



# Gadgets

Martin Kozianka

DOPaK 2018

# Gadgets – Eine kurze Wiederholung

## Erweiterung der DOCUMENTS-Funktionalität

- Übersichtsseiten, Formulare, Listen, Diagramme
- Erweiterung der DOCUMENTS Oberfläche
- Integration an vielen Stellen in DOCUMENTS möglich

## Gadgets allgemein

- Keine clientseitige Installation nötig
- Keine Updateproblematik / Mandantenabhängigkeit
- Hohe Wiederverwendbarkeit
- Einfache vordefinierte Programmierschnittstelle
- Interaktion mit anderen Gadgets und mit DOCUMENTS

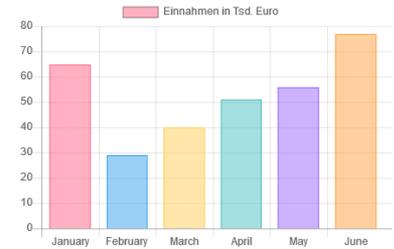
Mein Mappendeckel

Personalnummer: 00048



- **Name:** Schreiber
- **Vorname:** Willi
- **Email:** schreiber@dopa.ag
- **Telefon:** +49 (231) 958069 40

Einnahmen



Datenübernahme (Nordwind-DB)

Firma	Kontakt	Funktion	Ort	Land
Alfreds Futterkiste	Maria Anders	Vertriebsmitarbeiterin	Berlin	D
Ana Trujillo Emparedados y helados	Ana Trujillo	Inhaberin	México D.F.	M
Antonio Moreno Taquería	Antonio Moreno	Inhaber	México D.F.	M
Around the Horn	Thomas Hardy	Vertriebsmitarbeiter	London	G
Berglunds snabbköp	Christina Berglund	Einkaufsleitung	Luleå	S
Blauer See Delikatessen	Hanna Moos	Vertriebsmitarbeiterin	Mannheim	D
Blondel père et fils	Frédérique Citeaux	Marketingmanager	Strasbourg	F
Bólido Comidas preparadas	Martin Sommer	Inhaber	Madrid	S
Bon app'	Laurence Leibhan	Inhaber	Marseille	F
Bottom-Dollar Markets	Elizabeth Lincoln	Buchhalterin	Tsawassen	K
Er's Beverages	Victoria Ashworth	Vertriebsmitarbeiterin	London	G

Textfeld

Auswahlliste

Eins

Datumsfeld

Abschicken

Meine Zeitleiste 1

My bubble content.

My bubble content.

My bubble content.

# Gadgets – Eine kurze Wiederholung

## Portal-Scripting

- Gadgets sind Portalskripte
- Basieren auf der Gadget Scripting-API
- Die **gesamte** Portalscript API steht zur Verfügung

## Einschränkungen

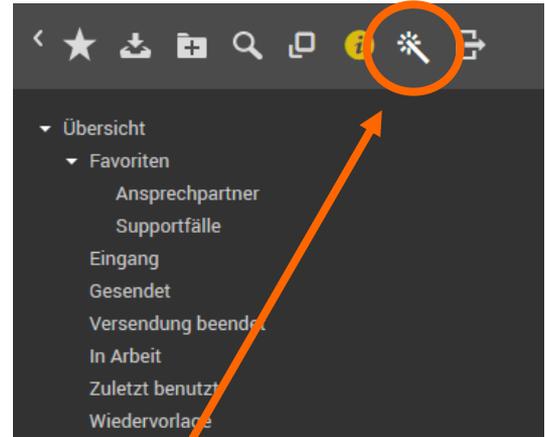
- Kein Ersatz für vorhandene Programmfunktionen
- Dienen nicht der Umgehung oder Veränderung von Berechtigungen in DOCUMENTS
- Keine Verwendung ohne Scripting-Kenntnisse

# Neue Integrationsmöglichkeiten 1/4

## Als Menübar-Icon

- Globale Eigenschaft: **globalGadgetConfig**
- Anzeige des Gadgets in der Listenansicht
- Alternative Anzeige im Dialog
  - Attribut **gadgetDialog: true** der gadgetConfig hinzufügen
- Konfiguration des Icons über das Attribut **gadgetIcon**
  - Verwendung des bekannten Syntax für entypo und ionicn Icons

```
{  
  gadgetScript: 'Gadget_EmailForm',  
  gadgetAction: 'showForm',  
  gadgetIcon: 'ionicons:ion-md-color-wand;top:-2px;font-size: 29px;',  
  gadgetDialog: true  
}
```

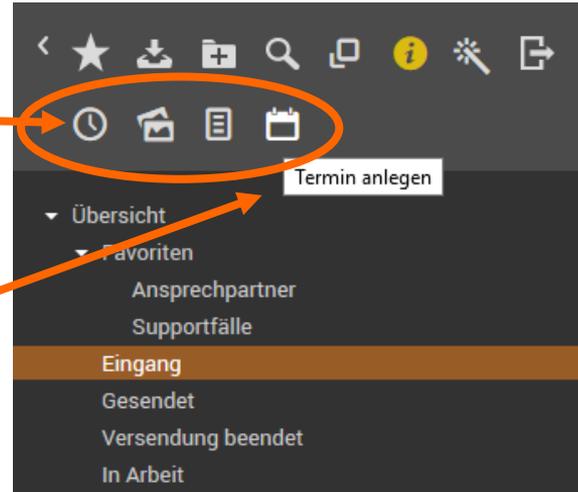


# Neue Integrationsmöglichkeiten 2/4

## Als Menübarleiste

- Globale Eigenschaft: **menubarGadgetConfigs**
- Leiste von Icons unterhalb der Menübar
- Konfiguration über ein Array von Gadget-Konfigurationen
- Attribute **gadgetIcon** und **gadgetDialog** können auch hier verwendet werden

```
[{  
  gadgetScript: 'Gadget_FileCreate',  
  gadgetAction: 'showForm',  
  gadgetIcon: 'entypo:clock',  
  gadgetDialog: true,  
  fileType: 'crmTask'  
}, ... {  
  gadgetScript: 'Gadget_FileCreate',  
  gadgetAction: 'showForm',  
  gadgetIcon: 'entypo:calendar',  
  gadgetDialog: true,  
  fileType: 'crmAppointment',  
  tooltip: 'Termin anlegen',  
}]
```



