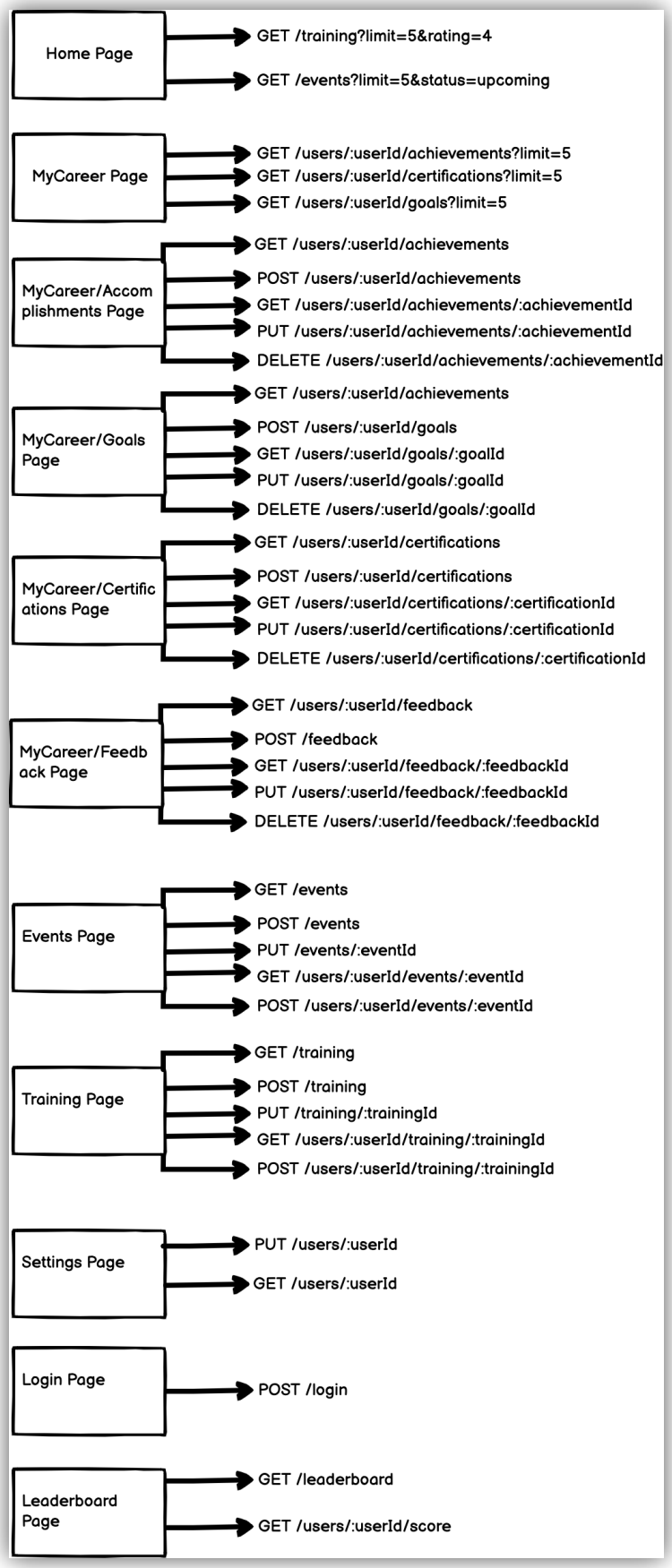
Endpoint Usage (Updated):

The below diagram showcases the pages within the Employee Development Dashboard and the necessary calls that will/could take place on the associated page.


