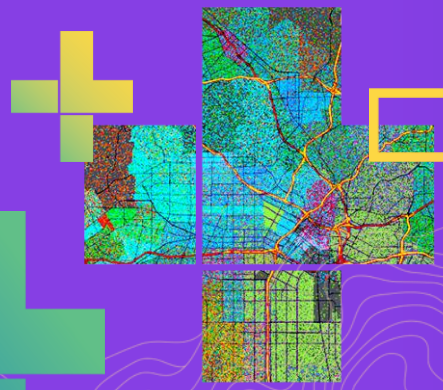
esri | THE SCIENCE OF WHERE™

Developer Primer: Svelte

Support and Services Spotlight Theater

Gavin Rehkemper @gavinrehkemper
Jacob Wasilkowski @jwasilgeo

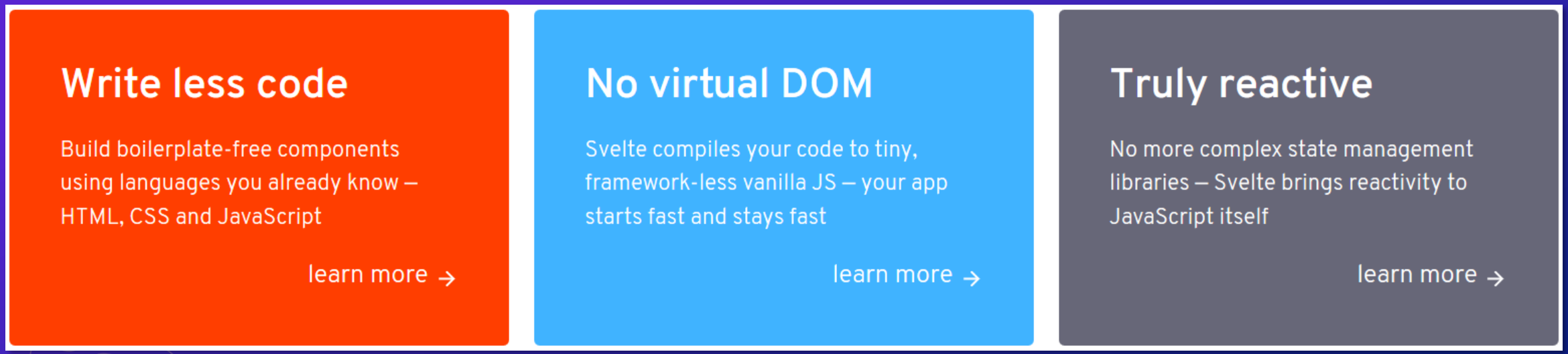
2020 ESRI DEVELOPER SUMMIT | Palm Springs, CA



Svelte: “Cybernetically enhanced web apps”

Bold claims

“Svelte is a radical new approach to building user interfaces. Whereas traditional frameworks like React and Vue do the bulk of their work in the *browser*, Svelte shifts that work into a *compile step* that happens when you build your app.”



Components

Component structure: JS, HTML, & CSS with a few helpers

Trigger efficient, granular updates by assigning to local variables. The compiler does the rest.

App.svelte

```
1  <script>
2    let count = 0;
3
4    function handleClick() {
5      count += 1;
6    }
7  </script>
8
9  <button on:click={handleClick}>
10   Clicked {count} {count === 1 ? 'time' : 'times'}
11 </button>
```

Result | JS output | CSS output

Clicked 0 times

Console CLEAR

Component HTML *logic* helpers: if / else blocks

App.svelte

```
<script>
  let user = { loggedIn: false };

  function toggle() {
    user.loggedIn = !user.loggedIn;
  }
</script>

{#if user.loggedIn}
  <button on:click={toggle}>
    Log out
  </button>
{:else}
  <button on:click={toggle}>
    Log in
  </button>
{/if}
```

Result | JS output | CSS output

Log in

Console CLEAR

Component HTML *logic* helpers: each blocks

App.svelte

```svelte
<script>
  let cats = [
    { id: 'J---aiyznGQ', name: 'Keyboard Cat' },
    { id: 'z_AbfPXTKms', name: 'Maru' },
    { id: 'OUtn3pvWmpg', name: 'Henri The Existential Cat' }
  ];
</script>

<h1>The Famous Cats of YouTube</h1>

<ul>
  {#each cats as { id, name }, i}
    <li><a target="_blank" href="https://www.youtube.com/watch?v={id}">
      {i + 1}: {name}
    </a></li>
  {/each}
</ul>
```

