

Profile Module Documentation

1. Overview

The **Profile** module is responsible for managing user profiles in the messaging application. It stores and manages user-related information such as:

- Full Name
- Bio
- Avatar (Profile Picture)
- Birth Date

The module also provides methods for **serialization** (`toJson`) and **deserialization** (`fromJson`), enabling easy data exchange with the server or local storage.

2. Class Definition

2.1 Class: Profile

Represents a user profile.

Constructors

Constructor	Description
<code>Profile()</code>	Default constructor, initializes all fields to empty strings.
<code>explicit Profile(const std::string& jsonStr)</code>	Initializes a profile object from a JSON string.

2.2 Getter and Setter Methods

Method	Description
<code>std::string getFullName() const</code>	Returns the full name.
<code>void setFullName(const std::string& name)</code>	Sets the full name.
<code>std::string getBio() const</code>	Returns the bio.
<code>void setBio(const std::string& bio)</code>	Sets the bio.
<code>std::string getAvatarUrl() const</code>	Returns the avatar URL.
<code>void setAvatarUrl(const std::string& url)</code>	Sets the avatar URL.
<code>std::string getBirthDate() const</code>	Returns the birth date.
<code>void setBirthDate(const std::string& date)</code>	Sets the birth date.

2.3 Serialization and Deserialization

Method	Description
<code>nlohmann::json toJson() const</code>	Converts the profile object to a JSON object.
<code>void fromJson(const nlohmann::json& j)</code>	Populates the profile object from a JSON object.
<code>void fromJson(const std::string& jsonStr)</code>	Populates the profile object from a JSON string.

2.4 Validation

Method	Description
<code>bool isValid() const</code>	Checks if the profile is valid. Currently, it ensures that the <code>fullName</code> field is not empty.

3. Dependencies

- [nlohmann/json](#) for JSON handling
- C++ Standard Library (`<string>`)