# Neue Integrationsmöglichkeiten 3/4

Als Bereich unterhalb der Menübar

- Globale Eigenschaft:  
**naviBarGadgetConfig**
- Hier ein Beispiel zur *Schnellanlage* eines Termin
  - Eingabe eines Termins als Textzeile
  - Skript versucht Termindaten und Titel auszulesen
  - Ein Gadgetformular wird mit den erkannten Werten befüllt und geöffnet

The screenshot shows a mobile application interface with a dark theme. At the top, there is a navigation bar with icons and a search field. Below the navigation bar, there is a calendar overview showing a date range from Monday 15-17. A text input field in the calendar overview contains the text "Mo 15-17 Besprechung DOPaK 2018". An orange circle highlights this text input field. Below the calendar overview, there is a "Termin anlegen" (Create Appointment) dialog. The dialog is pre-filled with the following information:

- von: 16.04.2018 15:00
- bis: 16.04.2018 17:00
- Titel: Besprechung DOPaK 2018
- Beschreibung: (empty text area)

At the bottom of the dialog, there are two buttons: "Anlegen und öffnen" and "Anlegen und schliessen".

# Neue Integrationsmöglichkeiten 4/4

## Im Monitor

- Anzeige im Monitor der Mappenansicht
- Eigenschaft am Mappentypen: **monitorGadgetView**
- Das Beispiel ist ein Gadget vom Typ Chart
  - Details finden sich in der Dokumentation und auf den Folien der DOPaK 2016
  - Monitor-Informationen: **Auf Register darstellen** aktivieren um die Daten des Monitors anzuzeigen

Monitor

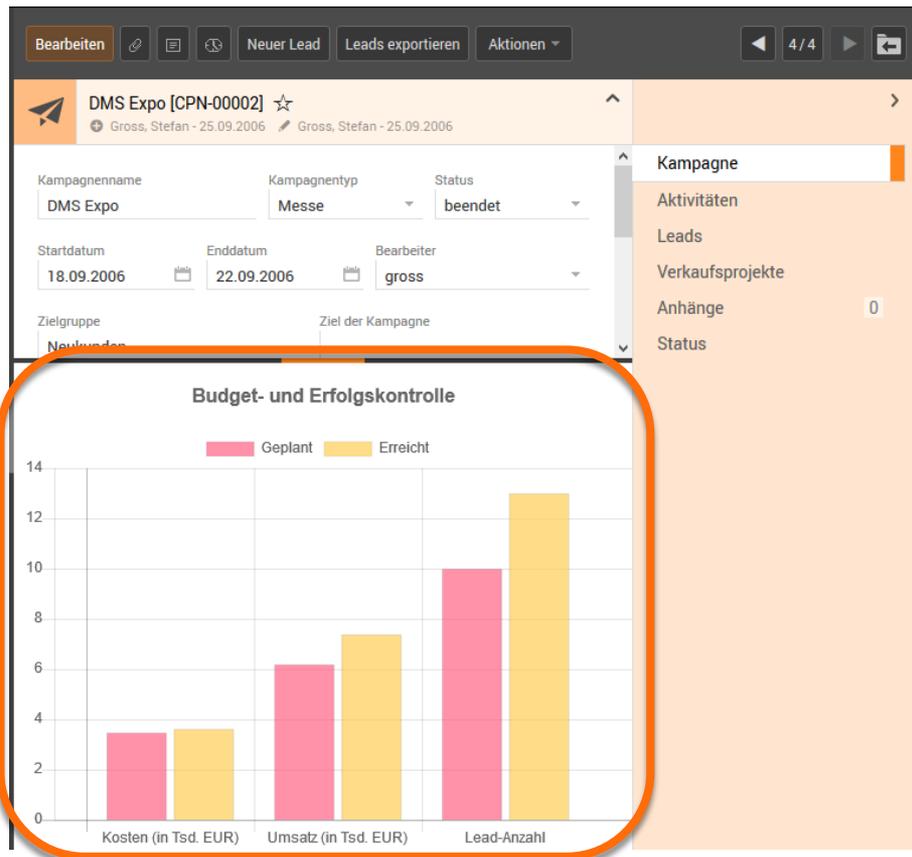
Monitor anzeigen

Alle Versendungen anzeigen

Umgekehrter Monitor

Monitor Kurzvorschau

Auf Register darstellen



# Allgemeine Erweiterungen

Vereinfachte Möglichkeit Daten an die Client-Funktionen zu übergeben.

- Die Methode `setContextData()` überträgt ein Datenobjekt an den Client
- Zugriff auf dieses Datenobjekt ist in den Client-Funktionen möglich

```
// Server (Portal-Skript)
var gadget = new otris.gadget.gui.Form();
// Setze Daten für die Client-Funktionen
gadget.setContextData({
  customData: "customValue",
  customFlag: false,
  gadgetId: gadgetContext.gadgetId
});
```

```
// Client-Funktion (Browser)
function onClickButton() {
  // Hole clientseitigen GadgetContext
  var gCxt = documentsContext.getGadgetContext();
  // Hole contextData
  var contextData = gCxt.getContextData();
  // Zugriff auf die übertragenen Werte
  if(contextData.bFlag === true) {
    alert(contextData.gadgetId);
  }
}
```

**Achtung:** Das serverseitige `gadgetContext` Objekt entspricht **NICHT** dem Objekt, das man über die Methode `documentsContext.getGadgetContext()` erhält.

