.
3. Geben Sie den Namen der CDS-View (oder Teile davon), gefolgt von einem „*“ als Suchmuster ein und drücken „Enter“.
4. Markieren Sie den gefundenen OData-Service. Bei Services, die aus CDS-Views generiert wurden, ergibt sich der Name des Service aus dem Namen der CDS-View und dem Suffix „_CDS“.
5. Fügen Sie den Service hinzu.
6. Der Service muss einem Paket zugeordnet werden. Wenn Sie den Service nicht auf weitere SAP-Systeme transportieren müssen, können Sie den Service auch als „lokales Objekt“ klassifizieren.
7. Mit Klick auf den grünen Haken wird der Service im Gateway verfügbar.

```

#@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Material Bestandsdaten'

@OData.publish: true

define view entity ZBA_MAT_STOCK
as select from nsdm_e_mard as d
join mara as a on d.matnr = a.matnr
{
  key d.matnr as MaterialID,
  key d.werks as PlantID,
  key d.lgort as StockID,
  @Semantics.quantity.unitOfMeasure: 'BaseUnit'
  d.labst as Available,
  a.meins as BaseUnit
}

```

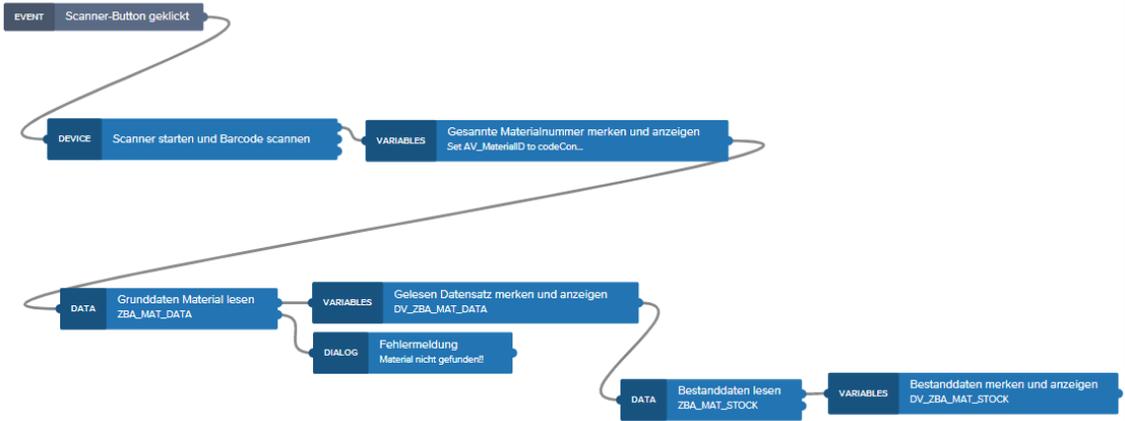
Text	Anfrage	Antwort	Server	URI	Laufzeit
Serverumgebung					
~request_line	GET	/sap/opu/odata/sap/ZBA_MAT_STOCK_CDS/?sap-ds-debug=true&sap-client=800 h...			
~request_method	GET				
~request_uri	/sap/opu/odata/sap/ZBA_MAT_STOCK_CDS/?sap-ds-debug=true&sap-client=800				
~path	/sap/opu/odata/sap/ZBA_MAT_STOCK_CDS/				
~path_translated	/sap/opu/odata/sap/ZBA_MAT_STOCK_CDS/				
~query_string	sap-ds-debug=true&sap-client=800				
~server_protocol	HTTP/1.0				
~server_name	192.168.70.122				
~server_port	44300				

1. Die CDS-View-Entity ZBA_MAT_STOCK stellt die Bestandsdaten zur Verfügung.
2. Beachten Sie, dass die Bestandsdaten nicht (wie vielleicht erwartet) aus der Tabelle MARD gelesen werden. Hintergrund: In S/4HANA ist für die Tabelle MARD die View NSDM_E_MARD als so genanntes Vertreterobjekt definiert.
3. Die frei verfügbare Menge (Feld LABST) wird unter dem Alias-Namen „Available“ in der View bereitgestellt, ergänzt um die Mengeneinheit, die per Join mit der Tabelle MARA zur jeweiligen Materialnummer bestimmt wird.
4. Die CDS-View-Entity ist als OData-Service publiziert.

App erstellen

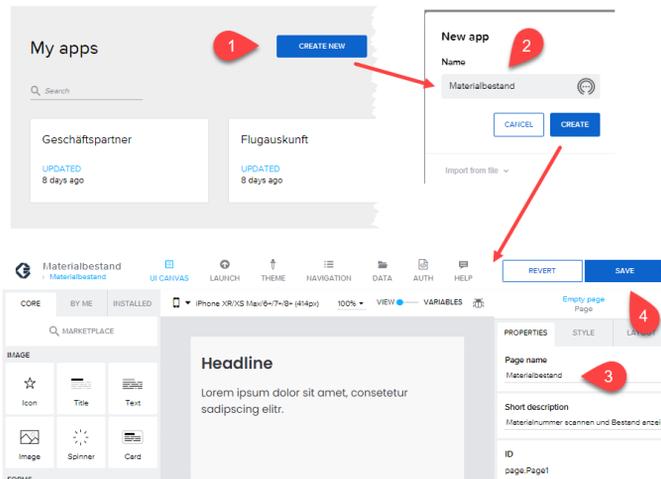
- Entwicklungsschritte-

1. Variablen für Ablage von Materialnummer, Benutzer und Kennwort
2. Button „Start Scanner“ und Feld für Anzeige der Materialnummer
3. Scanner starten und gescannte Materialnummer anzeigen
4. DataResource und Data-Variable für Zugriff auf Materialgrunddaten.
5. Felder für Anzeige der Materialdaten
6. Materialgrunddaten aus SAP-System lesen und anzeigen
7. DataResource und Data-Variable für Zugriff auf Bestandsdaten
8. Tabellendarstellung für Bestandsdaten vorbereiten
9. Bestandsdaten aus SAP-System lesen and anzeigen.

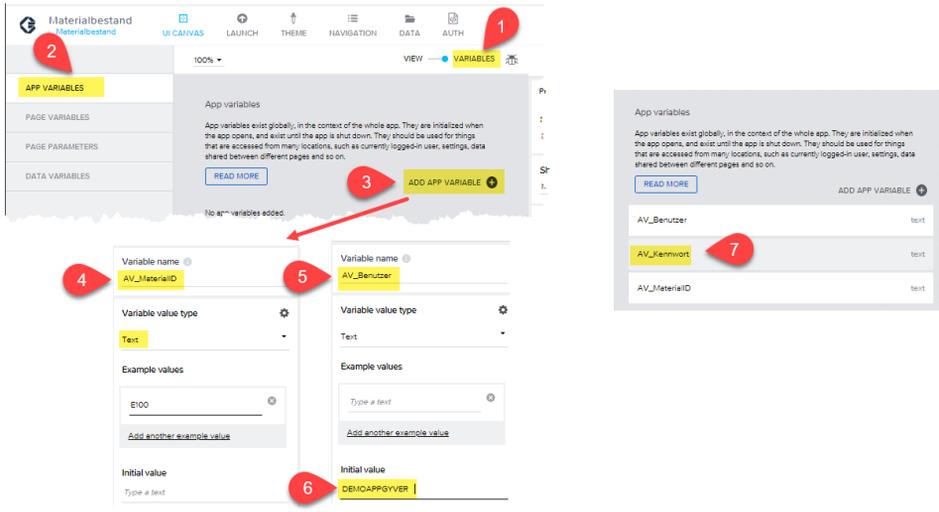


App erstellen

- Startseite und App-Variable -

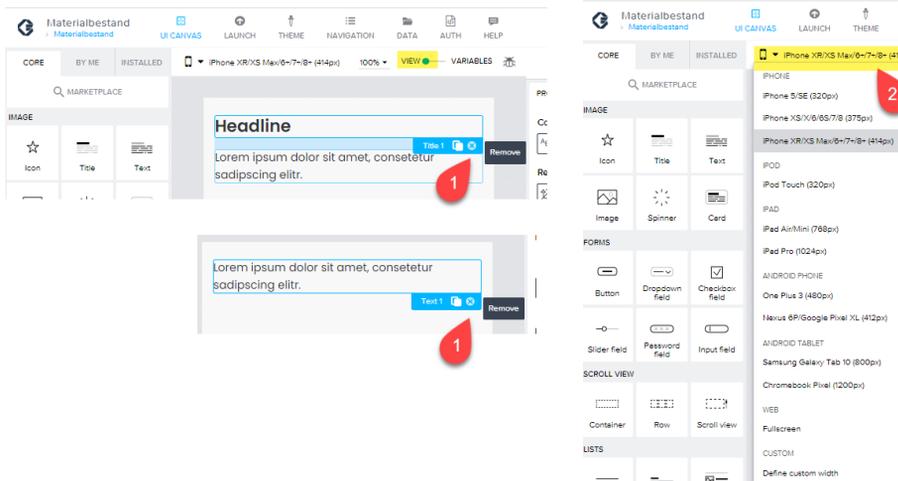


- Im AppGyver die Funktion „Create New“ starten.
- Namen der App angeben und „Create“. Es wird die Startseite der App angezeigt.
- Der Startseite einen Namen geben.
- Sichern.



1. Für die Startseite „Materialbestand“ die Variables-Sicht öffnen.
2. Es sollen 3 App-Variable definiert werden. Als Präfix für die Variablennamen wählen wir „AV_“, um die verschiedenen Variablentypen schon am Namen unterscheiden zu können.
3. Neue Variable Anlegen.
4. Die Variable AV_MaterialID soll die gescannte Materialnummer aufnehmen.
5. In der App muss an mehreren Stellen für einen Zugriff auf das SAP-System der Benutzername und das Kennwort angegeben werden. Die Variable AV_Benutzer soll den Benutzernamen für den Login-Vorgang enthalten.
6. Den Variablen-Wert, also den Benutzernamen, hinterlegen wir unter „Initial Value“.
7. Analog wird die Variable AV_Kennwort angelegt für die Ablage des Kennwortes.

