



Liebe Kolleginnen und Kollegen,

dieses Dokument soll Ihnen dabei helfen, die dpa-ElectionsLive-Applikationen nach Ihren Wünschen zu modifizieren. Wir erklären Ihnen anhand der Applikation zur Bundestagswahl (**bw\_de**) in kurzen Schritten, wie Sie zum Beispiel Schriften oder Farben ändern können. Zudem bekommen Sie einen Überblick über alle Einzel-Module, die Sie auf Ihrer Seite nutzen und die Sie ebenfalls individuell über CSS oder Variablen verändern können.

Sollten Sie darüber hinaus noch Fragen haben, wenden Sie sich gern über folgenden Verteiler an uns:  
[electionslive@dpa-info.com](mailto:electionslive@dpa-info.com)

## Dokumentenversionen

Thema	Bemerkung	Datum
Initialisierung		01.03.2017
Update	CSS-Anpassung, Einbindung in DOM	29.03.2017
Update	Widget-Optionen	24.05.2017
Update	Pym und RequireJS	10.07.2017
Update	Lieferumfang	18.07.2017
Update	Widget-Optionen	24.07.2017
Update	Widget-Optionen	05.10.2017



## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeine Informationen.....</b>	<b>3</b>
<b>2</b>	<b>Anlegen eines individuellen Layoutverzeichnisses.....</b>	<b>4</b>
<b>3</b>	<b>Integration der Applikation im iframe .....</b>	<b>5</b>
	3.1 Serverinterne Kommunikation .....	5
	3.2 Serverübergreifende Kommunikation .....	6
	a) Einbindung über Auto-Initialisierung .....	6
	b) Einbindung über Skriptaufruf:.....	7
	c) PYM.js und RequireJS:.....	7
<b>4</b>	<b>Integration der Applikation direkt in Ihrem CMS .....</b>	<b>8</b>
	4.1 Skripte manuell laden .....	8
	4.2. Skripte über RequireJS laden.....	9
<b>5</b>	<b>Widgets .....</b>	<b>10</b>
	5.1. Einzel-Widgets .....	10
	5.2. Widget-Kombinationen .....	10
<b>6</b>	<b>Individuelle Einstellungen .....</b>	<b>11</b>
	6.1 Widget-Optionen .....	11
	6.2 Konfiguration des eigenen Layouts .....	15
	6.2.1 Schrift .....	15
	6.2.2 Farben .....	15
	6.2.3 Weitere Änderungen .....	16
<b>7</b>	<b>Deeplinks .....</b>	<b>17</b>
<b>8</b>	<b>Social Media.....</b>	<b>18</b>
<b>9</b>	<b>IVW-Zählung .....</b>	<b>19</b>
<b>10</b>	<b>Google Analytics .....</b>	<b>20</b>



## 1 Allgemeine Informationen

### IDs und Termine der einzelnen Wahlen

Im Jahr 2017 stehen eine Vielzahl von politischen Wahlen an, die wir im Vorfeld mit Umfragen und am jeweiligen Wahlabend mit Ergebnissen begleiten.

Der Übersichtlichkeit halber beschreiben wir die Einbindung und die Anpassungsmöglichkeiten der Wahl-Applikationen anhand der Bundestagswahl (**bw\_de**).

Hinweis: Nicht jede der im Folgenden beschriebenen Möglichkeiten ist auf alle Wahlen übertragbar.

Veranstaltung	ID	Termin
Landtagswahl Saarland	lw_sl	26.03.2017
Landtagswahl Schleswig-Holstein	lw_sh	07.05.2017
Landtagswahl Nordrhein-Westfalen	lw_nw	14.05.2017
Parlamentswahl Niederlande	pw_nl	15.03.2017
Präsidentenwahlen Frankreich	pw_fr	23.04.2017/07.05.2017
Parlamentswahl Großbritannien	pw_gb	08.06.2017
Bundestagswahl	bw_de	24.09.2017
Landtagswahl Niedersachsen	lw_ni	15.10.2017
Nationalratswahl Österreich	pw_at	15.10.2017



## 2 Anlegen eines individuellen Layoutverzeichnisses

Wir übertragen das Verzeichnis **bw\_de** auf Ihren Server. Darin enthalten sind weitere Unterverzeichnisse für verschiedene Datentypen wie beispielsweise Ergebnisdaten, Bilder und Texte (.xml, .json, .js, .html, .jpg, .png, .svg). Bitte achten Sie darauf, dass alle entsprechenden MIME-Types auf Ihrem Server hinterlegt sind.

