

TIM Smartform Suite - DRAFT

1. Vorstellung und Abrenzung der TIM Smartform Suite von HTML Smartforms

- Konfigurieren statt Programmieren
- Schnell Smartforms erstellen. Validierung, Layout, Struktur
 - [Zugang zur TIM Smartform Suite](#)
- Ausbaustufen:
 - [TIM Smartform Designer](#)
 - [TIM Smartform Architect](#)
 - [TIM Smartform Expert](#)
 - Alle erstellten Konfigurationen können in allen Ausbaustufen geöffnet und je nach verfügbarem Funktionsumfang weiterbearbeitet werden.

2. Zugang zur TIM Smartform Suite

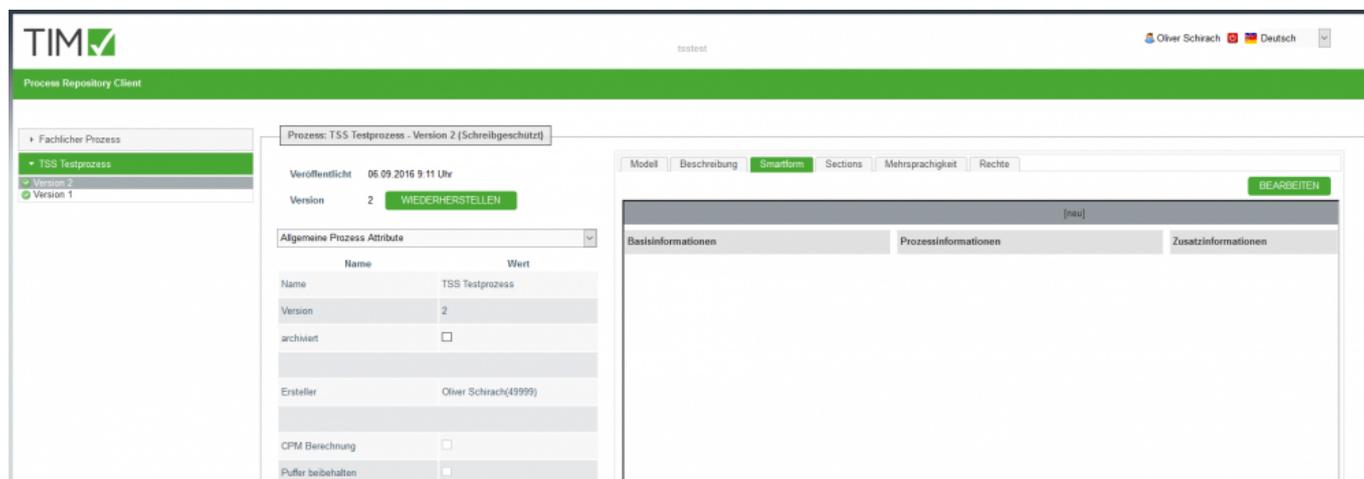
2.1 Systemrollen für den Zugang zur TSS

Die folgenden [Systemrollen](#) müssen im TIM Admin Client den entsprechenden Usern zugeordnet werden, so dass sie die Ausbaustufen der TIM Smartform Suite nutzen können:

- „smartform-designer“ für den Zugang zum [TIM Smartform Designer](#)
- „smartform-architect“ für den Zugang zum [TIM Smartform Architect](#)
- „smartform-expert“ für den Zugang zum [TIM Smartform Expert](#)

2.2 Zugang via Repository

Der Zugang zum Konfigurator der TSS erfolgt über das [TIM Repository](#). Im Tab Smartform des gewünschten Prozess befindet sich der BEARBEITEN-Button. Ein Klick auf ihn öffnet den Konfigurator.



2.3 Systemrolle für die Verwaltung von TSS Ressourcen

Alle Datei-Artefakte, die für die Erstellung von Smartforms benötigt werden, werden in der Datenbank zentral abgelegt. Zur Verwaltung der Artefakte wurde eine Oberfläche im Admin Client integriert.

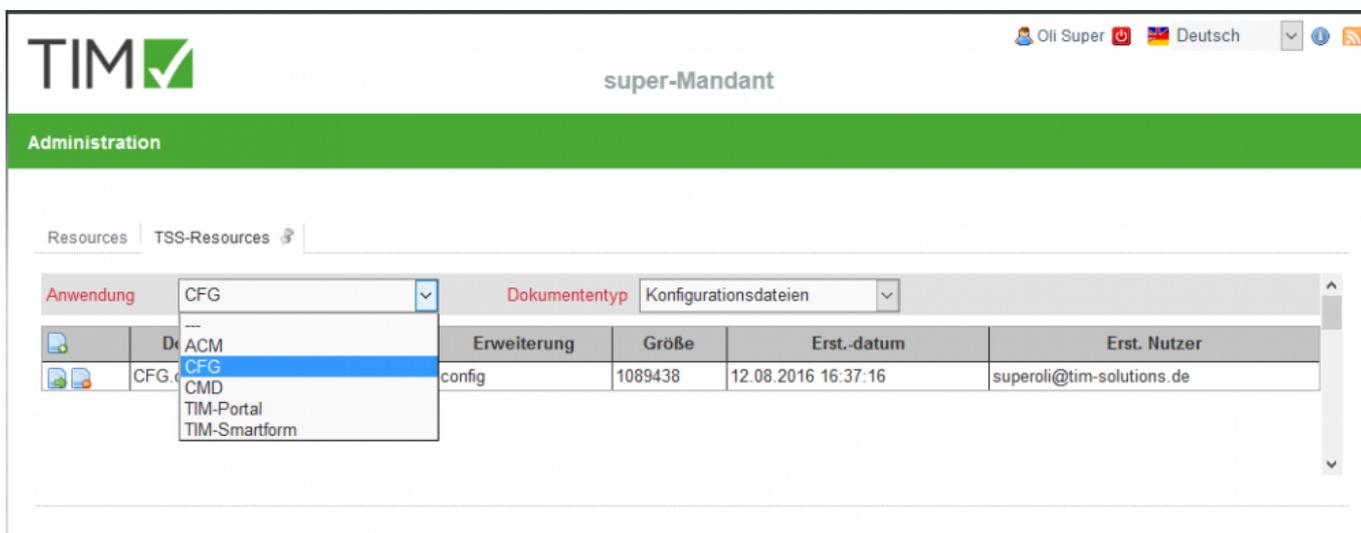
Die folgende Rolle wird zur [Verwaltung von TSS-Ressourcen](#) (z.B. Konfigurationsdateien, csv-Dateien, css-Dateien, etc.) in Ihrem Mandanten oder im super-Mandant benötigt:

- „resource-manager“ (Der Nutzer, der resource-manager ist, benötigt nicht die Rolle admin)

2.3.1 Globale Ressourcen im super-Mandant

Im super-Mandanten müssen nach der Installation/ einem Update der TSS die grundlegenden Konfigurationen (Dokumententyp Konfigurationsdateien) für die Applikationen CMD und CFG hochgeladen werden

- Die Konfigurationen für ACM, TIM-Portal und TIM-Smartform sind Bestandteil des Produktes. Sie können aber auf diesem Weg individualisiert zur Verfügung gestellt werden
- Über den Dokumententyp CSS-Dateien können Sie eine CSS-Datei zur Definition Ihres Cls (Highlightfarben, Akzente, etc.) global für den Konfigurator (CFG) und die Darstellung der Smartforms (TIM Smartform) zur Verfügung stellen.
 - Sie wird beim Upload automatisch in css.css umbenannt



- Über den Dokumententyp Logo-Dateien können Sie eine PNG-Datei Ihres Unternehmenslogos zur Verfügung stellen. Sie wird beim Upload automatisch in logo.png umbenannt und oben links im Konfigurator angezeigt

2.3.2 Mandantenspezifische Ressourcen

In Ihrem Mandanten können Sie analog Artefakte verwalten, die nur innerhalb des Mandanten gelten und verfügbar sind

- Hierzu zählt analog eine CSS-Datei zur Gestaltung des mandantenspezifischen Cls (Highlightfarben, Akzente, etc.)

