Toggly SvelteKit

Server hooks · loads/actions · signed snapshot · hydration

Install

```
npm i @ops-ai/toggly-sveltekit
```

.ENV

```
TOGGLY_APP_KEY=          # backend
PUBLIC_TOGGLY_APP_KEY=   # Front-end App Key
TOGGLY_ENVIRONMENT=Production
PUBLIC_TOGGLY_ENVIRONMENT=Production
```

Mark flags **Available to Client SDK**. Same environment on both sides.

hooks.server.ts

```
const client = createTogglyClient({
  appKey: env.TOGGLY_APP_KEY,
  environment: env.TOGGLY_ENVIRONMENT
    ?? 'Production',
  verifySignatures: true,
  featureDefaults: { 'new-dashboard': false },
})
await client.init()

export const handle = createTogglyHandle({
  client,
  context: () => ({ identity: '' }),
  frontend: {
    appKey: publicEnv.PUBLIC_TOGGLY_APP_KEY,
    environment: publicEnv.PUBLIC_TOGGLY_ENVIRONMENT
      ?? 'Production',
    expose: ['new-dashboard', 'api-v2'],
    featureDefaults: { 'new-dashboard': false },
  },
})
```

Do not `setIdentity` on the shared client. Default frontend snapshot is anonymous — project public identity via `clientContext`.

Load + hydrate

+LAYOUT.SERVER.TS

```
export const load = async (event) => ({
  toggly: await loadToggly(event),
  publicKey: env.PUBLIC_TOGGLY_APP_KEY ?? '',
  environment: env.PUBLIC_TOGGLY_ENVIRONMENT
    ?? 'Production',
})
```

+LAYOUT.SVELTE

```
const toggly = createToggly(data.toggly, {
  appKey: data.publicKey,
  environment: data.environment,
})
setContext('toggly', toggly)
$: toggly.update(data.toggly)
onMount(() => { void toggly.start() })
onDestroy(() => toggly.dispose())
```

Only `expose` keys are serialized. Do not return `event.locals.toggly` itself.

Feature + negate

```
<Feature {toggly} feature="new-dashboard">
  <p>New dashboard</p>
</Feature>
<Feature {toggly} feature="new-dashboard"
         options={{ negate: true }}>
  <p>Classic dashboard</p>
</Feature>
<Feature {toggly}
  feature={['new-dashboard', 'api-v2']}
  options={{ requirement: 'any' }}>
  <p>At least one enabled</p>
</Feature>
```

Same keys/requirement/entity/defaults on both blocks. Synchronous — no loading state.

Toggly SvelteKit

Server hooks · loads/actions · signed snapshot · hydration

Guards & entities

```
await requireFeature(event, 'enhanced-submit')
toggly.isEnabled('ExpressCheckout', {
  entity: {
    kind: 'Order', key: 'ord-vip',
    attributes: { Vip: true, Total: 150 },
  },
})
```

Frontend EntityGate without entity fails closed. Hiding UI does not block a direct action request.

Variants

```
// hooks.server.ts: frontend.enableVariants: true
// +layout.svelte: createToggly(data.toggly, {
//   ...opts, enableVariants: true })
toggly.getVariant('checkout-flow')
// { name, configurationValue? } | null
toggly.getVariantValue('checkout-flow')
```

Opt-in; fetches `/evaluated-variants-signed`. Set `enableVariants` on both server and browser.

Pitfalls

- Backend key stays in server-only modules; `PUBLIC_` is a Front-end App Key.
- After login/logout, invalidate the server load so the layout gets a new snapshot.
- Node adapter-node only for this server integration.
- Client Svelte (no SSR) stays on the JavaScript cheat sheet.

Resources

- Docs: `docs.toggly.io/sdks/javascript/sveltekit`
- JS / Svelte sheet: `/cheatsheets/javascript`
- Sample: `Toggly.Samples/sveltekit-sdk`