Home Page
GET /training?limit=5&rating=4
GET /events?limit=5&status=upcoming
MyCareer Page
GET /users/:userId/achievements?limit=5
GET /users/:userId/certifications?limit=5
GET /users/:userId/goals?limit=5
MyCareer/Accomplishments Page
GET /users/:userId/achievements
POST /users/:userId/achievements
GET /users/:userId/achievements/:achievementId
PUT /users/:userId/achievements/:achievementId
DELETE /users/:userId/achievements/:achievementId
MyCareer/Goals Page
GET /users/:userId/achievements
POST /users/:userId/goals
GET /users/:userId/goals/:goalId
PUT /users/:userId/goals/:goalId
DELETE /users/:userId/goals/:goalId
MyCareer/Certifications Page
GET /users/:userId/certifications
POST /users/:userId/certifications
GET /users/:userId/certifications/:certificationId
PUT /users/:userId/certifications/:certificationId
DELETE /users/:userId/certifications/:certificationId
MyCareer/Feedback Page
GET /users/:userId/feedback
POST /feedback
GET /users/:userId/feedback/:feedbackId
PUT /users/:userId/feedback/:feedbackId
DELETE /users/:userId/feedback/:feedbackId
Events Page
GET /events
POST /events
PUT /events/:eventId
GET /users/:userId/events/:eventId
POST /users/:userId/events/:eventId
Training Page
GET /training
POST /training
PUT /training/:trainingId
GET /users/:userId/training/:trainingId
POST /users/:userId/training/:trainingId
Settings Page
PUT /users/:userId
GET /users/:userId
Login Page
POST /login
Leaderboard Page
GET /leaderboard
GET /users/:userId/score

Endpoints/Descriptions (Updated):

The below is according to the swagger specification. It describes all of the endpoints, their parameters, and their responses.

```
swagger: "2.0"
info:
  description: |
    This is a document describing the interactions of that are possible with the Employee Development Dashboard microservice.
  version: 1.0.0
  title: Employee Development Dashboard
  contact:
    name: Devon King
    email: dking3@live.maryville.edu
host: virtserver.swaggerhub.com
basePath: /dking74/Employee-Development-Dashboard/1.0.0
tags:
- name: user
  description: Metadata of users in system
- name: training
  description: Access links of training videos
- name: event
  description: Access events that have been created by admin
- name: achievement
  description: Documents an achievement that a user completes
- name: certification
  description: Showcases certification that user completed
- name: goal
  description: Goals that a user has to achieve
- name: feedback
  description: Feedback that user receives/gives to other users
- name: score
  description: Score that user obtains from completing objectives and progressing themselves
schemes:
- https
- http
paths:
  /users:
    get:
      tags:
      - user
```

```
  summary: Get all users from the database
  operationId: getAllUsers
  produces:
  - application/json
  parameters:
  - name: first_name
    in: query
    description: The first name of a user
    required: false
    type: string
  - name: last_name
    in: query
    description: The last name of a user
    required: false
    type: string
  - name: status
    in: query
    description: The status of the users to return
    required: false
    type: boolean
  responses:
    "200":
      description: Successful retrieval of all users
    "500":
      description: Internal error within service
      schema:
        $ref: '#/definitions/ErrorResponse'
post:
  tags:
  - user
  summary: Add a new user to the system
  operationId: addUser
  consumes:
  - application/json
  produces:
  - application/json
  parameters:
  - in: body
    name: body
    description: User object that needs to be added to the system
    required: true
    schema:
      $ref: '#/definitions/User'
  responses:
```

```
    "201":
      description: Successfully created user
      schema:
        $ref: '#/definitions/User'
    "404":
      description: Body contained properties that were invalid
      schema:
        $ref: '#/definitions/ErrorResponse'
    "500":
      description: Unexpected server error occurred
      schema:
        $ref: '#/definitions/ErrorResponse'
  put:
    tags:
    - user
    summary: Update an existing user
    operationId: updateUser
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - in: body
      name: body
      description: User object that needs to be updated in store
      required: true
      schema:
        $ref: '#/definitions/User'
    responses:
      "200":
        description: Successfully able to update the user
      "400":
        description: Invalid ID supplied in the body of the request
        schema:
          $ref: '#/definitions/ErrorResponse'
      "404":
        description: User could not be found
        schema:
          $ref: '#/definitions/ErrorResponse'
/users/{userId}:
  get:
    tags:
    - user
    summary: Get a user by id
```

```
  description: Returns a single user
  operationId: getUserById
  produces:
  - application/json
  parameters:
  - name: userId
    in: path
    description: ID of user to return
    required: true
    type: integer
    format: int64
  responses:
    "200":