# Allgemeine Erweiterungen

## Zugriff auf die clientseitige Instanz von Gadgets

- Zugriff auf ein Client-Objekt für die folgenden Gadgettypen:
  - **Form**: Objekt vom Typ GadgetForm (Details dazu in der Gadget API)
  - **FullCalendar**: Zugriff auf das jQuery-Objekt (Ausführliche API auf <https://fullcalendar.io>)
  - **Chart**: Zugriff auf das Chart.js-Objekt (Ausführliche API auf <https://chartjs.org>)
- Der Zugriff auf das Client-Objekt erfolgt über den **clientseitigen** GadgetContext

```
// Hole das Client-Objekt eines Chart-Gadgets
chartInstance = documentsContext.getGadgetContext().getClientObject();

// Zugriff auf die gesamte API-Funktionalität der Library
// Für die Chart-Gadgets: https://www.chartjs.org/docs/

// Daten des Charts anpassen und aktualisieren (z.B: Daten nachladen)
chartInstance.update(config);

// „png data url“ erzeugen (Grafik als Bild abspeichern)
dataUrl = chartInstance.toBase64Image();
```

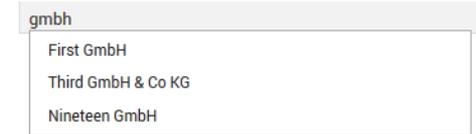
# Form Gadget Erweiterungen

## Autocomplete-Feld steht zur Verfügung

- Konfiguration über eine autocompleteConfig (analog zum Autocomplete auf Mappefeldern)

```
gadgetForm.addAutoCompleteField("autocompleteField",  
    "Autocomplete-Feld", null, {  
        scriptName: "autocompleteScript",  
        minQueryChars: 2  
    });
```

Autocomplete-Feld



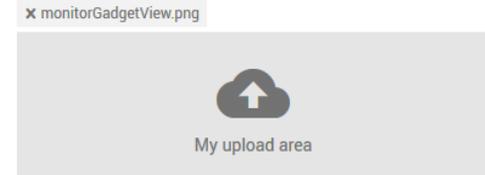
A screenshot of an autocomplete field. The input field contains the text "gmbh". Below the input field, a list of suggestions is displayed: "First GmbH", "Third GmbH & Co KG", and "Nineteen GmbH".

## Dropzone-Feld auch in Gadget-Formularen

- Funktionalität und Konfiguration analog zum Dropzone-Feld in Skriptdialogen

```
gadgetForm.addDropzone("myDropzone", "My dropzone", null, {  
    height: 120,  
    icon: "ionicons:ion-md-cloud-upload;font-size:64px;display:block;",  
    label: "My upload area,,  
});
```

My dropzone



A screenshot of a dropzone field. At the top, there is a small grey box containing the text "x monitorGadgetView.png". Below this, there is a large grey area with a dark grey cloud icon containing an upward-pointing arrow. Below the icon, the text "My upload area" is displayed.

# Form Gadget Erweiterungen

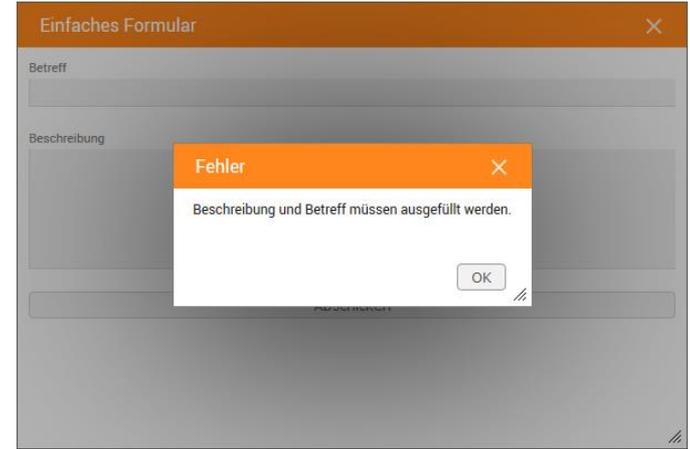
## Validatoren für Felder und Formulare

- Validatoren können auf Feldern und Formularen registriert werden.
- Die Validierung findet **clientseitig** statt

## Formular Validator

- Bei einem Fehler wird das Formular nicht abgeschickt und eine Fehlermeldung angezeigt.

```
gadgetForm.setFormValidator(function(field, gadgetForm, options) {  
    var errorMsg = "Beschreibung und Betreff müssen ausgefüllt werden.",  
        // Hole die Feldwerte (subject und description)  
        sub = (field.subject) ? field.subject.value[0] : undefined,  
        desc = (field.description) ? field.description.value[0] : undefined,  
  
        // Überprüfe ob die Felder gesetzt wurden  
        result = (subj && subj.length > 0 && desc && desc.length > 0);  
    return gadgetForm.createFormValidatorResult(result, errorMsg);  
});
```