Result | JS output | CSS output

The Famous Cats of YouTube

- 1: Keyboard Cat
- 2: Maru
- 3: Henri The Existential Cat

Console

Component HTML *logic* helpers: await promise blocks

App.svelte

```svelte
<script>
  let promise = getRandomNumber();

  async function getRandomNumber() {
    const res = await fetch(`tutorial/random-number`);
    const text = await res.text();

    if (res.ok) {
      return text;
    } else {
      throw new Error(text);
    }
  }

  function handleClick() {
    promise = getRandomNumber();
  }
</script>

<button on:click={handleClick}>
  generate random number
</button>

{#await promise}
  <p>...waiting</p>
{:then number}
  <p>The number is {number}</p>
{:catch error}
  <p style="color: red">{error.message}</p>
{/await}
```

Result | JS output | CSS output

generate random number

The number is 15

Console CLEAR

Props

~~Props~~

Props

App.svelte Nested.svelte +

```
1  <script>
2    import Nested from './Nested.svelte';
3  </script>
4
5  <Nested/>
```

Result JS output CSS output

Props

App.svelte Nested.svelte × +

```
1  <script>
2    let answer = 'DEV SUMMIT!';
3  </script>
4
5  <p>The answer is {answer}</p>
```

Result

The answer is DEV SUMMIT!

Props

App.svelte Nested.svelte × +

```
1  <script>
2    export let answer;
3  </script>
4
5  <p>The answer is {answer}</p>
```

Result JS output CSS output

The answer is undefined

Props

App.svelte | Nested.svelte

```
1  <script>
2    import Nested from './Nested.svelte';
3  </script>
4
5  <Nested answer={42}/>
```

Result | JS output | CSS output

The answer is 42

Events

- Remember REACTIVITY (from before) …

SAME Syntax for your own components!

App.svelte | Inner.svelte

```svelte
<script>
  import Inner from './Inner.svelte';

  function handleMessage(event) {
    alert(event.detail.text);
  }
</script>

<Inner on:message={handleMessage}/>
```

Result | JS output | CSS output

But how to I emit the event?

App.svelte Inner.svelte × +

```
1  <script>
2    import { createEventDispatcher } from 'svelte';
3
4    const dispatch = createEventDispatcher();
5
6    function sayHello() {
7      dispatch('message', {
8        text: 'Hello!'
9      });
10   }
11 </script>
12
13 <button on:click={sayHello}>
14   Click to say hello
15 </button>
```

Result JS output CSS output

Click to say hello

Stores

stores.js

```
1  import { writable } from 'svelte/store';
2
3  export const count = writable(0);
```

App.svelte stores.js +

```
1  <script>
2    import { count } from './stores.js';
3    let count_value;
4    const unsubscribe = count.subscribe(value => {
5      count_value = value;
6    });
7  </script>
8
9  <h1>The count is {count_value}</h1>
```

Result JS output CSS output

The count is 0

App.svelte stores.js

```svelte
<script>
  import { count } from './stores.js';
  let count_value;
  const unsubscribe = count.subscribe(value => {
    count_value = value;
  });
</script>

<h1>The count is {count_value}</h1>
<button on:click={() => {
  count.update(n => n + 1)
}}>Increment</button>
```

Result JS output CSS output

The count is 0

Increment

Load the ArcGIS API for JavaScript with esri-loader

- “npm install esri-loader”
- Rollup.config.js:

```
commonjs({
        namedExports: {
                // left-hand side can be an absolute path, a path
                // relative to the current directory, or the name
                // of a module in node_modules
                "esri-loader": ["loadModules"]
        }
}),
```

https://esri-svelte-example.surge.sh/

Load the ArcGIS API for JavaScript with esri-loader

- Same as other esri-loader apps!

https://esri-svelte-example.surge.sh/

```
<script>
  import { loadModules } from "esri-loader";
  import { onMount } from "svelte";

  export let title;
  export let centerText;

  let viewDiv; // this is set using "bind:this" down below in the HTML.
  // For more info see:https://svelte.dev/tutorial/bind-this

  // Svelte - onMount - https://svelte.dev/tutorial/onmount
  onMount(async () => {
    // Use esri-loader to load the EsriMap and MapView modules
    // // https://github.com/Esri/esri-loader#usage
    const esriLoaderOptions = { css: true };
    const [EsriMap, MapView] = await loadModules(
      ["esri/Map", "esri/views/MapView"],
      esriLoaderOptions
    );

    // Create the map
    const map = new EsriMap({
      basemap: "streets"
    });
```

