



# Süddeutsche Zeitung als digitale Ausgabe

Technische Spezifikationen für SZ-App- Anzeigen, Stand: 06.10.2015

## Inhaltsübersicht

1. Einleitung und Anlieferungstermine	S. 1
2. Anzeigen	
2.1. Bildanzeigen "statisch"	S. 2
2.2. HTML-Anzeigen	S. 4
3. Templates und Beispiele	S. 6

## 1. Einleitung

Die Apps der Süddeutschen Zeitung (SZ) platzieren Anzeigen auf Tablets (iPad, Android, Windows 8), Smartphones (iPhone, Android) und Desktop Computern (via WebApp). Die Displays der verwendeten Geräte unterscheiden sich dabei deutlich in Größe, Auflösung, und dem Verhältnis von Breite zu Höhe. Daher müssen sich die Anzeigen dem jeweiligen Bildschirmformat anpassen.

Um eine Anzeige in den Apps der Süddeutschen Zeitung zu schalten, können Sie entweder vollflächige Bilder liefern („Bildanzeigen“, Seite 2) oder eine komplette HTML-Anzeige zur Verfügung stellen (Seite 5). Der Speicherplatz pro Anzeige ist auf 2 MB beschränkt. Dieses Dokument beschreibt die technischen Spezifikationen für beide Arten von Anzeigen.

## Anlieferungstermin

Werbemittel	Werkstage vor ET	max. Datenmenge
statische Bild-Werbemittel	5	400 KB
interaktive Bild-Werbemittel	10	2 MB
HTML-Werbemittel	15	2 MB

Werbemittel-Lieferung erfolgt an [adsales@sueddeutsche.de](mailto:adsales@sueddeutsche.de)

## 2. Bildanzeigen

Diese verwenden vollflächige Bilder die alle Elemente wie Fotos, Grafiken und Texte enthalten. Eine *statische Bildanzeige* zeigt ein vollflächiges Bild mit einem optionalen Störer (Popup) Element an.

### 2.1. Bildanzeigen „statisch“

Statische Bildanzeigen verwenden jeweils ein Bild im Hoch- und Querformat, das in maximaler Größe zentriert angezeigt wird. Der übrige Platz wird durch einfarbige Balken gefüllt. Wenn nur ein Bild (hoch oder quer) verwendet wird, erscheint es im jeweils anderen Format mit einer eingeblendeten Aufforderung das Gerät zu drehen.

#### Bildformat

Es empfiehlt sich das Bildmaterial an das iPad Format anzupassen:

Bildauflösung (BxH)	Optimiert für iPad Retina
Hochformat: hoch.jpg	1536 x 2008 Pixel
Querformat: quer.jpg	2048 x 1496 Pixel

#### Hinweise

1. Die Bildfläche ist voll nutzbar und nicht durch Schaltflächen oder Navigationsleisten überdeckt.
2. Grafiken werden nicht skaliert, um diese gerätespezifisch einzupassen, werden an den Rändern weiße Balken platziert.

#### Verlinkung

Auf Wunsch kann die Anzeige mit einem vollflächigen Link versehen werden. Das Linkziel kann in Abhängigkeit vom Betriebssystem (iOS, Android, Win8) gesetzt werden, um z.B. direkt zu Apps in den unterschiedlichen App-Stores zu linken.

#### Störer

Ein Störer oder PopUp ist ein optionales grafisches Element, das nach einer definierten Zeitspanne erscheint um die Aufmerksamkeit des Benutzers zu erhöhen. Die Popup-Bilder haben die gleiche Größe wie die Hintergrundbilder, und können entweder mit transparentem Hintergrund (.png Datei) oder mit komplett gefüllter Bildfläche (.jpg Datei) angelegt werden.

## Zu liefernde Daten und Angaben

1. hoch.jpg und quer.jpg, Farbprofil sRGB, zusammen max. 800 KB
2. Balkenfarbe als HTML-Farbcode, z.B. #000000 (schwarz) oder #FFFFFF (weiß). Standardmäßig sind die Balken auf mobilen Geräten schwarz und in der WebApp weiß.
3. bei Verlinkung: externe URL für einheitliches Ziel oder getrennt für iOS, Android, Win8 und WebApp
4. für Störer: hoch\_popup.png und quer\_popup.png oder hoch\_popup.jpg und quer\_popup.jpg

## 2.2. HTML-Anzeigen

Das Werbemittel besteht aus einem kompletten HTML5-Projekt inklusive HTML, CSS, Javascript und Bildern. Alle Bestandteile müssen in einer Zip-Datei (max. 2 MB, nach Rücksprache ggf. auch mehr) übermittelt werden. Im Hauptverzeichnis muss die Datei „index.html“ liegen. Diese kann alle anderen Inhalte aus beliebigen Unterverzeichnissen nachladen. Das Werbemittel muss auf allen benutzten Plattformen (iOS, Android, Win8) funktionieren.

