

How to Setup a SvelteKit Project



Ryan Neil Parker · [Follow](#)

5 min read · Mar 10, 2024

Listen

Share

Introduction to SvelteKit



SvelteKit Logo

Overview of SvelteKit as a Web Development Framework

SvelteKit is a cutting-edge web development framework that has gained significant traction in recent times. It leverages the capabilities of the Svelte framework while providing additional features tailored for full-stack development.

Benefits of Using SvelteKit

- **Enhanced Performance:** SvelteKit utilizes a novel compilation process that translates Svelte components into highly optimized JavaScript code, resulting in exceptional performance and fast load times.
- **Developer-Friendliness:** SvelteKit's intuitive syntax and comprehensive documentation empower developers with a seamless and enjoyable



Search



In subsequent sections, we will embark on a comprehensive journey through SvelteKit, covering essential topics such as project setup, application creation, development and production workflows, and more. By the end of this tutorial, you will possess a strong foundation in SvelteKit and be equipped to harness its capabilities for your web development endeavors.

Prerequisites and Project Setup

Prerequisites

Before embarking on the journey of developing a SvelteKit application, it is essential to ensure your system meets the following requirements:

- Node.js version 16 or later
- npm version 6 or later

Installing Node.js and npm

To install Node.js, navigate to the official Node.js website and follow the instructions for your operating system. Once Node.js is installed, you will also have npm (Node Package Manager) at your disposal. Verify your installations by running the following commands in your command-line interface (CLI):

```
node -v  
npm -v
```

Setting Up a New Project

To create a new SvelteKit project, we will employ the “`degit`” command. It facilitates the instantiation of a project boilerplate from a git repository. Navigate to your desired project directory and execute the following command:

```
npx degit sveltejs/template my-sveltekit-project
```

This command will create a new directory named “`my-sveltekit-project`” containing the essential files and directories for a SvelteKit project.

Explanation of “`degit`”

“`degit`” is a command-line tool that allows you to initialize a new project by fetching a pre-existing template from a git repository. In our case, we are using the official SvelteKit template repository to create a new SvelteKit project.

Next Steps

With the necessary prerequisites in place and a new project set up, we can proceed to the next section of our guide, where we will delve into creating a basic Svelte application within our project.

Creating a Svelte App

Step 1: Creating a New File

Within the project directory, create a new file with the name `App.svelte`. This will serve as the main component of our Svelte application.

Step 2: Defining the Component

In the `App.svelte` file, we define the Svelte component as follows:

```
<script>
  export let name;
</script>

<h1>Hello, {name}!</h1>
```

This component expects a `name` property to be passed to it. When invoked, it will render a simple heading with the provided `name`.

Step 3: Using the Component

In the `index.svelte` file, we can use the `App` component by importing it:

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

<App name="SvelteKit User" />
```

This will render the `App` component with the `name` property set to `SvelteKit User`.

Step 4: Understanding the Syntax

a. Script Tag:

The ``<script>`` tag in Svelte is used to define the logic and state of the component. Here, we define a variable `name` that can be passed in as a property.

b. Export Statement:

`'export let name;'` marks `name` as a property that can be passed into the component when it is used.

c. Template Tag:

The ``<div>`` tag in Svelte represents the HTML structure of the component. It uses curly braces `({})` to interpolate the `name` property.

Running and Viewing the App

1. Starting the Development Server

Once your Svelte application is created, you can launch the development server by running the following command within your project directory:

```
npm run dev
```

This command starts a server on your local machine that enables you to view and test your application during development.

2. Accessing the App in a Browser

Once the development server is running, you can open your web browser and navigate to the following URL:

<http://localhost:3000>

This should display your Svelte application in the browser window.

3. Verifying App Functionality

After accessing the app in the browser, interact with its elements to ensure that it functions as expected. If you followed the steps correctly, you should see a simple Svelte application rendered on the screen.

4. Output

The output you see in the browser window will depend on the specific code you implemented in your Svelte app. If you followed the code examples provided in the previous section, you should see a basic heading and paragraph.

Building for Production

In the previous sections, we've established a solid foundation for developing with SvelteKit by setting up a project and creating a basic application. The next step in our journey is to prepare our app for deployment, a crucial phase known as "building for production."

Purpose of Building for Production

Building for production involves converting our development-ready code into a more optimized and streamlined package suitable for deployment on a live server or hosting platform. This process removes debugging tools, minifies code, and performs other optimizations to enhance performance and reduce file sizes.

Instructions on Building for Production

To build our SvelteKit application for production, follow these steps:

1. Ensure that your project is up-to-date by running `npm update`.
2. Open a terminal window in your project directory.
3. Run the following command:

```
npm run build
```

This command will launch the build process, during which you may see various messages and progress indicators.

Understanding the Output

Once the build is complete, you should see a new directory named `build` within your project folder. This directory contains the optimized and production-ready version of your application.

Inside the `build` directory, you'll find the following notable files:

- **index.html:** The main HTML file that serves as the entry point for your application.

- **client.js:** The bundled JavaScript code for the client-side (browser) portion of your app.
- **server.js:** The bundled JavaScript code for the server-side rendering (SSR) component.

You can now deploy these build files to a web server or hosting platform to make your application accessible to users.

Conclusion

Building for production is an essential step in the SvelteKit development process, ensuring that your application is optimized for performance and ready for deployment. By understanding the purpose and following the instructions outlined above, you can confidently prepare your SvelteKit app for production.

Conclusion

In this blog post, we have provided a thorough guide to setting up a SvelteKit skeleton project. We covered the following key steps:

- Installing Node.js and npm
- Creating a new project using the “`degit`” command
- Creating a basic Svelte application
- Running and viewing the application
- Building for production

By following these steps, you will have a solid foundation for developing SvelteKit applications.

I encourage you to explore further resources on SvelteKit and experiment with its capabilities. This innovative framework offers a powerful and efficient approach to web development, enabling you to create performant and user-friendly applications with ease.

Svelte

Sveltekit

Web Development

Programming

Software Development

[Follow](#)

Written by Ryan Neil Parker

24 Followers · 62 Following

I like to make beautiful things for the web, that just work

More from Ryan Neil Parker



In Dev Genius by Ryan Neil Parker

Building a Stylish Hello World with Svelte and FastAPI: A Step-by-Step Guide

Ready to embark on a fun and stylish web development journey with Svelte and FastAPI? Let's create a simple "Hello World" page that also...

Dec 7, 2023 47





Ryan Neil Parker

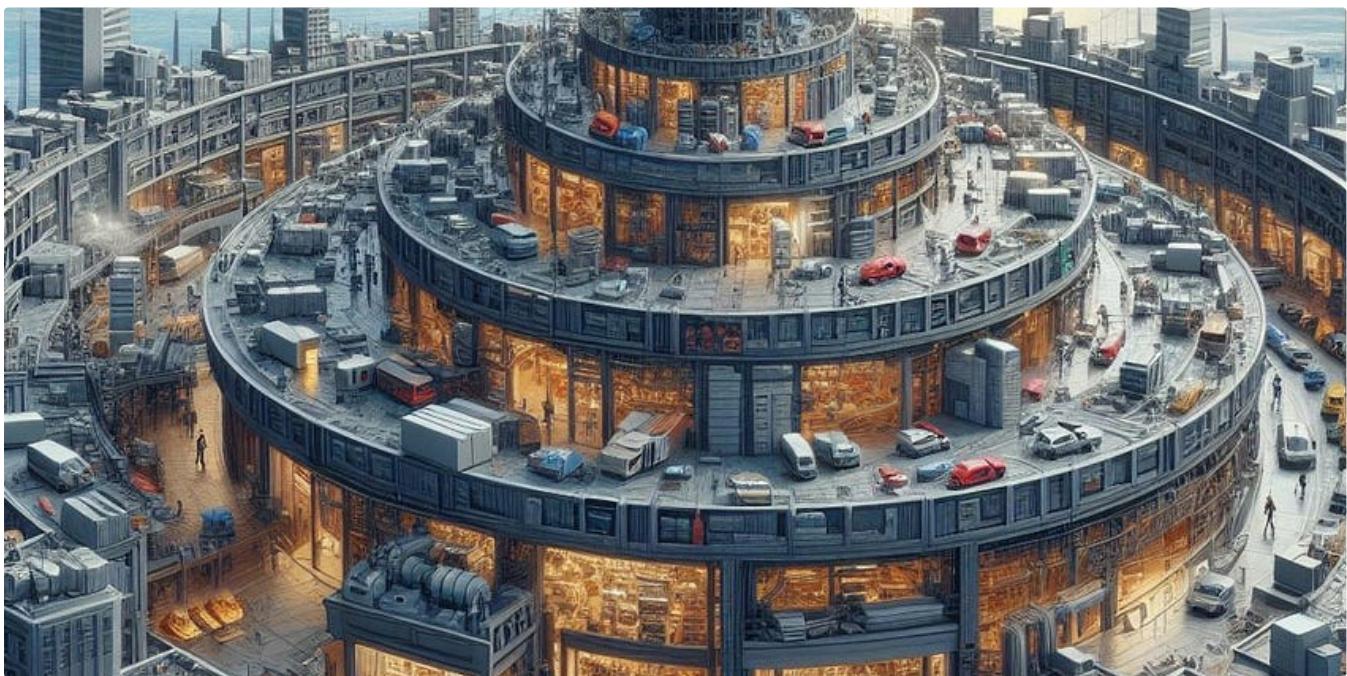
SA ID Fumble: How South Africa Managed to Incorrectly Apply the Luhn Algorithm

Picture this: You're sweating through a job interview, heart pounding like a rogue bass drum. The interviewer slides across your resume...

Jan 17

👏 2

💬 1



In Dev Genius by Ryan Neil Parker

The Magic of Data Sharing Between Components in Svelte!

Hey there, fellow Sveltians! 🚀 If you're delving into the enchanting world of Svelte, you're probably marveling at its simplicity and...

Dec 11, 2023 ⏱ 15 💬 1



 Ryan Neil Parker

Software Design Patterns: Building Robust and Maintainable Software

A quick reference of software design patterns and their use cases.

Mar 14 ⏱ 13



See all from Ryan Neil Parker

Recommended from Medium

Always Free
24 GB RAM + 4 CPU + 200 GB




 @harendravarma2
 @harendra21
 @hendra21

 Harendra

How I Am Using a Lifetime 100% Free Server