Im Verzeichnis **bw\_de** finden Sie auf Ihrem Server ein Anpassungsverzeichnis als ZIP-Datei **my\_bw\_de.zip**. Bevor Sie Ihre Layoutwünsche umsetzen, entpacken Sie bitte die mitgelieferte zip-Datei **my\_bw\_de**. Diese enthält bereits die notwendige Ordnerstruktur. Bitte legen Sie dieses Verzeichnis auf Ihrem Webserver an. Es kann umbenannt werden, muss allerdings **parallel** zum Standardverzeichnis **bw\_de** liegen. Anschließend sollten Sie die folgende Ordnerstruktur auf Ihrem Webserver vorliegen haben:

	<p><b>Verlinken</b> Sie auf die <b>sitemap.html</b> Datei in:  <b>my_bw_de → html</b></p> <p>Hier finden Sie die Liste aller Widgets, die Ihnen zur Verfügung stehen und die Sie im Layout anpassen können.</p> <p>Die Applikation lädt weiterhin alle Dateien, die in <b>bw_de</b> enthalten sind und dort aktualisiert werden.</p>
--	--

Abbildung 1: Verzeichnisstruktur des Anpassungsverzeichnisses

Im Folgenden erläutern wir die Implementierung des Dienstes **bw\_de** (Bundestagswahl)

Verzeichnisstruktur auf Webserver	Unterverzeichnis	Inhalte
my_bw_de		individuelles Layoutverzeichnis
(Name frei wählbar)	feed	style.css, config.xml
	html	Index-Dateien und sitemap.html
	js	click.js
	img	Leer, hier können Sie etwaige angepasste Bilder/Icons ablegen
bw_de		<b>Standardverzeichnis</b>
(Name <b>nicht</b> frei wählbar)	assets	jQuery u. versch. Plugins, css-Dateien, Label-Dateien, allgemeine Grafikdateien
	config	Konfigurationsdateien (JSON)
	feed	Alle Feeds, die in der Applikation dargestellt werden. (XML, JSON)
	img	Bilder (JPG, PNG, SVG)
	src	Source-Dateien der Grafik (JS, HTML)

**Hinweis:** Alle Anpassungen und Veränderungen Ihrerseits nehmen Sie bitte **nur** in Ihrem individuellen Layoutverzeichnis **my\_bw\_de** vor, denn dieses Parallelverzeichnis wird von unserem Promoter auch im Falle einer notwendigen Synchronisation niemals überschrieben. So ist gewährleistet, dass alle von Ihnen vorgenommenen Anpassungen nicht verändert werden.



### 3 Integration der Applikation im iframe

Wenn Sie die oben aufgezeigte Verzeichnisstruktur auf Ihrem Webserver angelegt haben, kann die Applikation über die Datei *my\_bw\_de/html/index.html* gestartet werden.

Welches Widget angezeigt werden soll, wird über die Variable „standalone“ gesteuert (Näheres dazu finden Sie in **Kapitel 5**). In den folgenden Beispielen wird das Widget „Prozentuale Stimmenverteilung“ eingebunden.

#### 3.1 Serverinterne Kommunikation

Wenn sich die Applikation auf dem gleichen Server befindet, wie die einbindende HTML-Seite, gehen Sie bitte nach folgendem Muster vor:

```
<iframe id='dpa_frame_bw_de'
src='[Ihr_Pfad]my_bw_de/html/index.html?standalone=bars&iframeId=dpa_frame_bw_de&sandbox=false'
height='850px' width='100%'...></iframe>
```

Setzen Sie für den iframe einen beliebigen Namen in **id**.

Die iframe-Id muss zusammen mit der Erlaubnis auf das übergeordnete Fenster zuzugreifen („sandbox=false“) an die Grafik übergeben werden. Das können Sie in „src“ über URL-Variablen tun (siehe Beispiel).

Sie können die Variablen auch direkt in der Datei, die Sie in Ihrem iframe einbinden, eintragen. Da die Variablen dann über data-Attribute an die Applikation übergeben werden, ist in diesem Fall eine andere Schreibweise zu beachten, aus „Camel-Case“ wird „Kebab-Case“, „iframeId“ wird zu „data-iframe-id“ (siehe auch **Kapitel 6**):

```
<body>
  <div data-app="dpa" data-competition-id="bw_de" data-standalone="bars" data-iframe-id="dpa_frame_bw_de" data-
  sandbox="false" data-path="../../bw_de/"></div>
</body>
```

So passt sich der iframe automatisch in der Höhe an, wenn sich die Höhe des Inhalts verändert.



### 3.2 Serverübergreifende Kommunikation

Wenn ihre Applikation auf einem anderen Server (im weiteren „Tool-Server“) liegt, als die HTML-Seite, in der die Applikation per iframe aufgerufen wird (im weiteren „CMS-Server), sieht die Einbindung etwas anders aus.

**Wichtig:** Arbeiten Sie bei unterschiedlichen Servern nicht wie unter 3.1. beschrieben mit der Variablen „iframeld“ (bzw. „data-iframe-id“). Die Applikation würde versuchen, die übergeordnete Seite anzusprechen und das Sicherheitssystem des Browser bricht in diesem Fall das Laden der Applikation ab.

Hintergrund ist, dass die Applikation dann nicht mehr auf das übergeordnete Fenster zugreifen darf („Same-Origin-Policy“). Um dieses Problem zu lösen, gehen wir den Weg über ein externes Skript, das die Kommunikation zwischen den beiden Servern erleichtert.