***Alle Anzeigen müssen auf einer der am Ende dieses Dokuments zur Verfügung gestellten Templates basieren, da diese Funktionen enthalten um die Anzeigeneinblendungen zu zählen.***

### Callbacks

Die Templates stellen mehrere Javascript callbacks zur Verfügung um die Anzeige zu steuern. Die Funktion **setupAd()** wird aufgerufen wenn alle Dateien durch den Preloader gefunden wurden und der DOM komplett geladen wurde. Die Funktion **startAd()** wird aufgerufen sobald der Benutzer die Anzeige auf den Bildschirm swiped. Diese wird beispielsweise benutzt um Animationen zu starten oder Event Handler zu registrieren. Die Funktion **stopAd()** wird aufgerufen sobald der Benutzer die Anzeige aus dem Sichtbereich herauswiped. Diese Callback-Funktion sollte beispielsweise alle laufenden Videos stoppen. Diese Funktion sollte auch die Anzeige für wiederholte Einblendungen in ihren Anfangszustand zurücksetzen.

Die Funktion `updateAdOrientation(orientation)` wird mit dem Argument `orientation == 'landscape'` oder `orientation == 'portrait'` durch das `resize` Event aufgerufen. Zu beachten ist dass `window.orientation` im Kontext dieser Funktion teilweise falsche Werte zurückliefert. Um diese Eigenschaft zu benutzen, muss ein eigener Eventhandler für `orientationchange` Events eingerichtet werden. Achtung: Der Winkel in `window.orientation` für eine gegebene Ausrichtung des Geräts stimmt bei iOS, Android und Windows nicht überein.

### Anpassung an verschiedene Bildschirmgrößen

Eine HTML-Anzeige kann wie eine liquide oder responsive Webseite angelegt werden und sich an die verschiedenen Bildschirmgrößen von Smartphones und Tablets anpassen. Viele Kunden bevorzugen es aber, die Anzeige mit einer festen Größe optimiert für iPad Bildschirme anzulegen, und benutzen zur Bildschirmanpassung eine automatische Skalierung der Anzeige mit einfarbigen Balken an den Seiten oder oben und unten. Am Ende dieses Dokuments steht ein entsprechendes Beispiel zur Verfügung.

### Orientation Change

Die HTML(5)-Werbung muss selbst auf eine Änderung der Ausrichtung des Gerätes reagieren, und dafür CSS media queries, den beschriebenen `updateAdOrientation` Callback oder einen eigenen Handler für `window.orientation` benutzen. Um schnelle und flüssige Orientation changes zu erzielen empfiehlt es sich soweit wie möglich auf CSS media queries zurück zu greifen. Beispiel:

CSS Media Queries:

```
@media screen and (orientation:portrait) { ... }
```

```
@media screen and (orientation:landscape) { ... }
```

## Videos

Unter iOS und Windows können Videos mit dem <video> Tag eingebunden und per Javascript gesteuert werden. Auf Android-Geräten funktioniert HTML5 Video nicht zuverlässig. Daher muss der native Vollbild-Videooplayer per Javascript aufgerufen werden, mittels *SZVideo.openVideoPlayer(URL)*;

Unter Android müssen Videos aus dem Internet gestreamt werden, da lokale Videodateien nicht eingebunden werden können. Für die meisten Videos erfordert die Speicherplatzbeschränkung auf 2 MB pro Anzeige ohnehin das Streaming von einer externen URL.

## Plattformabhängiges Verhalten

Alle in den Apps der SZ geschalteten Anzeigen werden auf mobilen Geräten mit iOS, Android oder Windows sowie Desktopcomputern via WebApp platziert und angezeigt. Per Javascript kann der User Agent String ausgewertet werden um die verwendete Plattform zu bestimmen, z.B. um verschiedene Bilder anzuzeigen oder bestimmte Hardwarefunktionen zu nutzen. Die SZ Templates benutzen so eine Abfrage und speichern die ermittelte Plattform in der Variable 'platform'. Die verwendeten Werte sind "android", "ios", "windows" und „webapp“.

## Browserkompatibilität

Die HTML-Werbung muss in allen gängigen Browsern und WebViews auf iOS, Android und Windows 8 funktionieren. Bei Benutzung von Funktionen wie Touch- oder Bewegungsevents die nicht auf allen Zielplattformen verfügbar sind, müssen geeignete Fallbacks eingebaut werden.

Folgende Probleme und Eigenheiten sind bekannt:

OS	Problem	Lösung
Android	CSS-Modifikationen im Event-Handler onResize werden erst beim nächsten Event ausgeführt	per setTimeout(..., 10) aus dem Event-Handler heraustrennen

OS	Problem	Lösung
Android + Windows	Erkennung von horizontalen Swipe-Gesten stört App-Navigation	Auf click Events ausweichen
Android	lokale gespeicherte Audio-Dateien spielen nicht	aus dem Internet streamen
Android	HTML5 Video tags funktionieren nicht zuverlässig	Die App ersetzt die Tags und fügt einen Play-Button ein. Dieser startet alle Videos in einem Fullscreen Player. Die Video Template enthält den dafür benötigten Code.

#### 4. Templates

Im Download-Link folgen drei Programmiervorlagen für HTML-Ads. Diese demonstrieren alle Funktionen, die für die Entwicklung von Anzeigen für die SZ wichtig und besonders sind.

1. "Empty" Vorlage für liquide Ads
2. „Resize“ Vorlage für das automatische skalieren und zentrieren für Anzeigen die auf iPad-Größe zugeschnitten sind.
3. „Video“ demonstriert wie Videos eingebunden werden müssen, damit sie auch in der Android-App funktionieren.

[http://sz-media.sueddeutsche.de/de/online/files/sz\\_html\\_ad\\_templates.zip](http://sz-media.sueddeutsche.de/de/online/files/sz_html_ad_templates.zip)