Get a server with 24 GB RAM + 4 CPU + 200 GB Storage + Always Free

Oct 26 6K 84



	JSON	YAML
<pre>er1</name> n</owner> 3456</created> /e</status></pre>	<pre>{ Servers: [{ name: Server1, owner: John, created: 123456, status: active }] }</pre>	<pre>Servers: - name: Serve owner: John created: 123 status: active</pre>

 Sam Atmaramani

The Comprehensive Guide to YAML, JSON, TOML, HCL (HashiCorp), XML & differences

The Comprehensive Guide to YAML and Its Alternatives

Jul 6 2



Lists



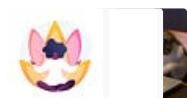
General Coding Knowledge

20 stories · 1780 saves



Coding & Development

11 stories · 925 saves



Stories to Help You Grow as a Software Developer

19 stories · 1499 saves



Leadership

62 stories · 495 saves



In Stackademic by Crafting-Code

I Stopped Using Kubernetes. Our DevOps Team Is Happier Than Ever

Why Letting Go of Kubernetes Worked for Us

Nov 19

3.4K

106



Elixir Web Scraping Guide

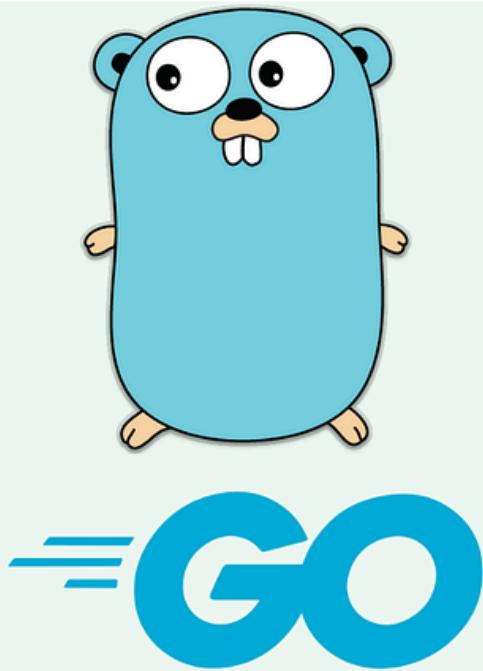


 Data Journal

Elixir Web Scraping in 2024: A Comprehensive Guide

In this guide, I'll walk you through using Elixir for web scraping. We'll go over the basics—setting up your environment, essential...

Nov 11  11  1

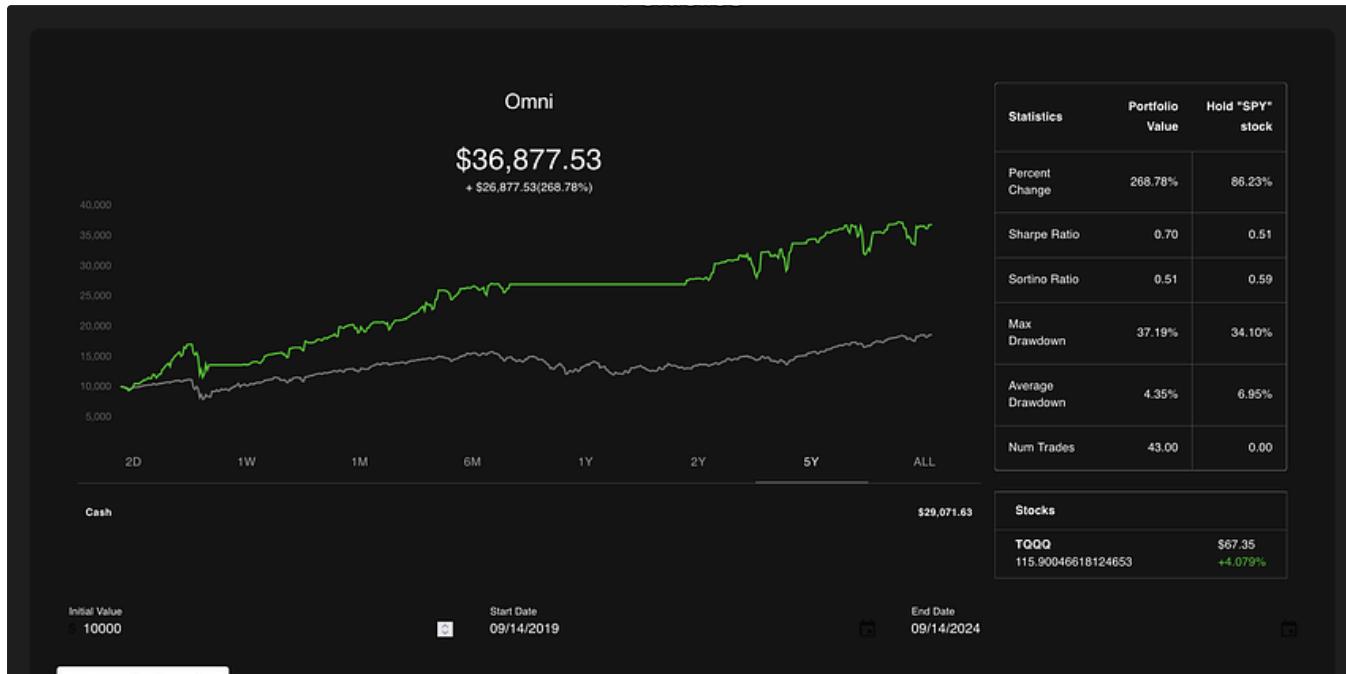


 Yaswanth

Why I'm Switching from Go to Python

Hey everyone! So, after spending years writing Go code (and loving it, honestly), I've decided it's time for a change. Yup, I'm switching...

★ Sep 3 ⚡ 491 🎧 14



In DataDrivenInvestor by Austin Starks

I used OpenAI's o1 model to develop a trading strategy. It is DESTROYING the market

It literally took one try. I was shocked.

★ Sep 15 ⚡ 6.7K 🎧 170



See more recommendations