# Form Gadget Erweiterungen

## Feld Validatoren

- Können an jedem Feld definiert werden
- Bei einem Fehler auf einem Feld wird das Formular nicht abgeschickt und die Felder markiert

```
// Feld „subject“ (Betreff) erstellen
element = gadgetForm.addTextField('subject', 'Betreff');

// Validator für das Feld setzen
element.setValidator(function(field, gadgetForm, options) {
    var msg = "Betreff muss gesetzt sein.",
        result = (field.value[0].length > 0);
    // gadgetForm ist das Client-Objekt des Formular-Gadgets
    // Ergebnis-Objekt mit Resultat und Fehlermeldung erstellen
    return gadgetForm.createFormValidatorResult(result, msg);
});
```

The screenshot shows a web form titled "Einfaches Formular" with a close button (X) in the top right corner. The form has two main sections: "Betreff" and "Beschreibung". The "Betreff" field is currently empty and highlighted with a red background, indicating a validation error. A small message box above the "Beschreibung" field displays the error message: "Betreff muss gesetzt sein.". The "Beschreibung" field contains the following text: "Seit der letzten DOPaK vor fast zwei Jahren haben wir DOCUMENTS umfangreich weiterentwickelt. Was es neues gibt und welche Tipps und Tricks wir Euch ans Herz legen möchten, erfahrt Ihr am ersten Konferenztage. Schon im Vorfeld hat die App für viel Aufsehen gesorgt. Die technischen Hintergründe und Anwendungsszenarien sind ein Schwerpunkt des zweiten Konferenztages." At the bottom of the form is a button labeled "Abschicken".

# Neues Gadget - PropertyGrid

## Geführte Eingabe von Schlüssel-Wert-Paaren

- Basiert auf der Gentable-Funktionalität (Invoice-Plugin)
- Speicherung der Daten als JSON-Objekt
- Für jeden Schlüssel kann ein JSON-Datenformat gewählt werden
  - string
  - number
  - boolean
- Editor- (Bearbeitung) und Renderer-Typen (Anzeige) des Gentables können genutzt werden
  - DATE2, NUMBER, CHECKBOX, ...



Farblaserdrucker BV-1909-B (2378913) ☆

👤 Schreiber, Willi - 12.04.2018 ✎ Schreiber, Willi - 18.04.2018

⬆

Produktnummer	Produktname
2378913	Farblaserdrucker BV-1909-B

### Produkteigenschaften

	Eigenschaft	Wert	
<input type="checkbox"/>	Gewicht (in kg)	9000000,78	
<input type="checkbox"/>	Wartungstermin	12.02.2020	
<input type="checkbox"/>	Aktiv	<input checked="" type="checkbox"/>	
<input type="checkbox"/>	Farbe	schwarz111111	
<input type="checkbox"/>	Farbe	gelb	
<input type="checkbox"/>	Maße (BxHxT in mm)	600x1240x360	

```

{
  "weight": 90,
  "maintenanceDate": "12.02.2020",
  "active": true,
  "colors": ["schwarz", "gelb"],
  "dimension": "600x1240x360"
}

```

# Neues Gadget - PropertyGrid

## Buttons im Bearbeitungsmodus

- Zeilen hinzufügen/enfernen und verschieben

## Auswahl des Schlüssels

- Schlüssel kann frei eingegeben werden (Textfeld)
- Auswahlliste für den Schlüssel

## Eingabe des Wertes

- Eingabe über den definierten Editor

Farblaserdrucker BV-1909-B (2378913)  
 Schreiber, Willi - 12.04.2018 Schreiber, Willi - 13.04.2018

Produktnummer: 2378913  
 Produktname: Farblaserdrucker BV-1909-B

Produkteigenschaften

<input type="checkbox"/>	Eigenschaft	Wert
<input type="checkbox"/>	Gewicht (in kg)	90
<input checked="" type="checkbox"/>	Wartungstermin	12.02.2020
<input type="checkbox"/>	Aktiv	
<input type="checkbox"/>	Farbe	
<input type="checkbox"/>	Farbe	
<input type="checkbox"/>	Maße (BxHxT in mm)	

Beschreibung

Calendar: FEBRUAR 2020  
 Mo Di Mi Do Fr Sa So  
 1 2  
 3 4 5 6 7 8 9  
 10 11 12 13 14 15 16  
 17 18 19 20 21 22 23  
 24 25 26 27 28 29

# Neues Gadget - PropertyGrid

```
gadgetGrid.setStore({
    type : "field",
    fieldName: "propertyGridData"
});
gadgetGrid.setEditable(gadgetContext.editMode);

gadgetGrid.setColumnOptions({
    keyLabel: "Eigenschaft",
    valueLabel: "Wert",
    keySelectionType: "SELECT"
}); // Weitere Konfigurationsmöglichkeiten in der API
gadgetGrid.addProperty("active", "boolean", "CHECKBOX", "Aktiv");
gadgetGrid.addProperty("weight", "number", "NUMBER", "Gewicht (in kg)");
gadgetGrid.addProperty("maintenanceDate", "string", "DATE2", "Wartungstermin");
gadgetGrid.addProperty("colors", "string[]", "TEXT", "Farbe");
gadgetGrid.addProperty("dimension", "string", "TEXT", "Maße (BxHxT in mm)");
```

Speicherort festlegen  
Mappenfild oder Skript möglich

Editiermodus für das Grid aktivieren

Spalten konfigurieren  
Art der Schlüsselauswahl festlegen

Schlüsselname

JSON-Typ (string, boolean, number)  
Angehängtes [] bewirkt Speicherung als Array

Gentable  
Editor/Renderer

Label für die  
Schlüsselauswahlliste

# Neues Gadget - Kalender

Kalendersuche aktivierbar

Verschiedene Ansichten  
(Tag, Woche, Monat, Jahr)

Konfigurierbare Klick-Aktion  
(Zum Beispiel: Mappe öffnen)

Neue Einträge über  
Doppelklick anlegbar

The screenshot displays the otris software calendar interface. The main calendar view shows a grid for April 2018. Callouts point to various features: 'Kalendersuche aktivierbar' points to the search icon in the top bar; 'Verschiedene Ansichten (Tag, Woche, Monat, Jahr)' points to the view selection buttons; 'Konfigurierbare Klick-Aktion (Zum Beispiel: Mappe öffnen)' points to the calendar grid; and 'Neue Einträge über Doppelklick anlegbar' points to a date cell. The right sidebar shows a detailed view of a meeting titled '[Termin] Eröffnung mit Keynote' with fields for status, location, description, and contact information.