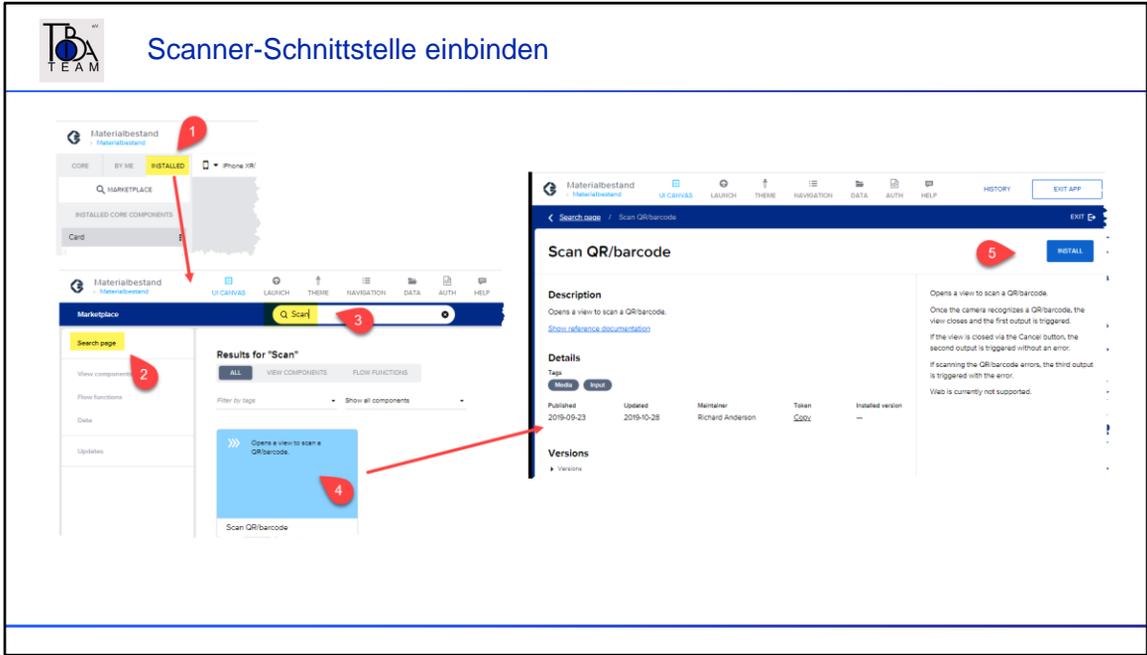
Hinweis: Für die Beispiel-App legen wir keine Anmeldeseite an, sondern geben Benutzernamen und Kennwort fest vor. **Achten Sie unbedingt darauf, dass der Wert für den Benutzernamen nicht mit angehängten Leerstellen versehen ist!**



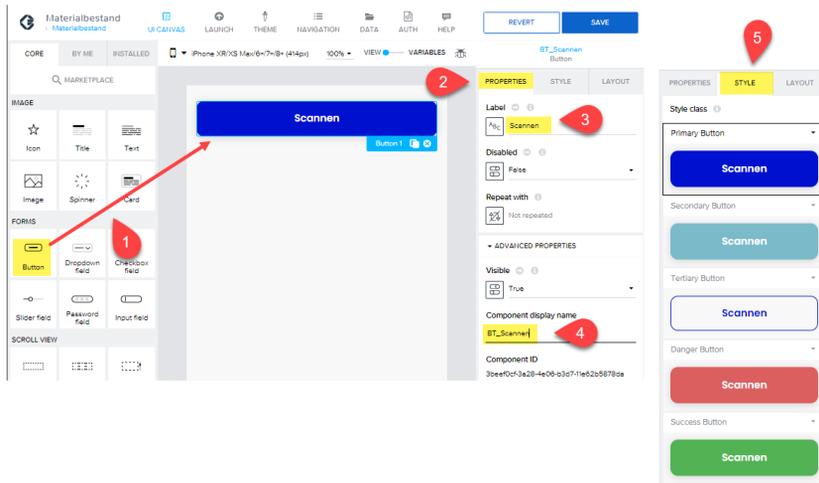
1. Wir löschen zunächst alle in der Startseite schon vorhandenen Elemente.
2. Die Größe der Startseite kann festgelegt werden. Die App ist natürlich im Prinzip auf allen mobilen Endgeräten startbar, auf denen die App installiert ist oder in der AppGyver-Preview-App getestet wird. Man muss halt nur kompromissbereit sein, wenn das benutzte Endgerät ein kleineres Display hat als die eingestellte Größe der Startseite.

App erstellen

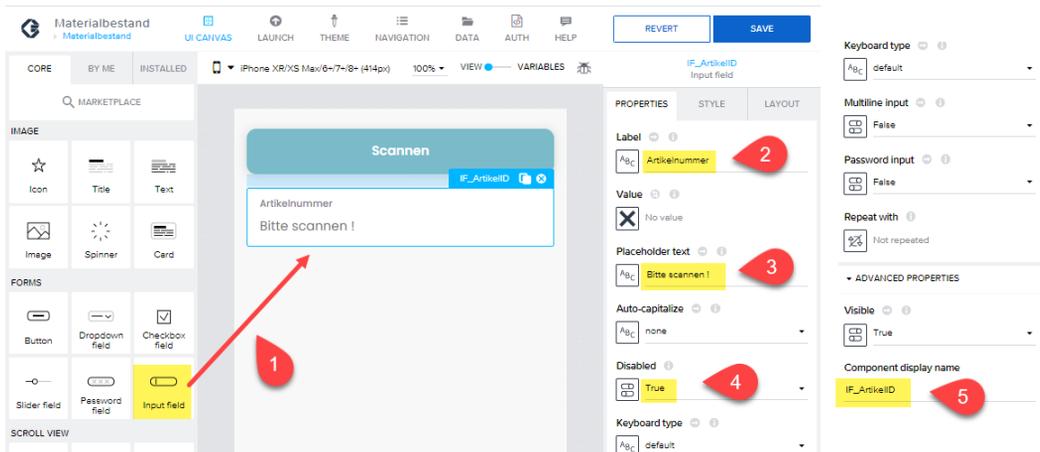
- Scanner anbinden -



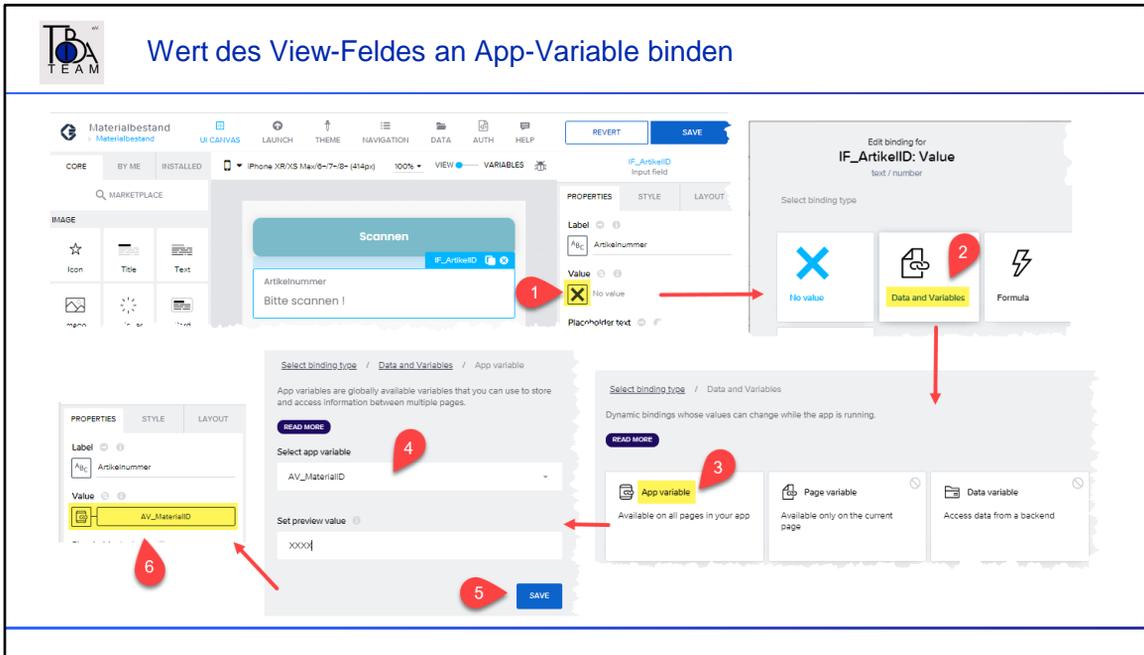
1. Öffnen Sie die Sicht „INSTALLED“.
2. Aktivieren Sie den Suchdialog.
3. Suchen Sie nach „Scan“
4. In der Trefferliste wird die Aktion für den Start eines Scanvorgangs angezeigt. Klicken Sie auf die Aktion.
5. Mit Klick auf „INSTALL“ wird die Scanner-Aktion der Apps zugeordnet.



1. Ziehen Sie per Drag & Drop ein Button-Forms-Element auf die Startseite.
2. Wählen Sie die Sicht PROPERTIES.
3. Legen Sie unter „Label“ die Beschriftung des Buttons fest.
4. Bei Bedarf können Sie den technischen Namen des Buttons anpassen (z.B. Präfix BT für Button).
5. In der Sicht STYLE kann die Farbe des Buttons verändert werden.

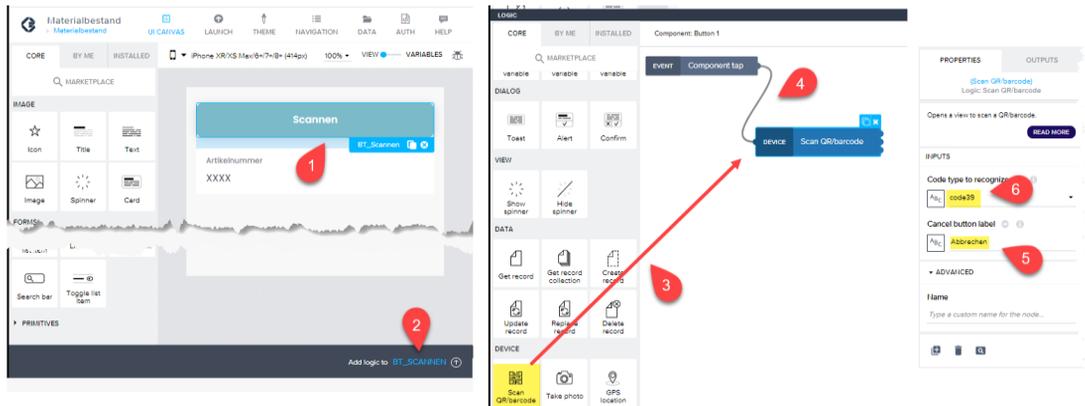


1. Ziehen Sie ein FORMS-Input field auf die Startseite unterhalb des Buttons. Wir verwenden ein „Input field“, obwohl die gesannte Artikelnummer nur angezeigt, und nicht eingegeben werden soll. Grund: Das FORMS-Element für „Input-Field“ sieht optisch einfach gut aus, da dem Feld auch gleich eine Überschrift zugeordnet werden kann.
2. Geben Sie die Beschriftung des Feldes ein.
3. Der „Placeholder text“ wird angezeigt, wenn das Eingabefeld leer ist.
4. Da die Artikelnummer nach einem Scan nur angezeigt und nicht bearbeitet werden soll, wird „Disabled“ auf „True“ gesetzt.
5. Bei Bedarf können Sie den Namen des Input-Feldes anpassen, z.B. mit Präfix „IF_“ für „Input Field“.

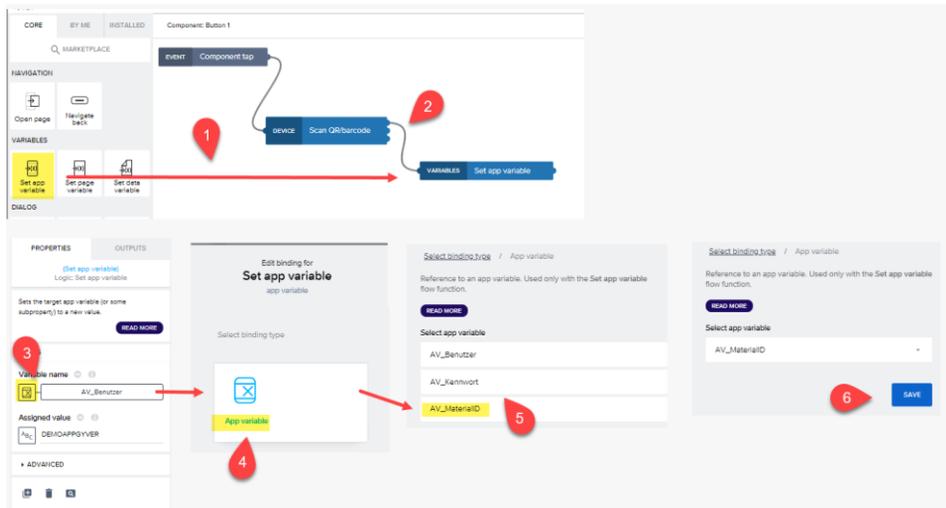


Der Wert des View-Feldes für die Artikelnummer soll an die App-Variable AV_MaterialID gebunden werden.

