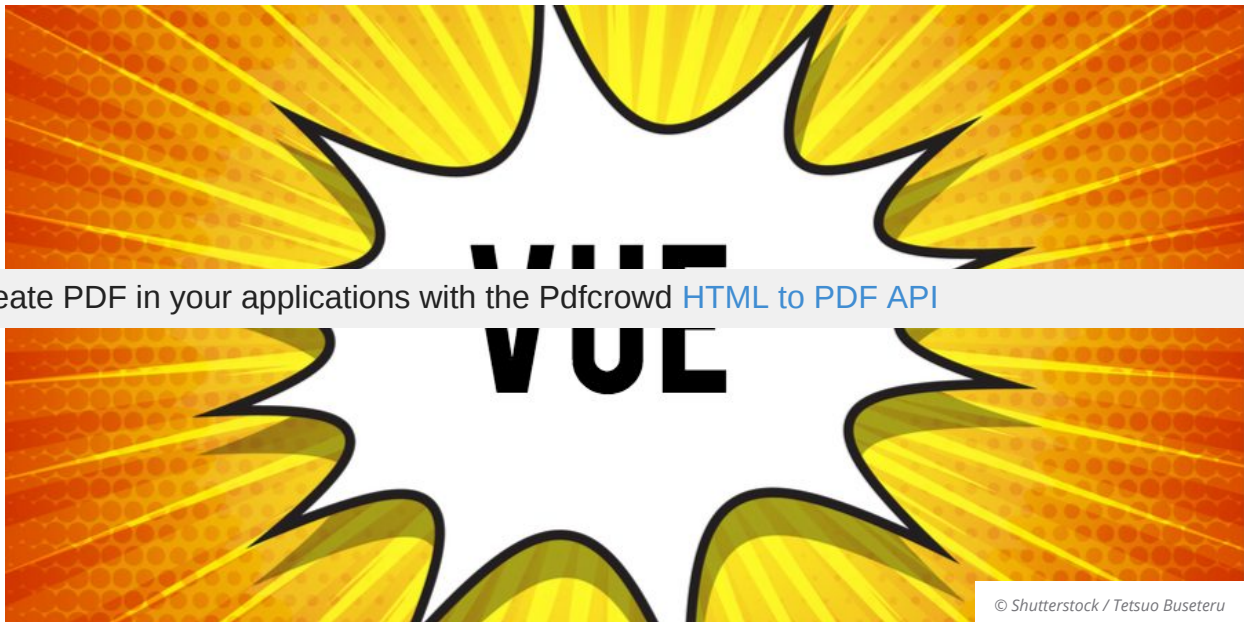


Siegeszug eines Frameworks

Vue.js Tutorial: Einführung in das JavaScript-Framework

👤 **Marc Teufel** ⌚ 2 Jahren online 💬 1 Kommentar

Bei Vue.js handelt es sich um ein Framework für Webanwendungen, mit dem man auf Basis von JavaScript komponentenorientiert entwickeln kann. Vue wirbt damit, dass es besonders schlank, anpassungsfähig und universell einsetzbar sei. Performant soll es auch noch sein, dieses Vue.js. Höchste Zeit also, dass wir uns dieses Framework einmal genauer ansehen. In einem zweiteiligen Tutorial geben wir eine systematische Einführung in das beliebte JavaScript Framework.



Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

© Shutterstock / Tetsuo Buseru

In dieser zweiteiligen Einführung in Vue.js wollen wir die Funktionsweise des JavaScript-Framework im Detail beleuchten. Im ersten Teil des Tutorials verschaffen wir uns einen Überblick und grenzen Vue.js gegenüber anderen Frameworks ab. In Teil 2 geht es dann zur Sache: Wir zeigen anhand von Code-Beispielen, wie die Entwicklung von Komponenten mit Vue.js funktioniert.

Tutorial Vue.js – Siegeszug eines Frameworks

Den Webanwendungen gehört die Zukunft. Wer sich die Entwicklung im Umfeld der Benutzeroberflächen in den letzten Jahren anschaut, muss eigentlich zu diesem Schluss kommen. Die zunehmende Digitalisierung in allen Lebensbereichen, die explosionsartige Verbreitung sozialer Netzwerke aller Art und das damit einhergehende Bereitstellen von zumeist mobilen, ressourcenschonenden UIs spielt Technologien und Sprachen wie HTML5, CSS und JavaScript positiv in die Hände. Sicherlich wird auch der klassische Rich Client, wie er in Umgebungen wie JavaFX oder der Eclipse Application Platform umgesetzt werden kann, nie ganz verschwinden, und das soll er auch nicht. Doch sind wir ehrlich, müssen wir uns schon eingestehen: Webanwendungen generell haben sich fast überall durchgesetzt. Grund ist die universelle Einsetzbarkeit der zugrunde liegenden Technologien auf so vielen unterschiedlichen Geräten: vom TV-Gerät, dem PC, Notebook, Tablet bis zum Mobiltelefon. Genau in diese Richtung wird es weitergehen.

Einführung in Vue.js: Das Tutorial zum Mitmachen

Teil 1: Einordnung und Überblick

Teil 2: Entwicklung von Komponenten mit Vue.js

Beschäftigt auch mit JavaScript!

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Insbesondere JavaScript profitiert von dieser Entwicklung. In der JavaScript-Welt passiert gerade extrem viel. Innovation und großes Engagement aus den diversen Entwicklercommunities sind klar erkennbar.

Webanwendungen gibt es schon lange, nur waren sie früher eben eher serverzentriert. Große Unternehmensanwendungen, die etwa auf Basis von Java EE umgesetzt wurden, nutzen hierzu oft Technologien wie Servlets, JSP und JavaServer Faces. All diesen Techniken ist gemein, dass sie ihre Ergebnisse (meist HTML-Seiten) serverseitig erzeugen und dieser Output auf dem Client lediglich dargestellt wurde (meist im Browser). Interaktivität auf dem Client? In vielen Fällen Fehlanzeige. Mehr noch: JavaScript war verpönt. Doch die Zeiten haben sich glücklicherweise geändert. Die Entwickler

begannen, ihre bestehenden Anwendungen mit JavaScript zu modernisieren, sie interaktiver und vor allem dynamischer zu gestalten. Die Möglichkeiten des Browsers auf dem Client wurden mehr und mehr mit einbezogen. So toll diese neu gewonnenen Freiheiten auch waren, so führten sie nicht selten zu dem Problem, dass die Codebasis einer solchen mit JavaScript modernisierten Webanwendung komplizierter wurde. Plötzlich tummelte sich viel JavaScript-Code in den unterschiedlichsten Dateien und an den unmöglichsten Stellen im Projekt. Die Wartung einer solchen Anwendung konnte da schon zur Herausforderung werden.