- Weiterhin Datenformate wie csv, txt, xls, png, etc. zur Integration als Datenquellen in Smartforms

3. TIM Smartform Designer

3.1 Zielgruppe & Projektscope

- Fachnutzer, die ein einfaches Smartform erstellen möchten
- Vorgegebene Templates
- Bewusst eingeschränkte Auswahl an gängigen Datentypen
- Anlegen und Anordnen von Attributen
- Nur einfache Smartforms, d.h. ohne Tabpages, Tabellen, Accordions, etc.
 - Für komplexere Formulare sind die Ausbaustufen [TIM Smartform Architect](#) und [TIM Smartform Expert](#) verfügbar

3.2 Erklärung der Oberfläche und Funktionen - Zielgruppe Fachnutzer!

3.2.1 Oberfläche und Bedienphilosophie

Im Fokus des Smartform Designers steht das Smartform.

- Es basiert auf einem Template und bietet die Möglichkeit, Attribute per drag and drop anzuordnen.
- Somit können Sie sich ihr Formular so gestalten, wie es zu Ihrem Prozess passt.
- Die TSS ermöglicht Ihnen, ohne viel Vorwissen, eine einheitliche Bedienphilosophie in Ihren Smartforms umzusetzen.
 - Durch einheitliche Strukturen für die Anordnung von Elementen, Validierungen, Pflichtfeldern und Editierbarkeit, etc. finden sich Ihre Endnutzer sehr schnell in allen Prozessen zurecht

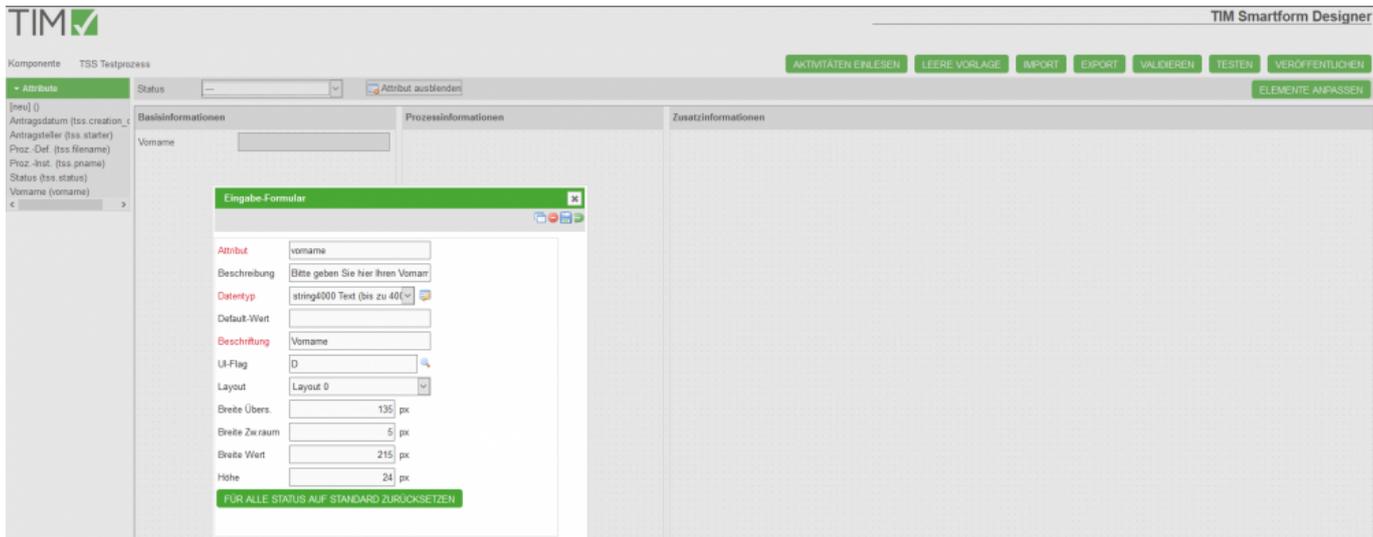


- Über den Button *Aktivitäten einlesen* können Sie die Aufgaben Ihres modellierten Prozess einlesen
- Über *Import* und *Export* können Sie eine bestehende Konfiguration als Grundlage importieren oder zum externen Backup/ Weiterverwendung in anderen Prozessen exportieren
- Über *Validieren* können Sie prüfen lassen, ob Ihre Konfiguration syntaktische Fehler enthält
- Über *Testen* können Sie sich live anschauen, wie Ihr Smartform in einer konkreten Aufgabe aussieht (Details siehe [3.2.3 Testmodus](#))
- Über *Veröffentlichen* sichern Sie Ihre Konfiguration. Alles, was Sie konfigurieren muss erst durch *Veröffentlichen* gesichert werden, bevor es im ToDo Client sichtbar wird.
 - Wenn Sie ohne *Veröffentlichen* Ihren Browser durch F5 neu laden oder ihn einfach schließen, dann sind die Änderungen unwiderrufbar weg
- Es empfiehlt sich ein häufiges *Veröffentlichen*, nachdem/ bevor Sie komplexe Änderungen am Smartform vornehmen
- Grundlegend wird das Smartform für den Wert „—“ in der Statusdropdown erstellt. Dies stellt die grundlegende Konfiguration dar.
 - Im Kapitel [3.2.4 Statusabhängige Konfiguration - Aktivitäten einlesen](#) ist beschrieben, wie Sie Ihr Formular für die einzelnen Prozessstatus und Aktivitäten im Prozess individualisieren können
- Per drag and drop können Attribut beliebig in der Reihenfolge innerhalb eines Elementes angeordnet, in ein anderes Element verschoben oder für einen Status ausgeblendet werden
 - Details siehe [3.2.2 Attribute](#)

3.2.2 Attribute

Im Smartform Designer werden die Attribute für die strukturierte Erfassung und Darstellung von Prozessinformationen angelegt.

- Ein neues Attribut wird durch einen Klick auf *[neu] ()* und drag&drop in das gewünschte Element angelegt.
 - Dies ist nur möglich, wenn die Status-Dropdown auf „—“ steht. (Details zum Statusabhängigen Konfigurieren siehe [Statusabhängige Konfiguration - Aktivitäten einlesen](#))
- Die Ablage erfolgt als TIM Prozessvariablen
- Die Attribute sind im Prozess mit `${technischer_Attributname}` für die Steuerung des Prozesses oder zur Verwendung in WordTemplates, Mails, etc. verfügbar
- Eine spätere Auswertbarkeit im Dashboard zur KVP ist gegeben
- Wenn ein neues Attribut auf ein Element gezogen wird, oder auf ein bereits platziertes Attribut geklickt wird, dann erscheint das Attribute-Popup zur Pflege der Eigenschaften:

