Cloud Contraptions
TRAINING & CONSULTING

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

Full-Stack Web Development with SvelteKit

Class Duration

5 days

Student Prerequisites

- JavaScript and HTML experience
- Python programming experience (FastAPI backend) or C# programming experience (ASP.NET Core backend)
- Experience with CSS and SQL is helpful, but not required

Target Audience

Full-stack developers and teams who want a SvelteKit front end with a dedicated API backend. Each delivery uses one backend track chosen by the team: Python with FastAPI, or C# with ASP.NET Core. Teams new to the front end can start with Svelte Essentials.

Description

This course teaches teams to build web apps with SvelteKit on the front end and a REST API on the back end. The first part covers runes, components, snippets, routing, load functions, form actions, hooks, and explicit environment variables. The FastAPI track builds an async Python API with Pydantic, SQLAlchemy, Alembic, and JWT authentication. The ASP.NET Core track builds a .NET API with EF Core, ASP.NET Core Identity, OpenAPI documents, and SignalR. Both tracks finish with typed OpenAPI clients, cookie-based auth, real-time updates, testing, and containerized deployment.

Learning Outcomes

- Build SvelteKit apps with runes, typed components, snippets, routing, and load functions.
- Build forms with form actions, validation, and progressive enhancement.
- Evaluate the experimental remote functions.

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

- Keep secrets on the server with hooks, server-only modules, and explicit environment variables.
- Build a REST API with FastAPI or ASP.NET Core.
- Generate a typed TypeScript client from the API's OpenAPI document.
- Use SvelteKit as a backend-for-frontend with consistent error handling.
- Secure the app with cookie sessions in SvelteKit and tokens in the API.
- Add real-time updates with SignalR, WebSockets, or server-sent events.
- Test with Vitest, Playwright, and backend integration tests.
- Containerize and deploy the full stack.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Students will need a free, personal GitHub account to access the courseware. Students will need permission to install Node.js, Visual Studio Code, Docker or Podman, and either Python or the .NET SDK on their computers. Also, students will need permission to install NPM packages, Python (pip or uv) or NuGet packages, and Visual Studio Code extensions. If students are unable to configure a local environment, a cloud-based environment can be provided.

Software versions: Svelte 5 and SvelteKit 3 on Node.js 22.17 or later; FastAPI on Python 3.12 or later, or ASP.NET Core on the .NET 10 (LTS) or .NET 11 SDK.

Training Topics

Introduction to Svelte and SvelteKit

- What is Svelte? What problem does it solve?
- The Svelte compiler and signal-based reactivity
- Routing, rendering, and server code in SvelteKit
- SvelteKit with a dedicated API backend

Development Environment

- Requirements
- Project scaffolding with `npx sv create`

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

- Adding tooling with `sv add`
- Project structure and `vite.config.ts`
- `#lib` imports
- Svelte extensions for VS Code and JetBrains
- TypeScript and `svelte-check`

Components and Templates

- Markup, expressions, and attributes
- `{#if}` and `{#each}` blocks
- `{#await}` and `{#key}` blocks
- Scoped CSS
- Props with `$props()` and TypeScript types
- Callback props and bindable props
- Snippets with `{#snippet}` and `{@render}`
- Attachments with `{@attach}`

Reactivity with Runes

- `$state`, `$derived`, `$effect`, and `$props`
- Deeply reactive objects and arrays
- Reactive classes and `.svelte.ts` modules
- Sharing state with context
- Page and navigation state with `$app/state`

Routing and Layouts

- File-based routing
- Route parameters and matchers
- Layouts and route groups
- Navigation and preloading

Loading Data

- Server load functions in `+page.server.ts`
- Universal load functions in `+page.ts`
- Streaming promises
- Invalidation and `refreshAll`

Forms and Mutations

- Named form actions
- Validation and `fail()`

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

- Progressive enhancement with `use:enhance`
- Experimental remote functions
- `query`, `form`, and `command`

Errors, Hooks, and Server Code

- `error()`, `redirect()`, and `+error.svelte`
- `handle`, `handleFetch`, and `handleError`
- `event.locals` for request-scoped data
- Server-only modules
- Explicit environment variables in `src/env.ts`

Connecting SvelteKit to the API

- Backend-for-frontend vs. browser calls
- Calling the API from server code
- Generating a typed client from OpenAPI
- Forwarding credentials with `handleFetch`
- Mapping API errors to `error()` and `fail()`
- CORS for direct browser calls

Authentication Across the Stack

- Session cookies vs. bearer tokens
- Login, registration, and logout
- Tokens in HTTP-only cookies
- Loading the current user in `handle`
- Protecting routes in layouts and hooks
- Role- and policy-based authorization

HTTP, REST, and OpenAPI Review

- HTTP verbs and URLs
- Path and query parameters
- Common headers and status codes
- REST conventions and resource design
- OpenAPI documents
- Problem Details error responses

FastAPI Track: Building the API

- Project setup with the FastAPI CLI
- Path operations and parameters

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved

To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

- Request and response models with Pydantic
- Errors with `HTTPException`
- Dependency injection with `Depends`
- Organizing APIs with `APIRouter`
- Async routes and background tasks
- Middleware and CORS
- Automatic OpenAPI documentation

FastAPI Track: Data Access

- SQLAlchemy async engine and sessions
- Defining models
- Database sessions as dependencies
- Migrations with Alembic
- CRUD operations from FastAPI routes

FastAPI Track: Security and Real-Time

- OAuth2 password flow with JWT
- Password hashing
- Security dependencies
- Scopes and roles
- WebSocket endpoints
- Server-sent events

ASP.NET Core Track: Building the API

- Configuration and options
- Dependency injection
- Request pipeline and middleware
- Logging
- Choosing minimal APIs or controllers
- Route groups and parameter binding
- `TypedResults`
- Validation and Problem Details
- Endpoint filters and action filters
- OpenAPI with `AddOpenApi` and `MapOpenApi`
- XML comments and document transformers
- Scalar or Swagger UI
- CORS

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved



To discuss this course and customizations:
Call: +1 434-509-6890 or Email: sales@cloudcontraptions.com

ASP.NET Core Track: Data Access

- EF Core with SQL Server or PostgreSQL
- `DbContext`, entities, and relationships
- Migrations
- Querying and modifying data with LINQ
- Dapper for hand-written SQL
- When the repository pattern helps

ASP.NET Core Track: Security and Real-Time

- ASP.NET Core Identity
- JWT bearer tokens
- Authorization policies and roles
- `RequireAuthorization` and `[Authorize]`
- SignalR hubs
- Groups and users
- SignalR from a Svelte component

Real-Time Updates in Svelte

- Connections with `$effect` cleanup
- Connections with attachments
- Live data in reactive state classes
- Reconnection and connection state

Testing

- Unit and component tests with Vitest
- Testing load functions and form actions
- FastAPI tests with `TestClient`
- ASP.NET Core tests with `WebApplicationFactory`
- End-to-end tests with Playwright

Deployment

- Building with `adapter-node`
- Containerizing the front end and API
- Same-origin deployment behind a reverse proxy
- Configuration and secrets per environment
- Health checks and OpenTelemetry

Copyright 2026 Cloud Contraptions LLC – All Rights Reserved