Auf der anderen Seite gingen Unternehmen wie Facebook, Twitter oder Google völlig neue Wege. Sie verlegten sich nämlich darauf, mehr und mehr Bestandteile einer Webanwendung direkt im Browser ablaufen zu lassen. Die Idee rund um die sogenannten Single Page Applications nahm konkrete Formen an. Google Docs oder Gmail sind gute Beispiele für diese Entwicklung.



Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)
Module Federation with Angular

by Manfred Steyer

**React Workshop: Single Page Applications Powered
by React**

by Sebastian Springer

The logo for Angular CAMP, featuring the word "Angular" in white and "CAMP" in red on a red, angular background.

Das 360°-Intensivtraining mit Angular-Koryphäe Manfred Steyer

17. – 19. August 2020 | Remote
30.09. – 02.10.2020 | Remote



Intensiv, nachhaltig & praxisbezogen: Alle Hintergründe zu Building-Blocks in Angular

mit Manfred Steyer (Freiberufler)

Als Online- oder Präsenztraining!

Das 360°-Intensivtraining mit Angular-Koryphäe Manfred Steyer
Präsentiert von Entwickler Akademie

Standards, Struktur und Kontinuität

Aber wie man es dreht und wendet: Egal ob wir Struktur in die mit JavaScript erweiterten Legacy-Webanwendungen bringen oder gleich eine ganze neue Single-Page-Anwendung schreiben möchten die komplett im Browser läuft: Frameworks sind wichtig. Frameworks helfen uns. Sie bringen Struktur, Standards und Kontinuität in unsere Anwendungsentwicklung. Daher ist es auch kein Wunder, dass gerade im innovationsgetriebenen Umfeld von JavaScript bis heute Frameworks wie Pilze aus dem Boden sprießen. Einige Frameworks für die Entwicklung von Webanwendungen auf Basis von

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Versenkung verschwunden. Werkzeugkästen wie Angular oder React sind mittlerweile echte Platzhirsche geworden. Gerade der zuletzt genannte Kandidat ist enorm erfolgreich und genießt immer mehr Beachtung. Warum das so ist, wird im Kasten „Web Components“ näher beleuchtet.

Web Components

Das Internet verändert sich seit 2010 rasant. Firmen wie Facebook, Twitter oder Google gingen dazu über, weitaus mehr als nur Webseiten oder Homepages zu veröffentlichen. Immer ausgereifter wurden deren Produkte. Vollständige Anwendungen, die interaktiv und dynamisch im Browser laufen

und sich nativ anfühlen, sind entstanden. Single Page Applications haben ihren Siegeszug angetreten.

Die Entwickler, die diese Anwendungen umsetzen mussten, standen jedoch vor großen Herausforderungen. Die Programmierung von Webanwendungen fand zu dieser Zeit hauptsächlich auf dem Server statt, JavaScript war verpönt, und der spartanische Satz von HTML-Elementen wie `<p>`, `<form>` oder `<div>` beschränkte die Entwickler zusätzlich. Letztendlich entwickelte man immer wieder neue Kombinationen der genannten Elemente und reicherte sie mit CSS oder JavaScript an. HTML5 hat die Situation zwar verbessert, doch es blieb der grundlegende Bedarf, wiederverwendbare Komponenten manifestiert in eigenen HTML-Elementen entwickeln zu können. Das W3C-Konsortium, bekannterweise verantwortlich für die Pflege und Weiterentwicklung von Standards wie HTML oder CSS, erkannte den Bedarf und stellte eine Erweiterung für HTML in Aussicht: Die Web Components wurden auf den Weg gebracht.

Leider ist es dem W3C über viele Jahre hinweg nicht gelungen, diesen neuen Standard voranzutreiben und unter die Leute zu bringen. Hier wurde eine wichtige Chance verspielt, denn in der Zwischenzeit sind große Firmen wie Google und Facebook aus der Not heraus dazu übergegangen, ihre eigenen Visionen von Web Components umzusetzen. So sind Angular, React und später Vue.js entstanden. All diese Frameworks folgen somit keinem echten Standard und sind keine offizielle Implementierung der Web Components. Sie verbreiten sich aber trotzdem sehr schnell. Wer letztendlich das Rennen

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

noch flächendeckend durchsetzen können wird, bleibt ebenso abzuwarten.

Vue.js – eine Einführung

Mit Vue.js [1] existiert ein weiteres, etwas kleineres Framework, dass zunehmend Aufmerksamkeit auf sich zieht. Es handelt sich dabei um ein Framework für Webanwendungen, mit dem wir auf Basis von JavaScript komponentenorientiert entwickeln können. Eigentlich ganz genauso, wie man es auch mit Angular oder React tun kann. Vue.js wirbt jedoch damit, dass es besonders schlank, anpassungsfähig und universell einsetzbar sei. Die Vue.js-Entwickler beschreiben ihr Framework ganz treffend: „An

approachable, versatile, performant web framework that helps to write maintainable and testable applications.“

Vue.js lässt sich einsetzen, um bestehende Webanwendungen zu modernisieren. Es ist kein Problem, zunächst ganz selektiv nur kleine Teile der Anwendung unter die Kontrolle von Vue.js zu bringen und dann Stück für Stück weitere Teile hinzuzunehmen. Neue Erweiterungen gleich auf Basis von Vue.js umzusetzen, wäre der nächste logische Schritt. Wer das Glück hat, ein ganz neues Projekt starten zu dürfen, kann dieses auf Wunsch auch komplett auf Basis von Vue.js aufbauen. Der Frameworkcharakter von Vue.js stellt dabei sicher, dass der Entwickler sich in einem festgelegten Rahmen bewegt. Dies spiegelt sich in einer nachvollziehbaren Projektstruktur und einem standardisierten, einheitlichen Programmiermodell wieder. Ein großer Mehrwert. All die gerade beschriebenen Möglichkeiten sind eine der Stärken von Vue.js, denn genau hier zeigt sich seine universelle Einsetzbarkeit (approachable, versatile). Wer mehr Geschäftslogik auf dem Client ausführen möchte (Stichwort Single Page Application), kann auch die komplette Anwendung auf Basis von Vue.js aufsetzen. Alle hierzu benötigten Komponenten bringt das Framework in Form der Module Vue Core, Vuex und Vue Router mit. Eine Beschreibung der einzelnen Module findet sich in Tabelle 1.