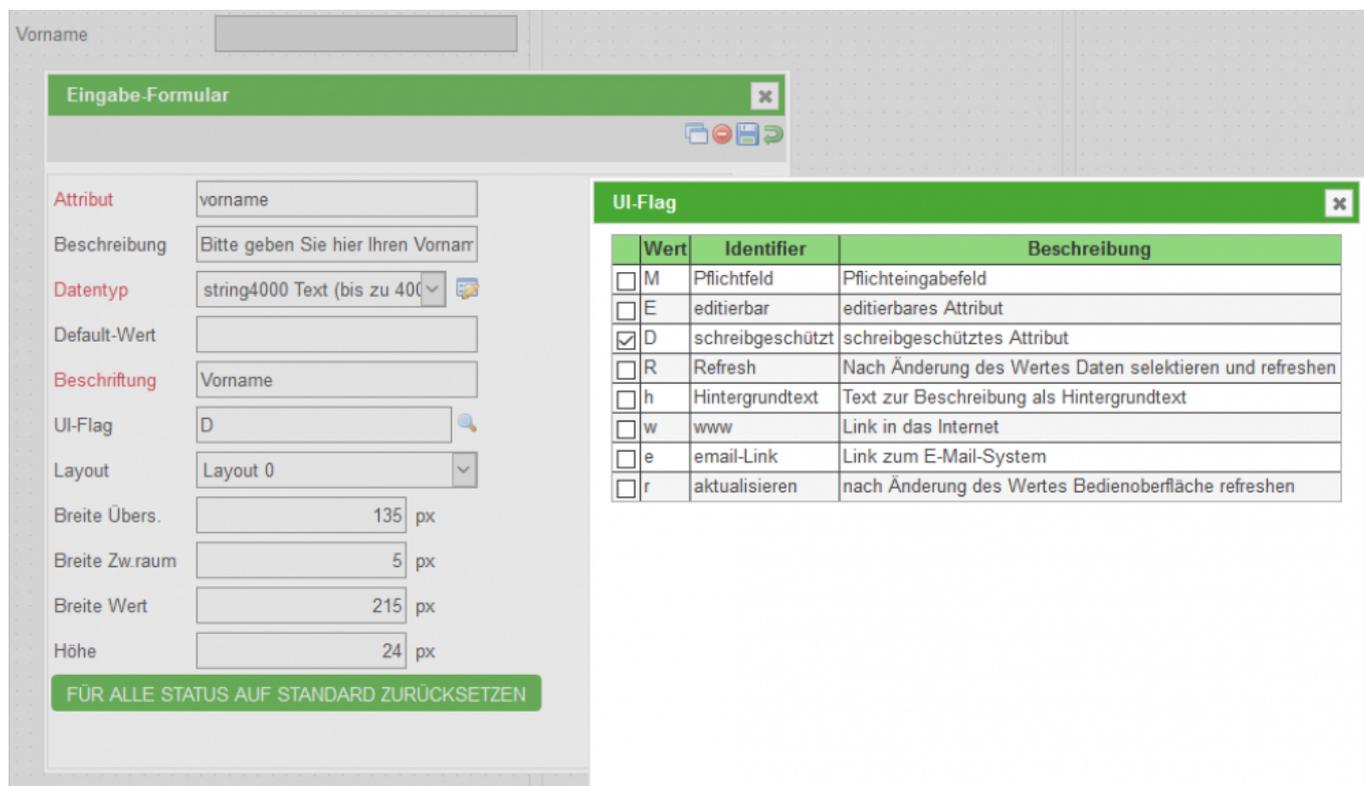


Feld	Beschreibung
Attribut	technischer Identifikator für das Attribut. Kann auch in Aufgabenmails mit $\${...}$ referenziert werden
Beschreibung	Erscheint als Tooltip, wenn der Nutzer mit der Maus über das Feld fährt
Datentyp	Auswahl eines Datentyps oder Erstellen eines eigenen zur individuellen Validierung (siehe Datentypen)
Beschriftung	Wird als Text links neben dem Eingabefeld dargestellt
UI-Flag	Definition der Editierbarkeit des Attributs (Details s.u.)
Layout	Auswahl eines Layouts zur Darstellung des Attributs (siehe Datentypen)
Breite Überschrift	Breite des Bereiches, in dem die oben angegebene Beschriftung dargestellt werden soll
Breite Zwischenraum	Abstand zwischen Beschriftung und Feld
Breite Wert	Breite des Eingabefeldes
Höhe	Höhe des Eingabefeldes. Bei allem über 24px wird die Layoutvariante „groß“ verwendet. (siehe Datentypen)
Buttons	Beschreibung
Für alle Status auf Standard zurücksetzen	Dieser Button erscheint, wenn die Statusdropdown auf „—“ steht. Damit können alle Status-spezifischen Varianten des Attributs zurückgesetzt werden
Auf Standard zurücksetzen	Dieser Button erscheint, wenn die Statusdropdown auf einem Status ungleich „—“ steht. Damit kann die Status-Variante auf die Grundkonfiguration des Attributs zurückgesetzt werden
„Zurücksetzen einzelner Attributvarianten auf Standard“-Icon	Jede Attributeigenschaft, die für einen ausgewählten Status von der Grundkonfiguration abweicht, wird fett dargestellt. Zudem erscheint daneben ein Icon, mit dem die betroffene Eigenschaft auf die Grundkonfiguration zurückgesetzt werden kann
Kopieren-Icon	Nutzt das geöffnete Attribut als Grundlage für eine Neuanlage. Die Felder <i>Attribut</i> und <i>Beschriftung</i> werden zurückgesetzt
Löschen-Icon	Löscht das Attribut aus der Konfiguration. Dieses Icon ist nur im Status „—“ verfügbar. Wenn Sie ein Attribut für einen Status nur ausblenden möchten, dann ziehen Sie es bitte per drag and drop auf die Fläche <i>Attribut ausblenden</i> neben der Statusdropdown. Sobald die Fläche rot wird, lassen Sie die linke Maustaste los. Das Attribut wird für den Status ausgeblendet, es verbleibt aber links in der Attributauswahlliste
Speichern-Icon	Speichert die getätigten Änderungen/ Neuanlage
Rückgängig-Icon	Setzt die getätigten Werte auf den Stand zurück, zu dem das Attribut das letzte mal gespeichert wurde

Hinweis: Details zum Statusabhängigen Konfigurieren siehe [Statusabhängige Konfiguration - Aktivitäten einlesen](#)

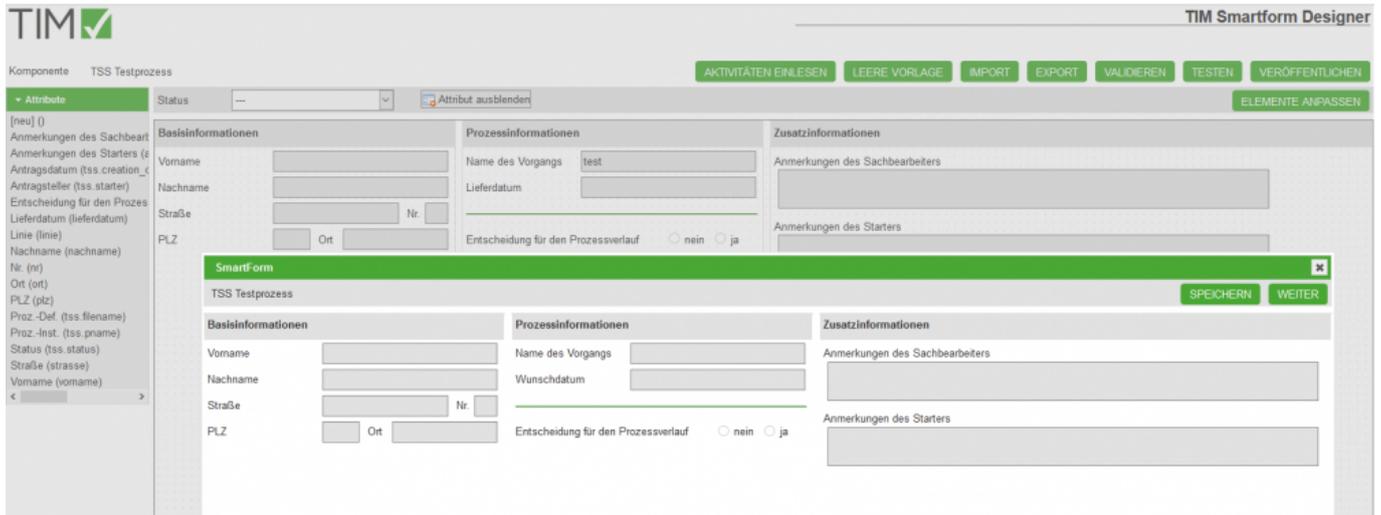
- Die Neuanlage, sowie alle getätigten Änderungen müssen durch einen Klick auf das Speichernsymbol abgeschlossen werden.
 - Das Popup kann danach geschlossen werden und das Attribut erscheint im gewünschten Element.
- Per drag and drop kann jedes Attribut beliebig in der Reihenfolge innerhalb eines Elementes angeordnet, oder in ein anderes Element verschoben werden
- Wenn ein Attribut ausgeblendet werden soll, dann muss es per drag and drop auf die Fläche *Attribut ausblenden* neben der Statusdropdown gezogen werden. Screenshot siehe [Statusabhängige Konfiguration - Aktivitäten einlesen](#)
 - Das Attribut bleibt auf diesem Weg in der Konfiguration erhalten, wird aber für den ausgewählten Status nicht mehr dargestellt