Sie finden das Skript **pym.min.js** in unserem Standardverzeichnis in **assets/js/pym.min.js**. Sie können die Datei auch an einem zentralen Ort ablegen, dann muss bei der Einbindung von mehreren Wettbewerben nicht jedes Mal der Pfad geändert werden.

Auf dem Tool-Server müssen Sie nichts weiter anpassen.

Für die Seite, auf der der iframe erstellt werden soll, gibt es zwei Möglichkeiten der Einbindung, diese werden im Folgenden erläutert. Bei beiden wird der iframe automatisch generiert.

**Wichtig:** Die Breite des generierten iframes beträgt immer 100%, Sie bestimmen also die Breite der Grafik über den Container, in dem der iframe erstellt wird. Die Höhe des iframes richtet sich nach der Höhe des Inhalts.

#### a) Einbindung über Auto-Initialisierung

Wir empfehlen diesen Weg der Einbindung, er ist der einfachste und am wenigsten fehleranfällig.

Sie definieren ein DIV oder einen <p>-Tag, in dem der iframe generiert werden soll. Sie können auch gleich mehrere Container definieren, entscheidend ist nur, dass das Skript **pym.min.js** nach dem letzten Container geladen wird (z.B. direkt über dem schließenden </body>-Tag).

Den Pfad zu der index.html der Applikation übergeben Sie dann über das Attribut „**data-pym-src**“. Dem DIV, bzw. dem <p>-Tag, eine Id zu vergeben ist bei dieser Art der Einbindung nicht erforderlich.

```
<div data-pym-src="http://kunde.toolserver/my_bw_de/html/index.html?standalone=bars"></div>
<script type='text/javascript' src='http://tool.musterzeitung.de/.../bw_de/assets/js/pym.min.js'></script>
```

**TIPP:** Dieses Vorgehen bietet sich z.B. auch für die Einbindung von Widgets im Artikel an.



### b) Einbindung über Skriptaufruf:

Das Skript **pym.min.js** muss auf der Seite vorliegen. Es genügt, das Skript einmal am Anfang zu laden (z.B. im <head>).

Definieren Sie das DIV, in dem der iframe generiert werden soll, und geben Sie ihm eine Id. Unterhalb des DIVs wird dann die Funktion aufgerufen und die DIV-Id und der Pfad zu der index.html der Applikation übergeben:

```
<script type='text/javascript' src='http://tool.musterzeitung.de/.../bw_de/assets/js/pym.min.js'></script>
<div id="dpa-application"></div>
<script type='text/javascript'>
  var pymApp = new pym.Parent("dpa-application",
  "http://kunde.toolserver/my_bw_de/html/index.html?standalone=bars", {});
</script>
```

### c) PYM.js und RequireJS:

**Wichtig:** Sollten Sie auf Ihrer Site mit **RequireJS** arbeiten, achten Sie darauf, auch **pym.min.js über require zu laden**. PYM.js arbeitet intern ebenfalls mit require und muss daher über Ihre eigene require-Instanz geladen werden. Der Aufruf sollte dann so aussehen:

```
<script type='text/javascript'>require(['http://tool.musterzeitung.de/.../bw_de/assets/js/pym.min.js'])</script>
```

Der Aufruf kann variieren, je nachdem, wie Sie Ihre require-Instanz benannt haben.



## 4 Integration der Applikation direkt in Ihrem CMS

Wenn Sie iframes vermeiden möchten, können Sie die Widgets auch direkt in Ihre HTML-Struktur integrieren. Das hat u.a. den Vorteil, dass Sie auf Ihre eventuell vorhandene jQuery-Version zurückgreifen können. Unsere ElectionsLive-Applikationen laufen mit **jQuery 3.1.1.**, sind aber **abwärts kompatibel bis jQuery 1.7.**

Ein Nachteil dieser Methode kann sein, dass sehr allgemeine Style-Vorlagen von Ihrer Seite unerwartete Effekte in unseren Applikationen hervorrufen.

### 4.1 Skripte manuell laden

Die Applikationen benötigen drei Skripte, die am besten bereits im <head> Ihrer HTML-Seite geladen werden. Wenn jQuery bereits vorhanden ist, können Sie den Skriptaufruf natürlich weglassen:

```
<script type="text/javascript" src="[ihr Pfad]/bw_de/assets/js/jquery-3.1.1.min.js"></script>
<script type="text/javascript" src="[ihr Pfad]/bw_de/src/Main.js"></script>
<script type="text/javascript" src="[ihr Pfad]/my_bw_de/js/click.js"></script>
```

**Wichtig:** Die Pfade zur jquery-3.1.1.min.js und zur Main.js müssen jeweils auf das Lieferverzeichnis „bw\_de“ verweisen, während die click.js aus Ihrem Anpassungsverzeichnis stammt.

Wenn Sie Widgets von unterschiedlichen Wahlen auf derselben Seite einbinden, müssen jquery-3.1.1.min.js und Main.js trotzdem nur einmal geladen werden. Auf welches Lieferverzeichnis Sie in dem Fall verweisen (bw\_de oder z.B. lw\_sh), ist egal.

