

# Documentazione

## SchoolAPIClient:

La classe SchoolAPIClient è responsabile della gestione della comunicazione con l'API di School Secretary. Conformata al protocollo SchoolSecretaryAPI, definendo metodi per varie interazioni con l'API, come il ping al server, il recupero delle informazioni sulle aule, l'aggiunta, l'aggiornamento e l'eliminazione di aule.

## Panoramica:

SchoolAPIClient agisce come un singleton, garantendo un'unica istanza condivisa per l'accesso all'API. Incapsula la logica per la costruzione e l'invio delle richieste HTTP al server di School Secretary e gestisce le risposte corrispondenti.

## Conformità al Protocollo:

- SchoolSecretaryAPI: Il protocollo definisce un insieme di metodi per interagire con l'API di School Secretary. SchoolAPIClient implementa questi metodi, fornendo funzionalità concrete per ciascuna chiamata API.

## URL di Base e Autenticazione:

L'URL di base e il token di accesso per l'API sono memorizzati come proprietà private. Questi valori vengono utilizzati per costruire gli URL per i diversi endpoint dell'API e per includere gli header di autorizzazione necessari nelle richieste HTTP.

## URLSession:

SchoolAPIClient utilizza la classe URLSession per effettuare richieste di rete asincrone. Incorpora un'istanza condivisa di URLSession, che fornisce un ambiente di rete comune per tutte le chiamate API.

## Metodi dell'API:

1. Ping: Verifica lo stato di connessione del server inviando una semplice richiesta di ping. Il gestore di completamento restituisce un Result indicando successo o errore.
2. Get Classrooms: Recupera un elenco di aule dal server. Il gestore di completamento restituisce un Result contenente un array di oggetti Classroom o un errore.
3. Add Classroom: Aggiunge una nuova aula al server. Il gestore di completamento restituisce un Result contenente l'oggetto Classroom aggiunto o un errore.

4. Update Classroom: Aggiorna una classe esistente sul server con informazioni modificate. Il gestore di completamento restituisce un Result contenente l'oggetto Classroom aggiornato o un errore.

5. Get Classroom: Recupera informazioni dettagliate su una specifica aula utilizzando il suo identificatore univoco. Il gestore di completamento restituisce un Result contenente l'oggetto Classroom richiesto o un errore.

6. Delete Classroom: Cancella un'aula dal server in base al suo identificatore univoco. Il gestore di completamento restituisce un Result indicando il successo dell'operazione di cancellazione o un errore.

Enum NetworkError:

È definita un'enum chiamata NetworkError per gestire potenziali errori che possono verificarsi durante le operazioni di rete. Include casi per situazioni in cui non viene ricevuto alcun dato o la risposta del server è considerata non valida.

Utilizzo:

Per interagire con l'API di School Secretary, gli sviluppatori possono utilizzare l'istanza condivisa di SchoolAPIClient e chiamare i suoi metodi, passando parametri e gestori di completamento appropriati.

Esempio:

```
SchoolAPIClient.shared.ping { result in
    switch result {
    case .success(let isConnected):
        print("Il server è (isConnected ? "connected" :
"disconnected")")
    case .failure(let error):
        print("Errore: (error.localizedDescription)")
    }
}
```

/\*

ClassroomManager

La classe ClassroomManager è responsabile della gestione e sincronizzazione delle informazioni sulle aule tra l'applicazione e il backend di School Secretary. Utilizza il Singleton SchoolAPIClient

per effettuare operazioni di rete come il recupero, l'aggiunta, l'aggiornamento e l'eliminazione di aule.

#### Panoramica:

ClassroomManager è un oggetto osservabile (ObservableObject) che tiene traccia delle aule locali e gestisce la comunicazione con il backend per garantire la coerenza dei dati. Espone metodi per eseguire operazioni CRUD sulle aule, garantendo la sincronizzazione con il backend School Secretary.

