knowit

Introduction to Svelte & SvelteKit

Svelte/Kit 101

knowit

Agenda

Introduction to Svelte & SvelteKit

1. Introduction
2. What is Svelte?
3. What is SvelteKit?
4. Hands on!
5. Questions & comments

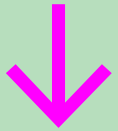
knowit

About me

- Originally from Muurame
- Tampere University 2014 - 2020, IT
- Cybercom/Knowit since 2019
- Love for sports
  - Freestyle skiing
  - Weightlifting
  - Padel
  - Bouldering
  - Skateboarding…

Disclaimer: Not a Svelte(Kit) expert

knowit



Svelte fundamentals

What is Svelte?

knowit

Svelte fundamentals

- Yet another JavaScript framework?!
  - Specifically, a compiler 🤓
- Vanilla JavaScript, simplified
- First released in 2016 by Rich Harris
- Current version 4 released in June 2023

knowit

Svelte fundamentals

- Compiled
  - Build optimized vanilla JavaScript
  - **No virtual DOM!**
- Compact
  - Tiny components
  - Tiny bundles
  - Less code to the end user
- Complete
  - Scoped styles
  - State management
  - Animations…

knowit

(More) Sustainable!

knowit

Svelte vs. React

SvelteComponent.svelte

```svelte
<script>
  let names = ["Matti", "Teppo"];
  let newName = "";

  function handleClick() {
    names = names.concat(newName);
    newName = "";
  }
</script>

{#each names as name, key}
  <p {key}>{name}</p>
{/each}

<input bind:value={newName} />
<button on:click={handleClick}>Add</button>
```

ReactComponent.jsx

```jsx
import { useState } from "react";

export function App() {
  const [names, setNames] = useState(["Matti", "Teppo"]);
  const [newName, setNewName] = useState("");

  function handleClick() {
    setNames(names.concat(newName));
    setNewName("");
  }

  function handleChange(event) {
    setNewName(event.target.value);
  }

  return (
    <>
      {names.map((name, key) => {
        return <p key={key}>{name}</p>
      })}

      <input value={newName} onChange={handleChange} />
      <button onClick={handleClick}>Add</button>
    </>
  );
}
```

knowit

Example: Store

Component.svelte

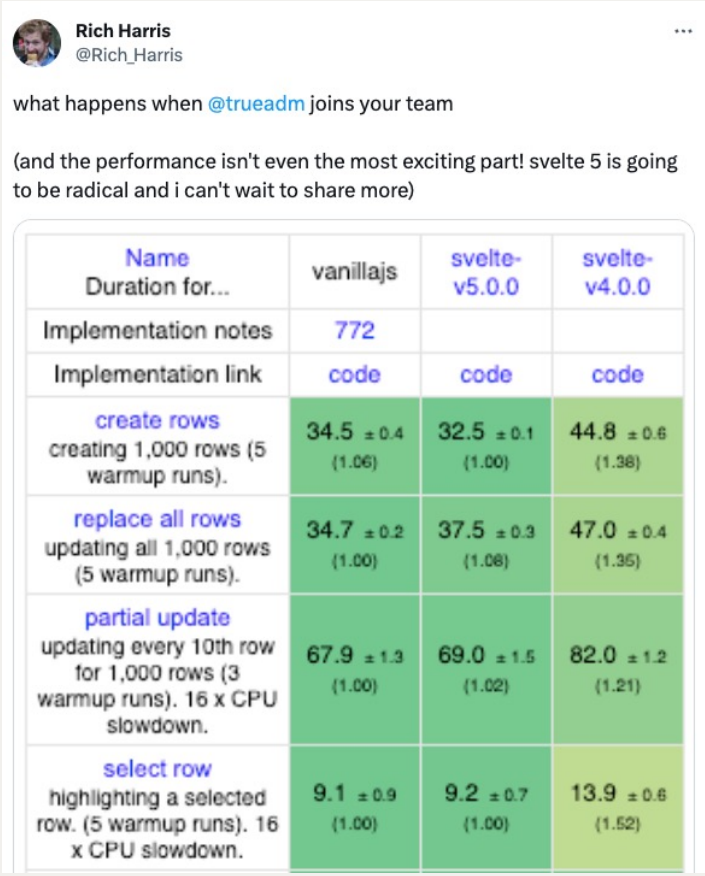
```
<script>
  import { createCountStore } from "./store.js";
  const count = createCountStore(0);
</script>

<h1>The count is {$count}</h1>
<button on:click={count.increment}>+</button>
<button on:click={count.decrement}>-</button>
```

store.js

```
import { writable } from "svelte/store";

export const createCountStore = (initialValue) => {
  const { update, subscribe } = writable(initialValue);

  const increment = () => update((state) => state + 1);
  const decrement = () => update((state) => state - 1);

  return {
    subscribe,
    increment,
    decrement,
  };
};
```

knowit

Future



Rich Harris
@Rich_Harris

what happens when @trueadm joins your team

(and the performance isn't even the most exciting part! svelte 5 is going to be radical and i can't wait to share more)