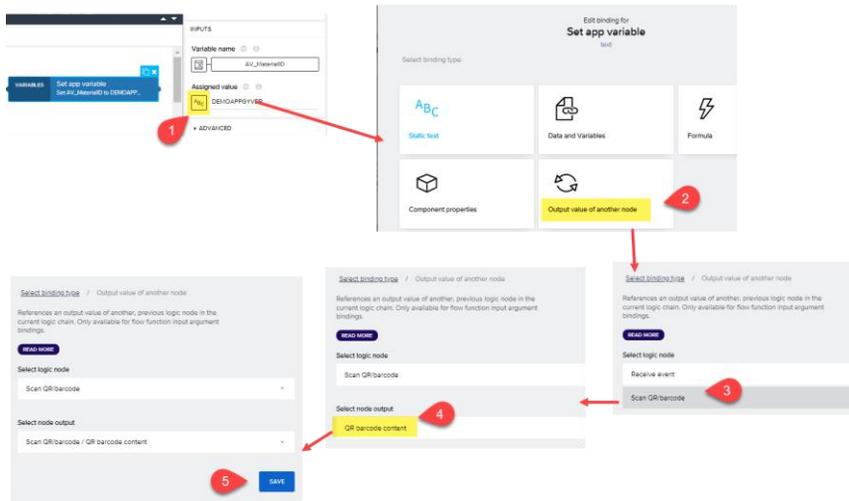
1. Klicken Sie auf das Icon für die Wertzuordnung.
2. Wählen Sie „Data and Variables“.
3. Es soll eine App Variable zugeordnet werden.
4. Wählen sie die App-Variable AV_MaterialID.
5. Sichern Sie die Zuordnung.
6. Die Bindung des Wertes an die App-Variable wird angezeigt.



1. Markieren Sie den Button „Scannen“
2. Öffnen Sie den Bereich für die Eingabe der Verarbeitungslogik.
3. Ziehen Sie das Device-Element „Scan QR/barcode“ in den Bereich für die Logik-Definition.
4. Verbinden Sie den Ausgang des Elements „EVENT Component tap“ mit dem Eingang des Elements „DEVICE Scan QR/barcode“.
5. Legen Sie eine Beschriftung für die Schaltfläche fest, die im Endgerät für das Abbrechen des Scan-Vorgangs angezeigt wird.
6. Bei Bedarf können Sie Angaben machen, welche Variante von QR/Barcode erkannt werden soll.



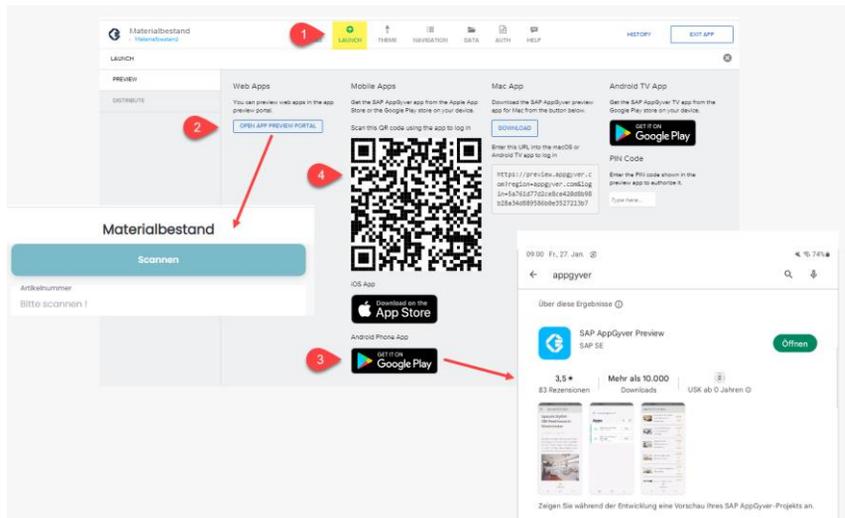
1. Ziehen Sie das Logik-Element „Set App Variable“ in den Logik-Bereich.
2. Verbinden Sie den obersten Ausgang des Scan-Elementes mit dem Eingang des „Set“-Elementes.
3. Starten Sie die Zuordnung einer App-Variablen.
4. Wählen Sie App-Variable.
5. Die Variable AV_MaterialID soll zugeordnet werden.
6. Sichern Sie die Zuordnung.



Starten Sie den Dialog für die Wertzuordnung.
 Wählen Sie „Output Value of another node“.
 Das Ergebnis des Scans soll ausgewertet werden.
 Der gescannte Wert wird zugewiesen.
 Sichern Sie die Zuordnung.

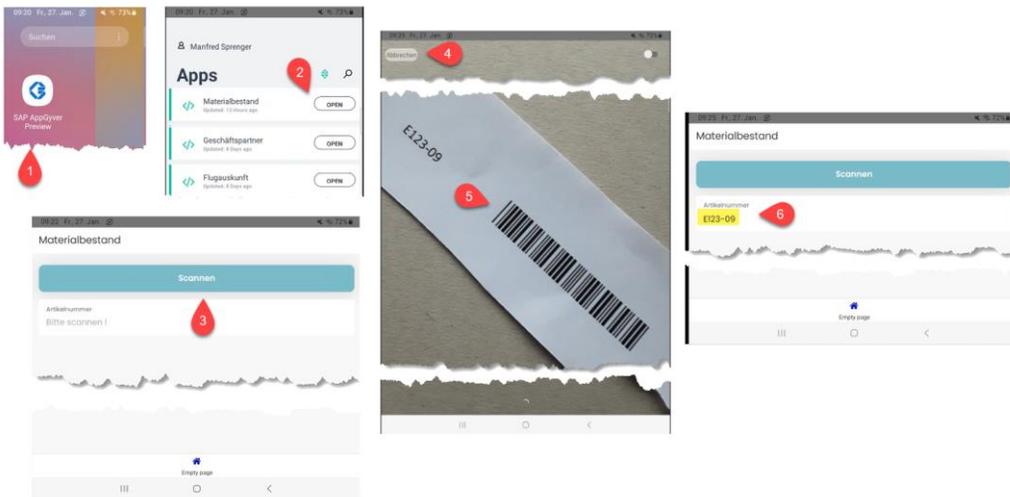
App erstellen

- Scanner testen -



Die App kann, ohne dass Sie über einen App-Store verteilen, in einer „Preview“ getestet werden.

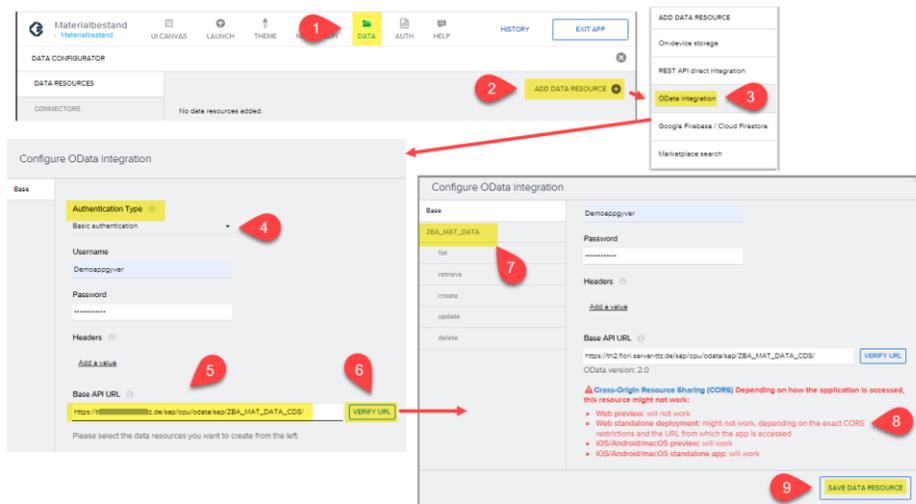
1. Wechseln Sie auf die Sicht „Launch“.
2. Mit „OPEN APP PREVIEW PORTAL“ kann die Preview im Browser gestartet werden. Im konkreten Fall sieht man die Startmaske, ein Test des Scan-Vorgangs ist aber z.B. auf einem PC nicht so ohne weiters möglich. Sie werden später noch sehen, dass Apps, die einen OData-Services eines SAP-Systems aufrufen, auch nur unter bestimmten Voraussetzungen im Preview Portal lauffähig sind.
3. Für mobile Endgeräte steht die „SAP Appgyver Preview“-App zur Verfügung, die Sie z.B. über den GooglePlay-Store auf dem Endgerät installieren können.
4. Beim ersten Start der App müssen Sie sich mit dem AppGyver-Preview-Server verbinden. Scannen Sie dazu den QR-Code. Damit wird die App mit den Login-Daten Ihrer AppGyver-Kennung verknüpft.



1. Öffnen Sie die SAP App Gyver Preview.
2. Starten Sie App.
3. Klicken Sie auf den Scannen-Button.
4. Den Scan-Vorgang können Sie durch die Abbrechen-Taste stoppen.
5. Scannen Sie den Barcode.
6. Der Scanner wird geschlossen und der gescannte Barcode im Feld „Artikelnummer“ angezeigt.