Die UI-Flags dienen zur Konfiguration der Editierbarkeit und können im Attribute-Popup entweder durch die Lupe in einem Popup ausgewählt, oder direkt eingegeben werden:



3.2.3 Testmodus

Der Testmodus dient dazu, das Smartform für einen ausgewählten Status zu testen, d.h. es öffnet sich nach Klick auf *TESTEN* ein Popup, in dem das Smartform voll funktional dargestellt wird. Dort können Sie prüfen, ob für den jeweiligen Status die konfigurierten Attribute sichtbar/ editierbar sind:



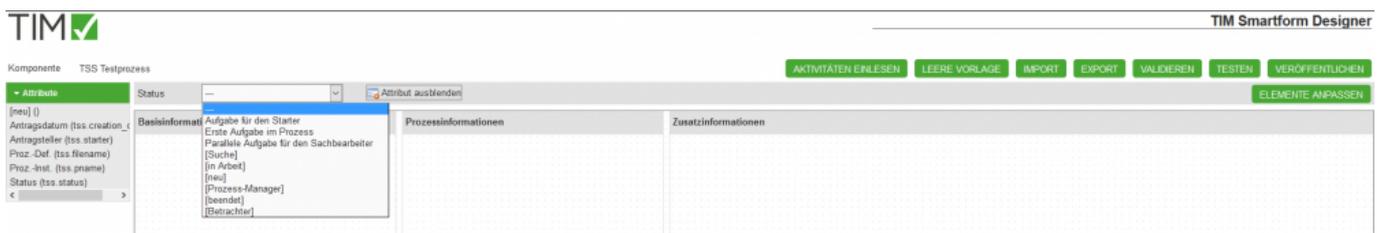
3.2.4 Statusabhängige Konfiguration - Aktivitäten einlesen

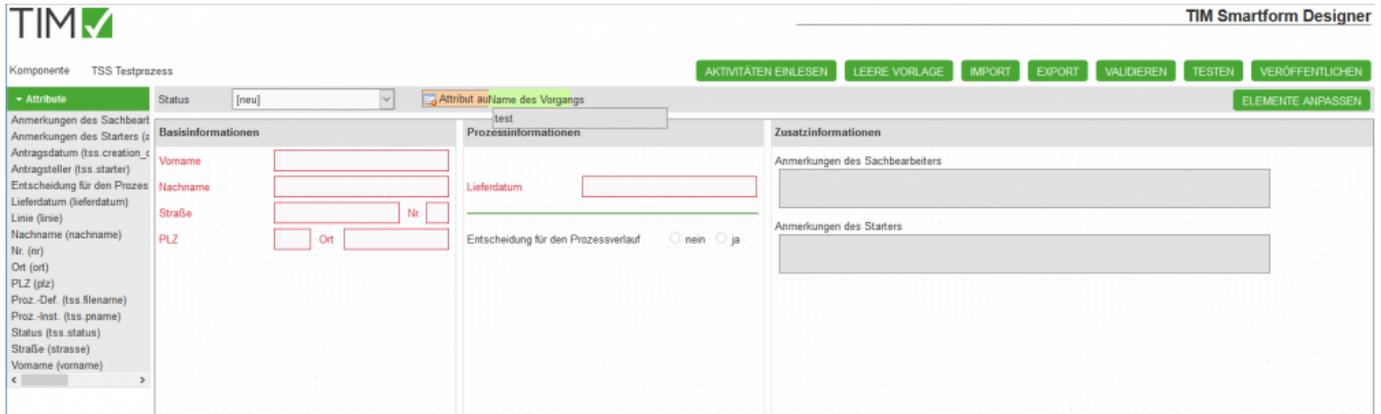
Nachdem Sie Ihr Smartform für den Status „—“ konfiguriert haben, können Sie es nun an die Besonderheiten der Aktivitäten im Prozess anpassen. Hierzu klicken Sie bitte auf **AKTIVITÄTEN EINLESEN**. Die Dropdown füllt sich mit den Namen der Aktivitäten.

- Nun können Sie eine Aktivität, oder einen der Standard-Status [neu, Suche, in Arbeit, Prozess-Manager, beendet, Betrachter] auswählen und das Formular entsprechend anpassen
 - Einfügen bestehender Attribute
- Verschieben oder ausblenden von bereits angeordneten Attributen
- Modifikation der Attributeigenschaften durch einen Klick auf ein Attribut. Details siehe [3.2.2 Attribute](#)

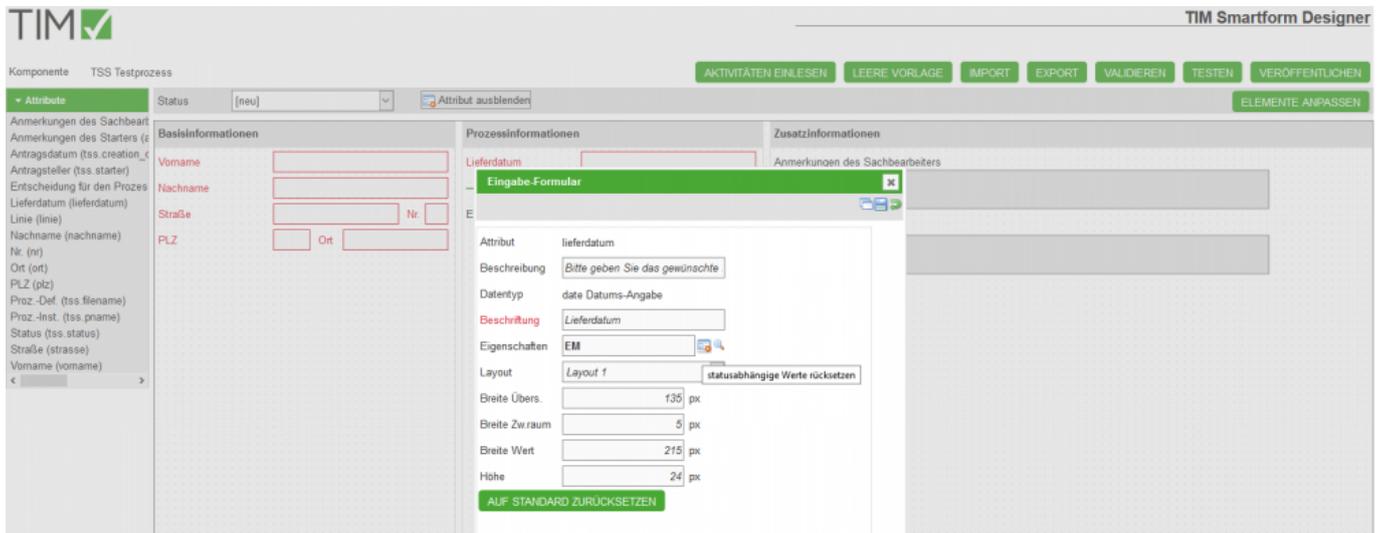
* z.B. können Sie dadurch einstellen, dass ein Attribut für eine konkrete Aktivität ein Pflichtfeld wird

Hinweis: Wenn Sie ein Attribut nur für einen konkreten Status benötigen, dann müssen Sie es zuerst für den Status „—“ anlegen, und dort auf *Attribut ausblenden* schieben. Wenn Sie nun den gewünschten Status auswählen, dann können Sie das Attribut links in der Attributliste auswählen und in das gewünschte Element schieben.