Load the ArcGIS API for JavaScript with esri-loader

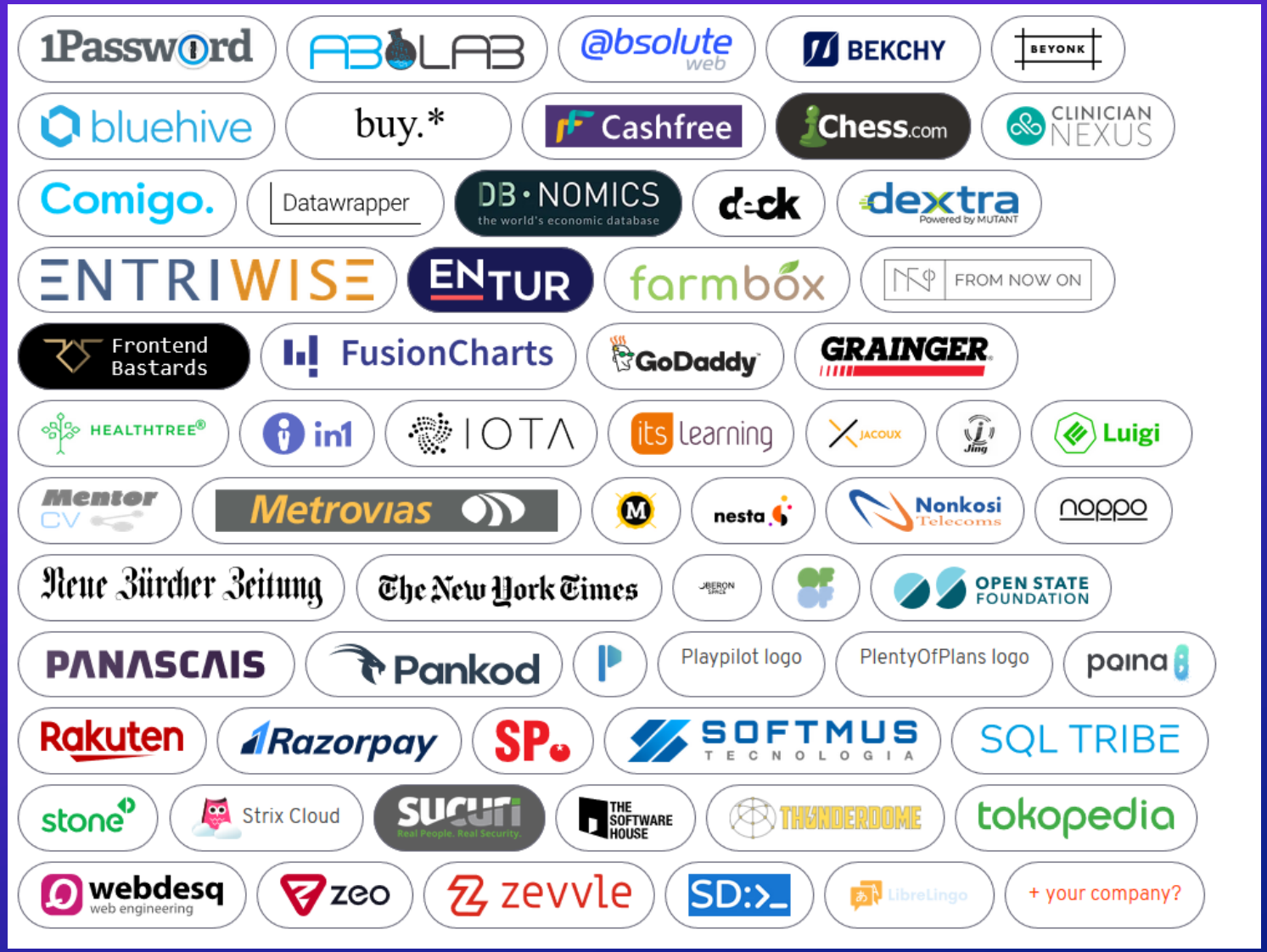
- **Wrap your map as a Svelte component!**

```
7     import EsriMapView from './EsriMapView.svelte';
8
9     export let basemap;
10    export let center;
11    export let scale;
12  </script>
13
14  <div class="card">
15    <p id={basemap} class="basemap-title">{basemap}</p>
16
17    <div class="esri-map-view-wrapper">
18      <EsriMapView basemap={basemap} center={center} scale={scale}></EsriMapView>
19    </div>
```

https://esri-svelte-basemaps-example.now.sh/

Real-World Apps?

- data4pakistan.com
- [others in progress]
- Yours here! Let us know!



Resources

- https://svelte.dev/
- Interactive tutorial https://svelte.dev/tutorial/
- Chat https://svelte.dev/chat
- Articles
  - Svelte – Svelte 3: Rethinking reactivity
  - Svelte – Write less code
  - DEV – Svelte Crash Course (in 10 pics!)
  - CSS Tricks – Getting Acquainted With Svelte, the New Framework on the Block
  - Felt Co-op – Why Svelte is our choice for a large web project in 2020

Bonus: Sapper!

https://sapper.svelte.dev/