App erstellen

- OData-Service für Materialgrunddaten anbinden -

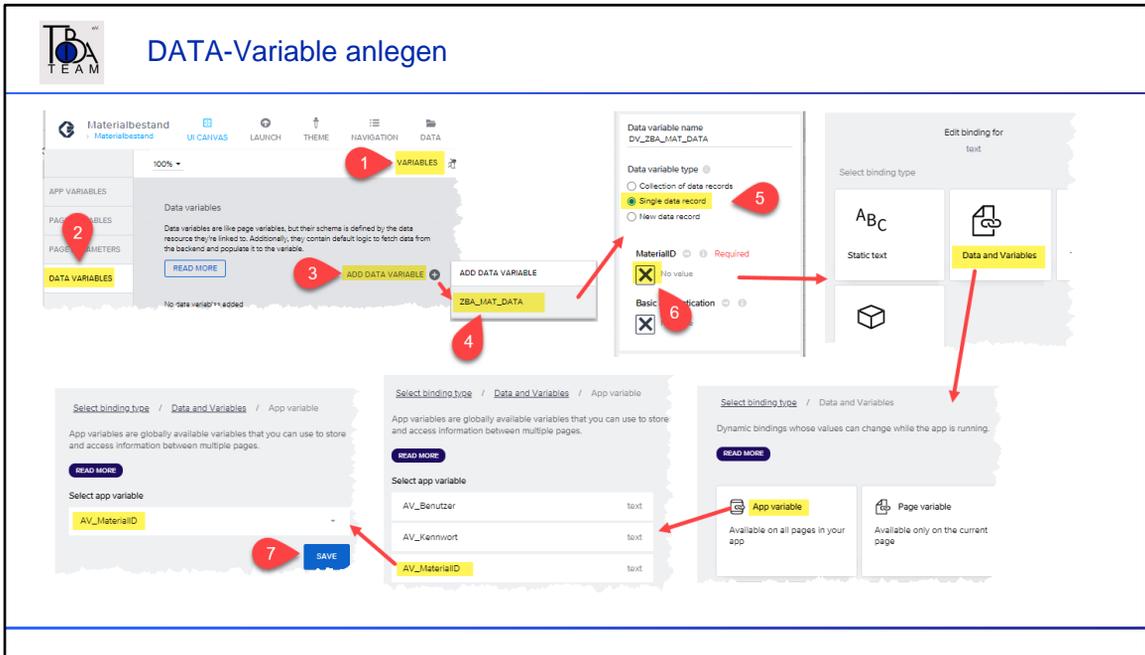


1. Wechsel Sie in die „DATA“-Sicht.
2. Fügen Sie eine „DATA RESOURCE“ hinzu.
3. Der Zugriff auf die Material-Grunddaten soll per OData-Service erfolgen.
4. Um die Definition des OData-Service aus dem SAP-System lesen zu können, ist eine Anmeldung am SAP-System per „Basic Authentication“ erforderlich. Geben Sie dazu den Benutzernamen und ein Kennwort ein. Der Benutzer muss über ausreichende Berechtigung für den Zugriff auf OData-Services verfügen. Die angegebenen Benutzerdaten werden nicht in der App gesichert und sind nur für den Abruf der Metadaten des OData-Service erforderlich. Für den konkreten Aufruf des OData-Service sind später noch einmal Anmeldedaten erforderlich.
5. Geben Sie die URL des OData-Services an (siehe Kapitel „Datenbereitstellung im SAP System“). Der konkret anzugebende Servernamen und die Portnummer (ausgegraut in der Abbildung) sind installationsabhängig.
6. Klicken Sie auf „VERIFY URL“. Es werden die Anmeldedaten und der Zugriff auf die Metadaten des OData-Service geprüft.
7. Bietet der OData-Service nur eine einzige Entität (z.B. nur eine CDS-View) an, wird diese bereits angezeigt. Hat der OData-Service mehrere Entitäten, müssen Sie noch die Entität in der eventuell angezeigten Auflistung markieren, die Sie konkret verwenden möchte.
8. Es wird u.U. eine CORS-Fehlermeldung angezeigt, die Sie darauf hinweist, dass der OData-Service nicht in der WebPreview lauffähig ist sondern nur auf einem mobilen Endgerät gestartet werden kann. Googlen Sie einmal nach „CORS-Error“. Sie werden sehen, was genau hinter diesem Fehler steckt und wie man ihn u.U. vermeiden kann (ist nicht trivial!).
9. Sichern Sie die Data Resource.

The screenshot shows the SAP Data Configurator interface. At the top, a table lists data resources, with 'ZBA_MAT_DATA' selected (1). Below, the 'Configure OData integration' dialog is open, showing the 'retrieve' button (2) and the 'TEST' tab (3). The 'Identifiers' section has 'E123-09' entered (4). The 'Authentication' section shows 'Basic authentication' with fields for 'Username' (DEMOAPPUSER) and 'Password'. The 'Selected Fields' section has a dropdown menu (5). The 'Test API call response' section shows a JSON response (6) and a warning about CORS. The 'SCHEMA' tab (7) lists available fields: MaterialID, MaterialDescription, MaterialGroup, and MaterialType.

Wir wollen testen, ob wir mit Hilfe der Data Resource die Grunddaten eines Materials abrufen können.

1. Klicken Sie auf die Data Resource.
2. Wir möchten die Daten eines einzelnen Materials lesen. Die Funktion „retrieve“ bietet die entsprechende Funktion an.
3. Klicken Sie auf den Tab-Reiter „TEST“.
4. Geben Sie Anmeldeinformationen für das SAP-System vor und eine Materialnummer. Die Felder, die Sie für den Einzelsatz-Zugriff angeben müssen, ergeben sich aus der Definition des OData-Services. Beachten Sie im Falle der Materialnummer, dass Sie bei rein numerischen Material-Nummer u.U. noch genügend führende Nullen angeben müssen.
5. Starten Sie den Lesevorgang mit RUN TEST.
6. Im Erfolgsfall werden die gelesenen Daten angezeigt. Die zuvor schon erwähnte CORS-Fehlermeldung können Sie ignorieren.
7. Ein Klick auf den Tab-Reiter „SCHEMA“ listet die zur Verfügung stehenden Felder. Diese ergeben sich aus der Definition des OData-Services.



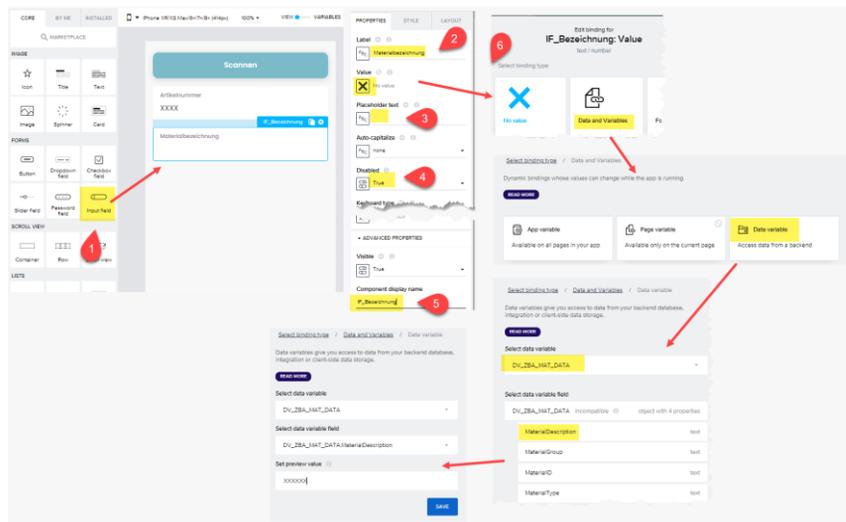
Die Materialdaten, die wir mit Hilfe der „Retrieve“-Funktion des OData-Service abrufen, müssen zur weiteren Auswertung innerhalb der App in eine Variable angelegt werden.

1. Wechsel Sie in die VARIABLES-Sicht der Startseite.
2. Wir wollen eine DATA VARIABLE anlegen. Solche Variablen sind immer gekoppelt an eine Data Resource.
3. Legen Sie eine neue Variable an.
4. Es werden jetzt die bereits in der App definierten Data Resources angeboten. Im konkreten Fall gibt es bisher nur die Data Resource ZBA_MAT_DATA,
5. Wir wollen die Variable nutzen, um genau einen gelesenen Datensatz abzulesen. Daher ist „Single data record“ die richtige Option.
6. Zum Lesen eines einzelnen Datensatzes ist die Angabe einer Materialnummer notwendig. Ordnen Sie der „MaterialID“ über den skizzierten Weg die App-Variable „AV_MaterialID“ zu.
7. Sichern Sie die Zuordnung der App-Variablen.

1. Wir passen den Namen der Data-Variablen an. Der Präfix „DV_“ soll andeuten, dass es sich bei der Variablen um eine Variable vom Typ „DATA“ handelt.
2. Einer Data-Variablen ist per Default eine Logik zugeordnet, die wir in unserem Beispiel nicht gebrauchen können. Klicken Sie zur Überarbeitung der Logik auf „Show logic for...“.
3. Die Standardlogik bewirkt, dass beim Aufruf der Seite sofort ein Versuch unternommen wird, über die Data-Resource einen Datensatz zu lesen. In unserem konkreten Fall macht ein Lesen aber erst dann Sinn, wenn die Materialnummer durch den Scandvorgang ermittelt wurde. Wir müssen daher die Standardlogik abändern. Klicken Sie dazu auf die Verbindungslinie zwischen „Page mounted“ und „Get record“. Wenn die Linie jetzt orange ist, können Sie sie durch Drücken der Löschen-Taste entfernen.
4. Die Standardlogik ist jetzt zwar nicht komplett gelöscht, das automatische Lesen aber nicht mehr aktiv.

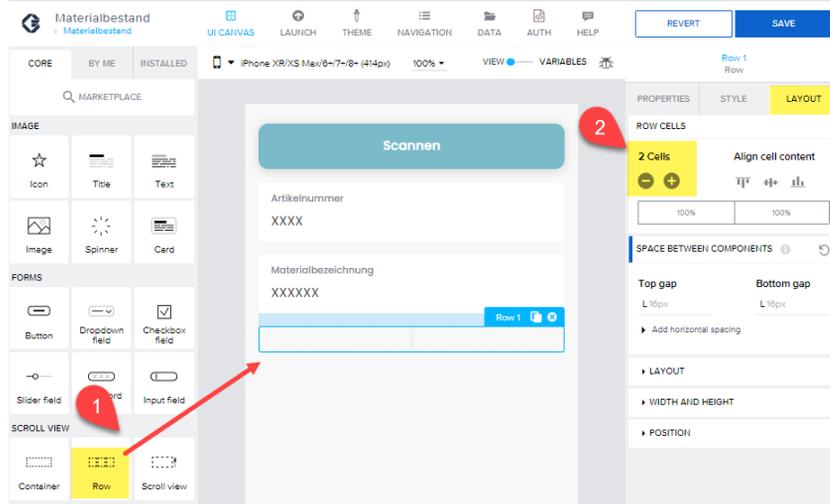
App erstellen

- Startseite erweitern: Felder für Materialgrunddaten -



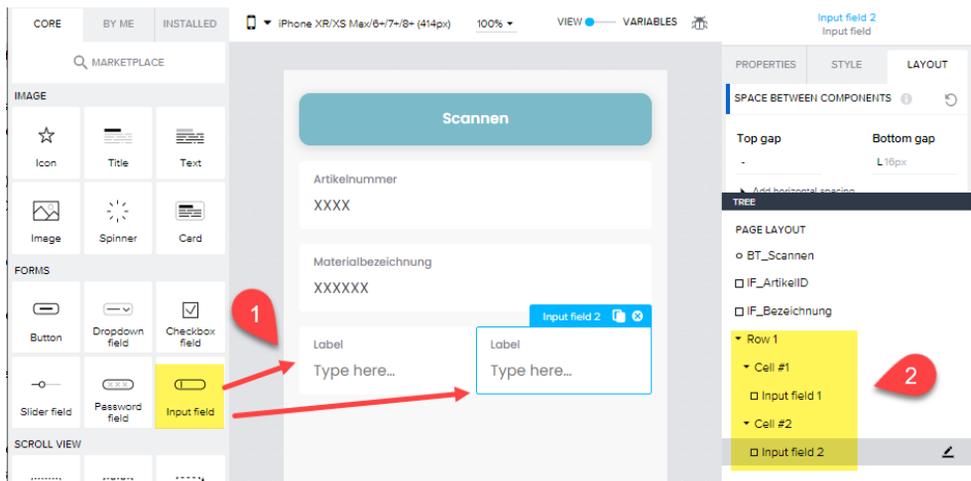
1. Ziehen Sie für die Materialbezeichnung ein Input-Feld in die Seite.
2. Geben Sie eine Beschriftung an.
3. Wenn Sie keinen Text anzeigen möchten, wenn das Feld einen leeren Inhalt hat, müssen Sie als „Placeholder Text“ ein Leerzeichen eingeben.
4. Die Einstellung Disabled = True stellt des Eingabefeld auf Read-Only ein.
5. Bei Bedarf geben Sie dem Feld einen eher „sprechenden“ Namen.
6. Über den angegeben Weg ordnen wir dem Eingabefeld die Komponente „MaterialDescription“ der Data-Variablen DV_ZBA_MAT_DATA zu.

Felder für Warengruppe und Materialart (1)



Die Felder für die Materialart und die Warengruppen sollen auf der Seite nebeneinander angeordnet werden.

1. Ziehen Sie einen „Row“-Element per Drag & Drop auf die Seite.
2. Unter LAYOUT können Sie die Anzahl Zellen einstellen, die in der Zeile zur Verfügung stehen sollen.



