

**jwTrainingAPI Documentation**

**By Jake Ward**

Student Number: 20808653

URL: <https://jw1448.brighton.domains/jwTrainingAPI>

Endpoint: **/jwTrainingAPI/article**

Method: **GET**

Description: Retrieves the details of a specific article from the database and renders it using an EJS template for articles.

Parameters:

* **articleId** (string, required): The ID of the article to retrieve.

Responses:

* **200 OK**: The article was successfully retrieved and rendered.
* **204 No Content**: The article ID provided does not match any article in the database.
* **400 Bad Request**: The articleId parameter was not provided or was empty.

Endpoint: **/jwTrainingAPI/login**

Method: **GET**

Description: Validates a user's login information using the jwTraining database. It takes the username and password as query parameters and returns the user's email, username, and job title if the credentials are valid.

Parameters:

* **username** (string, required): A string representing the user's username.
* **password** (string, required): A string representing the user's password.

Responses:

* **200 OK**: Returns the user's email, username, and job title in JSON format.
* **204 No Content**: If no user is found with the provided credentials.
* **400 Bad Request**: If the query parameters are missing or invalid.
* **500 Internal Server Error**: If there was an error while processing the request.

Example response:

|  |
| --- |
| {  "email": "example@gmail.com",  "username": "examplename1",  "job\_title": "intern" } |

Endpoint: **/jwTrainingAPI/signup**

Method: **POST**

Description: Creates a new user account on the jwTraining database. It takes the user's email, username, and password as request body parameters and returns the ID of the newly created user if successful.

Parameters:

* **email** (string, required): A string representing the user's email address.
* **username** (string, required): A string representing the user's desired username.
* **password** (string, required): A string representing the user's desired password.

Responses:

* **201 Created**: Returns the ID of the newly created user in JSON format.
* **400 Bad Request**: If the request body parameters are missing or invalid, or if the email or username are already in use.
* **500 Internal Server Error**: If there was an error while processing the request.

Example response:

|  |
| --- |
| {  "id": 10 } |

Endpoint: **/jwTrainingAPI**

Method: **DELETE**

Description: Would delete the resource located at /jwTrainingAPI but is instead rejected.

Parameters:

* **N/A**

Responses:

* **405 Method Not Allowed**: The request was rejected, as the method is not allowed.

Endpoint: **/jwTrainingAPI**

Method: **PUT**

Description: Would update the resource located at /jwTrainingAPI but is instead rejected.

Parameters:

* **N/A**

Responses:

* **405 Method Not Allowed**: The request was rejected, as the method is not allowed.

Endpoint: **/jwTrainingAPI/training**

Method: **GET**

Description: Retrieves the training articles assigned to a user. It requires a valid username and password in the request parameters. If the request is successful, it will return a status code of 200 and the user's assigned articles information in JSON format. If the user has no training articles assigned and not due or completed, it will return a status code of 204.

Parameters:

* **username** (string, required): The username of the authenticated user.
* **password** (string, required): The password of the authenticated user.

Responses:

* **200 OK**: The request was successful and returns the following JSON object.
* **204 No Content**: The request was successful, but there are no assigned articles.
* **400 Bad Request**: The request is invalid or missing required parameters.
* **401 Unauthorized**: The authentication credentials provided are invalid.
* **500 Internal Server Error**: An error occurred while processing the request.

Example Response:

|  |
| --- |
| {  "result": [  {  "articleId": 1,  "title": "Introduction to Data Science",  "description": "This article provides an overview of data science.",  "due\_date": "2023-03-31"  },  {  "articleId": 2,  "title": "Machine Learning Basics",  "description": "This article introduces basic concepts in machine learning.",  "due\_date": "2023-04-15"  },  ...  ],  "completed\_percentage": "25%" } |

Endpoint: **/jwTrainingAPI/training/search**

Method: **GET**

Description: Searches for articles in the database. It requires a valid username and password in the request parameters. The search results can be filtered by keyword, page, and page size to aid in pagination set up. If the request is successful, it will return a status code of 200 and the search results in JSON format. If there are no search results, it will return a status code of 204.

Parameters:

* **username** (string, required): The username of the authenticated user.
* **password** (string, required): The password of the authenticated user.
* **keyword** (string, optional): A keyword to search for in article titles, descriptions, and tags (default is a 0-length string).
* **page** (integer, optional): The page number of the search results to retrieve (default is 1).
* **pageSize** (integer, optional): The number of search results to include per page (default is 10).

Responses:

* **200 OK**: The request was successful and returns the following JSON object.
* **204 No Content**: The request was successful, but there are no matching search results.
* **400 Bad Request**: The request is invalid or missing required parameters.
* **401 Unauthorized**: The authentication credentials provided are invalid.
* **500 Internal Server Error**: An error occurred while processing the request.

Example Response:

|  |
| --- |
| {  "result": [  {  "articleId": 1,  "title": "Introduction to Data Science",  "tags": "data science, introduction",  "description": "This article provides an overview of data science."  },  {  "articleId": 2,  "title": "Machine Learning Basics",  "tags": "machine learning, basics",  "description": "This article introduces basic concepts in machine learning."  },  ...  ],  "overallTotal": 25,  "currentPage": 2,  "pageSize": 10  } |

Endpoint: **/jwTrainingAPI/assign\_role**

Method: **POST**

Description: Allows a user with an admin role to assign a new role to a target user.