      description: Successfully able to return user
      schema:
        $ref: '#/definitions/User'
    "400":
      description: Invalid ID supplied
      schema:
        $ref: '#/definitions/ErrorResponse'
    "404":
      description: User unable to be found with cre
      schema:
        $ref: '#/definitions/ErrorResponse'
put:
  tags:
  - user
  summary: Update an existing user in path
  operationId: updateSpecificUser
  consumes:
  - application/json
  produces:
  - application/json
  parameters:
  - name: userId
    in: path
    description: User id to delete
    required: true
    type: integer
    format: int64
  responses:
    "200":
      description: Successfully able to update the user
    "400":
```

```
      description: Invalid ID supplied in the body of the request
      schema:
        $ref: '#/definitions/ErrorResponse'
    "404":
      description: User could not be found
      schema:
        $ref: '#/definitions/ErrorResponse'
  delete:
   tags:
   - user
   summary: Deletes a specific user
   operationId: deleteUser
   produces:
   - application/json
   parameters:
   - name: userId
     in: path
     description: User id to delete
     required: true
     type: integer
     format: int64
   responses:
     "204":
       description: No content response indicating user has been deleted
     "400":
       description: Invalid ID supplied
       schema:
         $ref: '#/definitions/ErrorResponse'
     "404":
       description: User unable to be found
       schema:
         $ref: '#/definitions/ErrorResponse'
/users/{userId}/achievements:
  get:
   tags:
   - user
   - achievement
   summary: Get a users achievements
   description: Get all the achievements for a specific user
   operationId: getUserAchievements
   produces:
   - application/json
   parameters:
   - name: userId
```

```
    in: path
    description: User id to delete
    required: true
    type: integer
    format: int64
  - name: limit
    in: query
    description: The maximum number of results to return
    required: false
    type: integer
    format: int64
  responses:
    "200":
      description: Successful retrieval of user accomplishments
post:
  tags:
  - user
  - achievement
  summary: Create a user achievement
  description: Create a new achievement for a single user
  operationId: createUserAchievement
  consumes:
  - application/json
  produces:
  - application/json
  parameters:
  - name: userId
    in: path
    description: User id to delete
    required: true
    type: integer
    format: int64
  - in: body
    name: body
    description: Details of the achievement of user
    required: false
    schema:
      $ref: '#/definitions/Achievement'
  responses:
    "201":
      description: Successfully created user achievement
    "404":
      description: Unable to use the request parameters to create a new user achievement
      schema:
```

```
        $ref: '#/definitions/ErrorResponse'
/users/{userId}/achievements/{achievementId}:
  get:
    tags:
    - user
    - achievement
    summary: Get user achievement
    description: Get a specific achievement for a specific user
    operationId: getUserAchievement
    produces:
    - application/json
    parameters:
    - name: userId
      in: path
      description: User id to retrieve
      required: true
      type: integer
      format: int64
    - name: achievementId
      in: path
      description: Id of the achievement to update
      required: true
      type: integer
      format: int64
    responses:
      "200":
        description: User achievement returned successfully
      "404":
        description: Unable to retrieve specific user achievement
        schema:
          $ref: '#/definitions/ErrorResponse'
  put:
    tags:
    - user
    - achievement
    summary: Update a user achievement
    description: Update anachievement for specific user
    operationId: updatedUserAchievement
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - name: userId
```

```
      in: path
      description: User id to update
      required: true
      type: integer
      format: int64
    - name: achievementId
      in: path
      description: Id of the achievement to update
      required: true
      type: integer
      format: int64
    - in: body
      name: body
      description: Details of the achievement of user
      required: false
      schema:
        $ref: '#/definitions/Achievement'
    responses:
      "200":
        description: Successfully updated user achievement
      "404":
        description: Unable to use the request parameters to update an existing user
achievement
        schema:
          $ref: '#/definitions/ErrorResponse'
  delete:
    tags:
    - user
    - achievement
    summary: Delete user achievement
    description: Delete a specific achievement for a specific user
    operationId: deleteUserAchievement
    parameters:
    - name: userId
      in: path
      description: User id to retrieve
      required: true
      type: integer
      format: int64
    - name: achievementId
      in: path