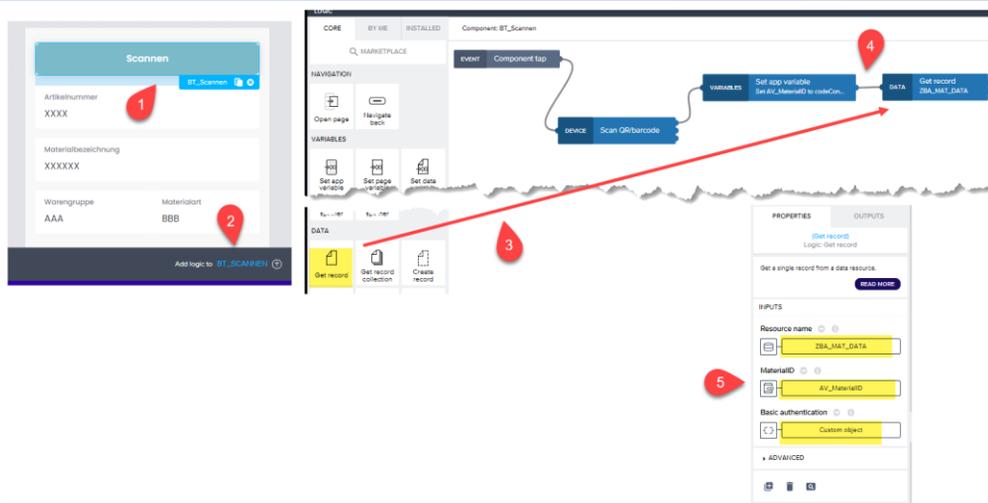
1. Ziehen Sie per Drag & Drop nacheinander jeweils ein Input-Field auf die Zellen des Row-Elementes.
2. Wenn Sie alles richtig gemacht haben, zeigt die TREE-Darstellung, dass die Input-Fields jeweils einer Zelle zugeordnet sind.

The image displays a mobile application interface on the left and two corresponding property panels on the right. The interface includes a 'Scannen' button, an 'Artikelnummer' field with the value 'XXXX', a 'Materialbezeichnung' field with the value 'XXXXXX', and two input fields: 'Warengruppe' with the value 'AAA' and 'Materialart' with the value 'BBB'. The property panels show the configuration for these fields. The first panel is for the 'Warengruppe' field, and the second is for the 'Materialart' field. Both panels have tabs for 'PROPERTIES', 'STYLE', and 'LAYOUT'. The 'PROPERTIES' tab is active in both. The 'Label' property is set to 'Warengruppe' and 'Materialart' respectively. The 'Value' property is set to a data binding: 'DV_ZBA_MAT_DATA.MaterialGro...' for 'Warengruppe' and 'DV_ZBA_MAT_DATA.MaterialType' for 'Materialart'. The 'Placeholder text' property is set to 'AaC' for both. The 'Auto-capitalize' property is set to 'none' for both. The 'Disabled' property is set to 'True' for both. The 'Keyboard type' property is set to 'None' for both. The 'Advanced Properties' section shows 'Visible' set to 'True' and 'Component display name' set to 'IF_Warengruppe' for the first field and 'IF_Materialart' for the second field.

Analog zur Materialbezeichnung werden Beschriftung, Name und Datenanbindung für die Eingabefelder angepasst.

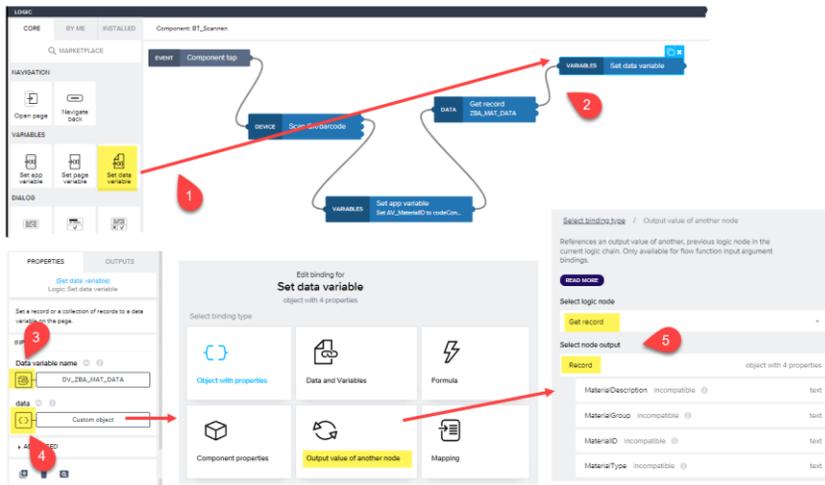
App erstellen

- Materialgrunddaten lesen und anzeigen -



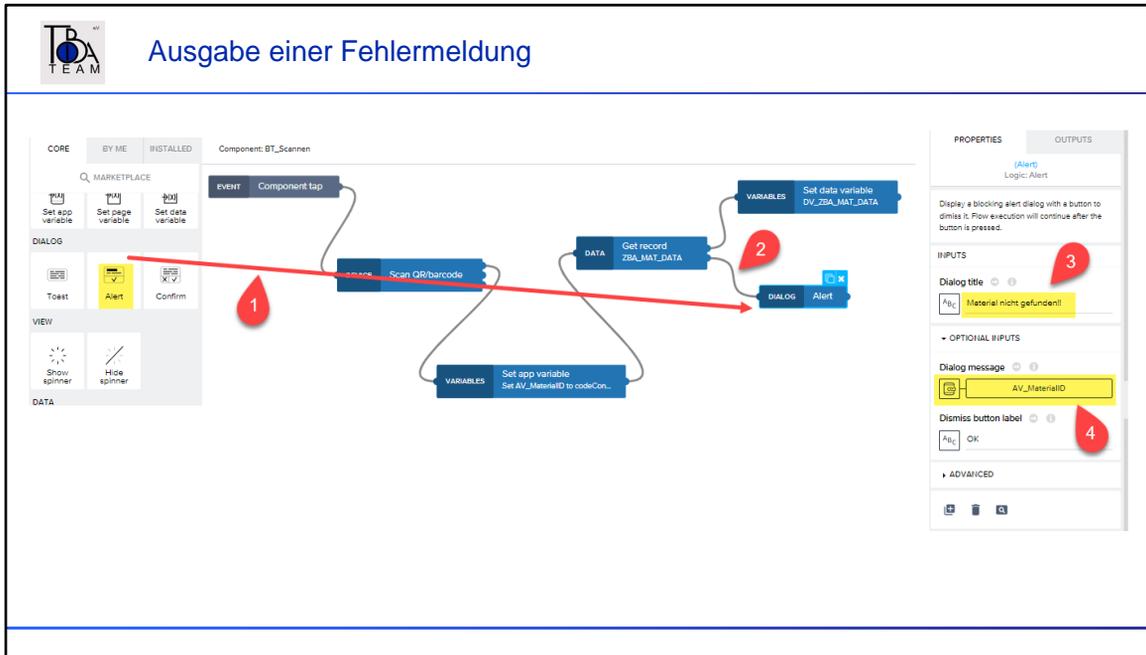
Die Logik für das Lesen der Materialgrunddaten nach dem Scannen soll ergänzt werden.

1. Klicken Sie in der Startseite auf den Scan-Button.
2. Öffnen Sie die Logik-Definition zum Button.
3. Ziehen Sie ein DATA-Element „Get Record“ per Drag & Drop in den Logik-Bereich.
4. Verbinden Sie die Ausgangsseite der Schritte „Set App Variable“ mit der Eingangsseite des Schrittes „Get Record“.
5. Ordnen Sie dem „Get Record“, den Namen der Data Resource zu (hier ZBS_MAT_DATA“, der „MaterialID“ die App-Variablen AV_MaterialID und der „Basis authentication“ die Anmeldedaten. Verwenden Sie dazu die App-Variablen AV_Benutzer und AV_Kennwort.



Der gelesene Datensatz muss nun an die DATA-VARIABLE übergeben werden, an die wir die Felder für die Materialgrunddaten gebunden haben.

1. Ziehen Sie das Logik-Element „Set data variable“ in den Logik-Bereich.
2. Verbinden Sie den oberen Ausgang des Schrittes „Get record“ mit dem Eingang des neuen Schrittes „Set data variable“.
3. Ordnen Sie die Data-VARIABLE „DV_ZBA_MAT_DATA“ zu.
4. Über „data“ weisen Sie über den angegebenen Weg das Element „Record“ des Logikknotens „Get Record“ zu (5). Damit wird der gelesene Datensatz an die DATA-VARIABLE übergeben.

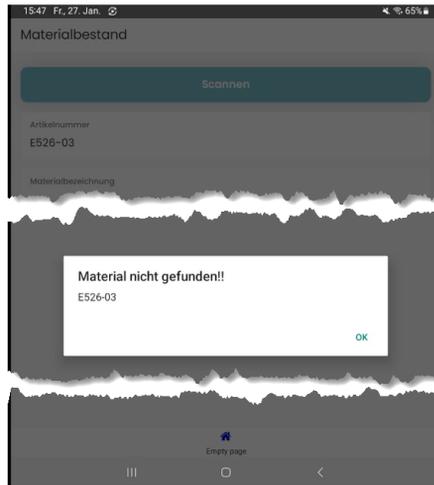
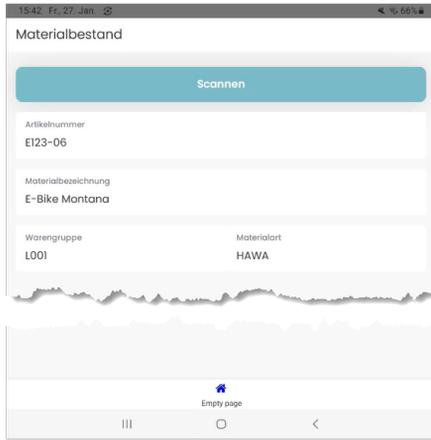


Es ist sicherlich nicht ausgeschlossen, dass ein Barcode zu einem Material gescannt wird, das nicht im SAP-System vorhanden ist. In diesem Fall soll eine Fehlermeldung ausgegeben werden.

- Ziehen Sie ein DIALOG-Element „Alert“ in den Logik-Bereich.
- Verbinden Sie den unteren Ausgang des „Get record“-Schrittes mit der Eingangsseite des Alert-Schrittes.
- Legen Sie die Überschrift der angezeigten Alert-Box fest.
- Der Einfachheit halber soll nur die Materialnummer als Meldungstext ausgegeben werden. Wenn Sie hier mehr Ansprüche haben, könnten Sie sich z.B. weitere (Page)-Variablen anlegen und diesen auch per Formeln komplexere Texte zuweisen. Ordnen Sie in diesem Fall dann die passende Variable als Dialog-Meldung zu.

App erstellen

- Test : Daten lesen und anzeigen -



App erstellen

- OData-Service für Materialbestand anbinden -

The screenshot displays the 'DATA CONFIGURATOR' interface. A table lists data resources:

RESOURCE ID	TYPE
ZBA_MAT_DATA	OData Integration

The 'Configure OData integration' dialog box is open, showing the following configuration details:

- Base: ZBA_MAT_STOCK
- Authentication Type: Basic authentication
- Username: Demopppgyver
- Base API URL: https://192.168.1.101:8080/odata/soap/ZBA_MAT_STOCK_CDS
- OData version: 2.0
- Warning: Cross-Origin Resource Sharing (CORS) Depending on how the application is accessed, this resource might not work.
- Notes: Web preview will not work; Web standalone deployment: might not work, depending on the exact CORS.

Wir binden den OData-Service für das Lesen des Materialbestandes als DATA RESOURCE an. Die Schritte dazu sind analog zu denen für die Anbindung des Services für die Material-Grunddaten.