Mo	Di	Mi	Do	Fr	Sa	So
13	26	27	28 08 Telefonat	29	30	31
14	2	3	4 09:30 Besprech. 14:15 Vertragsu.	5	6	7
15	9	10	11	12	13	14
16	16 08:15 DOPak 20	17	18	19 10 Datenschutz 13 Eröffnung mit 19 Abendverans	20 10 (Kater)frühst 10:30 otris macf	21
17	23 07:15 Telefonat	24	25	26	27	28
18	30	1	2	3 13 Projektdurchf	4 16 Projektdurchf	5
						6

**[Termin] Eröffnung mit Keynote**  
 Status: Offen  
 Ort: Dortmunder U  
 Beschreibung:  
 - Eröffnung mit Keynote  
 - Vorstellung DOCUMENTS 5.0d  
 - Tipps & Tricks  
 - VS Code & Scripting  
 - Gadgets & User Exits  
 - One more thing  
 Bearbeiter: schreiber von 19.04.2018 13:00 bis 19.04.2018 17:00  
 Bezugsinformation:  
 Kontakt: Firma  
 Verkaufsprojekt: Servicefall  
 Lead: Kampagne



# Neues Gadget - Kalender

## Kalender mit Daten befüllen

```
var colorMapping = {"Appointment" : "#e64a19", /*...*/ };
// files = FileResultset
for (var file = files.first(); file; file = files.next()) {

    // Typ der Mappe auslesen
    crmType = file.getAutoText("crmType.key");
    // Typ als Kategorie anlegen
    if(!events[crmType]) {
        events[crmType] = { color: colorMapping[crmType], entries: [] };
    }

    // Eintrag für die Mappe erzeugen und der Kategorie zuordnen
    events[crmType].entries.push({
        "id": file.getAutoText("id"),
        "title": file.getAutoText("crmSubject"),
        "start": file.crmDate,
        "end": file.crmEndDate
    });
}
```

Über die Mappen eines  
Ordners iterieren

Typ auslesen und als  
Kategorie anlegen

Eintrag der angelegten  
Kategorie zuordnen

Mappendaten in einen  
Kalendereintrag umwandeln

# Neues Gadget - Kalender

## Kalender konfigurieren und Daten setzen

```
var gadgetCalendar = new otris.gadget.gui.FullCalendar();
```

Gadget erzeugen

```
gadgetCalendar.setConfig({  
  // Wird bei einem Click auf ein Event ausgeführt  
  eventClick: function(calEvent, jsEvent, view) {  
    var fileId = calEvent.id;  
    documentsContext.openFileView(fileId);  
  }  
  // Weitere Konfigurationen (Button für Ansichtenwechsel)  
});
```

Konfiguration setzen

```
Object.keys(events).forEach(function(key) {  
  gadgetCalendar.addSource({id: key, color: events[key].color});  
});
```

Kategorien (source) dem  
Kalender hinzufügen

```
gadgetCalendar.setLazyLoad(false);  
gadgetCalendar.setEventLoader(function eventLoader(source) {  
  return events[source.id].entries;  
});
```

Einträge über eine  
Funktion zurückgeben 

# Neues Gadget - Kalender

Realisierung mit der JavaScript-Bibliothek FullCalendar



Verschiedene Ansichten konfigurierbar

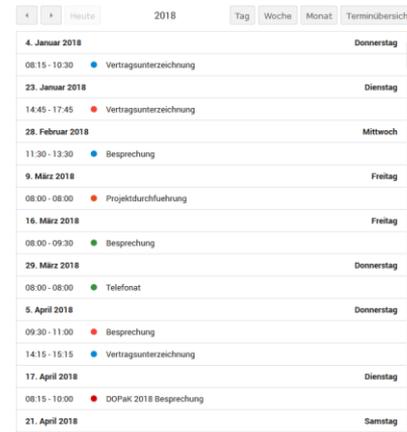
- Tages-, Wochen-, Monats- oder Jahresansicht
- Eigene Ansichten konfigurierbar

Einfache Erweiterung der Funktionalität

- Umfangreiche Konfigurationsmöglichkeiten
- Ausführlich dokumentierte Event-API
- Unterstützung für Drag&Drop

FullCalendar - A JavaScript event calendar. Customizable and open source.

