







## ***AuthProvider***

**Description:**

Abstract base class for authentication providers.

**Inherits from:**

abc.ABC

**Attributes:**

None.

**Methods:**

*authenticate*(username: str, credentialHash: str) → Session

**Description:**

Attempts to create a new session using the given username and credentials.

**Decorators:**

Virtual, static.

**Pre-conditions:**

N/A, depends on implementation.

**Post-conditions:**

If credentials are valid, creates and returns a new 'Session' object associated with the given user.

If credentials are invalid, returns 'None'.

**Side effects:**

Once initialized, the 'Session' object registers itself in the database by adding a new row to the 'sessions' table.

## DummyAuthProvider

**Description:**

Authentication provider which validates credentials against a local 'users' database table.

In a production environment, this probably wouldn't be used.

**Inherits from:**

*AuthProvider*

**Attributes:**

None.

**Methods:**

authenticate(username: str, credentialHash: str) → Session

**Description:**

Attempts to create a new session using the given username and credentials. Validates credentials against a local 'users' database table.

**Decorators:**

Static.

**Pre-conditions:**

Local database table 'users' exists.

**Post-conditions:**

If credentials are valid, creates and returns a new 'Session' object associated with the given user.

If credentials are invalid, returns 'None'.

**Side effects:**

Once initialized, the 'Session' object registers itself in the database by adding a new row to the 'sessions' table.

## UWM1LoginAuthProvider

**Description:**

Authentication provider which validates credentials against the UWM 1Login SSO gateway.

Implementation of this class is not within the scope of this course. Therefore, all methods will raise `NotImplementedError`. Hypothetical implementation details are documented below.

**Inherits from:**

*AuthProvider*

**Attributes:**

None.

**Methods:**

`authenticate(username: str, credentialHash: str) → Session`

**Description:**

Attempts to create a new session using the given username and credentials. Validates credentials against the UWM 1Login SSO gateway.

**Decorators:**

Static.

**Pre-conditions:**

None.

**Post-conditions:**

If credentials are valid, creates and returns a new 'Session' object associated with the given user.

If credentials are invalid, returns 'None'.

If the Shibboleth server cannot be reached, a `ConnectionError` exception is raised.

**Side effects:**

Once initialized, the 'Session' object registers itself in the database by adding a new row to the 'sessions' table.

## Session

### Description:

Holds information about the user's current session.

### Inherits from:

None.

### Attributes:

sessionID: uuid.UUID

**Description:** Unique key which represents this particular session.

sessionLastRenewed: datetime.datetime

**Description:** Date and time of the last server request made by this session. If the delta between the this time and the current time is longer than some defined expiration duration, the session should be invalidated.

user: User

**Description:** User which this session belongs to.

### Methods:

None.

## ***Notifiable***

**Description:**

Abstract base class for classes which describe individuals or groups who can be sent notifications.

**Inherits from:**

abc.ABC

**Attributes:**

None.

**Methods:**

*notify*(message: email.message.Message) → None

**Description:**

Sends the given message via the system mail service to the recipients represented by this object.

Note that this method assumes ‘best-effort’ delivery on behalf of the system mail service. Sent notifications will simply be added to the mail queue, without any regard to whether they are *actually* delivered.

**Decorators:**

Abstract.

**Pre-conditions:**

System mail service exists. ‘message’ is not None.

**Post-conditions:**

The given notification will be sent to the recipients represented by this object.

**Side effects:**

None.



## ***Assignable***

### **Description:**

Abstract base class for classes which describe individuals that can be assigned to Teachable objects (currently, instances of Class and Section).

### **Inherits from:**

abc.ABC

### **Attributes:**

assignedTo: List[Teachable]

**Description:** List of Teachable objects assigned to the individual associated with this object.

### **Methods:**

None.

## *User*

**Description:**

Abstract base class for users.

**Inherits from:**

*Notifiable*

**Attributes:**

userID: uuid.UUID

**Description:** Unique key which represents this particular user.

enabled: bool

**Description:** Indicates whether this user should be enabled or not. If enabled, the user can log in and interact with the system. If disabled, the user cannot log in, and any active sessions are immediately terminated on next connection.

name: str

**Description:** User's full name.

address: str

**Description:** User's full address.

phone: str

**Description:** User's primary phone number.

email: str

**Description:** User's primary email address – typically the one provided by UWM.

altEmail: str

**Description:** User's alternate email address.

**Methods:**

notify(message: email.message.Message) → None

Implemented from *Notifiable*.

**Description:**

Sends the given message via the system mail service to the user's primary and alternate email addresses.

Note that this method assumes 'best-effort' delivery on behalf of the system mail service. Sent notifications will simply be added to the mail queue, without any regard to whether they are *actually* delivered.

**Decorators:**

None.

**Pre-conditions:**

System mail service exists. 'message' is not None.

**Post-conditions:**

The given notification will be sent to the user's primary and alternate email addresses.

**Side effects:**

None.

## AdministratorUser

**Description:**

Administrative user. Can access and modify the entire system.

**Inherits from:**

*User*

**Attributes:**

None.

**Methods:**

None.

## InstructorUser

**Description:**

Instructor user. Can manage allocated TAs. Can manage assigned classes and courses.

**Inherits from:**

*User*

**Attributes:**

manages: List[TAUser]

**Description:** List of TAUsers that have been allocated to this instructor.

**Methods:**

None.

## TAUser

**Description:**

TA user. Can be allocated to professors and assigned to courses and sections.

**Inherits from:**

*User*

**Attributes:**

managedBy: List[InstructorUser]

**Description:** List of InstructorUsers that manage this user.

**Methods:**

None.

---

## ***Teachable***

### **Description:**

Abstract base class for classes which describe teachable objects that can be assigned to Assignable objects (currently, instances of InstructorUser and TAUser).

### **Inherits from:**

abc.ABC

### **Attributes:**

teachableID: uuid.UUID

**Description:** Unique key which represents this particular teachable object.

assigned: List[Assignable]

**Description:** List of Assignable objects that are assigned to this teachable object.

### **Methods:**

None.

## Course

**Description:**

A course in the catalog.

**Inherits from:**

*Teachable, Notifiable*

**Attributes:**

name: str

**Description:** Course's name / course code.

description: str

**Description:** Course's description.

sections: List[Section]

**Description:** List of this course's sections.

**Methods:**

notify(message: email.message.Message) → None

Implemented from *Notifiable*.

**Description:**

Sends the given message via the system mail service to all users assigned to this course and its sections.

Note that this method assumes 'best-effort' delivery on behalf of the system mail service. Sent notifications will simply be added to the mail queue, without any regard to whether they are *actually* delivered.

**Decorators:**

None.

**Pre-conditions:**

System mail service exists. 'message' is not None.

**Post-conditions:**

The given notification will be sent to all users assigned to this course and its sections.

**Side effects:**

None.

## Section

**Description:**

One section of a course in the catalog.

**Inherits from:**

*Teachable, Notifiable*

**Attributes:**

name: str

**Description:** Section's name / section code.

course: Course

**Description:** Course which this section belongs to.

**Methods:**

notify(message: email.message.Message) → None

Implemented from *Notifiable*.

**Description:**

Sends the given message via the system mail service to all users assigned to this section.

Note that this method assumes 'best-effort' delivery on behalf of the system mail service. Sent notifications will simply be added to the mail queue, without any regard to whether they are *actually* delivered.

**Decorators:**

None.

**Pre-conditions:**

System mail service exists. 'message' is not None.

**Post-conditions:**

The given notification will be sent to all users assigned to this section.

**Side effects:**

None.