The screenshot displays the SAP OData integration configuration and test results. The interface is divided into three main sections: CONFIG, TEST, and SCHEMA. The TEST tab is selected, showing the following details:

- Authentication type:** Basic authentication
- Username:** DEMOAPPGUYVER
- Password:** [Redacted]
- Conditions:** A condition is added: MaterialID = Equal = E510-00.
- Sort:** ADD SORT OPTION
- Test API call response:** A warning message states: "Cross-Origin Resource Sharing (CORS) Depending on how the application is accessed, this resource might not work." It lists conditions for Web preview, standalone deployment, and iOS/Android/macOS preview.
- Test Results:** The test is successful, showing a "Status: OK" and a JSON response for the OData service.

The JSON response is as follows:

```

A)";
  "uri": "https://192.168.70.122:44300/sap/opu/odata/sap/ZBA_MAT_STOCK_CDS/ZBA_MAT_STOCK(MaterialID='E510-00',PlantID='1010',StockID='1010')";
  "type": "ZBA_MAT_STOCK_CDS.ZBA_MAT_STOCKType"
},
{
  "MaterialID": "E510-00",
  "PlantID": "1020",
  "StockID": "1010",
  "Available": "15",
  "BaseUnit": "ST"
},
{
  "_metadata": {
    "id": "https://192.168.70.122:44300/sap/opu/odata/sap/ZBA_MAT_STOCK_CDS/ZBA_MAT_STOCK(MaterialID='E510-00',PlantID='1010',StockID='1010')";
    "uri": "https://192.168.70.122:44300/sap/opu/odata/sap/ZBA_MAT_STOCK_CDS/ZBA_MAT_STOCK(MaterialID='E510-00',PlantID='1010',StockID='1010')";
    "type": "ZBA_MAT_STOCK_CDS.ZBA_MAT_STOCKType"
  },
  "MaterialID": "E510-00",
  "PlantID": "1010",
  "StockID": "1010",
  "Available": "20",

```

Wir wollen testen, ob der Aufruf des OData-Services über die Data-Resource möglich ist.

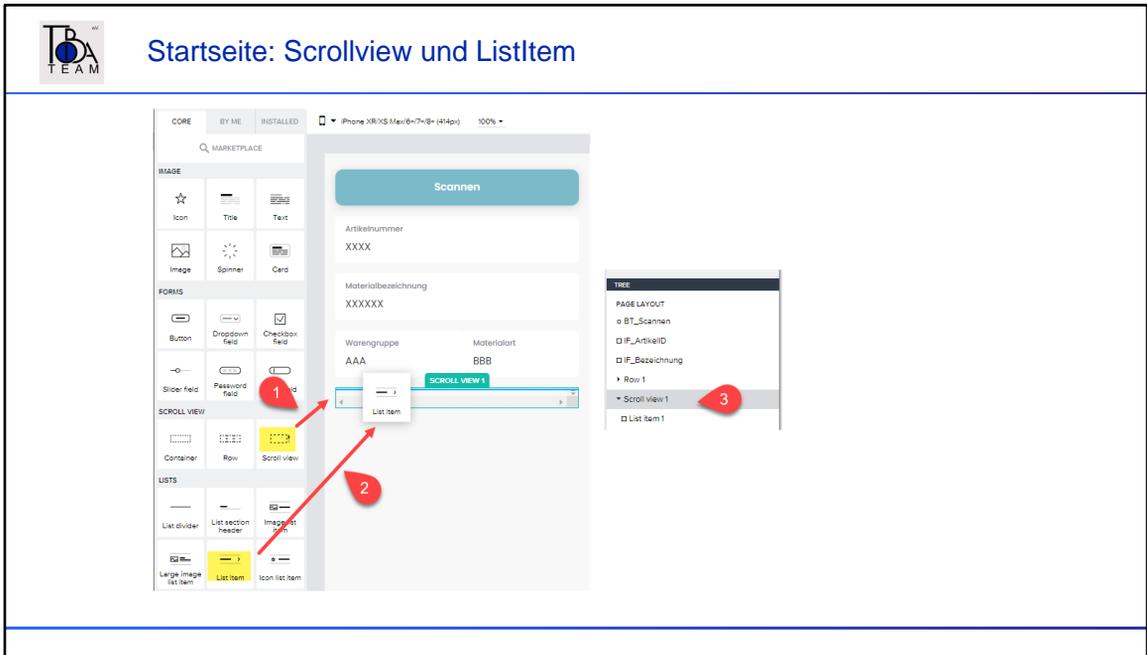
1. Klicken Sie auf den Tab-Reiter TEST.
2. Da es zu einem Material mehrere Bestandsdaten geben kann, testen wir die Funktion „list“, die wir später zum Lesen der Bestandsdaten verwenden möchten.
3. Da es u.U. sehr viele Bestandsdaten geben kann, fügen wir über ADD CONDITION eine Bedingung ein, mit der wir die Datenbeschaffung eingrenzen wollen.
4. Es sollen nur die Bestandsdaten eines Materials zum Test gelesen werden.
5. RUN TEST zeigt uns, dass der OData-Service aufgerufen werden kann und auch Bestandsdaten liefert.

Hinweise: Klicken Sie auf den Tab-Reiter SCHEMA, wenn Sie wissen möchten, welche Felder der Bestandsdaten Ihnen zur Verfügung stehen.

1. Wir legen eine neue Data-Variable mit Bezug zur Data-Resource ZBA_MAT_STOCK an.
2. Den Namen der Data-Variablen versehen wir mit dem Präfix „DV_“ (optional).
3. Achten Sie darauf, dass die Variable vom Typ „Collection of data records“ ist, da es nur so möglich ist, mehrere gelesene Zeilen zu verarbeiten.
4. Einer Data-Variablen vom Typ „Collection of data records“ ist per Default eine Logik zugeordnet, die wir in unserem Beispiel nicht gebrauchen können. Klicken Sie zur Überarbeitung der Logik auf „Add logic to...“ bzw. „Show logic for ...“.
5. Die Standardlogik bewirkt, dass beim Aufruf der Seite sofort ein Versuch unternommen wird, über die Data-Resource Daten zu lesen. In unserem konkreten Fall macht ein Lesen aber erst dann Sinn, wenn die Materialnummer durch den Scanvorgang ermittelt wurde. Wir müssen daher die Standardlogik abändern. Klicken Sie dazu auf die Verbindungslinie zwischen „Page mounted“ und „Get record collection“. Wenn die Linie jetzt orange ist, können Sie sie durch Drücken der Löschen-Taste entfernen.
6. Die Standardlogik ist jetzt zwar nicht komplett gelöscht, das automatische Lesen aber nicht mehr aktiv.

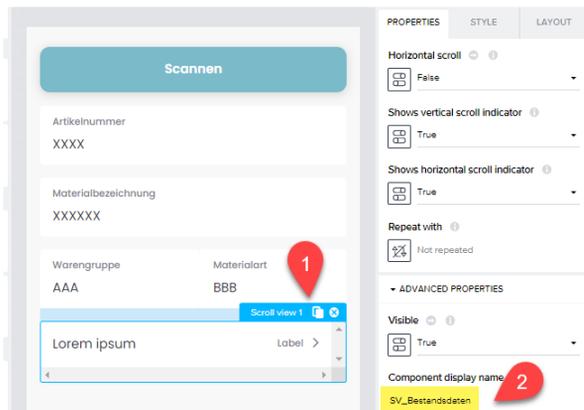
App erstellen

- Ausgabetabelle für Bestandsdaten -

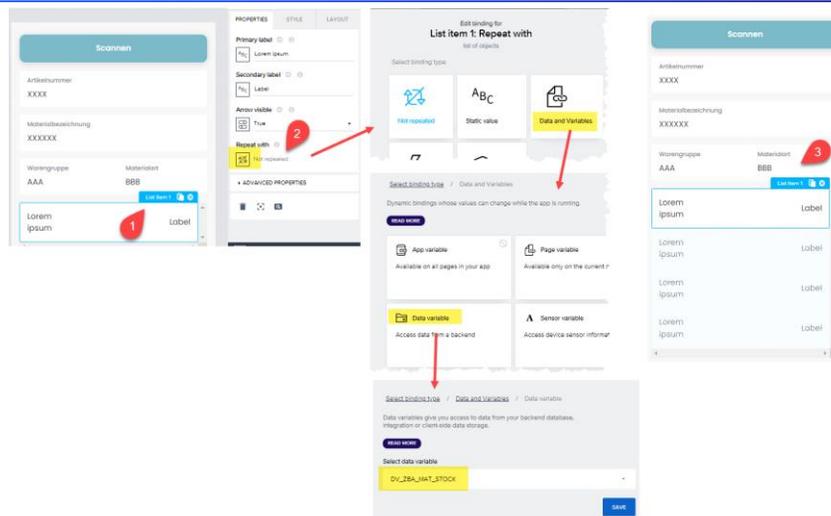


Auf der Seite sollen unterhalb der Materialdaten die Bestandsdaten in Form einer Tabelle ausgegeben werden.

1. Ziehen Sie zunächst ein „Scroll View“-Element auf die Seite. Dieses Element erlaubt es, mehrere Datenzeilen auszugeben. Sind mehr Zeilen vorhanden als im sichtbaren Bereich der Tabelle dargestellt werden können, steht ein Scrollbar zum Rollen der Tabellenzeilen zur Verfügung.
2. Ziehen Sie als nächstes ein „Line Item“ Element auf das „Scroll View“-Element.
3. Überprüfen Sie in der Treedarstellung, ob das „Line Item“-Element tatsächlich dem „Scroll-View“-Element untergeordnet ist.

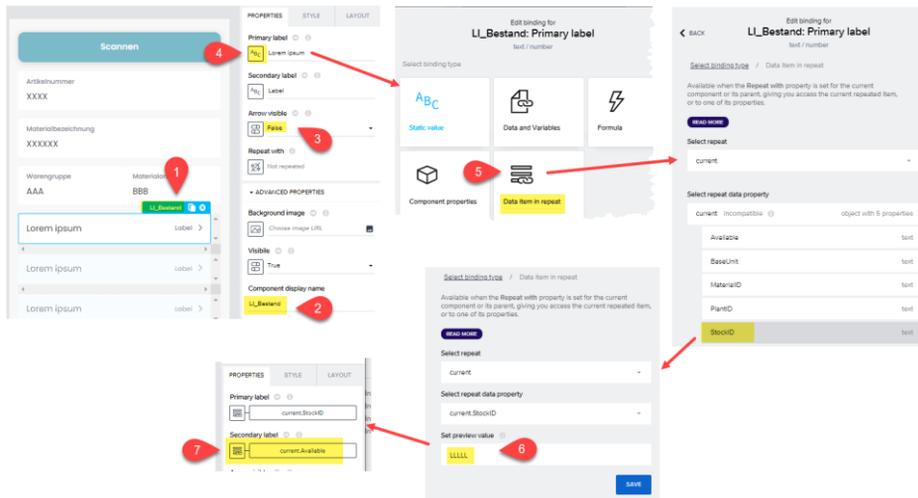


1. Markieren Sie die Scroll View (Klicken Sie z.B. alternativ auch in der Tree-Darstellung auf den Scrool View-Eintrag).
2. Auf dem PROPERTIES-Reiter passen wir zunächst den Namen der Scrool View an (Präfix „SV_“ für Scroll View).



Wir legen nun fest, dass das List Item die Zeilen der Datenvariable DV_ZBA_MAT_STOCK anzeigen soll. Da diese Datenvariable vom Type her mehrzeilig ist, hat dies zur Folge, dass auch das List Item mehrzeilig angezeigt wird, d. h. für jede von der Datenvariable gelieferte Zeile gibt es eine Listzeile.

1. Markieren Sie das List Item.
2. Ordnen Sie über „Repeat With“ die Daten-Variablen DV_ZBA_MAT_STOCK zu.
3. In der Scroll View werden jetzt beispielhaft mehrere Zeilen dargestellt.



