

msaSDK Module

.auth.auth.auth

Classes

Auth

Bases: `Generic[_UserModelT]`

Attributes

`backend` instance-attribute

```
backend = self.backend or AuthBackend(
    self, token_store or DbTokenStore(self.db)
)
```

`db` instance-attribute

```
db = db or self.db
```

`pwd_context` instance-attribute

```
pwd_context = pwd_context
```

`user_model` instance-attribute

```
user_model = user_model or self.user_model
```

Functions

`__init__`

```
__init__(
    db: Union[AsyncDatabase, Database],
    token_store: BaseTokenStore = None,
```

```

    user_model: Type[_UserModelT] = User,
    pwd_context: CryptContext = CryptContext(
        schemes=["bcrypt"], deprecated="auto"
    ),
)

```

authenticate_user async

```

authenticate_user(
    username: str, password: Union[str, SecretStr]
) -> Optional[_UserModelT]

```

create_role_user async

```

create_role_user(role_key: str = 'admin') -> User

```

get_current_user

```

get_current_user()

```

requires

```

requires(
    roles: Union[str, Sequence[str]] = None,
    groups: Union[str, Sequence[str]] = None,
    permissions: Union[str, Sequence[str]] = None,
    status_code: int = 403,
    redirect: str = None,
    response: Union[bool, Response] = None,
) -> Callable

```

AuthBackend

Bases: `AuthenticationBackend`, `Generic[_UserModelT]`

Attributes

auth instance-attribute

```

auth = auth

```

token_store instance-attribute

```

token_store = token_store

```

Functions

`__init__`

```
__init__(auth: Auth, token_store: BaseTokenStore)
```

`attach_middleware`

```
attach_middleware(app: FastAPI)
```

Add auth middleware

`authenticate` async

```
authenticate(
    request: Request,
) -> Tuple[Auth, Optional[_UserModelT]]
```

`get_user_token` staticmethod

```
get_user_token(request: Request) -> Optional[str]
```

AuthRouter

Bases: `MSARouterMixin`

Attributes

`auth` instance-attribute

```
auth = auth or self.auth
```

`router_prefix` class-attribute

```
router_prefix = '/auth'
```

`schema_user_info` instance-attribute

```
schema_user_info = (
    self.schema_user_info
    or schema_create_by_schema(
        self.auth.user_model,
        "UserInfo",
    )
)
```

```
        exclude={"password"},
    )
)
```

schema_user_login_out class-attribute

```
schema_user_login_out: Type[UserLoginOut] = UserLoginOut
```

Classes

OAuth2

Bases: OAuth2PasswordBearer

Functions

`__call__` async

```
__call__(request: Request) -> Optional[str]
```

Functions

`__init__`

```
__init__(auth: Auth = None)
```

route_gettoken property

```
route_gettoken()
```

route_logout property

```
route_logout()
```

route_userinfo property

```
route_userinfo()
```

router_path

```
router_path() -> str
```

Last update: September 13, 2022

Created: September 13, 2022