Name Duration for...	vanillajs	svelte-v5.0.0	svelte-v4.0.0
Implementation notes	772		
Implementation link	code	code	code
create rows creating 1,000 rows (5 warmup runs).	34.5 ± 0.4 (1.06)	32.5 ± 0.1 (1.00)	44.8 ± 0.6 (1.38)
replace all rows updating all 1,000 rows (5 warmup runs).	34.7 ± 0.2 (1.00)	37.5 ± 0.3 (1.08)	47.0 ± 0.4 (1.35)
partial update updating every 10th row for 1,000 rows (3 warmup runs). 16 x CPU slowdown.	67.9 ± 1.3 (1.00)	69.0 ± 1.5 (1.02)	82.0 ± 1.2 (1.21)
select row highlighting a selected row. (5 warmup runs). 16 x CPU slowdown.	9.1 ± 0.9 (1.00)	9.2 ± 0.7 (1.00)	13.9 ± 0.6 (1.52)

- Version 5 hype!
  - Improvements to performance
  - Improvements to reactivity
    - Svelte Runes (i.e., Signals)
    - https://svelte.dev/blog/runes

knowit

SvelteKit fundamentals

What is SvelteKit?

knowit

SvelteKit fundamentals

- “Metaframework” for Svelte
  - / Think Next for React
  - / Or Nuxt for Vue
- Gentle(r) learning curve
- Version 1 released in December 2022

knowit

SvelteKit fundamentals

- Out-of-the-box solutions for common use-cases
  - Routing
  - Preloading
  - CSR, SSR, SEO, SSG, SPA…
- Adapters for deployment
  - Node
  - Cloudflare
  - Netlify
  - Vercel…

```
my-project/
├ src/
│ ├ lib/
│ │ ├ server/
│ │ │ └ [your server-only lib files]
│ │ └ [your lib files]
│ ├ params/
│ │ └ [your param matchers]
│ ├ routes/
│ │ └ [your routes]
│ ├ app.html
│ ├ error.html
│ ├ hooks.client.js
│ ├ hooks.server.js
│ └ service-worker.js
├ static/
│ └ [your static assets]
├ tests/
│ └ [your tests]
├ package.json
├ svelte.config.js
├ tsconfig.json
└ vite.config.js
```

knowit

Project structure

- The only required parts
  - src/routes
  - app.html

```
my-project/
├ src/
│ ├ routes/
│ │ ├ blog/
│ │ │ ├ [slug]/
│ │ │ │ ├ +page.svelte
│ │ │ │ └ +page.server.js
│ │ ├ about/
│ │ │ ├ +page.svelte
│ │ │ ├ +page.js
│ │ │ └ +page.server.js
│ │ ├ +page.svelte
│ │ ├ +layout.svelte
│ │ ├ +layout.js
│ │ ├ +layout.server.js
│ │ └ +error.svelte
: :
```

knowit

Routing

- File based
- Route files prefixed with “+”
- Hierarchial

knowit

Example 1: Load function

/routes/blog/[slug]/+page.svelte

```svelte
<script>
  export let data;
</script>

<h1>{data.blog.title}</h1>
```

/routes/blog/[slug]/+page.server.js

```js
import { error } from "@sveltejs/kit";

export async function load({ params }) {
  const blog = await blogs.find((blog) =>
    blog.id === params.slug
  );

  if (!blog) throw error(404);

  return {
    blog,
  };
}
```

knowit

Example 1.5: Load function (TypeScript)

/routes/blog/[slug]/+page.svelte

```svelte
<script lang="ts">
  export let data;
</script>

<h1>{data.blog.title}</h1>
```

/routes/blog/[slug]/+page.server.ts

```ts
import { error } from "@sveltejs/kit";

export async function load({ params }) {
  const blog = await blogs.find((blog) =>
    blog.id === params.slug
  );

  if (!blog) throw error(404);

  return {
    blog,
  };
}
```

How it works: https://svelte.dev/blog/zero-config-type-safety

knowit

Example 2: Form actions

/routes/blog/new/+page.svelte

```
<form method="POST">
  <input name="title" />
  <input name="description" />
  <button type="submit">
    SUBMIT
  </button>
</form>
```

/routes/blog/new/+page.server.js

```
export const actions = {
  default: async ({ request }) => {
    const data = await request.formData();
    try {
      await db.createBlog({
        title: data.get("title"),
        description: data.get("description"),
      });
    } catch (e) {
      // Handle error
    }
  },
};
```

knowit

Example 2.5: Named form actions

/routes/blog/new/+page.svelte

```
<form method="POST" action="?/test">
  <input name="title" />
  <input name="description" />
  <button type="submit">
    SUBMIT
  </button>
</form>
```

/routes/blog/new/+page.server.js

```
export const actions = {
  test: async ({ request }) => {
    const data = await request.formData();
    try {
      await db.createBlog({
        title: data.get("title"),
        description: data.get("description"),
      });
    } catch (e) {
      // Handle error
    }
  },
};
```

knowit

Hands on time!

https://tinyurl.com/bdfsx2s7

knowit

Questions / comments?

knowit

Additional resources

- Huntabyte
  - https://www.youtube.com/@Huntabyte
- Joy of Code
  - https://www.youtube.com/@JoyofCodeDev