Es wichtig zu verstehen, dass grundsätzlich nur Vue Core zwingend erforderlich ist, die anderen Module können je nach Bedarf hinzugenommen werden. Um eine Vue.js-Anwendung zu schreiben, ist im

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

JavaScript-Libraries der einzelnen Komponenten von der Vue.js-Webseite [1] herunterladen und über den <Script>-Tag in die eigene Anwendung einbinden. Praktisch macht das heute aber niemand mehr, denn wie es zum Einbinden von Bibliotheken in der Java-Welt Werkzeuge wie Maven oder Gradle gibt, finden sich in der JavaScript-Welt ähnliche Tools. Recht gut durchgesetzt hat sich das Werkzeug npm, der Node Package Manager. Hierbei handelt es sich um ein Kommandozeilenwerkzeug, das automatisch bei der Installation von Node.js zur Verfügung gestellt wird. Wie in Maven die pom.xml die Anwendungskonfiguration und den Build steuert, erstellt npm eine Datei mit dem Namen package.json. In dieser Datei sind ganz ähnliche Informationen zusammengefasst wie in einer pom.xml, allen voran die Abhängigkeiten, die das betreffende Projekt hat, und Informationen

zum Build der Anwendung. Will man ernsthaft Single-Page-Anwendungen mit JavaScript entwickeln, kommt man um ein Werkzeug wie npm nicht herum. Jeds Modul von Vue.js lässt sich mithilfe von npm installieren, wie genau, ist ebenfalls in Tabelle 1 zusammengefasst. Um mit einem neuen Projekt zu beginnen, legen wir im einfachsten Fall einen leeren Projektordner an und initialisieren das Projekt aus diesem Ordner heraus mit dem Kommando

npm init

Nach ein paar Fragen erhalten wir die initiale package.json-Datei. Im Anschluss installieren wir – immer aus dem Projektverzeichnis heraus – die benötigten JavaScript-Module aus Tabelle 1. Die Module werden dann automatisch aus dem Internet geladen, im Projekt abgelegt, und es wird ein entsprechender Eintrag in die package.json eingefügt. Noch bequemer geht das Erstellen eines neuen Vue.js-Projekts jedoch mit dem Werkzeug vue-cli. Dieses lässt sich ebenfalls mit NPM installieren und bringt eine ganze Reihe von vorgefertigten Projektvorlagen mit, aus denen man sein Projekt erzeugen kann. Mit dem Befehl

vue init <Vorlagen-Name> <Projekt-Name>

wird dann ganz automatisch das Projektverzeichnis auf Basis der Vorlage angelegt, die entsprechenden Abhängigkeiten werden im Hintergrund heruntergeladen, Beispiel Quellcode wird hinzugefügt, und die Entwicklung kann losgehen. Auf der GitHub-Seite von vue-cli [2] findet sich eine recht

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

es sich, mit dem Template simple zu beginnen. In größeren Projekten wird man sicherlich irgendwann auch einen Bundle Manager wie webpack [3] hinzuziehen, daher sollte sich ein Blick auf die beiden Vorlagen webpack simple und webpack lohnen.

Modul	Beschreibung	Installation
Vue Core	Die Kernkomponente, die wir immer benötigen, wenn wir eine Anwendung auf Basis von Vue.js schreiben möchten. Implementiert das Vue-Objekt mit Data Binding und bringt alle Werkzeuge mit, um	<code>npm install --save vue</code> <code><script src="/path/to/vue.js"></code> <code></script></code>

	Komponenten entwickeln zu können.	
Vuex	<p>Vuex ist ein State Manager. Er hält die Komponenten einer Anwendung zusammen und kontrolliert ihren Zustand, verwaltet Daten und tauscht sie zwischen den Komponenten aus. Vuex lässt sich mit der JavaScript-Bibliothek Redux vergleichen, die im Umfeld von React oft eingesetzt wird. Im Detail unterscheiden sich die beiden State Manager jedoch. In Vuex lassen sich Daten nur nach bestimmten Regeln verändern oder wenn sich die Komponenten in einem festgelegten Zustand befinden. Ferner charakterisiert Vuex der gleiche reaktive Charakter wie im Core auch. Das heißt, ändern sich Daten innerhalb des State Managers, werden verschiedene Lifecycle-Methoden durchlaufen, in die sich der Entwickler einklinken kann.</p>	<pre>npm install --save vuex</pre> <pre><script src="/path/to/vuex.js"></pre> <pre></script></pre>
Vue Router	<p>Wer eine Single Page Application programmiert, benötigt für den Wechsel zwischen den einzelnen Seiten einen sogenannten Router. Vue Router stellt an sich den Anspruch, extrem einfach in der Bedienung zu sein. Hierzu</p>	<pre>npm install --save vue-router</pre> <pre><script src="/path/to/</pre> <pre>vue-router.js"></script></pre>

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

	<p>einem JavaScript-Objekt konfigurieren, in dem die Routen und deren Verknüpfung zu den dahinterliegenden Komponenten hinterlegt sind, die beim Aktivieren einer Route angezeigt werden sollen.</p>	
Vue CLI	<p>Kommandozeilenwerkzeug, um neue Vue.js-Projekte aus Vorlagen zu erstellen. Am besten beginnt man mit der Vorlage simple und macht dann mit den webpack-Vorlagen weiter. Es besteht auch die Möglichkeit, eigene Projektvorlagen zu definieren und einzubinden.</p>	<pre>npm install -g vue-cli</pre>

