# Utils Module Architecture



**Utils Module**

**© OAuthAuthentication**
- ☐ disp: Disp
- ☐ debug: bool
- ☐ success: int
- ☐ error: int
- ☐ runtime_manager_initialised: RuntimeManager
- ☐ database_link: SQL
- ☐ boilerplate_non_http_initialised: BoilerplateNonHTTP
- ☐ boilerplate_responses_initialised: BoilerplateResponses
- ☐ boilerplate_incoming_initialised: BoilerplateIncoming
- ☐ server_headers_initialised: ServerHeaders
- ☐ __init__(success, error, debug)
- ☐ _generate_oauth_authorization_url(provider): Union[int, str]
- ☐ _exchange_code_for_token(provider, code): Union[int, dict]
- ☐ _refresh_oauth_token(provider, refresh_token): Union[int, dict]
- ☐ _get_user_info_from_provider(provider, token): Union[int, dict]
- ○ renew_oaths(): None

**© PasswordHandling**
- ☐ disp: Disp
- ☐ debug: bool
- ☐ success: int
- ☐ error: int
- ☐ salt_rounds: int
- ○ __init__(error, success, debug)
- ○ hash_password(password): str
- ○ check_password(password, password_hash): bool

**© «Final» ServerManagement**
- ☐ disp: Disp
- ☐ error: int
- ☐ success: int
- ☐ debug: bool
- ☐ runtime_manager: RuntimeManager
- ☐ runtime_control: RuntimeControl
- ☐ database_link: SQL
- ☐ background_tasks_initialised: BackgroundTasks
- ☐ boilerplate_responses_initialised: Optional[BoilerplateResponses]
- ☐ server_headers: ServerHeaders
- ☐ documentation_handler_initialised: Optional[DocumentationHandler]
- ○ __init__(error, success, debug)
- ○ is_server_alive(): bool
- ○ is_server_running(): bool
- ○ shutdown(background_tasks): Response
- ☐ _perform_shutdown(): None

**© «module» constants**
- ○ ENV: Dict[str, Optional[str]]
- ○ TOML_CONF: dict

- ○ TAB_USER_OAUTH_CONNECTION: str
- ○ TAB_CONNECTIONS: str
- ○ TAB_VERIFICATION: str
- ○ REDIRECT_URI: str
- ○ EMAIL_VERIFICATION_DELAY: int

- ○ _get_environement_variable(env, var_name): str
- ○ _get_toml_variable(toml, section, key, default): Any

**OAuth Flow:**
1. Generate authorization URL
2. User authorizes at provider
3. Exchange code for token
4. Get user information
5. Store tokens in database
6. Periodically renew tokens

**Password Security:**
- Bcrypt with salt rounds
- Constant-time comparison
- Automatic salt generation
- Prevents timing attacks
- Prevents rainbow tables

**Shutdown Process:**
1. Schedule in background
2. Send response to client
3. Close database
4. Stop background tasks
5. Graceful server stop

**Configuration Sources:**
- .env file (secrets)
- os.environ (system vars)
- config.toml (structure)

Priority: .env > os.environ

**OAuth Providers**

**© Facebook OAuth**
- ○ authorization_url
- ○ token_url
- ○ user_info_url

**© GitHub OAuth**
- ○ authorization_url
- ○ token_url
- ○ user_info_url

**© Google OAuth**
- ○ authorization_url
- ○ token_url
- ○ user_info_url
- ○ client_id
- ○ client_secret
- ○ scope

Providers configured
in database table:
TAB_USER_OAUTH_CONNECTION

**Security Libraries**

**© uuid**
- ○ uuid4(): UUID

**© bcrypt**
- ○ gensalt(rounds): bytes
- ○ hashpw(password, salt): bytes
- ○ checkpw(password, hashed): bool

**Core Dependencies**

**© SQL**
- ○ get_data_from_table(): Union[int, List]
- ○ insert_data_into_table(): int
- ○ remove_data_from_table(): int
- ○ is_connected(): bool
- ○ close(): None
- ○ datetime_to_string(): str
- ○ string_to_datetime(): datetime

**© RuntimeManager**
- ○ get(cls): Any
- ○ set(cls, **kwargs): None
- ○ get_if_exists(cls, default): Any

**© RuntimeControl**
- ○ continue_running: bool
- ○ server: Optional[Any]
- ○ graceful_shutdown(): None

**© BackgroundTasks**
- ○ safe_start(): int
- ○ safe_stop(): int

**External APIs**

**© requests**
- ○ post(url, data, headers): Response
- ○ get(url, headers): Response

**© ENV**

**© TOML_CONF**

Edge labels: integrates, integrates, integrates, state generation, password hashing, database operations, password verification, retrieves instances, database management, manages instances, server control, HTTP requests, background jobs, loads, loads