- API Dokumentation: <https://fullcalendar.io/docs>



# Neues Gadget - Wizard

## Geführte Dateneingabe erstellen

- Konfiguration über die Definition von einzelnen Schritten
- Jeder Schritt enthält ein eigenes Gadget
- Zugriff auf die bisher gesammelten Daten in jedem Gadget

## Beispiel

- Benutzereigenschaften setzen
- **1. Schritt:** Anpassung der Darstellung (Skin und Signalfarbe)
  - HTML-Gadget

The screenshot shows a wizard window titled "Benutzereinstellungen anpassen" with a close button (X) in the top right corner. The interface is divided into three main sections:

- Übersicht (Overview):** Located on the left, it shows a progress indicator for three steps: "Darstellung" (highlighted in orange), "Mappenbearbeitung", and "Login". A callout bubble points to this section with the text "Übersicht Zeigt den Fortschritt".
- Infobereich (Info Area):** Located on the right, it provides information for the current step. It contains the text "Mit den beiden Einstellungen Skin und Signalfarbe lässt sich die Darstellung von Documents beeinflussen." A callout bubble points to this area with the text "Infobereich Information zum aktuellen Schritt".
- Inhaltsbereich (Content Area):** This is the main configuration area for the current step, highlighted with an orange border. It contains two sections:
  - Skin:** Two buttons labeled "Standard" and "Helles Grau".
  - Signalfarbe (Signal Color):** A grid of 16 color swatches. The first swatch (grey) is selected and has a small 'X' icon on it.
 A callout bubble points to this entire area with the text "Inhaltsbereich Gadget des aktuellen Schrittes".
- Buttonleiste (Button Bar):** Located at the bottom right, it contains three buttons: "Abbrechen" (Cancel), "Zurück" (Back), and "Weiter" (Next). A callout bubble points to this bar with the text "Buttonleiste Navigation im Wizard".

# Neues Gadget - Wizard

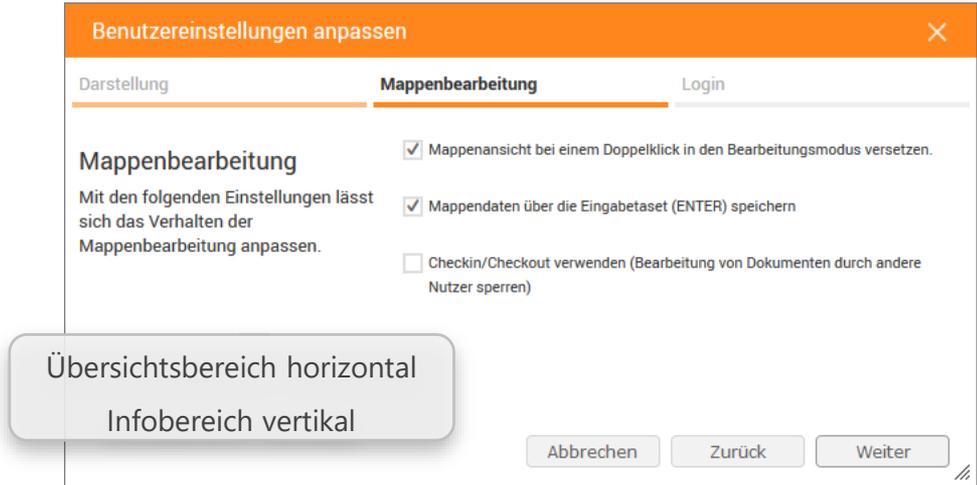
## Beispiel

- **2. Schritt:** Mappenbearbeitung
  - Formular Gadget mit 3 Checkboxes

```
fileEditOnDoubleClick
fileCommitOnEnter
hasCheckInCheckOut
```

## Anpassbare Darstellung

- **Übersicht** und **Infobereich** können ein- und ausgeblendet werden
- Ausrichtung der beiden Bereiche ist jeweils **oberhalb** oder **links** vom Inhalt möglich



```
gadgetWizard.showNavigation({
    enable: true,
    position: "top",
});
gadgetWizard.showInfoContent({
    enable: true,
    position: "left",
    width: "35%"
});
```

# Neues Gadget - Wizard

## Definition eines Schrittes

```
// Wizard-Gadget erstellen
gadgetWizard = new otris.gadget.gui.Wizard();

// Gadget-Konfiguration den Schritt
gadgetConfig = {
    gadgetScript: 'Gadget_UserSettings',
    gadgetAction: 'stepFile'
};
// Schritt mit id label und Gadget-Konfiguration erzeugen
step = new otris.gadget.gui.WizardStep("file", "Mappbearbeitung", gadgetConfig);
// Informationen hinzufügen
step.addInfo({title: "Mappbearbeitung", text: "Mit den ...anpassen."});
// Schritt dem Wizard Objekt hinzufügen
gadgetWizard.addStep(step);
```

Wizard-Gadget erstellen

Gadget für den Inhaltsbereich des Schrittes

ID des Schrittes  
(Zur Speicherung der Daten)

Informationen für den Schritt  
(Anleitungen/Erklärungen)

# Neues Gadget - Wizard

## Speicherung der Daten eines Schrittes

- Die Daten von Formular-Gadgets werden automatisch gespeichert
- Andere Gadget-Typen können über eine Client-Funktion Daten speichern

## Verwendung der Wizard Daten

- Zugriff auf die bisher gespeicherten Daten in jedem Schritt
- Möglichkeit abhängige Schritte (Formulare) zu erstellen
- Zugriff über die Gadget API

```
// Die Daten eines Schrittes über die ID holen
var stepData = otris.gadget.WizardData.getStepData(stepId);

// Die gesamten Daten des Wizards abrufen
var data = otris.gadget.WizardData.getData();

// Zugriff auf die Daten über die IDs der Schritte
systemUser.setAttribute("$fileEditOnDoubleClick", data.file.fileEditOnDoubleClick);
```