Tabelle 1: Die wichtigsten Module und Bestandteile von Vue.js

Die Ursprünge von Vue.js

Im Gegensatz zu Angular und React, hinter denen große Firmen stehen, wurde Vue.js von einer Einzelperson konzeptioniert und entwickelt. Der ursprüngliche Entwickler, Evan Yu, war zunächst bei Google angestellt und beschäftigte sich dort mit den unterschiedlichsten Webtechnologien. Insbesondere von Chrome Experiments [4] war er fasziniert, weil er hier erkannte, welches Potenzial im Browser generell und insbesondere in der Programmiersprache JavaScript steckt. Er brachte sich JavaScript selbst bei und arbeitete schließlich im Entwicklungsteam von Angular mit. Bei Angular begeisterten ihn die Möglichkeiten rund um das Data Binding und die Art und Weise, wie man mit dem DOM den Browser umgehen, ihn beeinflussen und verändern kann, ohne direkt auf ihn zugreifen zu müssen. Evan fragte sich, ob es möglich wäre, nur einen ganz kleinen zentralen Teil aus Angular zu extrahieren, den Teil, der ihn wirklich interessierte: Deklaratives Data Binding. Er wollte den Angular-Quellcode aber nicht kopieren, sondern ihn selbst gestalten und implementieren. So startete er, zunächst nur für sich selbst, ein kleines Experiment, um ein solches deklaratives Data Binding auf die Beine zu stellen. Das war der Beginn von Vue.js. Evan arbeitete an diesem Experiment und spürte instinktiv, dass das, was er da schuf, gut sein musste und Potenzial hatte, weil er es selbst gerne einsetzte. Im Februar 2014 veröffentlichte er die erste Version auf GitHub. Bereits in der ersten

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Downloadzahlen gingen rasant nach oben.

Abgrenzung zu Angular und React

Vergleicht man Vue.js mit anderen Frameworks wie Angular oder React, wird man vermutlich zu dem Schluss kommen, dass es wohl am meisten Ähnlichkeit mit React hat. Evan Yu bezeichnet es auch als progressives Framework. Die Idee dahinter ist, dass der Vue Core lediglich aus einer Implementierung für Data Binding besteht und darüber hinaus die Möglichkeit bietet, Komponenten zu programmieren. Das wars. Vue.js stellt an sich selbst also nur den Anspruch, ganz spezielle Probleme lösen zu wollen, sich zu fokussieren – im direktem Vergleich zu React jedoch mit einem sehr viel größeren Blick auf universelle Einsetzbarkeit. Für Vue.js

reicht es im Prinzip, wenn der Anwender HTML, JavaScript und CSS beherrscht. Er kann dann direkt loslegen. In React dagegen wird er um das Erlernen spezieller Erweiterungen wie etwa JSX nicht herumkommen.

Das Vue.js-Framework wurde ganz bewusst sehr schlank gehalten, während sich andere Frameworks wie beispielsweise Angular sehr breit aufstellen und quasi für fast jedes Problem, das im Rahmen der Anwendungsentwicklung auftreten kann, eine Lösung an Bord hat. Vue.js legt seinen Fokus auf Schlankheit und Performanz, zusätzliche Frameworkeigenschaften wie Routing oder State Management können in Form von Modulen bei Bedarf hinzugenommen werden.

Letztendlich lässt sich auch gar nicht sagen, welches Framework das beste ist. Egal ob Angular, React oder Vue.js – jeder Werkzeugkasten hat seine individuellen Stärken und Schwächen, und jedes Entwicklungsteam muss selbst das richtige Paket finden. Evan Yu sagt über sich selbst in diesem Zusammenhang, dass er sich dem Druck, sich mit anderen messen zu müssen, ganz bewusst entzieht. Es sei ihm lediglich wichtig, für die Entwickler, die sein Framework schätzen und es gerne nutzen, einen guten Job zu machen und sie zufriedenzustellen.

Progressives und Reaktiv

Um einen Überblick über Vue.js zu erhalten, empfiehlt es sich, ein erstes Projekt mit der Vorlage simple und dem Kommandozeilenwerkzeug vue-cli zu erstellen:

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Diese Vorlage erzeugt einen Projektordner vue-test und legt dort lediglich eine index.html ab. Nehmen wir dagegen beispielsweise die Vorlage webpack-simple, werden deutlich mehr Dateien erzeugt, etwa eine package.json mit einigen Abhängigkeiten und eine webpack-Konfiguration. Im Projekt vue-test dagegen gibt es keine package.json, die Abhängigkeit zur Vue.js-Bibliothek wird hier über eine einfache <script>-Direktive aufgelöst. Schaut man sich die index.html an, bekommt man dennoch einen recht guten Einblick, wie Vue.js grundsätzlich funktioniert.

Von zentraler Bedeutung ist hier das globale Vue-Objekt, das Data Binding implementiert. Es kann ein oder mehrere Vue-Objekte geben. Ein Vue-

Objekt wird an ein HTML-Element gebunden und übernimmt ab diesen Zeitpunkt die Kontrolle über dieses Element beziehungsweise diesen Bereich im DOM des Browsers. Das Vue-Objekt wird in der Form konfiguriert, dass ein JavaScript-Objekt übergeben wird, das eine überschaubare Anzahl vordefinierter Properties enthalten muss: `el` und `data`. Ferner lassen sich benutzerdefinierte Funktionen und sogenannte Lifecycle-Methoden hinzufügen. Mit der Property `el` binden wir das Vue-Element an das HTML-Element. Im Beispielprojekt wird das Vue-Objekt an das `<div>`-Element mit der ID `app` gebunden. Von zentraler Bedeutung ist die `data`-Property. Hier liegen alle Daten, die von Vue.js verwaltet werden. Sobald hier Daten vorliegen, werden sie im Reactivity System von Vue.js mit einbezogen, was bedeutet, dass Vue.js bei jeder Änderung des Datenbestands eine automatische Aktualisierung der betroffenen HTML-Elemente, die in Zusammenhang mit den Daten stehen, im DOM auslöst.