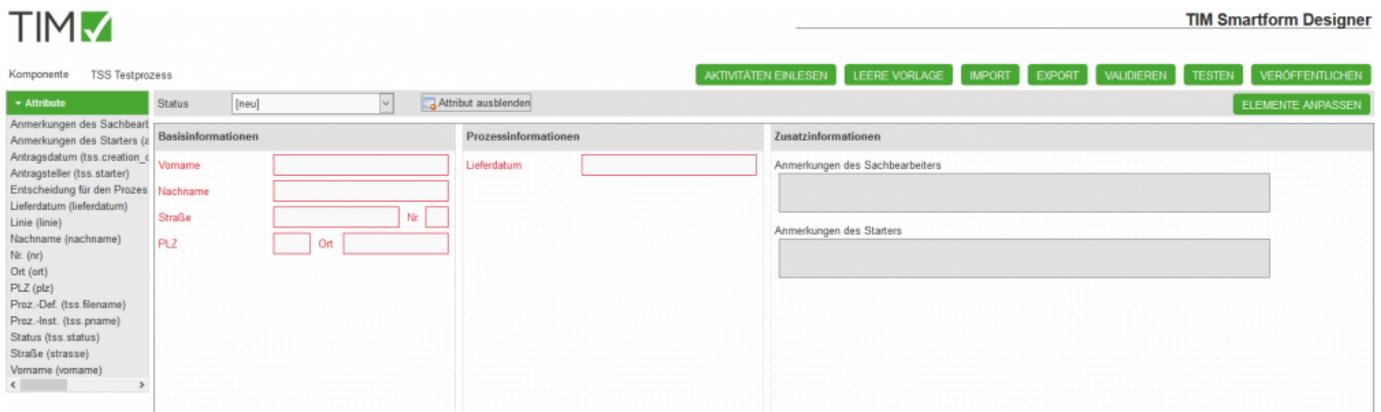




- Im Attribute-Popup werden statusspezifische Konfigurationen durch **fett** dargestellt. Konfigurationen, die sich aus dem Status „—“ ergeben werden *kursiv* dargestellt

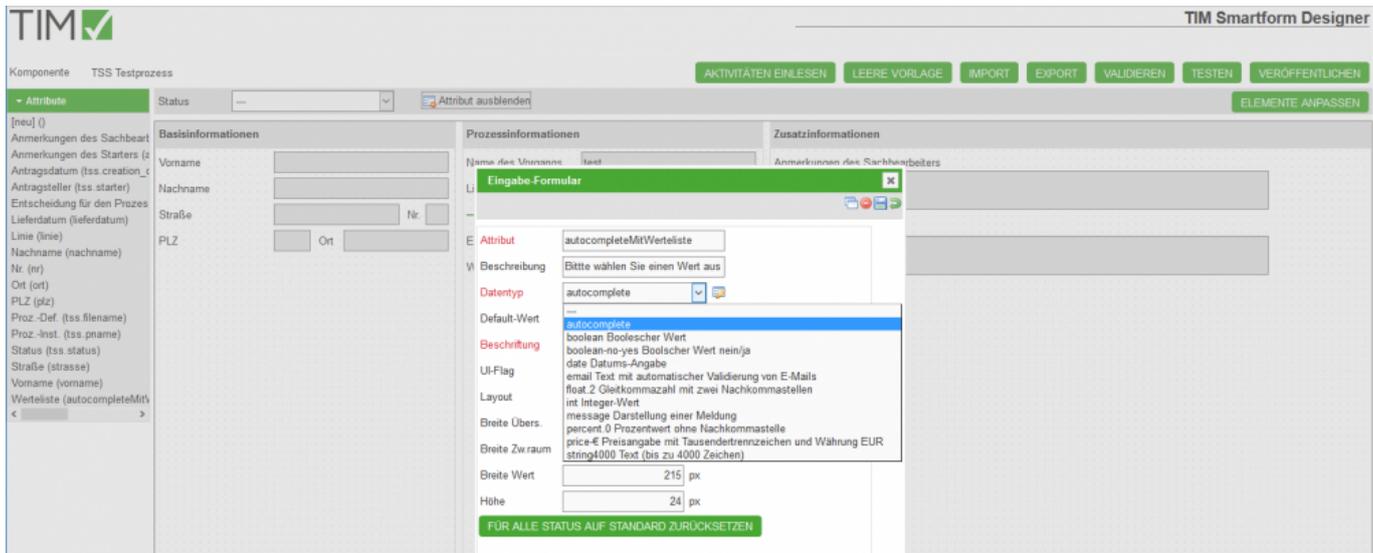


- Beispielhafte Anpassung des oben erstellten Smartforms für den Status [neu], welcher zum Prozessstart gilt



3.2.5 Datentypen

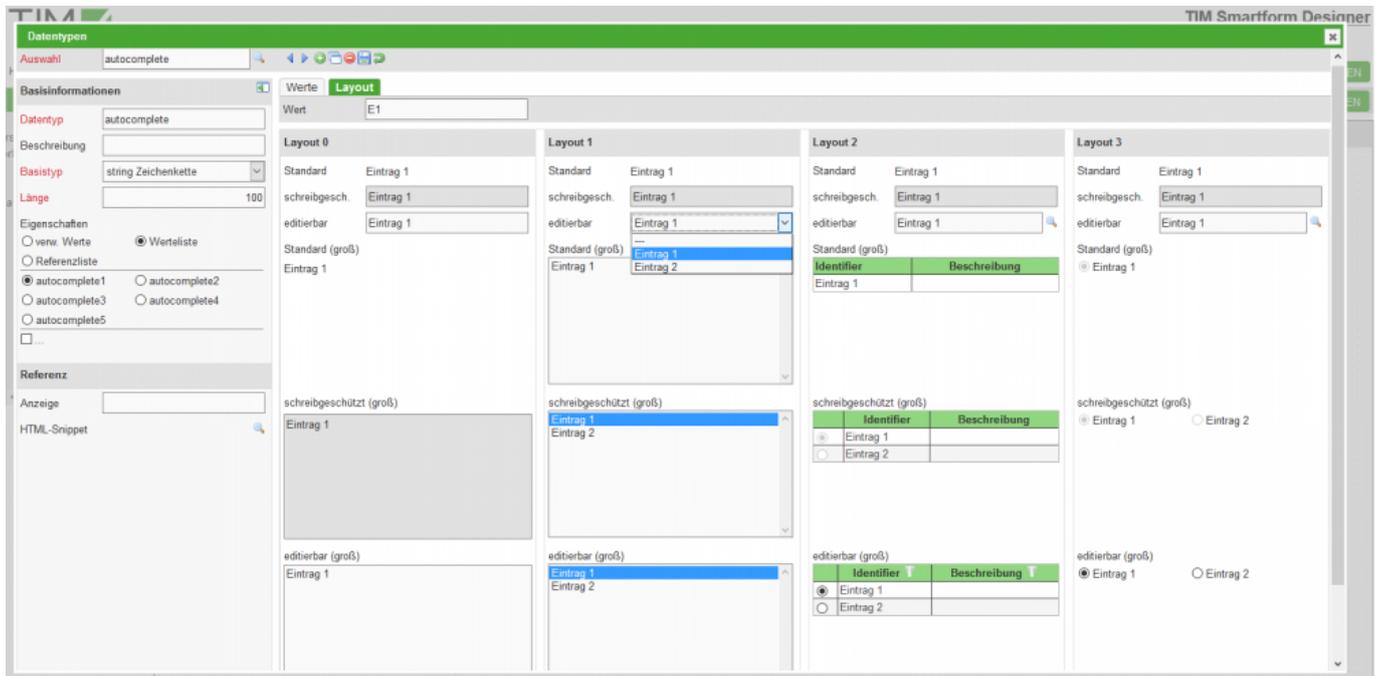
Im Attribute-Popup können Sie aus einer Liste vordefinierter Datentypen auswählen:



- Im Smartform Designer wurde diese Liste bewusst auf die in der folgenden Tabelle dargestellten Typen eingeschränkt

Datentyp	Beschreibung	Beispielhafte Verwendung
boolean	Boolescher Wert	Als Checkbox
decision-no-yes	Boolescher Wert nein/ja	Als Radio Button mit Ausprägung Ja/ Nein
date	Datums-Angabe	Als Textfeld mit Date Picker
email	Text mit automatischer Validierung von E-Mails	Als Textfeld
float.2	Gleitkommazahl mit zwei Nachkommastellen	Als Textfeld, welches den Input versucht, zu einer Zahl mit 2 Nachkommastellen umzuwandeln. Alle anderen Eingaben werden verweigert
int	Ganze Zahl	Als Textfeld, welches den Input versucht, zu einer Zahl umzuwandeln. Alle anderen Eingaben werden verweigert
message	Darstellung einer Meldung	Zur Darstellung von Hinweistexten
percent.0	Prozentwert ohne Nachkommastelle	Als Textfeld mit fester Einheit % und Validierung auf Zahlen
price-€	Preisangabe mit Tausendertrennzeichen und Währung EUR	Als Textfeld mit fester Einheit EUR und Validierung auf Zahlen mit 2 Nachkommastellen
string4000	Zeichenkette mit einer Maximallänge von 4000 Zeichen	Als ein- oder mehrzeiliges Textfeld

