Polymer at Google I/O 2016

At Google I/O 2016, we launched the brand-new Polymer App Toolbox for building Progressive Web Apps.

Polymer Team , 2016-05-19

The mission of the Polymer Project is to make it easy to use the cutting-edge web platform to build modern, high-quality applications.

The web was not initially designed for these kinds of apps. HTML provided a set of tags that were great for marking up documents, but not so great for building applications.

Over the years, we've used JavaScript to fill this gap. We've invented clever models for structuring our apps, built libraries and frameworks to implement them, and layered lots of other useful features on top. We've also constructed elaborate tool chains to build and serve and bundle and split our JavaScript code. These innovations have filled the gaps in the web platform, but they've also added cost in the form of complexity, developer lock-in and JavaScript overhead.

This overhead was a fine tradeoff for the desktop web, where big beefy machines with fast CPUs are connected via wire or WiFi to the internet. But the advent and reach of the *mobile* web fundamentally changes the equation. Flaky, slow data connections are the norm. Underpowered CPUs and tight on-device memory are the norm. And as the next billion users come online, many of them are coming online with the most challenging combinations of data and device.

Fortunately, the web platform is evolving to meet these challenges:

- Thanks to Web Components, developers have a way to extend HTML itself: to unlock the powerful component model that's baked directly into the browser and build an application out of granular, low-overhead components.
- Thanks to HTTP/2, developers have a way to efficiently deliver granular dependencies over the network without incurring the high overhead of multiple round-trips.
- Thanks to Service Worker, developers have a way to reliably cache these granular components, to ensure their web applications can still function and perform reliably even in flaky network conditions.

Today's web platform is a first-class platform for developing and delivering apps. The Polymer Project's goal is to make it easy to leverage all of these new features to build modern web applications with minimal overhead and maximal performance.

Today at Google I/O, we announced a number of new updates and tools for effectively building modern Progressive Web Apps, by using what the platform provides to achieve great performance.

Web Component progress: Custom Elements and Shadow DOM v1

If you have been following the Web Component specs, you know it has taken a long time to grow broad cross-browser support for the new APIs.

But thanks to deep collaboration between all major browser vendors, the specs have reached a state where they are seeing wide browser agreement—with the latest "v1" versions of Custom Elements and Shadow DOM.

And implementation of these v1 specs is already well underway. Firefox is nearing an implementation of Shadow DOM v1, Chrome is well underway with implementations of Custom Elements and Shadow DOM v1, and Safari has implemented Shadow DOM v1 in nightly and is prototyping the Custom Elements v1 implementation.

Broad native browser support for Web Components is no longer a matter of *if* but a matter of *when*, and *when* could be as soon as later this year!

The Polymer App Toolbox

The Polymer App Toolbox is a set of loosely-coupled components and tools to make it easy to create a Progressive Web Application using the modern platform:

- app-layout elements for responsive UI structure.
- app-route for mapping URL's to views.
- app-localize-behavior for translating and localizing UI strings.
- app-storage for turnkey integration between any data store and Polymer's idiomatic data-binding system.
- Polymer CLI as a simple command-line tool to support the entire development lifecycle, from scaffolding out a project to building it for production.

The PRPL Pattern

Three cutting-edge new features of the web platform—Web Components, HTTP/2 + Server Push, and Service Worker—all work seamlessly together to provide a totally new and amazingly efficient way to deliver applications to users.

We call this the "PRPL Pattern", and the Polymer App Toolbox and Polymer CLI make it easy to build an application to use this strategy for delivery. The PRPL pattern stands for:

- **Push** components critical for initial route
- **Render** the initial route ASAP
- **Pre-cache** components for remaining routes
- **Lazy-load** and create next routes on-demand

Shop App

Using the Polymer App Toolbox and the PRPL Pattern, it is possible to create complex Progressive Web Apps that start fast and stay fast.

The Polymer "Shop App" is built using the toolbox and served using the PRPL pattern, and demonstrates techniques for constructing a modern, multi-view Progressive Web App.

`carbon` elements become `app` elements

As part of this launch, the previously-announced `carbon-*` component product line has become the "app" product line. There was only one released `carbon` element—`carbon-route`—which has been renamed with a major version bump to `app-route`.

This decision was not made lightly: we had received a plethora of feedback that the "Periodic Table" component naming was confusing, especially for new Polymer developers. We wanted to take the opportunity with a new product line to align with a clearer, more straightforward name.

Polymer Summit 2016

Last but not least, we announced the next Polymer Summit! We'll be following up last year's Polymer Summit in Amsterdam with an event this fall in London.

To be notified when registration opens, sign up with your email address at g.co/polymersummitinfo.

We hope to see you there - and in the meantime, happy componentizing!