      description: Id of the achievement to update
      required: true
      type: integer
```

```
      format: int64
    responses:
      "204":
        description: User achievement successfully deleted
      "404":
        description: Unable to retrieve specific user achievement
        schema:
          $ref: '#/definitions/ErrorResponse'
  /users/{userId}/certifications:
    get:
      tags:
      - user
      - certification
      summary: Get a users certifications
      description: Get all the certifications for a specific user
      operationId: getUserCertifications
      produces:
      - application/json
      parameters:
      - name: userId
        in: path
        description: User id to delete
        required: true
        type: integer
        format: int64
      - name: limit
        in: query
        description: The maximum number of results to return
        required: false
        type: integer
        format: int64
      responses:
        "200":
          description: Successful retrieval of user certifications
    post:
      tags:
      - user
      - certification
      summary: Create a user certication
      description: Create a new certification for a single user
      operationId: createUserCertification
      consumes:
      - application/json
      produces:
```

```
    - application/json
    parameters:
    - name: userId
      in: path
      description: User id to delete
      required: true
      type: integer
      format: int64
    - in: body
      name: body
      description: Details of the achievement of user
      required: false
      schema:
        $ref: '#/definitions/Certification'
    responses:
      "201":
        description: Successfully created user certification
      "404":
        description: Unable to use the request parameters to create a new user certification
        schema:
          $ref: '#/definitions/ErrorResponse'
/users/{userId}/certifications/{certificationId}:
  get:
    tags:
    - user
    - certification
    summary: Get user certification
    description: Get a specific certification for a specific user
    operationId: getUserCertification
    produces:
    - application/json
    parameters:
    - name: userId
      in: path
      description: User id to retrieve
      required: true
      type: integer
      format: int64
    - name: certificationId
      in: path
      description: Id of the certification to get
      required: true
      type: integer
      format: int64
```

```
  responses:
    "200":
      description: User certification returned successfully
      schema:
        $ref: '#/definitions/Certification'
    "404":
      description: Unable to retrieve specific user certification
      schema:
        $ref: '#/definitions/ErrorResponse'
put:
  tags:
  - user
  - certification
  summary: Update a user certification
  description: Update certification for specific user
  operationId: updatedUserCertification
  consumes:
  - application/json
  produces:
  - application/json
  parameters:
  - name: userId
    in: path
    description: User id to update
    required: true
    type: integer
    format: int64
  - name: certificationId
    in: path
    description: Id of the certification to update
    required: true
    type: integer
    format: int64
  - in: body
    name: body
    description: Details of the certification of user
    required: false
    schema:
      $ref: '#/definitions/Certification'
  responses:
    "200":
      description: Successfully updated user certification
    "404":
      description: Unable to use the request parameters to update an existing user certification
```

```
      schema:
        $ref: '#/definitions/ErrorResponse'
  delete:
    tags:
    - user
    - certification
    summary: Delete user certification
    description: Delete a specific certification for a specific user
    operationId: deleteUserCertification
    parameters:
    - name: userId
      in: path
      description: User id to retrieve
      required: true
      type: integer
      format: int64
    - name: certificationId
      in: path
      description: Id of the certification to update
      required: true
      type: integer
      format: int64
    responses:
      "204":
        description: User certification successfully deleted
      "404":
        description: Unable to retrieve specific user certification
        schema:
          $ref: '#/definitions/ErrorResponse'
/users/{userId}/goals:
  get:
    tags:
    - user
    - goal
    summary: Get a users goals
    description: Get all the goals for a specific user
    operationId: getUserGoals
    produces:
    - application/json
    parameters:
    - name: userId
      in: path
      description: User id to delete
      required: true
```

```
    type: integer
    format: int64
  - name: limit
    in: query
    description: The maximum number of results to return
    required: false
    type: integer
    format: int64
  responses:
    "200":
      description: Successful retrieval of user goals
 post:
  tags:
  - user
  - goal
  summary: Create a user goal