- Über einen Klick auf das Symbol rechts neben der Datentyp-Dropdownliste im Attribute-Popup können Sie durch die Layoutvarianten der Datentypen blättern

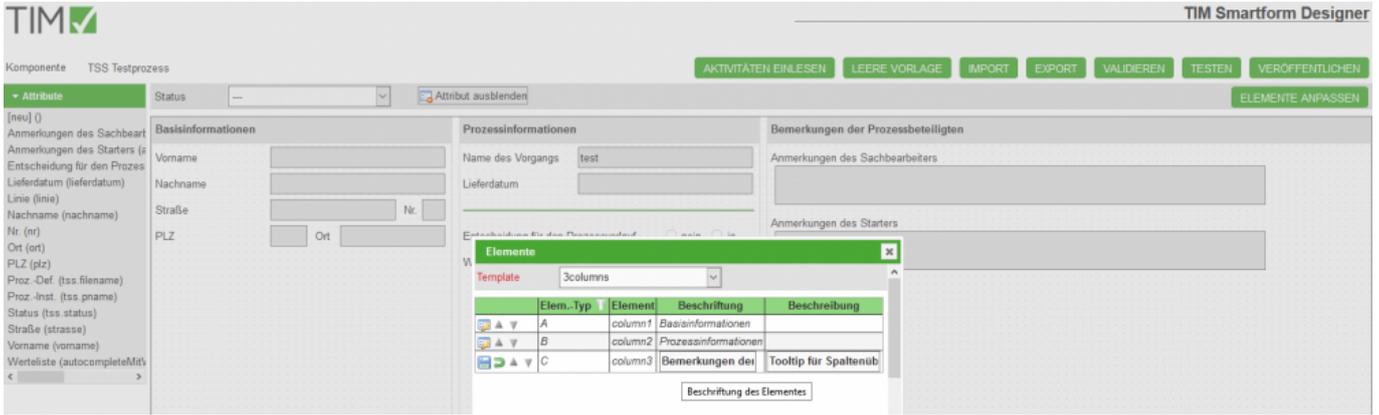


- zum anderen können Sie im Popup *Datentypen* einen eigenen Datentyp durch Klick auf das PLUS-Icon anlegen
 - Eigene Datentypen basieren dabei auf den vorhandenen Grunddatentypen
 - Durch die Angabe von Längenbeschränkungen oder Eingabemasken kann somit eine Validierung, z.B. auf Eingabe eines syntaktisch gültigen Buchungskreis `XX_098_zt`, konfiguriert werden
- Details zur Anlage von Datentypen siehe [6.4 Modul Datentypen](#)
 - Komplexe Datentypen z.B. für die Integration von CSV Listen als Autocomplete oder Dropdowns, etc. können im [TIM Smartform Architect](#) definiert werden
 - Angelegte Datentypen tauchen dann in der Dropdownliste *Datentyp* im Attribute-Popup mit auf und können durch den Designer verwendet werden

3.2.6 Templates

Grundlegend basiert das Smartform auf einem Template, welches das grobe Layout vorgibt. Serienmäßig wird das Template *3columns* vorgegeben. Wenn Sie ein individuelles Template für Ihren Prozess benötigen, so kann dies im Smartform Architect oder Expert erfolgen. Hierzu öffnet ein Nutzer mit der Rolle *smartform-architect* oder *smartform-expert* Ihre Konfiguration und erstellt ein Template. Anschließend können Sie edas Template via *ELEMENTE ANPASSEN* auswählen und Ihre Attribute darin anordnen **Hinweis:** Der Smartform Architect richtet sich an Fachuser. Deswegen ist eine Pflege von komplexen Layoutelementen, wie z.B. Tabs und Platzhalter für die Darstellung von Tabellen, nicht im Designer pflegbar

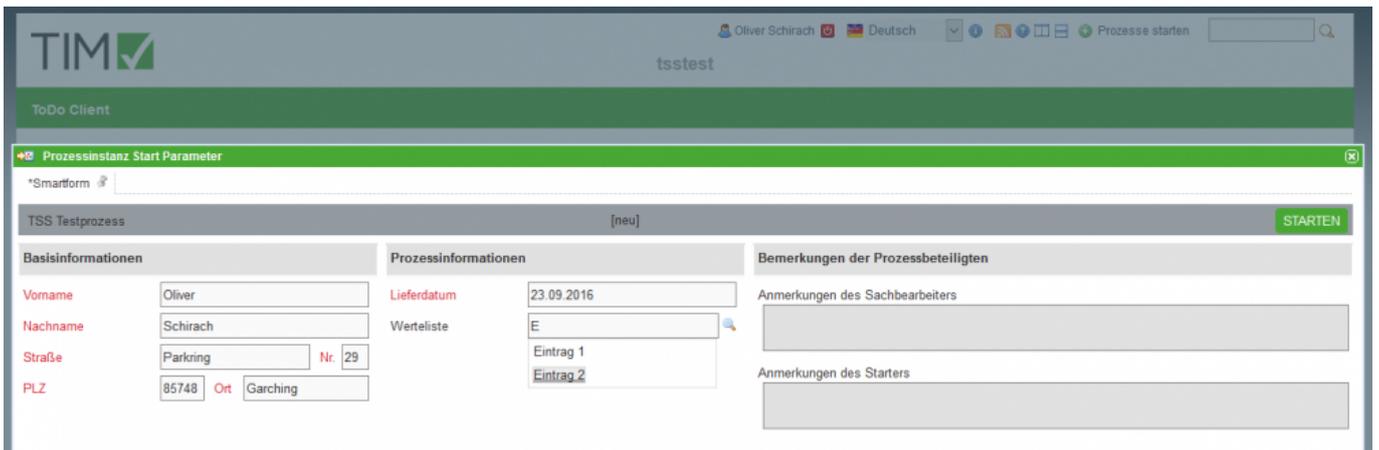
- Via *ELEMENTE ANPASSEN* können Sie zudem die Überschriften der Template-Elemente anpassen:



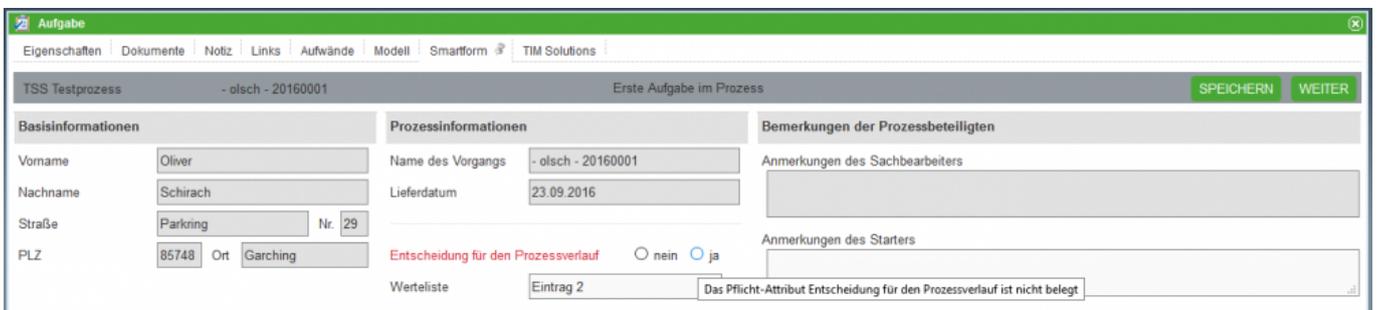
3.2.7 Ergebnis im ToDo Client

Der folgende Screenshot zeigt das konfigurierte Smartform einmal für den Prozessstart und einmal für die erste Aufgabe. Sie können sehen, dass teilweise Attribute ausgeblendet, oder rot als Pflichtfelder hervorgehoben sind:

- Smartform für den Start eines Vorgangs „TSS Testprozess“:



- Smartform für die Erledigung der ersten Aufgabe „Erste Aufgabe im Prozess“ Vorgangs „TSS Testprozess“:



3.2.8 Config zum Selbertesten

Die Konfiguration, anhand der die Screenshots erstellt wurden, können Sie hier einsehen und für Ihre Tests verwenden: [Beispielkonfiguration](#)

4. TIM Smartform Architect

4.1 Zielgruppe & Projektscope

- IT Berater
- Erstellung komplexer Smartforms mit Datenstrukturen, Tabpages, etc.
- Integration von strukturierten Daten aus csv, txt, xls oder Datenbanktabellen
- Erstellung von Übersetzungen für mehrsprachige Smartforms in DE und EN
- Erstellung von Datentypen, Templates und Meldungen für einen Prozess
- Einfacher Zugang zu Groovy für showInterface zum modifizieren der Benutzeroberfläche für Sonderkonstellationen oder Feldabhängigkeiten
- ALLES, was für einen Prozess definiert wird, d.h. Templates, Datentypen, etc. ist nur innerhalb dieses Prozesses gültig und existent!

4.2 Erklärung der Oberfläche und Funktionen

- Überblick: Menü; Sidebar; WYSIWYG
 - Details siehe [6. Module des TIM Smartform Architect und Expert](#)
-

5. TIM Smartform Expert

5.1 Zielgruppe & Projektscope

- IT Berater und Entwickler
- Tiefe Systemintegration
- Komplexe Konfigurationsmöglichkeiten
- Groovy extended. Auch zur Verwendung im Prozess via CBAGenericActionHandler möglich
- ALLES, was für einen Prozess definiert wird, d.h. Templates, Datentypen, etc. ist nur innerhalb dieses Prozesses gültig und existent!

5.2 Erklärung der Oberfläche und Funktionen

- Überblick: Menü; Sidebar; WYSIWYG
 - Details siehe [6. Module des TIM Smartform Architect und Expert](#)
-

6. Module des TIM Smartform Architect und Expert

6.1 Modul Bedienoberfläche

6.2 Modul Templates

6.3 Modul Tabellen

- dau- immer entsprechend des Dateityps, oder Datenbankzugriff
- Bei Ablage als Prozessvariable (BL-Flag P) immer dau-tim verwenden. Achtung: Max. 4000 Zeichen in einer Prozessvariable erlaubt. Dies schließt Steuerzeichen der Tabelle mit ein. Bitte nur zum fixen Testen verwenden

6.4 Modul Datentypen

Auflistung aller Basistypen und Details dazu

Reguläre Ausdrücke

- Stellen Pattern erlaubter Zeichenketten dar.
- Ist im Feld für den regulären Ausdruck ein Pattern eingegeben, ist der Attributwert nur erlaubt, wenn er dem Pattern entspricht.
- Syntax von regulären Ausdrücken ist nicht einfach – bitte Literatur verwenden, z.B. <http://www.regular-expressions.info/quickstart.html>.

Masken

- Geben dem Nutzer definierte Pattern zur Eingabe vor und lehnen Falscheingaben ab:
 - 0 .. Vorzeichen und Ziffern
 - 9 .. nur Ziffern
 - a .. Buchstaben (A-Z,a-z) sowie '-' und '_'
 - z .. nur Buchstaben (A-Z,a-z)
 - .. Buchstaben (A-Z,a-z) und Ziffern sowie '-' und '_'

6.5 Modul Meldungen

6.6 Modul Übersetzung

6.7 Modul funktionale Struktur

6.8 Verwaltung von TSS-Ressourcen

7. praktische Tipps für die gängigsten Smartformanforderungen

7.1 Attribute in Taskliste darstellen

- Attribute, die in der Taskliste angezeigt werden sollen, sind sog. Indexfelder
- Es gibt 10 Felder, die mit beliebigen Attributen belegt werden können
- Modul Tabellen - tim_process auswählen - Tab Attribute - Attribut auswählen - rechts in der Sidebar unter Geschäftslogik das BL-Flag auf 1 oder 2 ... oder 0 setzen - speichern - Veröffentlichen - Echten Prozess zum Testen starten
- Die Spaltenüberschriften können mittels dem folgenden Eintrag in der custom.js für jeden Mandanten angepasst werden:
 - `gadget.languages.de.standard_smartformindex1= 'Überschrift für Index 1 in DE';`
 - `gadget.languages.en.standard_smartformindex2= 'Überschrift für Index 1 in EN';`

7.2 Besondere Eigenschaften von Attributen

- Zusätzliche Flags zu „E“ oder „D“ für:
 - Mail (e): Es wird automatisch rechts neben dem Eingabefeld ein kleines Icon gerendert. Wenn der Nutzer auf das Icon klickt, dann öffnet sich das Systemstandardprogramm für den Versand einer Mail. Die Empfängeradresse wird dabei vorbelegt
 - HTML-Link (w): Es wird automatisch rechts neben dem Eingabefeld ein kleines Icon gerendert. Wenn der Nutzer auf das Icon klickt, dann öffnet sich der Systemstandardbrowser und die eingegebene Webseite wird geladen.
- Zusätzliche Flags zu „E“:
 - Placeholder (h): Der eingegebene Text aus dem Feld Beschreibung wird nicht nur als Tooltip verwendet, sondern auch als Hintergrundtext in editierbaren einzeiligen Textfeldern. Sobald der Nutzer etwas eintippt, verschwindet der Hintergrundtext
 - refresh (r): Dies führt dazu, dass nach Eingabe durch einen Nutzer in das Feld, das Smartform neu geladen wird. Alle Daten bleiben erhalten. Dadurch wird aber die Groovy Methode `showInterface` erneut ausgeführt. Dort abgelegte Funktionen können nun basierend auf der Eingabe des Nutzers z.B. Attribute ein-/ ausblenden oder Pflichtfeldbedingungen individualisieren
 - Werden diese beiden Flags in Kombination mit „D“ und/oder ohne „E“ genutzt, dann haben sie in dem Moment keine Wirkung
- Umbruch Labelbreite zu Feldbreite:
 - Per default wird ein Attribut immer wie folgt gerendert: Label + Zwischenabstand + Eingabefeld
 - Wenn nun das Eingabefeld einmal unter dem Label erscheinen soll, dann muss die Breite des Labels, d.h. Breite Übers., einfach breiter eingestellt werden. Wenn nun Breite Übers. + Breite Zw.raum + Breite Wert > Breite des Elements, in dem das Attribut platziert wird, dann bricht das Eingabefeld automatisch nach unten um