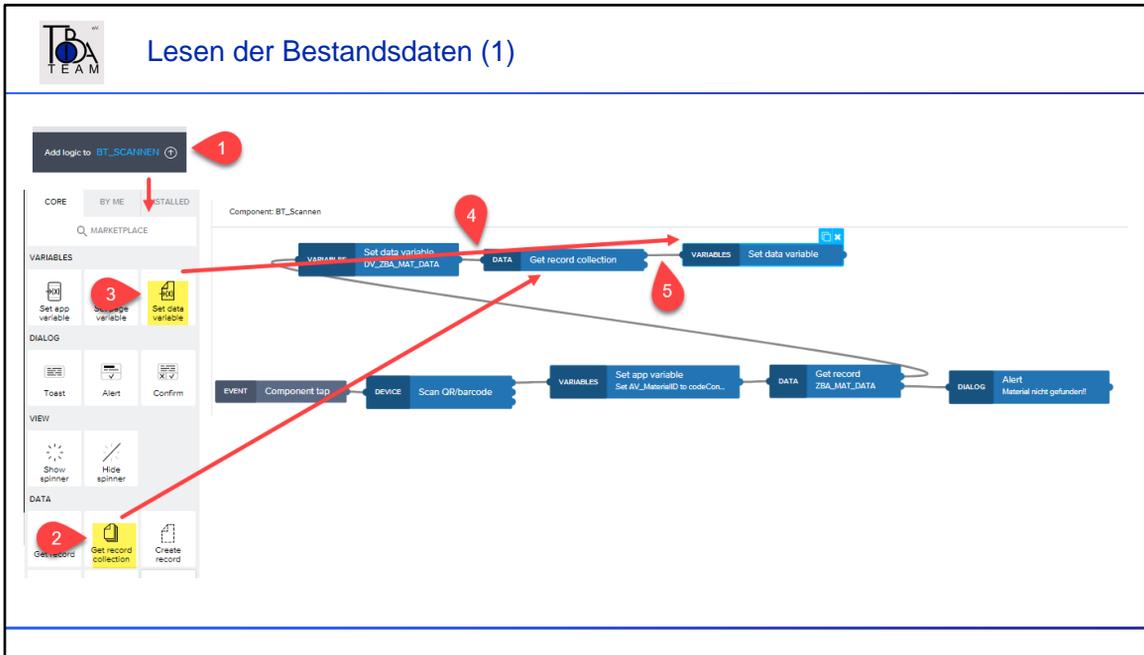
Das Layout des List-Items soll für einen Test zunächst einmal grob definiert werden.

1. Markieren Sie das List Item, das in der Scroll View eingebettet ist.
2. Passen Sie bei Bedarf den Namen des Items an (z.B. Präfix „LI_“ für List Item).
3. Da wir keine Detailseite zu einer Bestandszeile anbieten wollen, setzen wir „Arrow Visible“ auf „False“.
4. Öffnen Sie nun die Wertbindung für das 1. Listfeld (Primary Label).
5. Ordnen Sie dem Feld über „Data Item in repeat“ die Komponente „StockID“ zu.
6. Passen Sie Vorschauwert an, der in der Seite angezeigt werden soll.
7. Wiederholen Sie die Zuordnung für das Feld „Secondary Label“. Hier soll die Komponente „Available“ angezeigt werden.

Hinweis: Wir werden die Dartsellung der Zeile später noch anpassen und weitere Felder ergänzen.

App erstellen

- Lesen der Bestandsdaten -



Wir ergänzen die Logik für den Scannen-Button um das Lesen der Bestandsdaten.

1. Öffnen Sie zunächst den Logik-Editor für den Button BT_SCANNEN.
2. Ziehen Sie ein Data-Element „Get record collection“ in den Editor Bereich.
3. Ergänzen Sie die Logik um ein „Set data variable“.
4. Verbinden Sie den Ausgang von „Set Data variable DV_ZBA_MAT_DATA“ mit dem Eingang von „Get record collection“.
5. Verbinden Sie den Ausgang von „Get record collection“ mit dem Eingang des neu hinzugefügten „Set data variable“.

The screenshot shows the configuration of two logic nodes in SAP Business Builder:

- Get record collection:**
 - Resource name: ZBA_MAT_STOCK (1)
 - Basic authentication: Custom object (2)
 - Filter condition: Custom condition (3)
 - Condition type: Property (MaterialID) compared to (AV_MaterialID) with type 'equal' (4)
- Set data variable:**
 - Data variable name: DV_ZBA_MAT_STOCK (5)
 - Record collection: Get record collection (6)
 - Logic node: Get record collection (7)
 - Output: Collection of records (7)

Ergänzen wir zuerst das Logik-Element zum Lesen der Bestandsdaten (Get Record Collection).

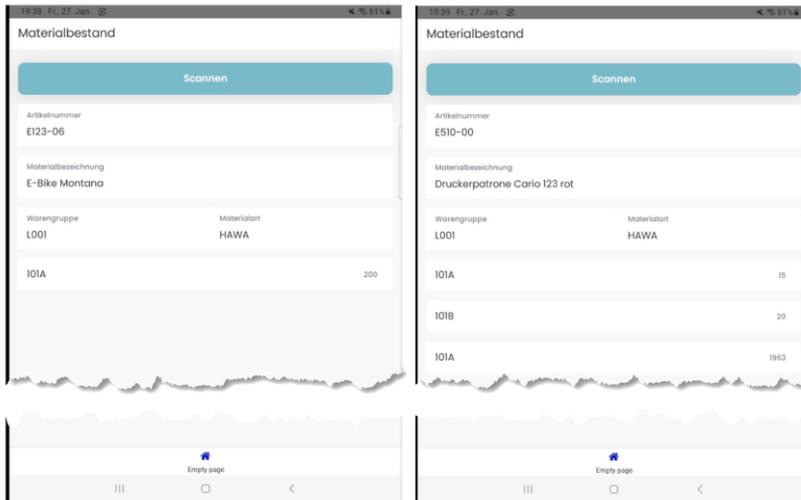
1. Ordnen Sie die Data Resource ZBA_MAT_STOCK unter „Resource Name“ zu.
2. Füllen Sie wie zuvor die „Basis authentication“ mit den Anmeldedaten.
3. Die Filterbedingung für das Lesen der Bestandsdaten muss ergänzt werden.
4. Zugeordnet wird die Bedingung, dass die MaterialID der Bestandsdaten mit dem Wert der App-Variablen AV_MaterialID übereinstimmen muss.

Auch die Parameter des neuen Elementes „Set data variable“ müssen angepasst werden.

5. Geändert werden soll die Data-Variable DV_ZBA_MAT_STOCK
6. Die Daten für die Variable sollen übernommen werden aus dem Ergebnis der Schrittes „Get record Collection“ (7).

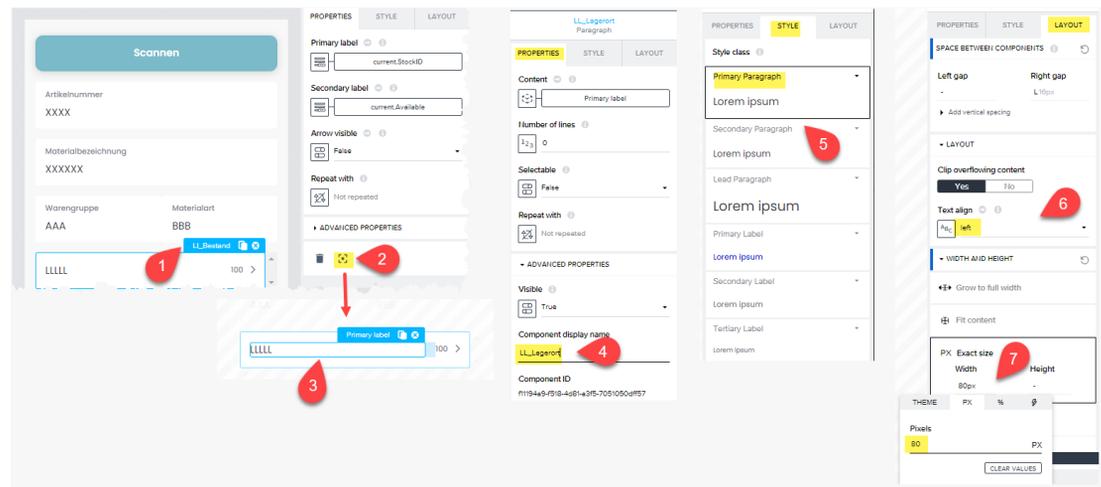
App erstellen

- Test : Bestandsdaten lesen und anzeigen -



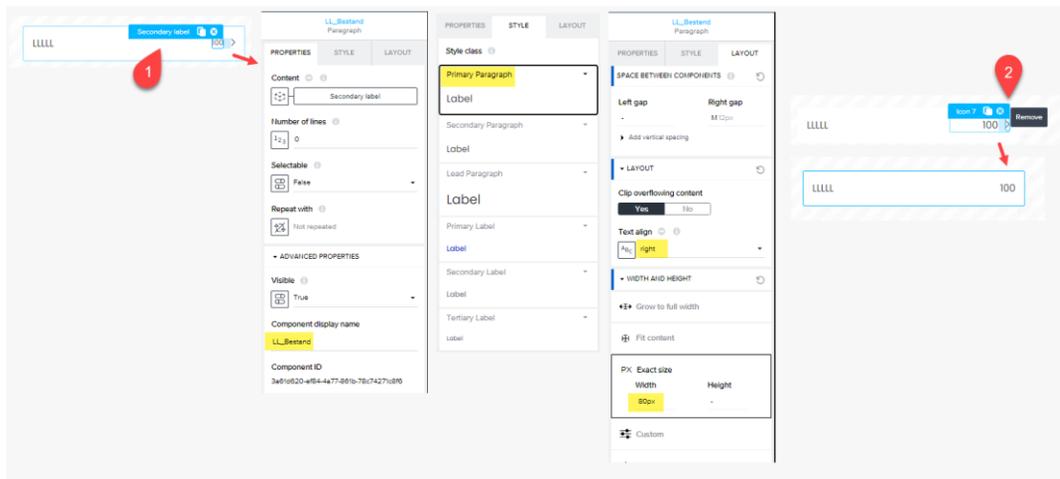
App erstellen

- Layout der Bestandsanzeige anpassen -

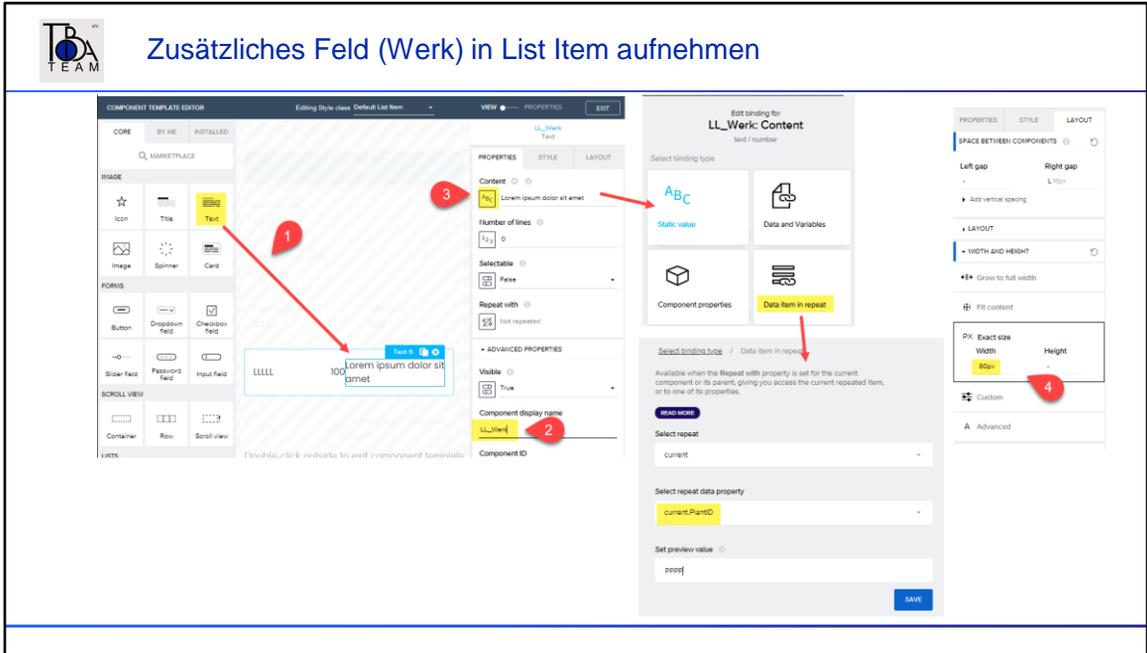


Das Layout eines List Items kann angepasst werden. Dies ist insbesondere dann notwendig, wenn Sie beispielsweise zusätzliche Felder anzeigen oder die Feldreihenfolge ändern möchten.