Funktionen mit eigener Geschäftslogik (im Beispiel die Funktion `humanizedURL`) lassen sich durch Einfügen als Element der Auflistung `methods` einbringen. Dies können beliebig viele Funktionen sein. Öffnen wir im Browser (Chrome, Firefox) die Entwicklungstools, haben wir zur Laufzeit die Möglichkeit, über die JavaScript-Konsole das Vue-Objekt zu analysieren, zu verändern, Methoden darauf aufzurufen etc. Das vereinfacht den Umgang und die Entwicklung von Vue.js-Anwendungen ungemein. Im Beispielprojekt wird das Vue-Objekt in einer Variable `app` gehalten, folglich können wir zur Laufzeit in der JavaScript-Konsole direkt auf das `app`-Objekt zugreifen. Die Bestandteile der Daten (`data`) sind direkt als Properties im

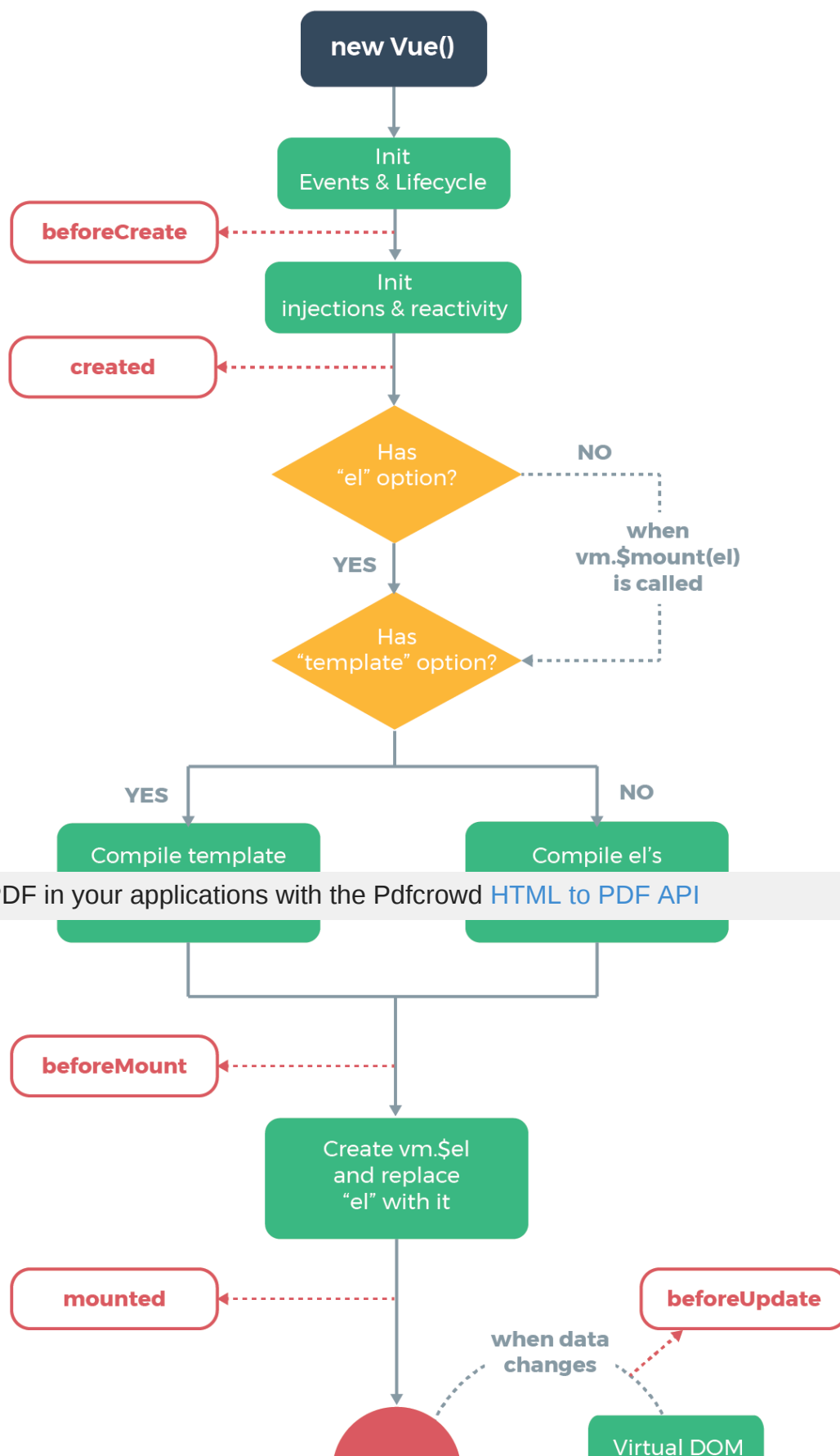
Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