Fügen Sie das DIV für das Widget an der Stelle ein, an der Sie es anzeigen möchten. Passen Sie auch hier den Pfad zum Lieferverzeichnis „bw\_de“ an.

**Achtung:** Geben Sie zusätzlich noch den Pfad zu Ihrem Anpassungsverzeichnis an („data-path-config“), da die Applikation von dort die style.css und die config.xml lädt:

```
<div data-app="dpa" data-competition-id="bw_de" data-standalone="bars" data-path="[ihr Pfad]/bw_de/" data-path-
config="[ihr Pfad]/my_bw_de/feed/"></div>
```





## 4.2. Skripte über RequireJS laden

„**RequireJS** ist eine Open-Source-Implementierung für asynchrone Moduldefinition in JavaScript. Es dient dazu JavaScript-Dateien nicht beim Öffnen einer Webseite, sondern erst, wenn diese benötigt werden, zu laden.“ (Quelle: Wikipedia)

**Wenn Sie RequireJS auf Ihrer Website nicht verwenden, können Sie dieses Kapitel überspringen.**

Wenn doch:

Wir verwenden für die Widgets „Koalitionsrechner“ und „Sitzverteilung“ das Plugin *Chart.js*. Dieses kommt als Bundle und lädt intern die benötigten Skripte über RequireJS.

Sollten Sie ebenfalls damit arbeiten, ist es zwingend notwendig, dass das Chart-Bundle von Ihnen mit require vorgeladen wird.

Damit das Plugin dadurch nicht zweimal geladen wird, übergeben Sie außerdem noch „data-use-require=true“ an die Applikation.

Beispiel der Einbindung des Koalitionsrechners:

```
<div data-app="dpa" data-competition-id="bw_de" data-standalone="coalition" data-path="[ihr Pfad]/bw_de/" data-path-config="[ihr Pfad]/my_bw_de/feed/" data-use-require="true"></div>
```

```
<script>
require(["[ihr Pfad]/bw_de/assets/js/jquery-3.1.1.min.js", "[ihr Pfad]/bw_de/assets/js/Chart.bundle.min.js", "[ihr Pfad]/bw_de/src/Main.js", "[ihr Pfad]/my_bw_de/js/click.js"]);
</script>
```

Sollte Ihnen sonst noch etwas im Zusammenhang mit unseren Applikationen und RequireJS auffallen, melden Sie sich gern bei uns.



## 5 Widgets

Über die Datei **my\_bw\_de/html/sitemap.html** können Sie sich eine Übersicht über die Widgets verschaffen, die Ihnen für die Wahlen zur Verfügung stehen. Nicht alle Widgets sind für jede Wahl verfügbar.

Alle Applikationen können über dieselbe **index.html** aufgerufen werden. Die Variable „standalone“ steuert, welche Applikation gestartet wird.

### 5.1. Einzel-Widgets

Name	standalone=	bw_de	lw_sl	lw_sh	lw_nw	pw_nl	pw_fr
<b>Prozentuale Stimmenverteilung</b> (klassisch)	bars	✓	✓	✓	✓		
<b>Prozentuale Stimmenverteilung</b> (Querbalken)	barsList	✓	✓	✓	✓	✓	
<b>Sitzverteilung</b>	seats	✓	✓	✓	✓	✓	
<b>Gewinne/Verluste</b>	profitsLosses	✓	✓	✓	✓		
<b>Direktwahlfrage</b>	ranking	✓	✓	✓	✓	✓	✓
<b>Koalitionsrechner</b>	coalition	✓	✓	✓	✓		
<b>Wahlkreise</b>	votingAreas	✓	✓	✓	✓		
<b>Galerie</b> (Kanzler)		✓					

Beispiel **Sitzverteilung**:

Verlinken Sie auf [my\\_bw\\_de/html/index.html?standalone=seats](my_bw_de/html/index.html?standalone=seats)

Das Widget „**Galerie**“ ist ein eigenständiges Produkt und wird nicht auf Ihrem Server abgelegt, sondern bei uns gehostet. In Ihrer **sitemap.html** finden Sie einen Link zur Setup-Seite. Dort können Sie Ihre Einstellungen vornehmen (Größe des Widgets, IVW-Pixel, ...) und einen embed-Code generieren.

### 5.2. Widget-Kombinationen

Sie haben außerdem die Möglichkeit, einzelne Widgets zu einer Applikation zusammenzufassen.

