Web Components v1 - the next generation

By Taylor Savage

Taylor is a PM on the Chrome Team, focusing on Web Components and Polymer.

Ever wanted to build your own self-contained JavaScript component, that you can easily use across multiple projects or share with other developers regardless of what JavaScript framework they use? Web Components may be for you.

Web Components are a set of new web platform features that let you create your own HTML elements. Each new custom element can have a custom tag like `<my-button>`, and have all the goodness of built-in elements - custom elements can have properties and methods, fire and respond to events, and even have an encapsulated style and DOM trees to bring along their own look and feel.

```
<paper-progress></pa
```

By providing a platform-based, low-level component model, Web Components let you build and share re-usable elements for everything from specialized buttons to complex views like datepickers. Because Web Components are built with platform API's which include powerful encapsulation primitives, you can even use these components within other JavaScript libraries or frameworks as if they were standard DOM elements.

You may have heard of Web Components before - an early version of the Web Components spec - v0 - was shipped in Chrome 33.

Today, thanks to broad collaboration between browser vendors, the next-generation of the Web Components spec - v1 - is gaining wide support. Chrome supports the two major specs that make up Web Components - Shadow DOM and Custom Elements - as of Chrome 53 and Chrome 54 respectively. Safari shipped support for Shadow DOM v1 in Safari 10, and has completed implementation of Custom Elements v1 in WebKit. Firefox is currently developing Shadow DOM and Custom Elements v1, and both Shadow DOM and Custom Elements are on the roadmap for Edge.

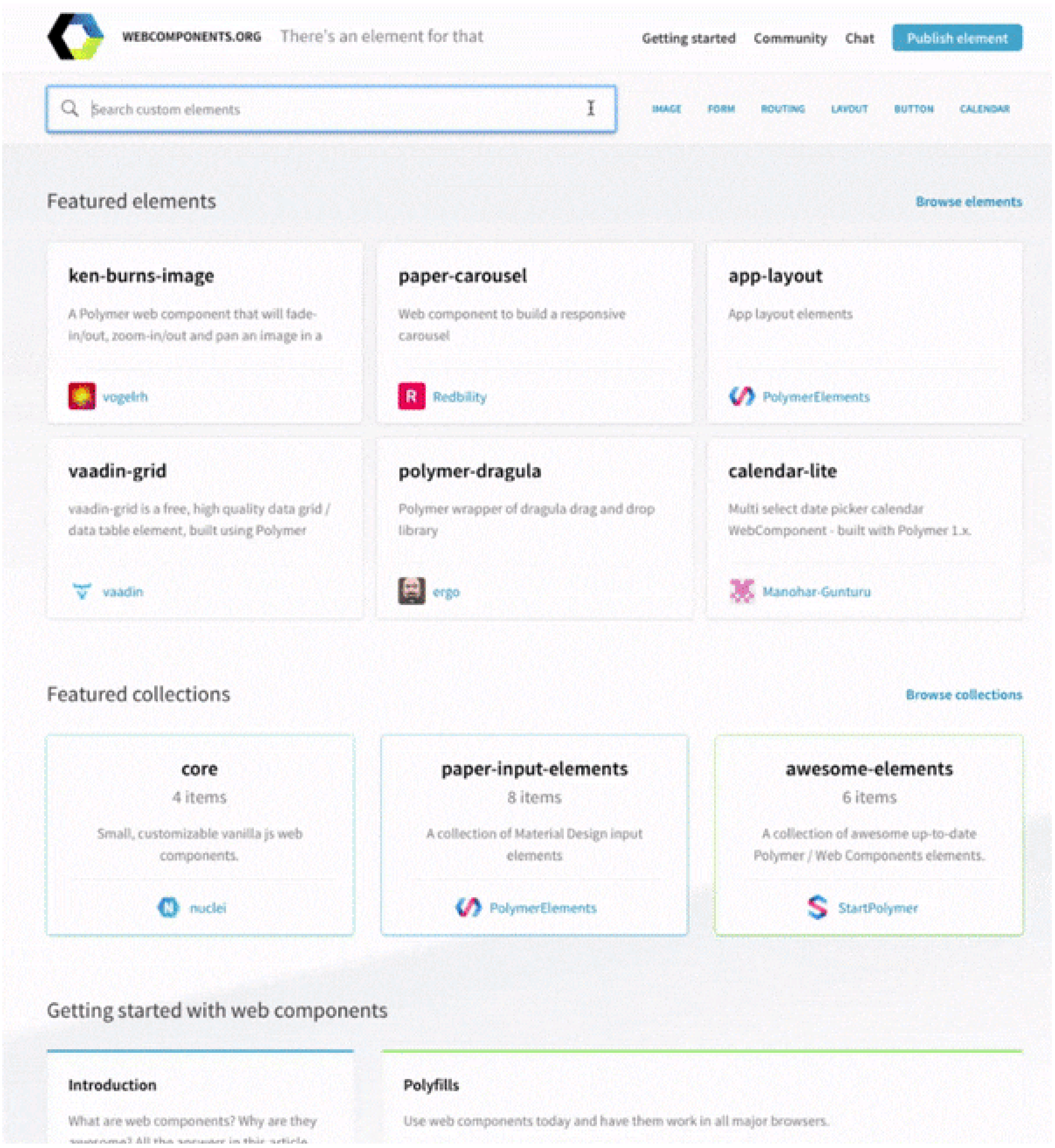
To define a new custom element using the v1 implementation, you simply create a new class that extends `HTMLElement` using ES6 syntax and register it with the browser:

```
class MyElement extends HTMLElement {...}
window.customElements.define('my-element', MyElement);
```

The new v1 specs are extremely powerful - we've put together tutorials on using Custom Elements v1 and Shadow DOM v1 to help you get started.

webcomponents.org

The Web Component community is also growing in leaps and bounds. We're excited to see the launch of an updated webcomponents.org site - the focal point of the web components community - including an integrated catalog for developers to share their elements.


WEBCOMPONENTS.ORG There's an element for that

Getting started Community Chat Publish element

Search custom elements

IMAGE FORM ROUTING LAYOUT BUTTON CALENDAR

Featured elements

Browse elements

ken-burns-image
A Polymer web component that will fade-in/out, zoom-in/out and pan an image in a
vogelrh

paper-carousel
Web component to build a responsive carousel
Redbility

app-layout
App layout elements
PolymerElements

vaadin-grid
vaadin-grid is a free, high quality data grid / data table element, built using Polymer
vaadin

polymer-dragula
Polymer wrapper of dragula drag and drop library
ergo

calendar-lite
Multi select date picker calendar WebComponent - built with Polymer 1.x.
Manohar-Gunturu

Featured collections

Browse collections

core
4 items
Small, customizable vanilla js web components.
nuclei

paper-input-elements
8 items
A collection of Material Design input elements
PolymerElements

awesome-elements
6 items
A collection of awesome up-to-date Polymer / Web Components elements.
StartPolymer

Getting started with web components

Introduction
What are web components? Why are they awesome? All the answers in this article

Polyfills
Use web components today and have them work in all major browsers.

The webcomponents.org site contains information about the Web Components specs, updates and content from the web components community, and displays documentation for open-source elements and collections of elements built by other developers.

You can get started building your first element using a library like Google's Polymer, or just use the low-level Web Component API's directly. Then publish your element on webcomponents.org.

Happy componentizing!

Last updated July 2, 2018.