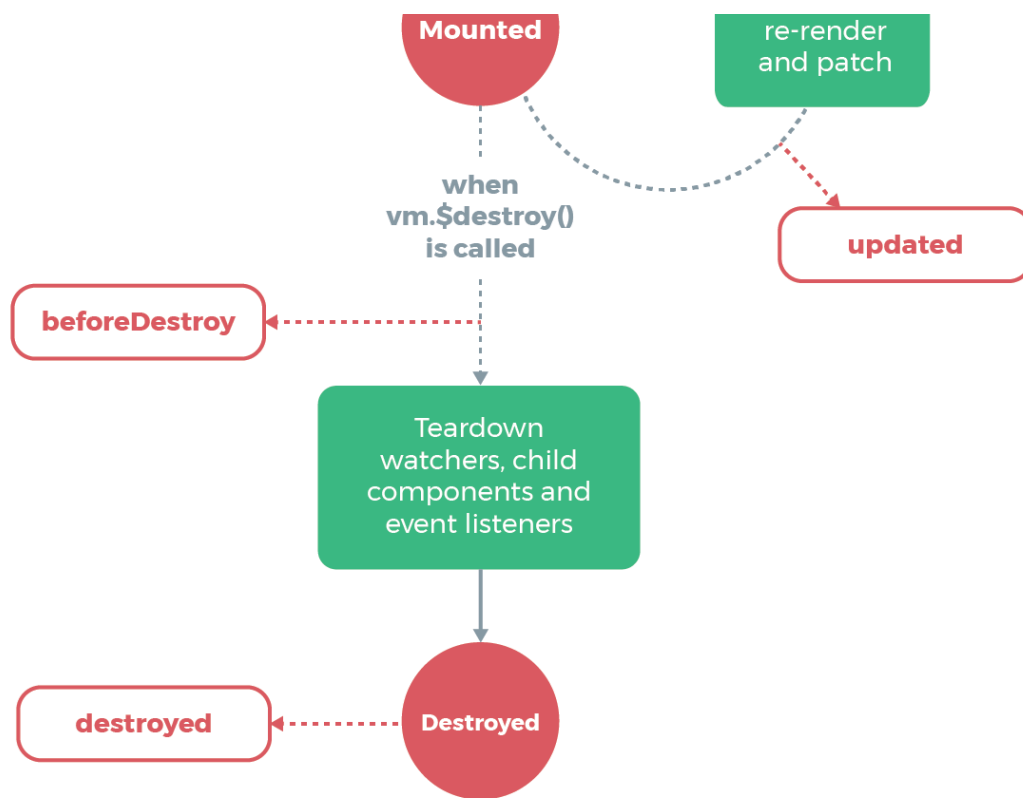
auch die eigenen Methoden stehen zur Verfügung, die Methode `app.humanizedURL` lässt sich problemlos direkt aufrufen.

Lifecycle und Watchers

Es geht noch besser. Wenn ein Vue-Objekt erzeugt und gestartet wird, durchläuft es einen Lebenszyklus, wie in Abbildung 1 dargestellt. Wichtig dabei ist zu verstehen, dass man sich in jeden Moment dieses Lebenszyklus mit den bereits erwähnten Lifecycle-Methoden (in der Abbildung rot dargestellt) einklinken kann. So kann der Anwender die Methode `mounted` verwenden, um sich informieren zu lassen oder zu reagieren, wenn das Vue-Objekt (oder die Vue-Komponente) fertig initialisiert ist. Das kann

nützlich sein, wenn er im Anschluss eine eigene Initialisierung vornehmen möchte. Auch für die Veränderung von Daten gibt es entsprechende Lifecycle-Methoden. Die Methoden `beforeUpdate` und `updated` könnten hier interessant sein.





* template compilation is performed ahead-of-time if using a build step, e.g. single-file components

Abb. 1: Der Vue.js-Lebenszyklus (Quelle: Offizielle Vue.js-Dokumentation)

Wie bereits erwähnt, lassen sich die Daten von Vue.js-Instanzen als einfache Properties anzeigen. Wir können sie aber auch verändern bzw. neu setzen, was zur Folge hat, dass der in der Abbildung dargestellte Lifecycle entweder erneut komplett oder teilweise durchlaufen wird. Würden wir in der Konsole z. B. `app.greeting = „Hallo Welt“` setzen, ändern sich die Daten auf der

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Methoden `beforeUpdate` und `updated` werden aufgerufen.

Vue.js-Instanzen bieten noch eine Reihe weiterer interessanter Instanz-Properties und -Methoden an, über die sich der Zustand der Vue.js-Instanz überwachen und abfragen lässt. Diese speziellen Properties und Methoden beginnen alle grundsätzlich mit dem Präfix `$`, damit sie von den benutzerdefinierten Daten und Methoden deutlich abgegrenzt sind. Zwei Beispiele: `app.$data` würde das komplette Data-Objekt der Instanz ausgeben und `app.$el` das HTML-Element, an dem diese Vue.js-Instanz gebunden ist. Hier empfiehlt es sich, etwa im Chrome-Browser in der Konsole der Developer Tools, einmal `app.$` einzugeben und auf die

Codevervollständigung zu warten. Es ist schon erstaunlich und interessant, was hier alles angeboten wird.

Schließlich lassen sich auch Watcher an eine Vue.js-Instanz binden. Dies erfolgt über die Hilfsmethode `$watch`:

```
app.$watch('greeting', function (newValue, oldValue) {  
  
  console.log('Old=' + oldValue + ', New=' + newValue);  
  
});
```

Das bietet dem Entwickler die Möglichkeit, sich über jede Veränderung, jeden Aufruf einer bestimmten Property oder Methode der Vue.js-Instanz informieren zu lassen und individuell zu reagieren. Der Watcher oben wird beispielsweise immer dann aufgerufen, wenn sich die entsprechenden Daten verändern.

Vue.js Tutorial – Teil 1: Zwischenfazit

Den Webanwendungen gehört die Zukunft. Deshalb ist man gut beraten, sich mit JavaScript zu beschäftigen. Diese Sprache ist weiter auf dem Vormarsch und mit Frameworks wie Angular, React oder Vue.js stehen Werkzeugkästen bereit, mit denen sich skalierbare, erweiterbare, performante und komponentenbasierte Webanwendungen schreiben lassen.

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

In diesem Tutorial haben wir die Grundlagen von Vue.js behandelt, gezeigt, wo das Framework herkommt und wie es sich zwischen den Platzhirschen Angular und React positioniert. Wir haben den grundlegenden Aufbau von Vue.js-Anwendungen beleuchtet und dessen Grundkonzepte dargestellt. Wie in den anderen Webframeworks, ist es natürlich auch mit Vue.js möglich, wiederverwendbare und eigenständige Komponenten zu entwickeln. Genau wie Angular und React folgt Vue.js hier nicht dem offiziellen Web-Components-Standard, sondern bringt eine eigene, ganz individuelle Implementierung von Komponenten mit. Im zweiten Teil dieser Einführung werden wir sehen, wie sich komponentenorientiert mit Vue.js entwickeln lässt.