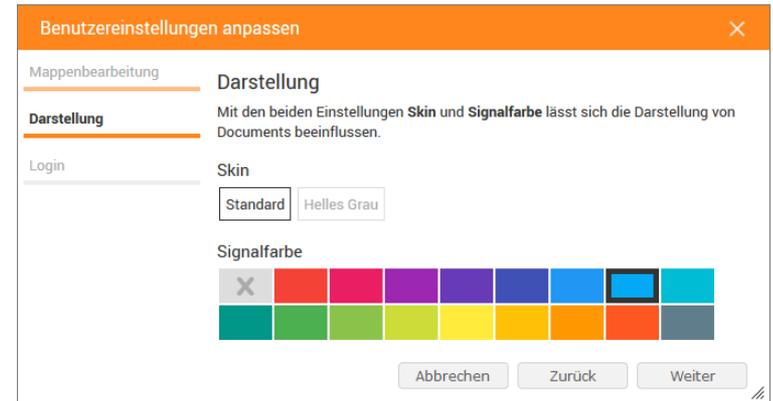
# Neues Gadget - Wizard

## Daten eines Schrittes über eine Client-Funktion speichern

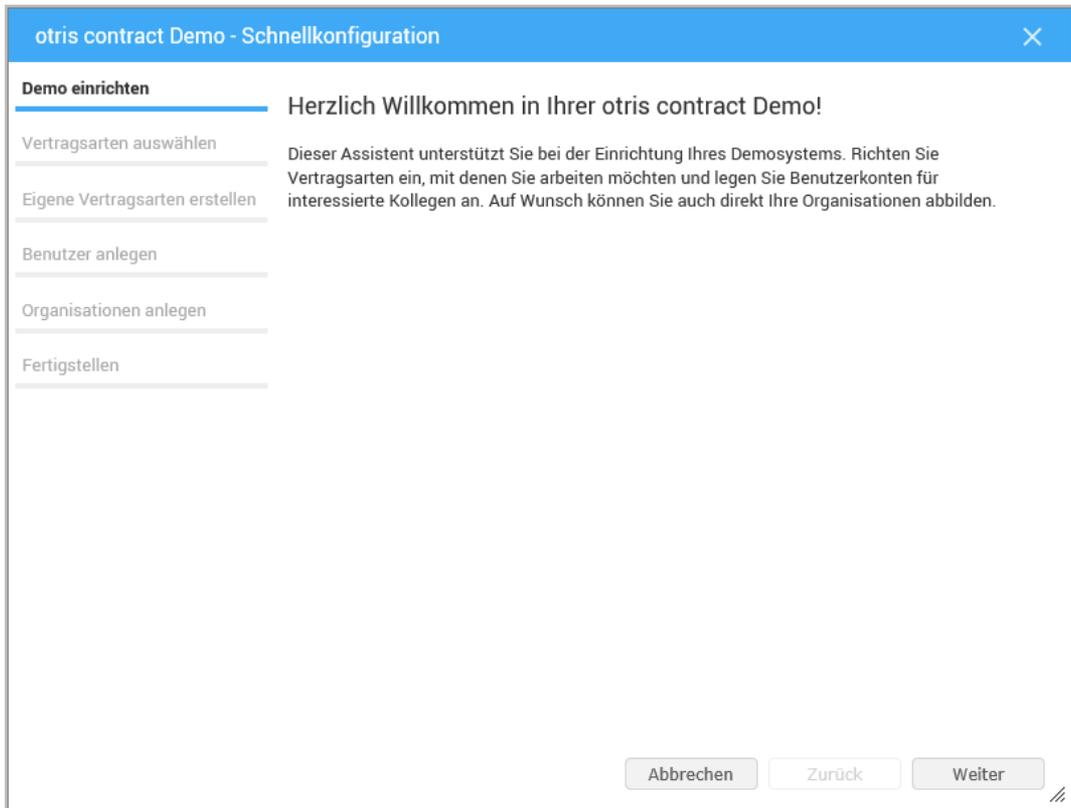
- HTML-Gadget mit einer Client-Funktion **customData**
- Wizard holt Daten über die konfigurierte Funktion
  - Im Beispiel wird das selektierte Skin und die selektierte Farbe übergeben

```
// Name der Client-Funktion als 4. Parameter übergeben
step = new otris.gadget.gui.WizardStep("skin", "Darstellung",
gadgetConf, "customData");
```

```
// Im Gadget des Schrittes
htmlGadget.addClientFunction(
    function customData() {
        // Im Gadget des Schrittes
        var data = collectCustomData();
        return data;
    }
);
```



# Contract Cloud Wizard



# Contract Cloud Wizard

otris contract Demo - Schnellkonfiguration ✕

Demo einrichten

**Vertragsarten auswählen**

Eigene Vertragsarten erstellen

Benutzer anlegen

Organisationen anlegen

Fertigstellen

**Vertragsarten auswählen**

Jeder Vertrag wird in otris contract einer bestimmten Vertragsart zugewiesen. Markieren Sie gewünschte Vertragsarten. Im nächsten Schritt können Sie individuelle Vertragsarten ergänzen.

Wählen Sie aus:

<input type="checkbox"/> Absichtserklärung (LOI)	<input type="checkbox"/> Kaufvertrag
<input type="checkbox"/> Arbeitsvertrag	<input type="checkbox"/> KFZ-Leasingvertrag
<input checked="" type="checkbox"/> Aufhebungsvertrag	<input type="checkbox"/> Leasingvertrag
<input type="checkbox"/> Bürgschaft	<input checked="" type="checkbox"/> Lizenzvertrag
<input checked="" type="checkbox"/> Dienstleistungsvertrag	<input type="checkbox"/> Mietvertrag
<input type="checkbox"/> Geheimhaltungsvereinbarung (NDA)	<input type="checkbox"/> Softwarepflegevertrag
<input checked="" type="checkbox"/> Gesellschaftsvertrag	<input checked="" type="checkbox"/> Telekommunikationsvertrag
<input type="checkbox"/> Handelsvertretervertrag	<input checked="" type="checkbox"/> Versorgungsvertrag
<input type="checkbox"/> Immobilienverwalter	<input type="checkbox"/> Versicherungsvertrag

Abbrechen Zurück Weiter

# Contract Cloud Wizard

otris contract Demo - Schnellkonfiguration ✕

Demo einrichten

Vertragsarten auswählen

**Eigene Vertragsarten erstellen**

Benutzer anlegen

Organisationen anlegen

Fertigstellen

## Eigene Vertragsarten erstellen

Hier können Sie neue Vertragsarten hinzufügen.  
Sie benötigen ein Kürzel, einen Namen für die Vertragsart und einen Namen für den Ordner, in dem alle Verträge dieser Art zu finden sind.  
Nehmen Sie im Zweifel einfach den Plural des Vertragsart-Namen. Das ist immer ein guter Ordnername!

