Toggly Electron

Main process client · preload bridge · renderer gates

Cheat Sheet

`@ops-ai/electron-feature-flags-toggly`

Electron 28+ · React 18+ optional

Install

```
npm i @ops-ai/electron-feature-flags-toggly
```

Frontend App Key in **main**. Keep `contextIsolation: true` and `nodeIntegration: false`.

Main

```
import { app, BrowserWindow, ipcMain }
  from 'electron'
import {
  closeToggly, initToggly, registerTogglyIpc,
} from '@ops-ai/electron-feature-flags-toggly/main'

await app.whenReady()
await initToggly({
  appKey: process.env.TOGGLY_APP_KEY,
  environment: 'Production',
  userDataPath: app.getPath('userData'),
  flagDefaults: { NewDashboard: false },
  verifySignatures: true,
})
const unregister = registerTogglyIpc(
  ipcMain, () ⇒ BrowserWindow.getAllWindows())
app.on('before-quit', () ⇒ {
  unregister(); closeToggly()
})
```

Preload

```
import { createRequire } from 'node:module'
const require = createRequire(import.meta.url)
const preload = require.resolve(
  '@ops-ai/electron-feature-flags-toggly'
  + '/preload/entry')

new BrowserWindow({
  webPreferences: {
    preload,
    contextIsolation: true,
    nodeIntegration: false,
  },
})
```

Create the window after init + IPC. Custom preload: bundle CommonJS and call `exposeToggly()`.

Renderer

```
import {
  evaluateFeatureGate, isFeatureOn, onFlagsUpdated,
} from '@ops-ai/electron-feature-flags-toggly'

const on = isFeatureOn('NewDashboard')
const any = evaluateFeatureGate(
  ['new-dashboard', 'beta-access'], 'any')
const unsub = onFlagsUpdated(() ⇒ render())
```

Missing preload: sync checks fail closed. `getFlags` / `setContext` reject.

React (optional)

```
import { Feature, useFeatureFlag }
  from '@ops-ai/electron-feature-flags-toggly/react'

<Feature featureKey="NewDashboard">
  <NewDashboard />
</Feature>
<Feature featureKey="NewDashboard" negate>
  <Classic />
</Feature>
```

Toggly Electron

Main process client · preload bridge · renderer gates

Context & entities

```
await setContext({
  identity: 'user-123',
  groups: ['beta'],
  claims: { plan: 'pro' },
})
isFeatureOn('ExpressCheckout', {
  kind: 'Order', key: 'order-9001',
  attributes: { Vip: true, Total: 149 },
})
```

One client per main process — not per window. Entity per check, not in global user context.

Offline & live updates

- `flagDefaults` when there is no key or first refresh fails. Last-known-good lives under `userData`.
- Omit `appKey` for defaults-only (no network).
- `enableLiveUpdates` defaults true when an app key is set. IPC broadcasts the boolean snapshot to renderer windows.

Pitfalls

- Initialize main + `registerTogglyIpc` before any renderer uses the bridge.
- Untrusted remote content: do not expose full `window.toggly` — use a narrower preload.
- `getFlags()` is a plain boolean snapshot — use entity-aware checks for ContextProperty flags.

Resources

- Docs: `docs.toggly.io/sdks/electron`
- Sample: `Toggly.Samples/electron-sdk`