



# Apex Universal Theme – neue Hürden mit dem Responsive Layout

Tobias Arnhold, IT Consulting und Richard Rieb, DB System

Seit April letzten Jahres ist Apex 5 verfügbar. Vor allem das Universal Theme und das Responsive Design lösten große Begeisterung aus. Die ersten Entwicklungen haben jedoch gezeigt, dass das resultierende Layout doch nicht immer optimal ist. Dieser Artikel zeigt, wie man mit einfachsten Lösungen die durch das Responsive Design hervorgerufenen Layout-Probleme im Universal Theme beheben kann.

Das Universal Theme (UT) ist das neue Nutzer-Interface für Apex-Anwendungen. Es wurde erstmalig mit Apex 5 ausgeliefert. Eines der Kernelemente im UT ist das Responsive Design und dessen Responsive Layout (RL). Die darin enthaltene wichtigste Änderung im Design ist die Umstellung der bis Apex 4.2 verwendeten tabellenbasierten Darstellung auf einen festen Zwölf-Spalten-DIV-Element-basierten Aufbau der Seite. Diese und weitere technische Neuerungen sorgen dafür, dass mit UT erstellte Apex-Anwendungen sofort auf jedem Bildschirmformat optimal

dargestellt werden. Diese Anpassungsfähigkeit bietet eine sehr leichte Integration von Tablet- und Handy-Nutzern.

Desktop-Nutzer haben jedoch mit einigen Hürden zu kämpfen. So werden Interactive Reports und Forms-Masken verzerrt dargestellt. Zusätzlich erscheinen einzelne Seiten-Elemente wie Breadcrumbs übermäßig platzintensiv. Nachfolgend ist anhand einzelner Fallbeispiele erklärt, wie diese Probleme umgangen werden und sich in Reports- und Forms-Masken die gewohnte Desktop-Darstellung wiederherstellen lässt.

## **Interactive Report**

Der Interactive Report ist eines der mächtigsten Werkzeuge von Apex und Kern der meisten Anwendungen. Er ermöglicht dem Endanwender, eigenständig individuelle Berichte zu definieren, und bietet eine Vielzahl von Funktionalitäten. Im Responsive Layout wird die Breite des Interactive Reports mindestens auf die Breite seines Containers ausgedehnt. Dabei werden alle angezeigten Spalten gleichmäßig auf diese Breite verteilt. Dies führt jedoch dazu, dass viele Spalten we-

Abbildung 1: Breite von Icon-Spalten im Interactive Report

sentlich breiter als ihr Inhalt angezeigt werden.

Insbesondere Spalten mit einem Icon sind davon betroffen, da diese nur eine vordefinierte, verhältnismäßig kleine Größe benötigen (siehe Abbildung 1).

Das Problem lässt sich mit zwei kleinen CSS-Erweiterungen lösen. Im Falle des „Bearbeiten“-Icons muss ein CSS-Snippet als Datei oder auf der betroffenen Seite „Page>CSS>Inline“ hinzugefügt werden (siehe Listing 1).

Anschließend muss dem Interactive Report noch die entsprechende CSS-Klasse durch „IR Region>Appearance>CSS Classes“: minimize-first-table-col mitgegeben werden. Bei allen anderen Spalten ist dem Report und der entsprechenden Spalte eine

```
.minimize-first-table-col td:first-child{
width: 1%;
overflow: visible;
}
```

Listing 1

```
"IR Region>Advanced>Static ID": reportid
"IR Region>Column>Advanced>Static ID": colid
```

Listing 2

```
#reportid th#colid, #reportid td[headers=colid]{
width: 1%;
overflow: visible;
}
```

Listing 3

Abbildung 2: Forms-Maske, visualisiert auf einem 24-Zoll-Bildschirm

ID zu erteilen (siehe Listing 2). Die Formatierung erfolgt durch ein CSS (siehe Listing 3).

## Forms-Masken

Forms-Masken dienen in erster Linie der Bearbeitung von Datensätzen. Damit sind sie Bestandteil nahezu jeder Anwendung. Das UT hat den grundsätzlichen Aufbau einer Apex-Forms-Maske nicht

verändert. Durch das RL wurde die Visualisierung der Seitenelemente und Regionen jedoch insbesondere bei der Nutzung von großen Bildschirmen negativ beeinflusst. So sind auch Forms-Masken durch die automatisierte Anpassung an die Bildschirmgröße verzerrt. Dies resultiert in Leerräumen innerhalb der Seite.

Der Grund für dieses Verhalten ist, dass das UT Objekte, wie Seiten und Re-

gionen, in zwölf Spalten unterteilt. Standardmäßig nehmen diese zwölf Spalten zusammen die gesamte Bildschirmbreite ein, wobei einzelne Objekte gleichmäßig auf die Spalten aufgeteilt sind. Bei zwei Regionen nimmt damit jede Region sechs Spalten und fünfzig Prozent der Bildschirmbreite ein (siehe Abbildung 2).

