3 - AuthService

Iagro CLP Local — AuthService

Cliente HTTP mínimo para o domínio **Auth** do backend NestJS. Não persiste tokens — responsabilidade de `SessionManager` + `main.py`.

Código: `src/services/auth_service.py`

📋 Resumo

Operação	Endpoint	Persistência
Login	POST `/auth/login`	`SessionManager.save_session`
Refresh	POST `/auth/refresh`	Loop ~90 s em `main`
Userinfo	GET `/auth/userinfo`	UI perfil operador

`API_BASE_URL` lido em import de `runtime_root()/.env` (fallback `<http://localhost:3000`).>

🎯 API pública

Método	Request	Response / efeito
`login(username, password, tenant_id?)`	JSON credentials; header `X-Tenant-ID` opcional	`access_token`, `refresh_token`, `tenant_id` resolvido
`refresh_token(refresh_token, tenant_id)`	Body refresh	Novo par de tokens
`get_user_info(access_token, tenant_id)`	Bearer	Perfil OIDC-like

`extract_tenant_from_token(jwt)`	—	Claim tenant ou None

`_extract_tenant_from_token` **(sem verificar assinatura)**

Ordem de claims tentadas: `tenant_id`, `tenantId`, `realm`, parsing de `iss`, etc. Usado quando o body de login não traz tenant explícito.

Implicação de segurança: extração de tenant é conveniência de cliente; autorização real é no backend com Bearer validado.

📁 Contratos HTTP

Login — request

```
{ "username": "string", "password": "string" }
```

Login — response (200/201)

Campos obrigatórios para o desktop:

Campo	Uso
`access_token`	`ApiService` Authorization
`refresh_token`	Loop de renovação
`tenant_id`	Body ou JWT derivado

Refresh

Mesmo tenant_id da sessão salva; falha 401 → UI pode forçar re-login.

🔄 Fluxo na aplicação

```
LoginWindow.submit
  → AuthService.login
  → SessionManager.save_session(tokens, tenant, user)
  → ApiService(API_BASE_URL, access_token, tenant_id)
  → ContainerManager.set_api_service (propaga a todos CLPContainer)
  → BranchSelector → SmartCaldas

Timer main._session_refresh_loop (~90s):
  → refresh_token
  → SessionManager.update
  → api_service.update_headers(Bearer, tenant)
  → _sync_session_tokens_to_api_service()
```

Sem refresh bem-sucedido, chamadas ERP começam a falhar com 401 — monitors logam e pulam ciclo.

🎯 Integração

Consumidor	Detalhe
`main.py`	Orquestra login, refresh, logout
`ApiService`	Mesma base URL; headers após login
`LoginWindow` (**11**)	Formulário credenciais + tenant opcional
`IAgroCLPAppCTk`	Indicador usuário / sessão

⚙️ Configuração

Variável	Efeito
`API_BASE_URL`	Host NestJS (sem path trailing obrigatório)

⚠️ Erros

HTTP / rede	Comportamento
401 login	Exceção credenciais inválidas
401 userinfo	Exceção token expirado
Timeout / connection	`Erro de conexão: …`
JWT sem tenant	`tenant_id=None` + warning; pode bloquear rotas multi-tenant

🔗 Referências

- **4 - ApiService** · **12 - API — contrato OpenAPI** (tag Auth)
- **11 - UI — IAgroCLPAppCTk**