  description: Create a new goal for a single user
  operationId: createUserGoal
  consumes:
  - application/json
  produces:
  - application/json
  parameters:
  - name: userId
    in: path
    description: User id to delete
    required: true
    type: integer
    format: int64
  - in: body
    name: body
    description: Details of the goal of user
    required: false
    schema:
      $ref: '#/definitions/Goal'
  responses:
    "201":
      description: Successfully created user goal
    "404":
      description: Unable to use the request parameters to create a new user goal
      schema:
        $ref: '#/definitions/ErrorResponse'
/users/{userId}/goals/{goalId}:
 get:
```

```
  tags:
  - user
  - goal
  summary: Get user goal
  description: Get a specific goal for a specific user
  operationId: getUserGoal
  produces:
  - application/json
  parameters:
  - name: userId
    in: path
    description: User id to retrieve
    required: true
    type: integer
    format: int64
  - name: goalId
    in: path
    description: Id of the goal to update
    required: true
    type: integer
    format: int64
  responses:
    "200":
      description: User goal returned successfully
    "404":
      description: Unable to retrieve specific user goal
      schema:
        $ref: '#/definitions/ErrorResponse'
put:
  tags:
  - user
  - goal
  summary: Update a user goal
  description: Update a goal for specific user
  operationId: updatedUserGoal
  consumes:
  - application/json
  produces:
  - application/json
  parameters:
  - name: userId
    in: path
    description: User id to update
    required: true
```

```
    type: integer
    format: int64
  - name: goalId
    in: path
    description: Id of the goal to update
    required: true
    type: integer
    format: int64
  - in: body
    name: body
    description: Details of the goal of user
    required: false
    schema:
      $ref: '#/definitions/Goal'
  responses:
    "200":
      description: Successfully updated user goal
    "404":
      description: Unable to use the request parameters to update an existing user goal
      schema:
        $ref: '#/definitions/ErrorResponse'
delete:
  tags:
  - user
  - goal
  summary: Delete user goal
  description: Delete a specific goal for a specific user
  operationId: deleteUserGoal
  parameters:
  - name: userId
    in: path
    description: User id to retrieve
    required: true
    type: integer
    format: int64
  - name: goalId
    in: path
    description: Id of the achievement to update
    required: true
    type: integer
    format: int64
  responses:
    "204":
      description: User goal successfully deleted
```

```
    "404":
      description: Unable to retrieve specific user goal
      schema:
        $ref: '#/definitions/ErrorResponse'
/users/{userId}/events:
  get:
    tags:
    - user
    - event
    summary: Get all user events
    description: Get the events that a user is related with
    operationId: getUserEvents
    produces:
    - application/json
    parameters:
    - name: userId
      in: path
      required: true
      type: string
    - name: status
      in: query
      description: The status of the event, whether 'registered' or 'attended'
      required: false
      type: string
    responses:
      "200":
        description: Successfully returned all events for a user
        schema:
          $ref: '#/definitions/UserEvent'
  post:
    tags:
    - user
    - event
    summary: Create user event
    description: Create a user to event relationship
    operationId: createUserEvent
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - name: userId
      in: path
      required: true
```

```
      type: string
    - in: body
      name: body
      required: true
      schema:
        $ref: '#/definitions/UserEvent'
    responses:
      "201":
        description: Successfully created user to event relationship
        schema:
          $ref: '#/definitions/UserEvent'
      "404":
        description: The body of the user-event request was poorly formed
        schema:
          $ref: '#/definitions/ErrorResponse'
/users/{userId}/events/{eventId}:
  get:
    tags:
    - user
    - event
    parameters:
    - name: userId
      in: path
      required: true
      type: string
    - name: eventId
      in: path
      required: true
      type: string
    responses:
      "200":
        description: Successfully retrieved specific user event
        schema:
          $ref: '#/definitions/UserEvent'
      "404":
        description: Unable to find userId/eventId combination
        schema:
          $ref: '#/definitions/ErrorResponse'
  put:
    tags:
    - user
    - event
    parameters:
    - name: userId
```

```
      in: path
      required: true
      type: string
    - name: eventId
      in: path
      required: true
      type: string
    - in: body
      name: body
      required: true
      schema:
        $ref: '#/definitions/UserEvent'
    responses:
      "200":
        description: Successfully updated user event
        schema:
          $ref: '#/definitions/UserEvent'
