Kontekstualna integracija kliničnih modulov v OI PAS

`OI PAS` uporablja Synapse SDK, napisan v jeziku TypeScript, ki omogoča integracijo kliničnih modulov. Modul teče v `<iframe>`, s PAS pa komunicira prek `postMessage`. Skrbi za življenjski cikel povezave, tipizirane klice metod, dostop do avtentikacijskega žetona in posredovanje sporočil.

Kako deluje

Synapse uporablja način delovanja gostitelj-odjemalec:

Stran	Razred	Vloga
Odjemalec (modul v iframe)	`SynapseClient`	Vzpostavi povezavo s PAS ter izpostavi `auth`, `init`, `close` in sporočanje.
Gostitelj (PAS)	`SynapseHost`	Sprejme povezavo modula in zagotavlja implementacije tipiziranih metod gostitelja. Modul ga potrebuje samo, če sam gosti podmodule.

Življenjski cikel

- Povezava: ko je iframe vstavljen v DOM, ima modul 30 sekund, da se naloži in pokliče `init()`. Sicer PAS prikaže napako, modul pa ostane skrit.
- Začetni kontekst: PAS ga posreduje prek URL parametrov na `src` iframa (npr. `?view=list&patientID=123`). Modul jih ob zagonu prebere.
- Konfiguracija: `init()` vrne `SynapseConfig` s končnima točkama `fhirEndpoint` in `ehrEndpoint` (korena strežnikov FHIR in OpenEHR), jezikom (`language`, npr. sl-SI) in okoljem (`environment`).
- Zapiranje: `close(data?)` gostitelju posreduje neobvezne podatke. Po zaprtju je povezava terminirana. S `setCanClose(bool)` modul sporoči, ali ga je varno zapreti.

Avtentikacija in napake

Modul pridobi žeton z metodo `client.auth.getToken()`. Vrne prazen niz, če ga gostitelj nima, in ne preverja poteka, zato ob odgovoru 401 modul pokliče `auth.refreshToken()`. Vse metode lahko zavrnejo klic: `init()` po 30 sekundah.

Metode gostitelja

Primarno predlagamo uporabo metod: `init`, `setCanClose`, `close`, `auth.getToken`, `auth.refreshToken`. Metode `sendMessage`, `addMessageListener` se uporablja samo v primeru, da tipizirane metode primera ne pokrivajo.

Varnost

PAS naloži modul samo, če je njegov izvor naveden v `moduleOrigins` v konfiguraciji PAS. Sicer modula ne naloži in mu ne izda žetona. Sporočila iz drugih izvorov zavrže. Pri `targetOrigin` je v produkciji potrebno uporabiti natančen izvor (npr. `https://modules.pas.onko-i.si`), nikoli `*`.

Primer

```
import { SynapseClient } from '@pas/synapse';

const client = new SynapseClient();
const config = await client.init();         // 30 s časovna omejitev
const token = await client.auth.getToken();

// ... delo z modulom prek tipiziranih metod gostitelja ...

await client.close({ status: 'completed' });
```