Name	standalone=	standaloneComponents=
<b>Prozente/Sitze</b>	resultsCombi	bars,seats
<b>Prozente/Gewinne&amp;Verluste</b>	resultsCombi	bars,profitsLosses
<b>Sitze/Gewinne&amp;Verluste der Prozente</b>	resultsCombi	seats,profitsLosses
<b>Sitze/Gewinne&amp;Verluste der Sitze</b>	resultsCombi	seats,profitsLosses&value=seats

Beispiel **Prozente/Sitze**:

Verlinken Sie auf

[my\\_bw\\_de/html/index.html?standalone=resultsCombi&standaloneComponents=bars,seats](my_bw_de/html/index.html?standalone=resultsCombi&standaloneComponents=bars,seats)



## 6 Individuelle Einstellungen

Sie können Ihre Grafik nach eigenen Vorstellungen und Ideen anpassen. Über Variablen können Sie teilweise das Verhalten der Applikation ändern, über CSS können Sie Farben, Schrift und Layout an Ihr CI angleichen. Wir haben Ihnen dazu in der **style.css** in **my\_bw\_de/feed** die wichtigsten Klassen zur Anpassung bereits vorbereitet.

### 6.1 Widget-Optionen

Über das Setzen von Variablen können Sie die Applikationen an Ihre Bedürfnisse anpassen. Sie können diese Variablen über die Adresszeile an die Applikation übergeben, oder direkt über das DIV der Applikation.

#### Beispiel „resultType“:

Übergabe per URL-Var:

*my\_bw\_de/index.html?resultType=live*

Übergabe direkt über das DIV:

```
<div data-app="dpa" data-competition-id="bw_de" data-standalone="bars" data-result-type="live" data-path="[Ihr Pfad]/bw_de/"></div>
```

**Wichtig:** Beachten Sie die unterschiedlichen Schreibweisen der Variablen. Da in data-Attributen kein Camel-Case erlaubt ist, wird hier mit Bindestrichen gearbeitet, anstatt mit Großbuchstaben. In der folgenden Liste werden Ihnen immer beide Schreibweisen angegeben.

Name	Optionen (grün=Default)	Beschreibung
<i>resultType</i> data-result-type	"auto", "live", "survey"	"auto" (Default, erst Umfragen, automatischer switch auf Live-Ergebnisse, sobald sie vorliegen), „live“ (NUR Live-Ergebnisse, leer, wenn noch nichts da ist) oder „survey“ (nur Umfrage-Ergebnisse) möglich.
<i>autoResetTime</i> data-auto-reset-time	"15"	Bestimmt den Zeitraum in Minuten, in dem die Applikation von „auto“ auf „live“ umspringt; der Default-Wert ist „15“. Das bedeutet z.B. für die Bundestagswahl: In den 15 Minuten, bevor die Prognose erwartet wird (also ab 17:45 Uhr) zeigt die Applikation nicht mehr die letzte Umfrage an, sondern einen leeren Screen.
<i>maxParties</i> data-max-parties	Number	Mit „maxParties“ kann eingestellt werden, wie viele Balken höchstens angezeigt werden sollen. Steht „others“ nicht auf „false“, werden alle weiteren Parteien auf die Sonstigen draufgerechnet. Steht „maxParties“ also auf 6, werden 5 Parteien und die Sonstigen angezeigt.



Name	Optionen (grün=Default)	Beschreibung
<i>others</i> data-others	"true", "false"	Steht „others“ auf „false“, werden 6 Parteien und keine Sonstigen angezeigt. Wenn „maxParties“ nicht gesetzt wird, „others“ aber auf „false“ steht, werden alle vorhandenen Parteien außer den Sonstigen angezeigt. Ist keine von beiden Variablen gesetzt, werden alle Parteien, inkl. Sonstige, angezeigt.
<i>difference</i> data-difference	"true", "false"	Default ist "true". Wenn "difference" auf false steht, werden die Tendenzen in "bars" und "seats" nicht angezeigt.
<i>differenceType</i> data-difference-type	"value", "icon"	Default ist "icon". Je nach Wert werden bei den Differenzen entweder die Icons (Pfeil nach oben, Pfeil nach unten, Punkt) oder die Zahl angezeigt.
<i>differenceSource</i> data-difference-source	"election", "survey"	Default ist "survey". Hier wird beim Berechnen der Gewinne/Verluste auf die vorherige Umfrage zurückgegriffen. Wenn es keine gibt, dann auf das Ergebnis der letzten Wahl. Bei "election" wird immer das Ergebnis der letzten Wahl als Berechnungsgrundlage genommen.
<i>value</i> data-value	"seats", "percent"	Kann den Widgets "bars" und "barsList" übergeben werden, dann werden statt der Prozentwerte die Sitze angezeigt.
<i>displayMap</i> data-display-map	"true", "false"	Kann dem Widget votingAreas übergeben werden. Es wird nur der Zustand "false" abgefragt, dann wird die Karte nicht angezeigt.
<i>view</i> data-view	"small"	Kann dem Widget "ranking" übergeben werden, dann wird die "Bauchbindenversion" angezeigt, sofern nicht mehr als zwei Einträge in den Daten vorhanden sind.
<i>widgetHeader</i> data-widget-header	"true", "false"	Kann jedem Widget übergeben werden, um die Kopfzeile (inkl. Social- und Aktualisieren-Button!) auszublenden.
<i>widgetFooter</i> data-widget-footer	"true", "false"	Kann jedem Widget übergeben werden, um die Fußzeile einzublenden. In der Fußzeile können über weitere Variablen der Text und eine Weiterverlinkung eingestellt werden.
<i>widgetFooterText</i> data-widget-footer-text	"Ihr Text"	Wenn die Fußzeile eingeblendet ist, kann über diese Variable der Text gesetzt werden.
<i>pathClick</i> data-path-click	URL	Wenn die Fußzeile eingeblendet ist, kann über diese Variable eine URL übergeben werden, auf die bei Klick auf die Fußzeile verlinkt wird.