      "400":
        description: Unable to find userId/eventId combination
        schema:
          $ref: '#/definitions/ErrorResponse'
      "404":
        description: Bad request body sent
        schema:
          $ref: '#/definitions/ErrorResponse'
/users/{userId}/training:
  get:
    tags:
    - user
    - training
    summary: Get all user training videos
    description: Get the training videos that a user is related with
    operationId: getUserTraining
    produces:
    - application/json
    parameters:
    - name: userId
      in: path
      required: true
      type: string
    - name: status
      in: query
      description: The status of the video, whether 'watched' or 'saved' or 'starred'
      required: false
```

```
      type: string
    responses:
      "200":
        description: Successfully returned all videos for a user
        schema:
          $ref: '#/definitions/UserTraining'
  post:
    tags:
    - user
    - event
    summary: Create user training
    description: Create a user to training video relationship
    operationId: createUserTraining
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - name: userId
      in: path
      required: true
      type: string
    - in: body
      name: body
      required: true
      schema:
        $ref: '#/definitions/UserTraining'
    responses:
      "201":
        description: Successfully created user to training relationship
        schema:
          $ref: '#/definitions/UserTraining'
      "404":
        description: The body of the user-training request was poorly formed
        schema:
          $ref: '#/definitions/ErrorResponse'
/users/{userId}/training/{trainingId}:
  get:
    tags:
    - user
    - training
    summary: Get user training video
    operationId: getOneUserTraining
    parameters:
```

```
  - name: userId
    in: path
    required: true
    type: string
  - name: trainingId
    in: path
    required: true
    type: string
  responses:
    "200":
      description: Successfully retrieved specific user training video
      schema:
        $ref: '#/definitions/UserTraining'
    "400":
      description: Unable to find userId/trainingId combination
      schema:
        $ref: '#/definitions/ErrorResponse'
put:
  tags:
  - user
  - training
  parameters:
  - name: userId
    in: path
    required: true
    type: string
  - name: trainingId
    in: path
    required: true
    type: string
  - in: body
    name: body
    required: true
    schema:
      $ref: '#/definitions/UserTraining'
  responses:
    "200":
      description: Successfully updated user training
      schema:
        $ref: '#/definitions/UserTraining'
    "400":
      description: Unable to find userId/training combination
      schema:
        $ref: '#/definitions/ErrorResponse'
```

```
    "404":
      description: Bad request body sent
      schema:
        $ref: '#/definitions/ErrorResponse'
/events:
  get:
    tags:
    - event
    summary: Get all the events
    description: Get all the events that are in system
    operationId: getAllEvents
    produces:
    - applicaton/json
    parameters:
    - name: status
      in: query
      description: If the event is open or not
      required: false
      type: string
    - name: date
      in: query
      description: The date of the event
      required: false
      type: string
    responses:
      "200":
        description: Successfully returned all events
  post:
    tags:
    - event
    summary: Create event
    description: Create a new event in system
    operationId: createEvent
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - in: body
      name: body
      description: Properties needed for new event
      required: false
      schema:
        $ref: '#/definitions/Event'
```

```
    responses:
      "201":
        description: Successfully created new event
        schema:
          $ref: '#/definitions/Event'
      "404":
        description: Unable to create a new event with information provided
        schema:
          $ref: '#/definitions/ErrorResponse'
  /events/{eventId}:
    get:
      tags:
      - event
      summary: Get one event
      description: Get a single event from the event id
      operationId: getOneEvent
      produces:
      - application/json
      parameters:
      - name: eventId
        in: path
        required: true
        type: string
      responses:
        "200":
          description: Successfully returned the single event
          schema:
            $ref: '#/definitions/Event'
        "404":
          description: Unable to find the specific event
    delete:
      tags:
      - event
      summary: Delete event
      description: Delete a single event from the event id
      operationId: deleteEvent
      parameters:
      - name: eventId
        in: path
        required: true
        type: string
      responses:
        "204":
          description: Successfully returned the single event
```

```
    "404":
      description: Unable to find the specific event
/training:
  get:
    tags:
    - training