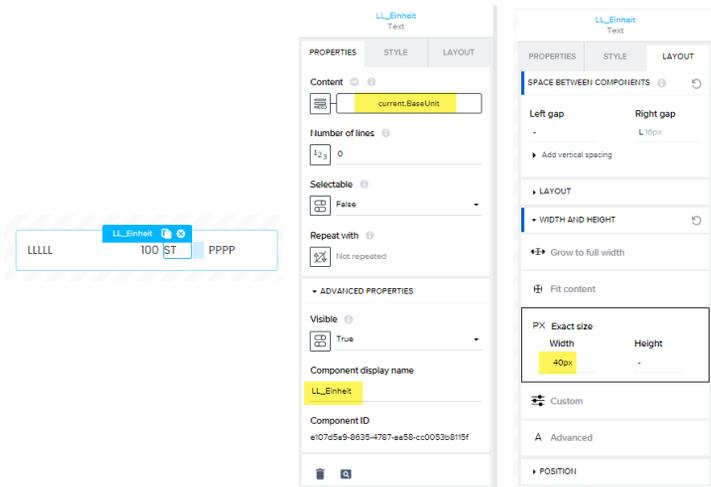
1. Markieren Sie das List Item.
2. Öffnen Sie den „Component template Editor“.
3. Markieren Sie das Feld (Label), das Sie bearbeiten möchte.
4. Bei Bedarf können Sie den Namen des Feldes anpassen.
5. Auf dem Tab-Reiter STYLE stehen Ihnen verschiedene Muster für die Anpassung der Schriftgröße zur Verfügung.
6. Der Tab-Reiter LAYOUT bietet Ihnen u.a. die Möglichkeit, die horizontale Ausrichtung des Textes einzustellen.
7. Die Feldbreite wird bei der Anzeige der Seite auf dem Endgerät per Default automatisch berechnet und berücksichtigt dabei z.B. die zur Verfügung stehende Breite der Anzeige. Sie können aber, wie gezeigt, die Breite eines Feldes auch exakt bemessen.



1. Das Bestandsfeld wird angepasst, die Schriftgröße, die horizontale Ausrichtung und die Feldbreite festgelegt.
2. Das nicht benötigte Icon wird gelöscht.



- Wir wollen ein zusätzliches Feld in die Listzeile aufnehmen.
- Ziehen Sie ein Text-Element in die Listzeile (Reihenfolge der Felder wird später angepasst!).
- Geben Sie dem Feld einen sprechenden Namen.
- Binden Sie das Feld an die Komponente PlantID.
- Passen Sie die Breite des Feldes an.



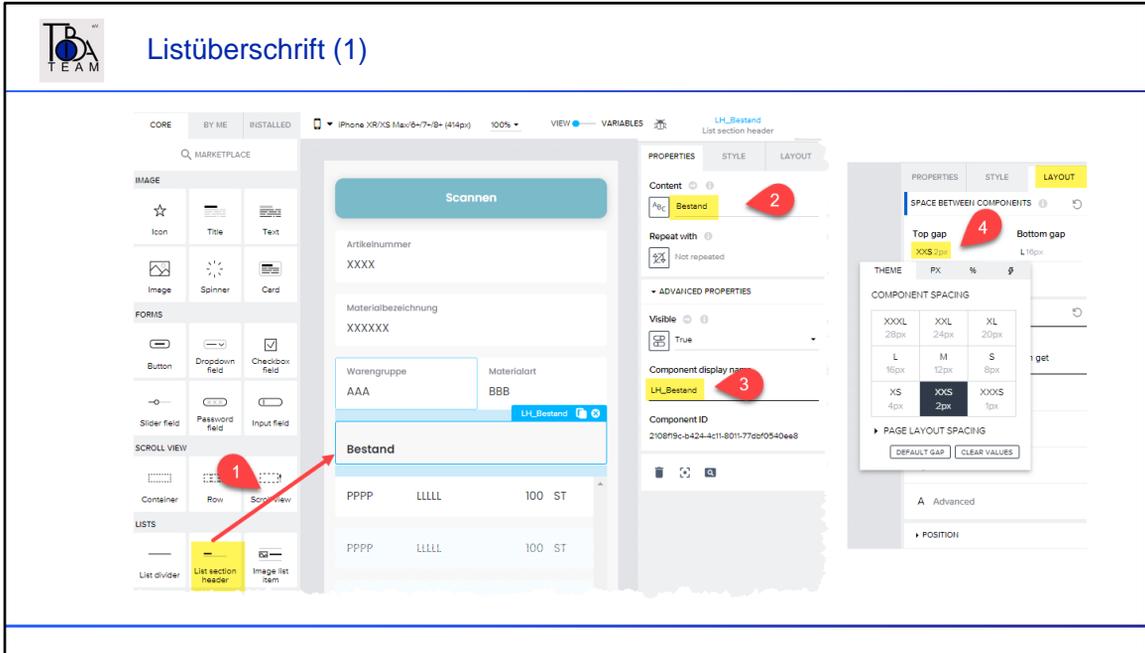
Zusätzlich wird das Feld LL_Einheit in die Zeile aufgenommen. Es wird an die Komponente BaseUnit gebunden.

The screenshot illustrates the process of adjusting the field order in the TREE view. It shows three columns of a hierarchical structure. The middle column has a red arrow pointing up and a red circle with '1' next to it, indicating a drag action. Below, a table shows a row with 'Container 3' selected, and a red arrow with a '2' points to a table below it, indicating a drop action.

Materialbestand

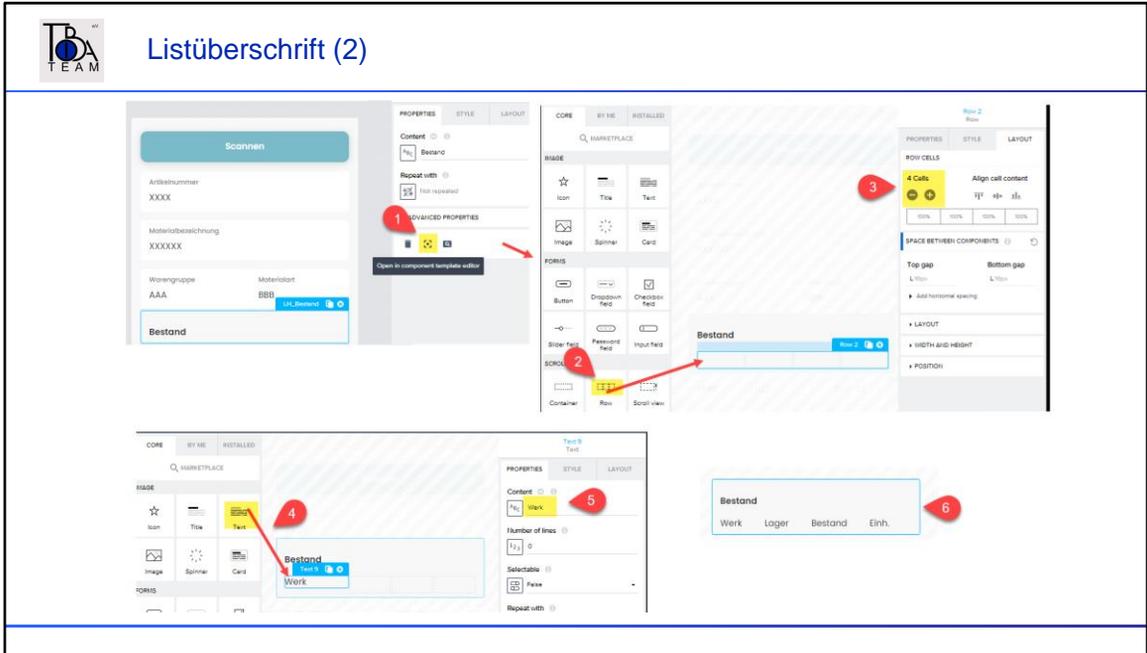
Scannen			
Artikelnummer	E510-00		
Materialbezeichnung	Druckerpatrone Cario 123 rot		
Warengruppe	Materialart	HAWA	
1020	101A	15	ST
1010	101B	20	ST
1010	101A	1963	ST

- Die Feldreihenfolge können Sie am schnellsten in der TREE-Auflistung per Drag & Drop verändern.
- Nicht benötigte Elemente können gelöscht werden.



Wir wollen eine Listüberschrift ergänzen-

1. Ziehen Sie einen „List section header“ per Drag & Drop auf die Seite und platzieren ihn vor die Scroll View.
2. Geben Sie die Beschriftung ein.
3. Ändern Sie bei Bedarf den Namen der Komponente.
4. Passen Sie, falls erforderlich, die Abstände zu den übrigen Seitenelementen an.



Die Listüberschrift soll um die Spaltenüberschriften ergänzt werden.

1. Markieren Sie die Überschrift und öffnen den Komponenten-Editor.
2. Ziehen sie ein Row-Element in das Überschriften-Element.
3. Erhöhen Sie die Anzahl der Zellen auf 4.
4. Ziehen Sie ein Text-Element in die erste Zelle des Row-Elementes.
5. Ändern Sie den Text.
6. Wiederholen Sie die Schritte 4 und 5 für die restlichen Spaltenüberschriften.

The image shows a design tool interface with several property panels and a table. Red arrows indicate adjustments being made to the table header. The panels show 'Width' set to 80px and 'Text align' set to 'right' and 'center'. The table has columns: Bestand, Werk, Lager, Bestand, Einh. with values: P PPP, L LLLL, 100, ST.

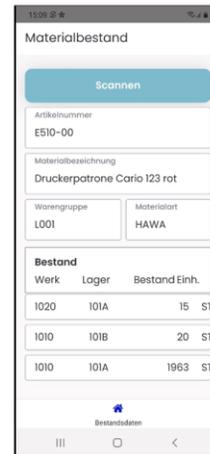
Bestand	Werk	Lager	Bestand	Einh.	
P	PPP	L	LLLL	100	ST
P	PPP	L	LLLL	100	ST

Einige letzte Korrekturen an der Überschrift,
Die Breite der Texte wurde angepasst, damit die Überschriften möglichst über den Werten in den Spalten stehen.

Tablet S6



Galaxy A40



Die Darstellung der Seite ist vom Endgerät abhängig, die Rendering nicht noch ganz perfekt.