### Signup API

#### User Module

* **UserRepository**
  + Manages operations for user entities in the database.
  + Method: **createUser(userDetails: User): Promise<void>**
* **UserService**
  + Organizes user-related operations, utilizing UserRepository.
  + Method**: signupUser(userDetails: User): Promise<void>**

#### Default Data Module

* **DefaultDataRepository**
  + Manages default data operations for a user in the database.
  + Method**: populateDefaultData(userId: string): Promise<void>**
* **DefaultDataService**
  + Modifies default data operations, utilizing DefaultDataRepository.
  + Method**: populateDefaultData(userId: string): Promise<void>**

#### Directory Module

* **DirectoryRepository**
  + Manages file system operations for creating user directories.
  + Method: **createUserDirectory(userId: string): Promise<void>**
* **DirectoryService**
  + Updates directory-related operations, utilizing DirectoryRepository.
  + Method: **createUserDirectory(userId: string): Promise<void>**

#### Notification Module

* **NotificationService**
  + Sends notifications to users based on different scenarios.
  + Methods:
    - **notifyUserOnSuccess(userId: string): Promise<void>**
    - **notifyUserOnError(userId: string, errorMessage: string): Promise<void>**

#### Authentication Module

* **SignupService**
  + Organizes the signup process.
  + Method: **signupUser(userDetails: User): Promise<void>**
* **SignupController**
  + Handles HTTP requests related to the signup process.
  + Method: **signupUser(request: Request, response: Response): Promise<void>**