

REST

Based on the codebase created during the previous module, implement follow REST API (as a RestController):

1. Trainee Registration (POST method)
 - a. Request
 - I. First Name (required)
 - II. Last Name (required)
 - III. Date of Birth (optional)
 - IV. Address (optional)
 - b. Response
 - I. Username
 - II. Password
2. Trainer Registration (POST method)
 - a. Request
 - I. First Name (required)
 - II. Last Name (required)
 - III. Specialization (required) (Training type reference)
 - b. Response
 - I. Username
 - II. Password
3. Login (GET method)
 - a. Request
 - I. Username (required)
 - II. Password (required)
 - b. Response
 - I. 200 OK
4. Change Login (PUT method)
 - a. Request
 - I. Username (required)
 - II. Old Password (required)
 - III. New Password (required)
 - b. Response
 - I. 200 OK
5. Get Trainee Profile (GET method)
 - a. Request
 - I. Username (required)
 - b. Response
 - I. First Name
 - II. Last Name
 - III. Date of Birth
 - IV. Address
 - V. Is Active
 - VI. Trainers List

1. Trainer Username
 2. Trainer First Name
 3. Trainer Last Name
 4. Trainer Specialization (Training type reference)
6. Update Trainee Profile (PUT method)
 - a. Request
 - I. Username (required)
 - II. First Name (required)
 - III. Last Name (required)
 - IV. Date of Birth (optional)
 - V. Address (optional)
 - VI. Is Active (required)
 - b. Response
 - I. Username
 - II. First Name
 - III. Last Name
 - IV. Date of Birth
 - V. Address
 - VI. Is Active
 - VII. Trainers List
 1. Trainer Username
 2. Trainer First Name
 3. Trainer Last Name
 4. Trainer Specialization (Training type reference)
7. Delete Trainee Profile (DELETE method)
 - a. Request
 - I. Username (required)
 - b. Response
 - I. 200 OK
8. Get Trainer Profile (GET method)
 - a. Request
 - I. Username (required)
 - b. Response
 - I. First Name
 - II. Last Name
 - III. Specialization (Training type reference)
 - IV. Is Active
 - V. Trainees List
 1. Trainee Username
 2. Trainee First Name
 3. Trainee Last Name
9. Update Trainer Profile (PUT method)
 - a. Request
 - I. Username (required)
 - II. First Name (required)
 - III. Last Name (required)

- IV. Specialization (read only) (Training type reference)
 - V. Is Active (required)
 - b. Response
 - I. Username
 - II. First Name
 - III. Last Name
 - IV. Specialization (Training type reference)
 - V. Is Active
 - VI. Trainees List
 - 1. Trainee Username
 - 2. Trainee First Name
 - 3. Trainee Last Name
- 10. Get not assigned on trainee active trainers. (GET method)
 - a. Request
 - I. Username (required)
 - b. Response
 - I. Trainer Username
 - II. Trainer First Name
 - III. Trainer Last Name
 - IV. Trainer Specialization (Training type reference)
- 11. Update Trainee's Trainer List (PUT method)
 - a. Request
 - I. Trainee Username
 - II. Trainers List (required)
 - 1. Trainer Username (required)
 - b. Response
 - I. Trainers List
 - 1. Trainer Username
 - 2. Trainer First Name
 - 3. Trainer Last Name
 - 4. Trainer Specialization (Training type reference)
- 12. Get Trainee Trainings List (GET method)
 - a. Request
 - I. Username (required)
 - II. Period From (optional)
 - III. Period To (optional)
 - IV. Trainer Name (optional)
 - V. Training Type (optional)
 - b. Response
 - I. Training Name
 - II. Training Date
 - III. Training Type
 - IV. Training Duration
 - V. Trainer Name
- 13. Get Trainer Trainings List (GET method)
 - a. Request

- I. Username (required)
 - II. Period From (optional)
 - III. Period To (optional)
 - IV. Trainee Name (optional)
- b. Response
 - I. Training Name
 - II. Training Date
 - III. Training Type
 - IV. Training Duration
 - V. Trainee Name
- 14. Add Training (POST method)
 - a. Request
 - I. Trainee username (required)
 - II. Trainer username (required)
 - III. Training Name (required)
 - IV. Training Date (required)
 - V. Training Duration (required)
 - b. Response
 - I. 200 OK
- 15. Activate/De-Activate Trainee (PATCH method)
 - a. Request
 - I. Username (required)
 - II. Is Active (required)
 - b. Response
 - I. 200 OK
- 16. Activate/De-Activate Trainer (PATCH method)
 - a. Request
 - I. Username (required)
 - II. Is Active (required)
 - b. Response
 - I. 200 OK
- 17. Get Training types (GET method)
 - a. Request -no data
 - b. Response
 - I. Training types
 - 1. Training type
 - 2. Training type Id

Notes:

1. During Create Trainer/Trainee profile username and password should be generated as described in previous modules.
2. Not possible to register as a trainer and trainee both.

3. All functions except Create Trainer/Trainee profile. Should be executed only after Trainee/Trainer authentication (on this step should be checked username and password matching).
4. Implement required validation for each endpoint.
5. Users Table has parent-child (one to one) relation with Trainer and Trainee tables.
6. Training functionality does not include delete/update possibility via REST
7. Username cannot be changed.
8. Trainees and Trainers have many to many relations.
9. Activate/De-activate Trainee/Trainer profile not idempotent action.
10. Delete Trainee profile is hard deleting action and bring the cascade deletion of relevant trainings.
11. Training duration have a number type.
12. Training Date, Trainee Date of Birth have Date type.
13. Is Active field in Trainee/Trainer profile has Boolean type.
14. Training Types table include constant list of values and could not be updated from the application.
15. Implement error handling for all endpoints.
16. Cover code with unit tests.
17. Two levels of logging should be implemented - transactions and each operation transaction level - which endpoint was called, which request came and the service response - 200 or error and response message + at this level, a transactionId is generated, by which you can track all operations for this transaction the same transactionId can later be passed to downstream services.
18. Implement error handling.
19. Document methods in RestController file(s) using Swagger 2 annotations.