Name	Optionen (grün=Default)	Beschreibung
<i>pathClickTarget</i> data-path-click-target	"_top", "_blank", "_self", "_parent"	Als Erweiterung zu pathClick gibt es noch pathClickTarget, wo Sie bestimmen können wo der neue Inhalt angezeigt wird. _blank ist dabei der default-Wert. Mit _blank geben Sie an, dass der pathClick Link in einem neuen Tab/Fenster geöffnet wird. _self würde bedeuten, dass der Link sich im selben Tab/Fenster öffnet und _parent meint das Fenster, welches den iframe beinhaltet, in dem die Applikation läuft.
<i>updateTime</i> data-update-time	"60"	Der default-Wert für die automatischen Updates ist 60 Sekunden. Sie können diesen über die Variable beliebig anpassen. 0 steht dabei für keine automatischen Updates.
<i>width</i> data-width	"640px", "100%"	Mit width können Sie auch außerhalb von CSS die Breite des Widgets festlegen. Dabei können Sie mit festen „px“-Werten arbeiten, aber auch mit prozentualen Werten.
<i>customFont</i> data-custom-font	"false", "true"	Der default-Wert ist false und beinhaltet unseren mitgelieferten Font „Roboto“. Falls Sie einen eigenen Font verwenden wollen, übergeben Sie „customFont=true“. Damit wird unser Font nicht mehr mitgeladen und Sie sparen Traffic.
<i>singleSource</i> data-single-source	"" "source02"	Damit kann ein Widget auf eine Quelle festgelegt werden. Es wird auch dann kein Flyout mehr generiert, wenn Daten aus mehreren Quellen vorliegen. Übergeben wird die Quellen-Id. Beispiel: singleSource=source02 → „source02“ ist eine live-Quelle, wenn resultType auf "survey" steht, passiert hier nichts. Erst wenn resultType auf "live" steht (entweder, weil live-Daten vorhanden sind oder weil der resultType ebenfalls übergeben wurde) wird das Flyout nicht gebaut und nur "ZDF" bleibt übrig. Achtung: Greift auch, wenn noch keine Daten vorhanden sind! Die Quelle steht dann nicht in der Überschrift, das Widget steht auf "Keine Daten vorhanden". Die Verwendung dieser Variablen ist dann sinnvoll, wenn zwei Quellen gegenübergestellt werden sollen (zweimal das gleiche Widget einbinden und unterschiedliche Quellen übergeben). Zu beachten ist hier, dass das Endergebnis immer Vorrang hat! Wenn also z.B. über den Wahlabend ARD und ZDF gegenübergestellt wurden, werden beide Widgets das Endergebnis anzeigen, sobald es vorliegt.



Name	Optionen (grün=Default)	Beschreibung
<i>state</i> data-state	"" "1", "2", ... "16",	<p>Kann dem Widget votingAreas übergeben werden (nur Bundestagswahl!). Der Wert ist der amtliche Gemeindeschlüssel der Bundesländer (1 → Schleswig-Holstein, ... 16 → Thüringen), ohne führende Null. Wird „state“ übergeben, wird die Applikation mit Zoom auf das entsprechende Bundesland starten.</p> <p>Wenn Sie zusätzlich noch eine Wahlkreis-Id als Deeplink übergeben (s.u. Kapitel 7), startet die Applikation immer mit diesem Wahlkreis, unabhängig davon, ob anderswo Ergebnisse vorliegen.</p> <p>Wird kein Wahlkreis übergeben, startet die Applikation mit dem ersten Wahlkreis des Bundeslandes (Reihenfolge alphabetisch, Bayern → Wahlkreis Altötting), solange noch gar kein Wahlkreisergebnis vorliegt. Sobald ein Wahlkreisergebnis in dem entsprechenden Bundesland einläuft, startet die Applikation immer mit dem aktuellsten Wahlkreis des Bundeslandes.</p>
<i>deepLink</i> data-deep-link	String	Falls es Ihnen nicht möglich ist, gewünschte Deeplinks mit „#“ ans Ende der URL zu hängen (s.u. Kapitel 7), können Sie den Deeplink auch über die Variable übergeben. Die Schreibweise des zu übergebenden Strings bleibt gleich.
<i>customClass</i> data-custom-class	String	Hier können Sie einen individuellen Klassennamen übergeben, über den die Applikation dann ansprechbar ist. So können Sie z.B. über Ihre style.css dem gleichen Widget unterschiedliche Layouts geben.