Um dieses Verhalten zu umgehen, muss das responsive Verhalten reglementiert werden. Eine Lösung besteht darin, den Regionen mitzugeben, in welcher Spalte sie beginnen sollen. Dadurch

```
@media (min-width:550px) {
  .col-600 {
    min-width: 600px !important;
    max-width: 600px !important;
  }
}
```

Listing 4

```
.t-Breadcrumb-item{
  font-size: 1.2rem !important;
}
.t-Breadcrumb-item.is-active{
  font-size: 2rem !important;
  line-height: 1.2em !important;
}
```

Listing 5

```
.t-BreadcrumbRegion-body {
  padding-top: 10px;
  padding-bottom: 10px;
}
```

Listing 6



Abbildung 3: Breadcrumb im UT

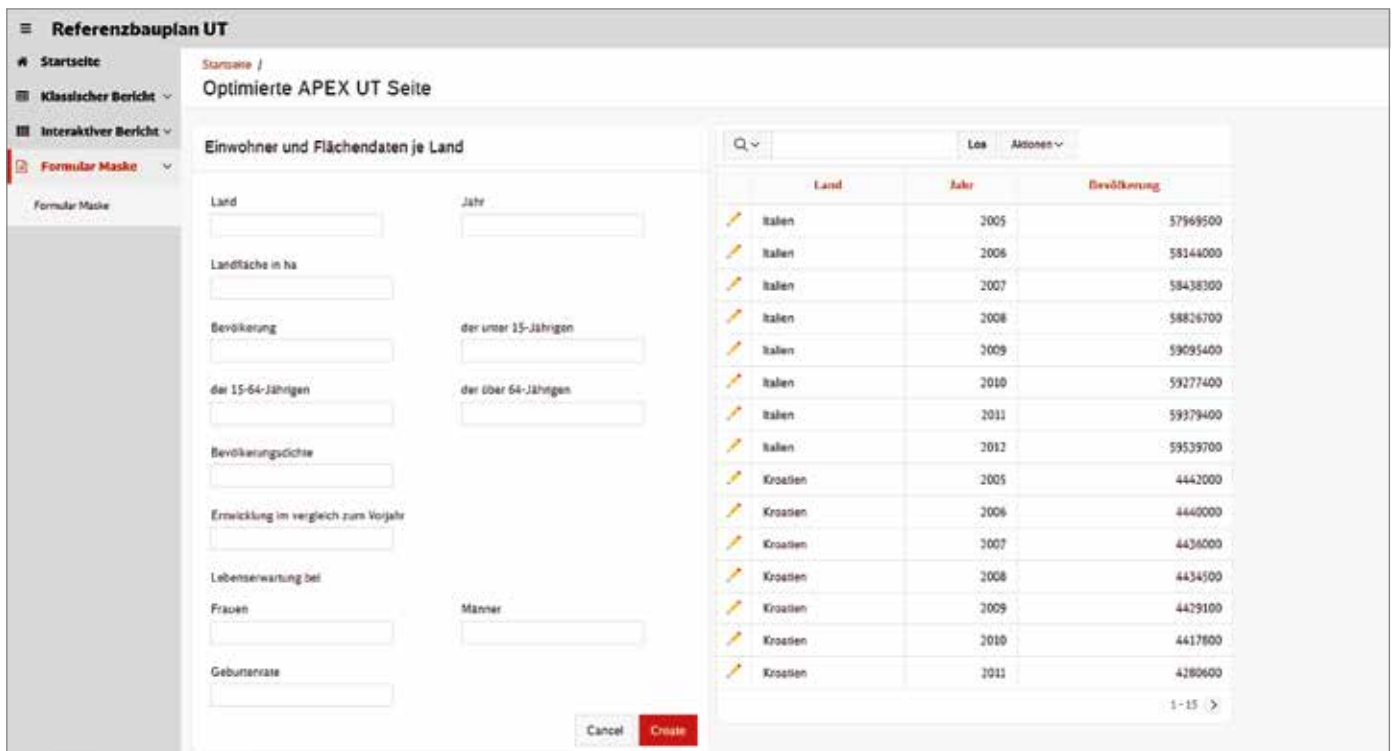


Abbildung 4: Ergebnis mit allen Design-Anpassungen

werden die Breitenverhältnisse entsprechend den Definitionen berechnet. Bei den links ausgerichteten Regionen ist dies nicht notwendig, da diese automatisiert in der ersten Spalte beginnen. Um beispielsweise zu definieren, dass die erste Region lediglich fünfundzwanzig Prozent der Breite einnimmt, ist die zweite Region mit „Region Source>Grid>Column“:4“ auf Spalte 4 zu definieren. Dies führt jedoch auch bei kleineren Bildschirmgrößen zu einer Verkürzung der Region, die in vielen Fällen nicht mehr optimal ist. Daher empfiehlt es sich, die Breite der Regionen über CSS-MEDIA-Queries zu definieren (siehe Listing 4).

Im Anschluss ist beiden Regionen noch mit „Region Source>Appearance>CSS Classes“: col-600“ die entsprechende CSS-Klasse zuzuweisen. Dadurch werden die Regionen untereinander angezeigt, sobald eine kleine Bildschirmgröße dafür sorgt, dass die Mindestbreite im RL unterschritten ist.

## Breadcrumbs

Breadcrumbs zeigen Nutzern, in welcher Verzweigung der Anwendung sie sich gerade befinden, und ermöglichen es, direkt zu einer übergeordneten Seite zu springen. Sie werden üblicherweise oben auf der Seite in einer Textzeile angezeigt. Das UT nutzt Breadcrumbs auch als Seitenüberschriften. Dabei wird ihnen übermäßig viel Platz auf der Seite zugewiesen, der durch andere Bereiche effektiver genutzt werden könnte (siehe Abbildung 3).

Dies liegt daran, dass Breadcrumbs eine Schriftgröße von 2,4 beziehungsweise 1,4 rem im Fall des Navigationspfads haben und einen Innenabstand von 24 px definieren. Um Breadcrumbs optimaler darzustellen, muss zunächst die Schriftgröße mittels CSS angepasst werden (siehe Listing 5). Ein CSS verringert den Innenabstand (siehe Listing 6). Abbildung 4 zeigt die beschriebenen Beispiele noch einmal im Zusammenspiel.



Tobias Arnhold  
tobias-arnhold@hotmail.de



Richard Rieb  
richard.rieb@deutschebahn.com

# DOAG 2016 BI

8. - 9. Juni 2016 in Bonn