    summary: Get all the training videos
    description: Get all the training videos that are in system
    operationId: getAllTraining
    produces:
    - applicaton/json
    parameters:
    - name: title
      in: query
      description: The title of the video
      required: false
      type: string
    - name: category
      in: query
      description: The category of the video type
      required: false
      type: string
    responses:
      "200":
        description: Successfully returned all training videos
        schema:
          $ref: '#/definitions/Training'
  post:
    tags:
    - training
    summary: Create training video
    description: Create a new training video in system
    operationId: createTraining
    consumes:
    - application/json
    produces:
    - application/json
    parameters:
    - in: body
      name: body
      description: Properties needed for new video
      required: false
      schema:
        $ref: '#/definitions/Training'
```

```
    responses:
      "201":
        description: Successfully created new video
        schema:
          $ref: '#/definitions/Training'
      "404":
        description: Unable to create a new video with information provided
        schema:
          $ref: '#/definitions/ErrorResponse'
  /training/{trainingId}:
    get:
      tags:
      - training
      summary: Get one video
      description: Get a single video from the training video id
      operationId: getOneVideo
      produces:
      - application/json
      parameters:
      - name: trainingId
        in: path
        required: true
        type: string
      responses:
        "200":
          description: Successfully returned the single training video
          schema:
            $ref: '#/definitions/Training'
        "404":
          description: Unable to find the specific video
    delete:
      tags:
      - training
      summary: Delete training video
      description: Delete a single video from the event id
      operationId: deleteVideo
      parameters:
      - name: trainingId
        in: path
        required: true
        type: string
      responses:
        "204":
          description: Successfully returned the single video
```

```
    "404":
      description: Unable to find the specific video
/user/{userId}/score:
  get:
    tags:
    - user
    - score
    summary: Get the score of a single user
    operationId: getUserScore
    parameters:
    - name: userId
      in: path
      required: true
      type: string
    responses:
      "200":
        description: Successfully able to get user score
        schema:
          $ref: '#/definitions/Score'
  put:
    tags:
    - user
    - score
    summary: Update the score of a single user
    operationId: updateUserScore
    parameters:
    - name: userId
      in: path
      required: true
      type: string
    - in: body
      name: body
      required: true
      schema:
        $ref: '#/definitions/Score'
    responses:
      "200":
        description: Successfully able to update user score
        schema:
          $ref: '#/definitions/Score'
/feedback:
  post:
    tags:
    - user
```

```
    - feedback
    summary: Create feedback for user
    operationId: createFeedback
    parameters:
    - in: body
      name: body
      required: true
      schema:
        $ref: '#/definitions/Feedback'
    responses:
      "201":
        description: Successfully created feedback
        schema:
          $ref: '#/definitions/Feedback'
  /users/{userId}/feedback:
    get:
      tags:
      - user
      - feedback
      summary: Get a users feedback
      description: Get the feedback that is submitted on behalf of a user
      operationId: getUserFeedback
      parameters:
      - name: userId
        in: path
        required: true
        type: string
      responses:
        "200":
          description: Successfully retrieved user feedback
          schema:
            $ref: '#/definitions/Feedback'
  /leaderboard:
    get:
      tags:
      - user
      - score
      summary: Get the leaderboard results
      operationId: getLeaderboard
      parameters:
      - name: limit
        in: query
        required: false
        type: string
```

```
    responses:
      "200":
        description: Successfully retrieved leaderboard results
        schema:
          $ref: '#/definitions/Score'
  /login:
    post:
      tags:
      - user
      summary: Login to the portal with credentials
      operationId: loginToPortal
      parameters:
      - in: body
        name: body
        required: true
        schema:
          $ref: '#/definitions/LoginInput'
      responses:
        "200":
          description: Successfully was able to login to portal
definitions:
  LoginInput:
    type: object
    properties:
      username:
        type: string
      password:
        type: string
  User:
    type: object
    properties:
      userId:
        type: integer
        format: int64
      username:
        type: string
      email:
        type: string
      first_name:
        type: string
      last_name:
        type: string
      phone:
        type: string
```

```
    status:
      type: integer
      format: int32
      description: The status of the user
    score:
      type: integer
      format: int32
      description: The score of the user as they achieve accomplishments
Achievement:
  type: object
  properties:
    achievementId:
      type: integer
      format: int64
    userId:
      type: integer
      format: int64
      description: The id of the user that registered the achievement
    title:
      type: string
      description: The title of achievement
    summary:
      type: string
      description: The full description of the achievement
    completed_date:
      type: string
      description: The date the achievement was completed
    other_comments:
      type: string
      description: Other comments the user wishes to make about the achievement
Certification:
  type: object
  properties:
    certificationId:
      type: integer
      format: int64
    userId:
      type: integer
      format: int64
      description: The user who has the goal
    name:
      type: string
      description: Name of certification
    description:
```

```
      type: string
      description: Details of the certification
    link:
      type: string
      description: Link to certification
Goal:
  type: object
  properties:
    goalId:
      type: integer
      format: int64
    userId:
      type: integer
      format: int64
      description: The user who has the goal
    summary:
      type: string
      description: The outline of the goal
    completion_date:
      type: string
      description: The date the goal is to be achieved by
Event:
  type: object
  properties:
    eventId:
      type: integer
      format: int64
    title:
      type: string
      description: The main title of the event
    summary:
      type: string
      description: The detailed description of the event
    organizers:
      type: array
      description: The individuals who are response for contacting and organizing event
      items:
        type: string
    registered:
      type: number
      description: The number of people registered for the event
    capacity:
      type: number
      description: The number of people who can register for event
```

```
    location:
      type: string
      description: Where the event is taking place
Feedback:
  type: object
  properties:
    feedbackId:
      type: integer
      format: int64
    userId:
      type: integer
      format: int64
      description: The user that feedback is given to
    submittedBy:
      type: string
      description: The user that is submitting feedback
    feedbackType:
      type: integer
      format: int64
      description: The type of feedback being given
    submittedDate:
      type: string
      description: The date that the feedback is submitted
    description:
      type: string
      description: The text associated with the feedback
Score:
  type: object
  properties:
    userId:
      type: integer
      format: int64
      description: The user that has the score
    value:
      type: integer
      format: int64
      description: The value of the score for the user
UserEvent:
  type: object
  properties:
    userEventId:
      type: integer
      format: int64
    userId:
```

```
      type: integer
      format: int64
      description: The user that is associated with event
    eventId:
      type: integer
      format: int64
      description: The event that is being associated with user
    status:
      type: string
      description: The status of the user-event relationship
Training:
  type: object
  properties:
    trainingId:
      type: integer
      format: int64
    title:
      type: string
      description: The title of the video
    url:
      type: string
      description: The url to embed training video in
    keywords:
      type: array
      description: The keyword for tracking of the training video
      items:
        type: string
    category:
      type: string
      description: The specific category of the video
    views:
      type: number
      description: The number of views on the video
    rating:
      type: number
      description: The rating of the video
    numRatings:
      type: number
      description: The number of times the video has been rated
UserTraining:
  type: object
  properties:
    userTraingingId:
      type: integer
```

```
      format: int64
    userId:
      type: integer
      format: int64
      description: The user that is associated with video
    trainingId:
      type: integer
      format: int64
      description: The video that is being associated with user
    status:
      type: string
      description: The status of the user with video
  ErrorResponse:
   type: object
   properties:
    statusCode:
      type: integer
      format: int32
    type:
      type: string
    message:
      type: string
    details:
      type: string
externalDocs:
  description: Learn more about this page and Swagger
  url: http://swagger.io
```