Parameters:

* **username** (string, required): A required string representing the username of the admin user.
* **password** (string, required): A required string representing the password of the admin user.
* **targetuser** (string, required): A required string representing the username of the target user to assign the new role.
* **newrole** (string, required): A required string representing the new role to assign to the target user.

Responses:

* **200 OK**: Roles was successfully assigned, returns affected rows in JSON format.
* **204 No Content**: If no user is found with the provided target username, no roles were assigned.
* **400 Bad Request**: If the query parameters are missing or invalid.
* **401 Unauthorized**: If the credentials were not representative of an administrative account.
* **500 Internal Server Error**: If there was an error while processing the request.

Example response:

|  |
| --- |
| {  "affectedRows": 1, } |

Endpoint: **/jwTrainingAPI/assign\_role**

Method: **POST**

Description: Allows the user with an admin role to assign an article to all users with a specific job title.

Parameters:

* **username** (string, required): The username of the admin making the request.
* **password** (string, required): The password of the admin making the request.
* **articleId** (string, required): The ID of the article to assign as training for users.
* **job\_title** (string, required): The job title of the users to assign the article to.
* **deadline** (string/integer, required) The number of days until the article is due.

Responses:

* **200 OK**: The article was assigned successfully. Returns an object with the affectedRows property indicating the number of rows updated.
* **204 No Content**: If no valid users were found, no articles were assigned to users.
* **400 Bad Request**: If the query parameters are missing or invalid.
* **401 Unauthorized**: If the credentials were not representative of an administrative account.
* **500 Internal Server Error**: If there was an error while processing the request.

Example response:

|  |
| --- |
| {  "affectedRows": 1, } |

Endpoint: **/jwTrainingAPI/stats**

Method: **GET**

Description: Gets statistics about the user's progress, including the number of completed, missed, and uncompleted articles, as well as information about the next closest article to be completed.

Parameters:

* **username** (string, required): The username of the user making the request.
* **password** (string, required): The password of the user making the request.

Responses:

* **200 OK**: Request was successful, and the results are displayed in the following JSON format.
* **400 Bad Request**: If the query parameters are missing or invalid.
* **401 Unauthorized**: If the credentials were incorrect.
* **500 Internal Server Error**: If there was an error while processing the request.

Example response:

|  |
| --- |
| {  "completedArticles": 10,  "missedArticles": 2,  "uncompletedArticles": 3,  "closestArticle": {  "articleId": 15,  "title": "Advanced JavaScript Techniques",  "description": "Learn some advanced techniques for working with JavaScript.",  "due\_date": "2023-04-01T23:59:59.000Z"  } } |

Endpoint: **/jwTrainingAPI/article/submit**

Method: **POST**

Description: Submits the users answers for a quiz associated with an article. If the user's score is greater than or equal to 70%, the article is marked as completed in the database and their score is returned.

Parameters:

* **username** (string, required): The username of the user submitting their results.
* **password** (string, required): The password of the user submitting their results.
* **articleId** (string, required): The ID of the article.
* **answers** (string/array, required): An array that is converted to string by the API or a comma-separated string of integers representing the index of user's answers to the quiz questions.

Responses:

* **200 OK**: Request was successful, and the results are displayed in the following JSON format.
* **400 Bad Request**: If the query parameters are missing or invalid.
* **401 Unauthorized**: If the credentials were incorrect.
* **500 Internal Server Error**: If there was an error while processing the request.

Example response:

|  |
| --- |
| {  "percentScore": 85.7 } |

Endpoint: **/jwTrainingAPI/profile\_picture/upload**

Method: POST

Description: Uploads a profile picture for a user. It takes the user's username and password as body parameters, and the profile picture as a file. If the username and password are valid, the picture is saved in the /uploads/profilePictures directory with the format "username.fileExtension" and returns the picture's name, mimetype, and size in the response body.

Parameters:

* username (string, required): A string representing the user's username.
* password (string, required): A string representing the user's password.
* profilePicture (file, required): The user's profile picture.

Responses:

* 201 Created: The profile picture was successfully uploaded. Returns the name, mimetype, and size of the uploaded picture in the response body.
* 400 Bad Request: If the query parameters are missing or invalid, or if the file format is incorrect.
* 500 Internal Server Error: If there was an error while processing the request.

|  |
| --- |
| {  "status": **true**,  "message": "Image has been uploaded",  "data": {  "name": "testusername1.png",  "mimetype": "image/png",  "size": 164008  }  } |

Endpoint: **/jwTrainingAPI/profile\_picture/download**

Method: GET

Description: Downloads the user's profile picture. It takes the user's username and password as query parameters, searches for the picture in the /uploads/profilePictures directory, and returns the picture in the response body with the correct content type header.

Parameters:

* username (string, required): A string representing the user's username.
* password (string, required): A string representing the user's password.

Responses:

* 200 OK: Returns the user's profile picture in the response body with the correct content type header.
* 400 Bad Request: If the query parameters are missing or invalid.
* 404 Not Found: If the user does not have a profile picture uploaded.
* 500 Internal Server Error: If there was an error while processing the request.