## 6.2 Konfiguration des eigenen Layouts

In der mitgelieferten Datei `my_bw_de/feed/style.css` können Sie alle individuellen Einstellungen vornehmen. Die wichtigsten Klassen haben wir dort für Sie bereits vorbereitet.

### 6.2.1 Schrift

Die verwendete Schriftart und die Schriftfarbe stellen Sie hier ein:

```
.dpa_app
{
  font-family      : Roboto, Arial, sans-serif;
  color            : #1a222f;
  fill             : #1a222f;
}
```

**Wichtig:** Wenn Sie eine eigene Schriftart verwenden, denken Sie daran, „`customFont=true`“ an die Applikation zu übergeben, damit unser Font nicht noch zusätzlich zu Ihrem geladen wird.

Die **Schriftgrößen** werden über Klassen eingestellt, so können Sie schnell einzelne Schriftgrößen anpassen oder vereinheitlichen:

```
.dpa_fontSize11    { font-size:11px; }
.dpa_fontSize12    { font-size:12px; }
.dpa_fontSize13    { font-size:13px; }
.dpa_fontSize14    { font-size:14px; }
.dpa_fontSize15    { font-size:15px; }
.dpa_fontSize16    { font-size:16px; }
.dpa_fontSize18    { font-size:18px; }
```

### 6.2.2 Farben

Die Hintergrundfarbe des **Headers** und der Überschriften, sowie die Farbe der darin liegenden Texte und Icons, ändern Sie über die Klasse „`backgroundColorHighlight`“:

```
.dpa_app .backgroundColorHighlight
{
  background-color: #882E2E;
  color           : #FFFFFF;
  fill            : #FFFFFF;
}
```

Der **Content-Bereich** ist mit einem leichten Grau eingefärbt. Wenn Sie das anpassen möchten, ändern Sie die folgende Klasse:

```
.dpa_appSA .dpa_body
{
  background-color : #F5F5F5;
}
```



Die Klassen für die **Parteifarben** finden Sie ebenfalls bereits vorbereitet in Ihrer style.css vor:

```
/* ##### SPD ##### */  
.bg_p100 {background-color : #f50000;}  
...
```

Wenn Ihnen dort eine Partei fehlt, können Sie die zugehörige Partei-Id aus der Labels-Datei `bw_de/assets/feed/de/labels.bw_de.xml` erschließen. Ansonsten melden Sie sich gerne bei uns.

### 6.2.3 Weitere Änderungen

In Ihrer style.css finden Sie noch weitere Optionen vorbereitet.

Wenn Sie Fragen dazu haben oder noch tiefer gehende Änderungen an den Applikationen vornehmen wollen, sprechen Sie uns bitte an, wir unterstützen Sie gern.





## 7 Deeplinks

In einigen Widgets können Sie direkt mit einer festgelegten Startansicht einsteigen, z.B. mit einem bestimmten Umfrageinstitut oder einer Wunschkoalition beim Koalitionsrechner.

Um Ihre gewünschte Startansicht festzulegen, erweitern Sie ihre URL mit dem entsprechenden Deeplink.

Beispiel 1: → **Einstieg mit der letzten Umfrage von Infratest im Koalitionsrechner**

`.../my_bw_de/html/index.html?standalone=coalition#/surveySource01`

Beispiel 2: → **Einstieg mit der letzten Umfrage von Infratest und der Koalition SPD-Grüne im Koalitionsrechner**

`.../my_bw_de/html/index.html?standalone=coalition#/surveySource01-p100,p300`

Sie können Deeplinks auch über die **Variable "deepLink"** übergeben: `&deepLink=surveySource01`

Die Liste der verfügbaren Deeplinks finden Sie in der nachfolgenden Tabelle:

Deeplinks	Navigationspunkt / Inhalte	Verfügbar für
<p><i>INSTITUTE</i></p> <p><code>#/surveySource01</code>  <code>#/surveySource02</code>  <code>#/surveySource03</code>  <code>#/surveySource04</code>  <code>#/surveySource05</code>  <code>#/surveySource06</code>  <code>#/surveySource07</code></p> <p><code>#/source01</code>  <code>#/source02</code></p>	<p>Umfrage Infratest  Umfrage Forschungsgruppe Wahlen  Umfrage Forsa  Umfrage Emnid  Umfrage Allensbach  Umfrage INSA  Umfrage GESS</p> <p>Prognose/Hochrechnungen ARD  Prognose/Hochrechnungen ZDF</p>	<p>bars, barsList, seats, coalition, profitsLosses, ranking, resultsCombi</p>
<p><i>KOALITIONEN</i></p> <p><code>#/p100,p200</code>  <code>#/p100,p300,p500</code>  ...</p>	<p>Große Koalition  Rot-Rot-Grün</p> <p>→ Grundsätzlich können Koalitionen zwischen allen Parteien gewählt werden, die Sitze erhalten haben.</p>	<p>coalition</p>
<p><i>Wahlkreise</i></p> <p><code>#/wk1</code>  <code>#/wk2</code>  ...</p>	<p>(Beispiel Bundestagswahl)  Schleswig-Flensburg  Nordfriesland  ...</p> <p>→ Eine Liste der Wahlkreiszuordnungen finden sie in <b>bw_de/config/areas.json</b>.</p>	<p>votingAreas</p>



## 8 Social Media

Sie haben die Möglichkeit, ihre Grafik auch an soziale Netzwerke wie **Twitter**, **Facebook** oder **Google+** anzubinden oder eine vorgefertigte **E-Mail** anzubieten. Damit können Ihre User aktuelle Ergebnisse direkt mit dem Verweis auf Ihre Grafik verbreiten.