#### Proprietà:

- classrooms: Una variabile pubblicata che tiene traccia delle aule locali.
- showAddClassroomSheet: Una variabile che indica se mostrare o meno la schermata per l'aggiunta di una nuova aula.
- networkManager: Un'istanza condivisa di SchoolAPIClient per la gestione delle operazioni di rete.

#### Metodi Principali:

1. getClassrooms(): Restituisce un array di aule locali.
2. addClassroom(classroom:completion:): Aggiunge una nuova aula sia localmente che nel backend. Il gestore di completamento restituisce un Result indicando il successo o l'errore dell'operazione.
3. updateClassroom(classroom:completion:): Aggiorna le informazioni di un'aula sia localmente che nel backend. Il gestore di completamento restituisce un Result indicando il successo o l'errore dell'operazione.
4. deleteClassroom(classroom:): Elimina un'aula sia localmente che nel backend.
5. syncWithBackend(completion:): Sincronizza le informazioni sulle aule locali con il backend. Recupera le aule dal backend e le aggiorna localmente, garantendo l'assenza di duplicati. Il gestore di

completamento restituisce un eventuale errore durante la sincronizzazione.

Metodi di Supporto:

- `fetchClassroomsFromBackend(completion:)`: Recupera le informazioni sulle aule dal backend.

Utilizzo:

Per utilizzare `ClassroomManager`, è possibile creare un'istanza e chiamare i suoi metodi per interagire con le informazioni sulle aule. Ad esempio, per aggiungere un'aula, è possibile chiamare il metodo `addClassroom`.

Esempio:

```
let classroomManager = ClassroomManager()
classroomManager.addClass(classroom: newClassroom)
{ result in
    switch result {
    case .success:
        print("Aula aggiunta con successo")
    case .failure(let error):
        print("Errore durante l'aggiunta dell'aula:
(error.localizedDescription)")
    }
}
```

/\*

`ClassroomsListView`

La vista `ClassroomsListView` è responsabile della visualizzazione delle aule all'interno dell'applicazione `School Secretary`. Consente agli utenti di visualizzare, aggiungere, sincronizzare e navigare nelle informazioni sulle aule.

Panoramica:

`ClassroomsListView` è implementato utilizzando il framework `SwiftUI` ed è collegato a un oggetto osservabile `ClassroomManager`. Visualizza una lista di aule, consente l'aggiunta di nuove aule e

fornisce la funzionalità di sincronizzazione con il backend School Secretary.

#### Caratteristiche Principali:

- Visualizzazione delle aule: Mostra una lista di aule utilizzando la vista SwiftUI List, con ogni cella colorata in modo distintivo.
- Aggiunta di aule: Fornisce un pulsante "Add classroom" che attiva la schermata di aggiunta di una nuova aula.
- Sincronizzazione con il backend: Consente agli utenti di sincronizzare le informazioni sulle aule con il backend mediante la pressione di un pulsante "Sync".
- Navigazione alle informazioni dettagliate: Permette agli utenti di navigare alle informazioni dettagliate di ciascuna aula.

#### Metodi Principali:

- `syncWithBackend()`: Avvia la procedura di sincronizzazione con il backend School Secretary, aggiornando le informazioni sulle aule locali.
- `syncClassrooms()`: Gestisce la logica di fusione tra le aule locali e remote durante la sincronizzazione.
- `mergeClassrooms(local:remote:)`: Unisce le aule locali con quelle remote, garantendo l'assenza di duplicati.
- `updateLocalClassrooms(local:remote:)`: Aggiorna le aule locali con le informazioni più recenti provenienti dal backend.
- `addNewClassroomsToLocal(local:remote:)`: Aggiunge nuove aule locali al backend, se necessario.

#### Utilizzo:

ClassroomsListView viene utilizzata come parte dell'interfaccia utente principale dell'app School Secretary per gestire le

informazioni sulle aule. È integrata con ClassroomManager per gestire le operazioni CRUD e la sincronizzazione con il backend.