| **Auth Method** | **Use Case** | **How It Works** | **Pros** | **Cons** | **Libraries Used** |
| --- | --- | --- | --- | --- | --- |
| **OAuth2 + Bearer** | Web/Mobile token-based login | Client sends a bearer token in the header | - Standardized- Secure- Easy with Depends | - Requires login endpoint- Needs token generation | fastapi, OAuth2PasswordBearer |
| **JWT (JSON Web Token)** | Stateless authentication APIs | Encodes user info in a signed token (JWT) | - Stateless- Scalable- Compact | - Can be stolen- No built-in revocation | pyjwt, jose, fastapi |
| **API Key** | Internal APIs, dashboards | A fixed token passed in headers or query string | - Simple- Easy to test/debug | - Less secure- No user context | FastAPI built-in |
| **HTTP Basic** | Admin access or CLI | User sends Base64 encoded username:password in header | - Very simple- Built-in to browsers/CLI tools | - Password sent every time- Not ideal for production | fastapi.security.HTTPBasic |
| **OAuth2 Social Login** | Login via Google, GitHub, etc. | Redirect to provider -> authorize -> token | - User-friendly- No password storage | - Complex- Token exchange- Needs extra setup | authlib, fastapi-login |
| **Session Cookie** | Browser-based web apps | Stores a session ID in a cookie (server tracks session) | - Works like Django/Flask- Persistent sessions | - Server-side state- CSRF protection needed | starlette.middleware.sessions |
| **LDAP / Active Dir.** | Enterprise environment, org auth | Validates user against an external directory | - Centralized- Corporate-friendly | - Complex- Needs external LDAP server | ldap3, python-ldap |
| **Custom Header Auth** | Internal APIs, microservices | Uses custom headers like x-user-id or x-auth-token | - Flexible- Can pass custom data easily | - No standard- Easy to misuse | FastAPI native |