Kürzel	Name	Ordner	
KONSIL	Konsiliararztvertrag	Konsiliararztverträge	Entfernen
KONSIG	Konsignationslagervertrag	Konsignationslagerverträge	Entfernen
BH	Beleghebammenvertrag	Beleghebammenverträge ✕	Entfernen

Neu anlegen

Abbrechen Zurück Weiter

# Contract Cloud Wizard

otris contract Demo - Schnellkonfiguration ✕

Demo einrichten

Vertragsarten auswählen

Eigene Vertragsarten erstellen

**Benutzer anlegen**

Organisationen anlegen

Fertigstellen

**Benutzer anlegen**

Legen Sie Benutzerkonten für Kollegen an. Die Anmeldung erfolgt mit der E-Mail-Adresse als Benutzername und Kennwort. Ein persönliches Kennwort kann nach der ersten Anmeldung gesetzt werden.

Anrede	Vorname	Name	E-Mail	
Frau ▾	Laura	Leser	leser@otris.de	Entfernen
Herr ▾	Andreas	A...!		Entfernen

Benutzer anlegen

Abbrechen Zurück Weiter

# Contract Cloud Wizard

otris contract Demo - Schnellkonfiguration ✕

Demo einrichten

Vertragsarten auswählen

Eigene Vertragsarten erstellen

Benutzer anlegen

**Organisationen anlegen**

Fertigstellen

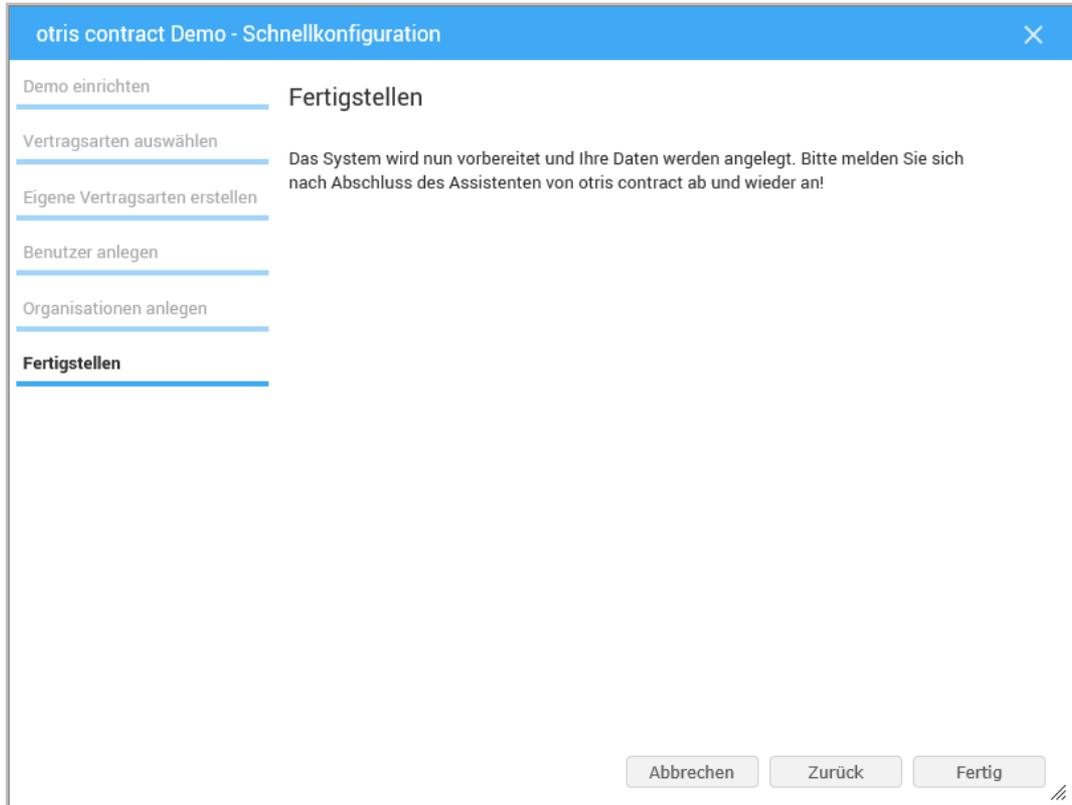
**Organisationen anlegen**

Organisationen dienen der strukturierten Ablage von Verträgen. Sie können verschiedene Unternehmen mit mehreren Ebenen abbilden und die Verträge dort ablegen. Bitte geben Sie zeilenweise Ihre Organisationen ein. Unterkategorien beginnen mit einem "-".

- DoPa AG
- Sales
- Sales Europe
- Sales America
- Legal
- Controlling
- DoPa Ltd.

Abbrechen Zurück Weiter

# Contract Cloud Wizard



Vielen Dank!

**Martin Kozianka**

[www.otris.de](http://www.otris.de)

otris software AG  
Königswall 21  
44137 Dortmund