Hier gehts zu Teil 2 unseres Vue.js Tutorials: **So entwickelt man Komponenten mit Vue.js**

Links:

[1] Offizielle Vue.js-Webseite: <https://vuejs.org/>

[2] vue-cli-Kommandozeilenwerkzeug: <https://github.com/vuejs/vue-cli>

[3] webpack: <https://webpack.js.org/>

[4] Chrome Experiments: <https://experiments.withgoogle.com/chrome>

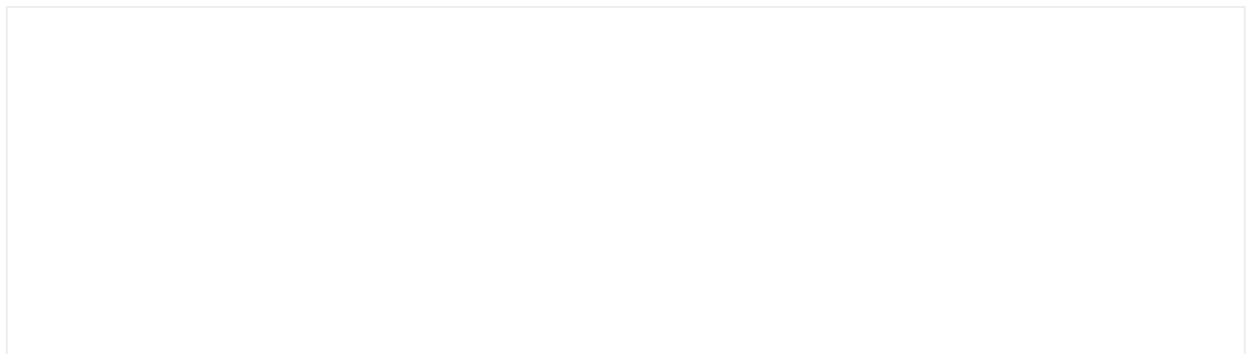


DOSSIER VUE.JS

JAVASCRIPT

VUE

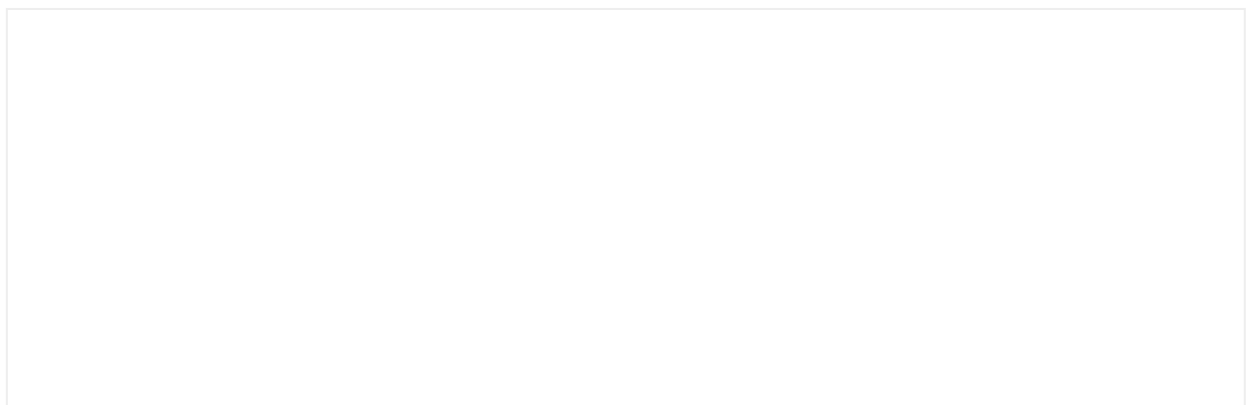
RELEVANTE BEITRÄGE



Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

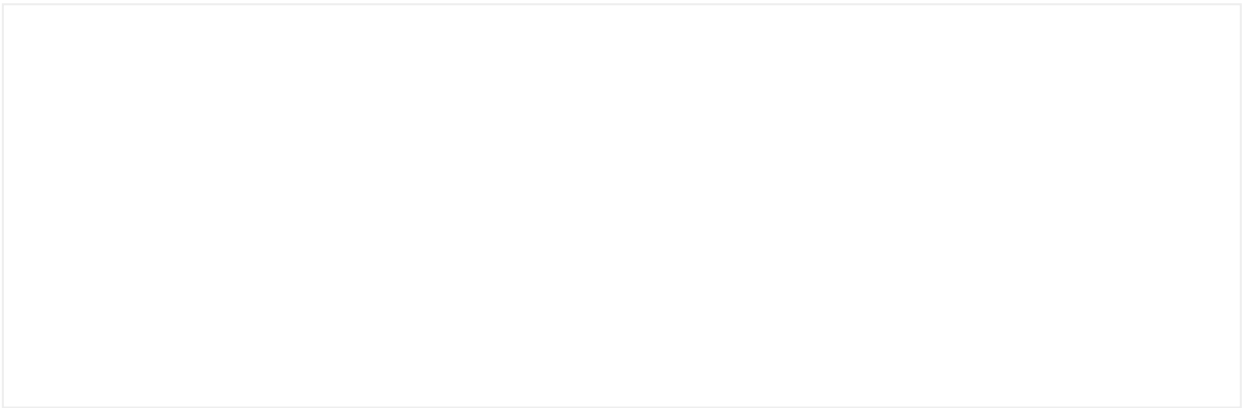
Progressive Web Apps ohne Framework: Geht das?

Von **Redaktion** / 2 Tagen online



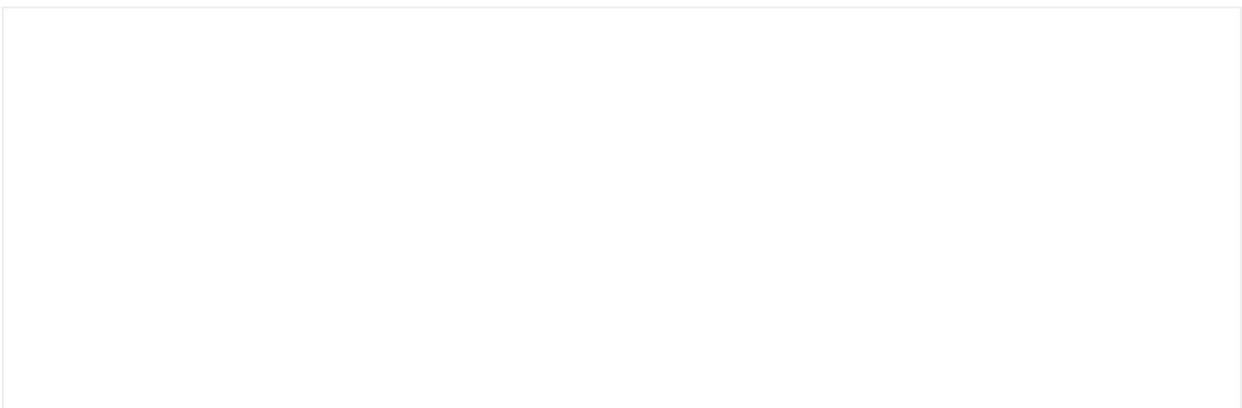
Vue.js, ECMAScript und ActiveJ – Unsere Highlights der letzten Woche