7.3 Einbinden einer csv-Datei für ein Autocomplete-/Dropdownfeld

- Zuerst muss im Modul Tabellen eine Tabelle angelegt werden, die genau so heißt, wie die vorliegende csv, txt oder xls
 - Die Tabelle bekommt die entsprechend ihrer Endung passende DAU zugeordnet.
 - Die BLU muss immer auf blu-basic gesetzt werden
 - Nun müssen in dieser Tabelle alle Spalten, die in der zugrundeliegenden Datei existieren, als Attribute angelegt werden
 - Dies ist analog zu den Attributen im Smartform auszufüllen
 - Wichtig: DA-Flag immer auf „SIU“ setzen
 - Indizes: Hier geben Sie an, welche Spalten zur eindeutigen Referenz (= Index 1) eines Datensatzes zusammengezogen werden müssen
 - Ref.-Typ:
 - Hier entscheiden Sie, welches Attribut mit „D“ automatisch als Bezeichnung, z.B. später in einer Dropdown verwendet wird
 - Hier entscheiden Sie, welches Attribut mit „I“ automatisch als fachlicher Kurzidentifizier verwendet wird
 - Hier entscheiden Sie, welches Attribut mit „V“ automatisch verwendet wird, wenn ein ausgewählter Eintrag gespeichert wird
 - Details siehe unten (Exemplarisches Ergebnis eines Dropdowneintrags im HTML: `<option value=„Attribut mit V“>„Attribut mit I“ „Attribut mit D“ </option>`)
- Nun muss die Datei via Tab Ressources im Admin Client hochgeladen werden. Hierbei ist der Datentyp Datendatei auszuwählen
 - Nach dem Upload bitte den Button Cache löschen klicken, damit die neueste hochgeladene Version verwendet wird
- Datentyp für Dropdown
 - Damit die Tabelle nun als Grundlage für eine Dropdown oder ein Autocompletefeld verwendet werden kann, muss hierfür ein datentyp angelegt werden
 - Basistyp: = Datentyp der Spalte aus der CSV, die als eindeutige Referenz dient
 - Eigenschaften: Referenzliste
 - Tabelle: vorher angelegte Tabelle
 - Attribut: Das Attribut aus der Tabelle, das in der DB gespeichert werden soll. Es muss eindeutig sein. Entspricht dem Attribut mit dem „V“
 - Filter: optional - hier kann mit der Syntax ATTRIBUT= die Ergebnismenge der Tabelle auf ein Attribut eingeschränkt werden. Mittels `ATTRIBUT={ATTRIBUT_IN_TIM_PROCESS}` kann auch auf den Inhalt eines anderen Attributs gefiltert werden, um somit eine Filterung zu ermöglichen, die vom Inhalt eines anderen Attributs abhängig ist. WICHTIG: Das Attribut, wonach gefiltert werden soll, muss das Flag „r“ haben, damit bei einem Wechsel des Wertes auch die Tabelle mit dem neuen Filter geladen wird
 - Anzeige:
 - leer lassen, dann zieht der Default, bestehend aus I und D
 - oder mit {Attribut} die Attribute auswählen, die angezeigt werden sollen. z.B.: {nachname}, {vorname} ergibt: „Schirach, Oliver“
- Alternativ kann die Tabelle auch als Grid eingebunden werden, um Datensätze tabellarisch darzustellen oder bearbeitbar zu machen

7.4 Erstellung von Tabpages

- Im Smartform muss ein Element mit der uie-tabpages-interfaces eingebunden werden.
 - Soll das erste enthaltene Tab direkt geladen werden, dann muss unter Eigenschaften das Flag „t“ ausgewählt werden
- Nun muss für jedes zu platzierende Tab ein Template erstellt werden
 - Achtung: Bitte vermeiden Sie eine Überschneidung der verwendeten Elementbuchstaben, da sonst beim drag&drop die platzierten Attribute in allen Tabs auftauchen, in denen der selbe Buchstabe existiert
 - Sollte eine Überschneidung von Buchstaben unumgänglich sein, so dürfen die Attribute, die in den betroffenen Buchstaben nur in einem Tab auftauchen, nicht per drag&drop platziert werden
 - Die Attribute müssten dann via Bedienoberflächen - Attribute in der tabellarischen Ansicht den jeweiligen Buchstaben zugeordnet werden. Achten Sie hierbei darauf, dass Sie sich im richtigen Tab befinden. Wenn in der Tabelle ein Elementbuchstabe eingegeben wird, dann erscheint das Attribut exakt nur in dem gewünschten Tab.
- Zur Kombination des Tabs mit dem Element im Smartform, muss aus dem Accordion in Bedienoberfläche - Editor der Eintrag „[neu] ()“ aus dem Untermenü „Inhalte“ auf das Element mit der uie-tabpages-interfaces gezogen werden
 - Hierbei erscheint links eine leere Sidebar, in der das einzufügende Tab konfiguriert wird:
 - Bedienoberfläche: technischer Name für das Tab
 - Beschriftung: im Smartform verwendeter Bezeichner für das Tab
 - übergeordnet: Smartform
 - Template: Das zu verwendende Template für das Tab auswählen
 - Menü / Inhalt: Das Element mit der uie-tabpages-interfaces auswählen
 - UIU: Wenn der Tab auch einfach nur Attribute enthält, die als Prozessvariablen in tim_process abgebildet werden, dann: uiu-selection
 - Wenn das Tab auf eine Tabelle zeigt, die mit dem Prozess in Verbindung steht, dann uiu-details
 - Eigenschaften: E, damit der Tab generell editierbar ist
 - zusätzlich n und a, wenn der Tab auf eine Tabelle zeigt, die mit dem Prozess in Verbindung steht und die darin befindlichen Attribute unmittelbar im Tab angeordnet werden. Sobald eine grid in dem Tab verwendet wird, so sind n und a nicht zu verwenden
 - Icons: keine notwendig. Sie kommen zum Tragen wenn über diesen Weg ein grid zur tabellarischen Ansicht/ Bearbeitung von Datensätzen eingebunden wird
 - Tabelle: tim_process, wenn einfach nur Prozessvariablen dargestellt werden sollen, oder eine andere fachliche Tabelle, wie gewünscht
 - Filter: Einschränkung einer fachlichen Tabelle auf gewisse Einträge. So könnten z.B. aus einer Stammdatentabelle Informationen angezeigt werden, die entsprechend nach einem oder mehreren Attributen gefiltert sind

7.5 Integration einer fachlichen Tabelle als Grid

- Achtung: verwendetes Template grid-Template unter Bedienoberfläche - Elemente - G - auf uie-grid-edit umstellen, sonst tauchen die Icons nicht auf

7.6 Erstellung eines Accordions

- Subelemente

7.7 Erstellung eines Groovy Codes, der beim Laden des Smartforms gezogen werden soll

- Im Modul Bedienoberfläche gibt es für jede Bedienoberfläche links in der Sidebar den Button „showInterface“
 - Dort kann Groovy Code hinterlegt werden, der im Moment des Seitenaufbaus ausgeführt wird
- Mit dem Flag r kann das Neuladen des Smartforms nach Eingabe eines Wertes in ein Feld aktiviert werden
 - Dadurch wird das Smartform neu geladen und showInterface erneut ausgeführt
 - Ebenso werden dynamische Filter in Datentypen oder Tabellen, die auf andere Attribute zugreifen, neu ausgeführt
 - Eingegebene Daten bleiben bei diesem Neuladen erhalten
- Beispiele:
 - Auslesen der Benutzersprache und setzen in das Attribut „sprache“:

```
def locale = FRAME.getProperty("locale_id").toUpperCase();
if (locale != null && locale.length() > 1 && locale.substring(0,2) !=
"DE") {
    DATA_ROW.setValue("mm_sprache", "EN");
}
```

- Setzen des UI-Flags für ein Attribut:

```
ICbaRow attribut_xxxx = FRAME.searchRow("dd_intattr", "interf_recid="
+ CONFIG_ROW.getRecid() + "&attr_id=TECHNISCHER_ATTRIBUTNAME");
attribut_xxxx.setValue("ui_flag", "D");
```

- Lesen eines Wertes für ein Attribut:

```
String wert_des_attributs_im_moment_des_seitenaufbaus =
DATA_ROW.getValue("TECHNISCHER_ATTRIBUTNAME");
```

- Setzen eines Wertes für ein Attribut:

```
DATA_ROW.setValue("TECHNISCHER_ATTRIBUTNAME",
"neuer_Wert_für_das_Attribut");
```

7.8 Konfiguration von statusabhängigen Attributeigenschaften

- Zurücksetzen einzelner Attributvarianten auf Standard
- „Für alle Status auf Standard zurücksetzen“ bzw. „Auf Standard zurücksetzen“ Schaltflächen
- Statusübersicht. Kopieren, Löschen, * **Schritt, der entfallen ist** *

8. Installation

- siehe [Installation](#)
-

9. Altes CBA Wiki

- [16.0 Übersicht](#)
- [16.1 Einführung](#)
- [16.2 Tabellen](#)
- [16.3 Datentypen](#)
- [16.4 Templates](#)
- [16.5 Bedienoberflächen](#)
- [16.6 Meldungen](#)
- [16.7 TIM-Anbindung](#)

From:

<https://wiki.tim-solutions.de/> - **TIM Wiki** / [NEW TIM 6 Documentation](#)

Permanent link:

<https://wiki.tim-solutions.de/doku.php?id=software:tss:start&rev=1477490133>

Last update: **2021/07/01 10:00**