In Ihrer Konfigurationsdatei **my\_bw\_de → feed → config.xml** steht Ihnen ein eigener Knoten mit dem Namen **social** zur Verfügung. In diesem Knoten finden Sie folgende Attribute:

```
<social url="">
  <mail visibility="true"><![CDATA[Hier könnte Ihr Text stehen.]]></mail>
  <gplus visibility="true"><![CDATA[Hier könnte Ihr Text stehen.]]></gplus>
  <facebook visibility="true"/>
  <twitter visibility="true"><![CDATA[Hier könnte Ihr Text stehen.]]></twitter>
  <whatsapp visibility="true"><![CDATA[Hier könnte Ihr Text stehen.]]></whatsapp>
</social>
```

In dem Attribut „**url**“ können Sie den Link eintragen, auf den der User verweisen soll. Wenn Sie nichts eintragen, wird bei direkter Einbindung in Ihr CMS die URL aus der eigenen Adresszeile verwendet und bei Einbindung über **iframe** mithilfe der **pym.js** die URL der einbindenden Seite.

Wenn Sie die Anbindung an einzelne Netzwerke abstellen wollen, ändern Sie in dem jeweiligen Attribut „**visibility**“ den Eintrag auf **"false"**. Wenn Sie bei allen „visibility“ auf „false“ stellen, wird der Social-Button nicht mehr angezeigt. Der Button für WhatsApp ist nur mobil sichtbar.

Für Twitter, Google+, WhatsApp und E-Mail können Sie zudem noch einen Text vorkonfigurieren:

```
<social url="http://dpa-infoactive.com">
  <mail visibility="true"><![CDATA[Lieber E-Mailkontakt,...]]></mail>
  <twitter visibility="true"><![CDATA[Liebe Twitterfreunde,...]]></twitter>
  ...
</social>
```



## 9 IVW-Zählung

Zur Zählung der Klicks haben wir ein externes JavaScript angelegt. Die click.js finden Sie im Verzeichnis *my\_bw\_de/js*.

```
function click(objTrac)
{
    //alert(objTrac.tag1 + "," + objTrac.tag2)

    // SZM 2.0
    var iam_data = {
        "st":"kunde",
        "cp":"btw-wahl-live",
        "oc":"btw-wahl-live",
        "mg":"yes",
        "sv":"i2"
    }

    iom.c(iam_data,1);
}
```

**Hinweis:** Bei der Einbindung des **SZM 2.0**-Tags<sup>1</sup> sind einige Dinge zu beachten. Standardmäßig wird der IVW-Pixel über *document.write* geladen. Bei einer AJAX<sup>2</sup>-Anwendung hat das aber zur Folge, dass der gesamte Inhalt überschrieben wird und die Seite damit leer ist.

Das kann mit zwei kleinen Änderungen des üblichen Codes verhindert werden:

1. Anstatt „in“ tragen AGOF<sup>3</sup>-Kunden bei „sv“ „i2“ ein.
2. Beim Aufruf von *iom.c()* wird ein weiterer Parameter<sup>4</sup> mitgegeben.

Beide Änderungen bewirken, dass der Pixel anstatt mit *document.write* mit der *appendChild()-Methode* ausgeliefert wird.

---

<sup>1</sup> SZM 1.5-Tags werden ab dem seit 1. September 2014 nicht mehr unterstützt.

<sup>2</sup> Unsere Grafiken nutzen die Web-Technologie AJAX (Konzept der asynchrone Datenübertragung) zur Nutzerinteraktion.

<sup>3</sup> AGOF: Arbeitsgemeinschaft Online Forschung ([www.agof.de](http://www.agof.de))

<sup>4</sup> (iam\_data,1) - Übertragung der Daten an den Messserver über die *appendChild()-Methode*



## 10 Google Analytics

Zur Einbindung von einem oder mehreren Google Analytic-Codes können Sie die **config.xml** nutzen. Die config.xml finden Sie im Verzeichnis **my\_bw\_de/feed**.

```
<ga>
  <tracking create="{trackingId: 'UA-XXXXXXXX-X'}" />
  <tracking init-pageview="true" create="{trackingId: 'UA-XXXXXXXX-X'}" />
</ga>
```

Jeder **<tracking>** Eintrag steht für eine Google-Analytics-Einbindung. Das Attribut **init-pageview** besagt hierbei, ob beim erstmaligen Aufruf der Applikation bzw. des Widgets ein pageView ausgeführt werden soll oder nicht.