Von [Jean Kiltz](#) / 3 Tagen online



International JavaScript Conference London – The Online Conference: Bis 30. Juli anmelden und sparen

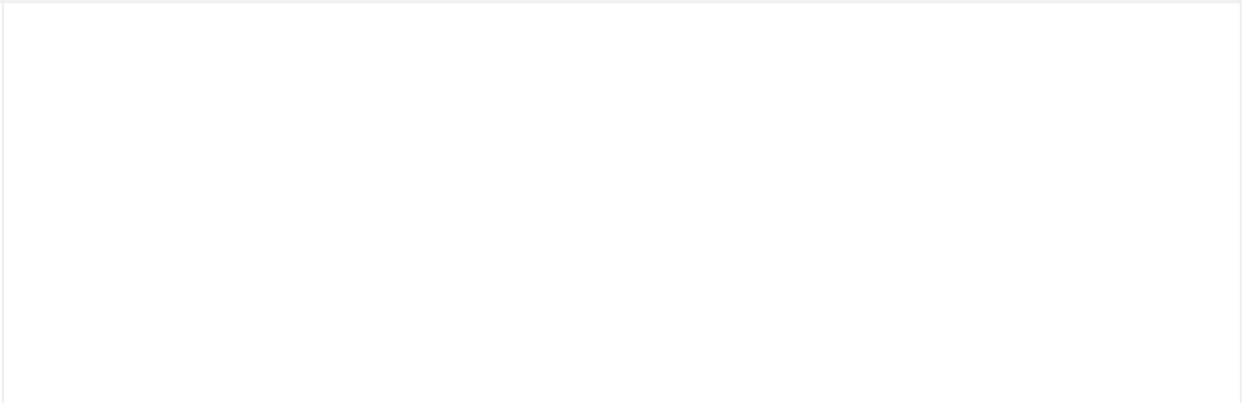
Von [Redaktion](#) / 3 Tagen online



TypeScript-Tipps für React-Entwickler

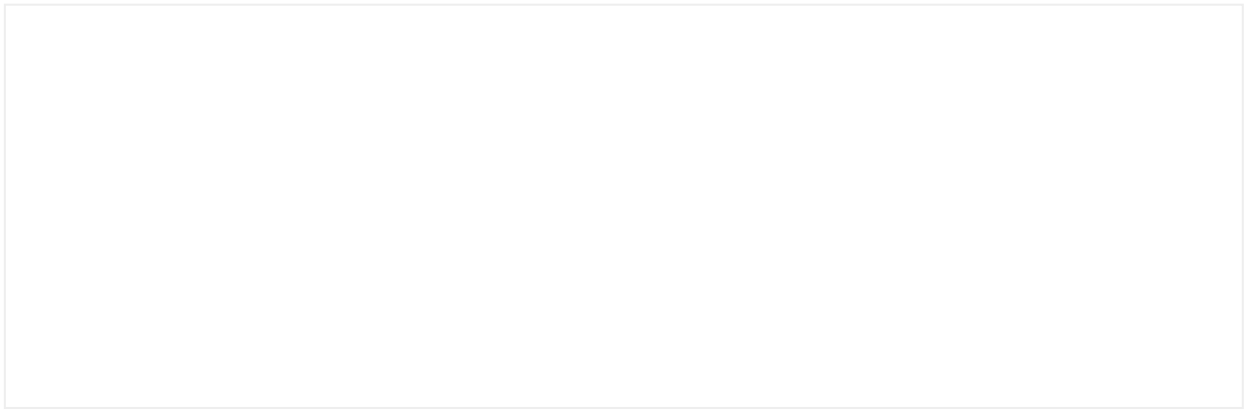
Von [Redaktion](#) / 1 Woche online

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)



ECMAScript 2021: Vier weitere Features stehen fest

Von [Ann-Cathrin Klose](#) / 1 Woche online



Vue.js 3: Release Candidate veröffentlicht

Von [Ann-Cathrin Klose](#) / 1 Woche online

Hinterlasse einen Kommentar

1 Kommentar auf "Vue.js Tutorial: Einführung in das JavaScript-Framework"

Beteilige dich an der Diskussion

✉ Subscribe ▼



Gast

Martin



Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

Einwandfrei. Danke für die Zusammenfassung und den Einstieg. Gerade bei vorhandenen Projekten welche mit jQuery umgesetzt wurden kann man schnell und einfach Vue integrieren und alten Code umstellen.

Antworten

🕒 2 Jahre her

ONLINE

entwickler.de

JAXenter.de

JAXenter.com

MAGAZINE

Entwickler Magazin

PHP Magazin

Windows Developer

Mobile Technology

Cloud Compendium

JavaScript Compendium

Java Magazin

Business Technology

KONFERENZEN

JAX

Create PDF in your applications with the Pdfcrowd [HTML to PDF API](#)

International PHP Conference

International JavaScript Conference

webinale

MobileTech Conference

Internet of Things Conference

DevOpsCon

API Conference

Machine Learning Conference

SERVICE

[Leserservice & Kontakt](#)

[Anzeigenservice & Werbung](#)

[Autor werden](#)

[Digital lesen](#)

[Social Media](#)

[Entwickler Akademie](#)

[Über S&S Media](#)

[Software & Support Media GmbH](#)

[Impressum](#)

[Datenschutz](#)

[Datenschutz-Shop](#)

[AGB](#)

[Lieferkonditionen](#)