11/27/2024

API Specification

Rafael Claudino Tavares

TEAM OR1ON

Table of Contents

API Specifications:

GET Requests:

GET	**users** needs token from user and to be an admin		
Retrieve all Users			
Parameters:	**Name**	**Type**	**Description**
* required		path	Get all Users
Responses:	**Code**	**Description / example if successful**	
	200	Return a list of Users	

GET	**users/{userID}** needs token from user		
Retrieve the User by ID			
Parameters:	**Name**	**Type**	**Description**
* required	userID	path	Id of the user to get the information
Responses:	**Code**	**Description / example if successful**	
	200	Returns the User Information with the ID that was provided.	
	404	User ID is incorrect	

GET	**users/{userID}/feedbacks** needs token from user		
Retrieve the all the Feedback by the User ID			
Parameters:	**Name**	**Type**	**Description**
* required	userId	path	Id of the user to get his feedbacks
Responses:	**Code**	**Description / example if successful**	
	200	Returns a list of all the Feedbacks from the selected User	
	404	User ID is incorrect	

GET	**routes** needs token from user		
Retrieve all Routes			
Parameters:	**Name**	**Type**	**Description**
* required		path	Get all Routes
Responses:	**Code**	**Description / example if successful**	
	200	Return a list of Routes	

GET	**routes/{routeID}** needs token from user		
Retrieve the Route by ID			
Parameters:	**Name**	**Type**	**Description**
** required*	routeID	path	Id of the route to get the information
Responses:	**Code**	**Description / example if successful**	
	200	Returns the Route Information with the ID that was provided.	
	404	Route ID is incorrect	

GET	**routes/{routeId}/highlights** needs token from user		
Retrieve the all the Highlight by the Route ID			
Parameters:	**Name**	**Type**	**Description**
** required*	routeId	path	Id of the route to get the information
Responses:	**Code**	**Description / example if successful**	
	200	Returns a list of all the Highlights from the selected Route	
	404	Route ID is incorrect	

GET	**routes/{routeId}/highlights/{highlightID}** needs token from user		
Retrieve the selected Highlight by the Route ID			
Parameters:	**Name**	**Type**	**Description**
** required*	routeId	path	Id of the route to get the information
	highlightID	path	Id of the selected Highlight
Responses:	**Code**	**Description / example if successful**	
	200	Returns the selected Highlight from the selected Route	
	404	Route ID is incorrect	

GET	**highlights** needs token from user		
Retrieve all Highlights			
Parameters:	**Name**	**Type**	**Description**
** required*		path	Get all Highlights
Responses:	**Code**	**Description / example if successful**	
	200	Return a list of Highlights	

GET	**highlights/{highlightID}** needs token from user		
Retrieve the Highlight by ID			
Parameters:	**Name**	**Type**	**Description**
* required	highlightID	path	Id of the highlight to get the information
Responses:	**Code**	**Description / example if successful**	
	200	Returns the Highlight Information with the ID that was provided.	
	404	Highlight ID is incorrect	

GET	**highlights/{highlightID}/feedbacks** needs token from user		
Retrieve the Feedbacks from selected highlights the by ID			
Parameters:	**Name**	**Type**	**Description**
* required	highlightID	path	Id of the highlight to get his feedbacks
Responses:	**Code**	**Description / example if successful**	
	200	Returns all the Feedbacks Information with the ID that was provided.	
	404	Highlight ID is incorrect	

GET	**sugestions** needs token from user		
Retrieve all Sugestions			
Parameters:	**Name**	**Type**	**Description**
* required		path	Get all Sugestions
Responses:	**Code**	**Description / example if successful**	
	200	Return a list of Sugestions	

GET	**sugestions/{sugestionID}** needs token from user		
Retrieve the Sugestion by ID			
Parameters:	**Name**	**Type**	**Description**
* required	sugestionID	path	Id of the Sugestion to get the information
Responses:	**Code**	**Description / example if successful**	
	200	Returns the Sugestion Information with the ID that was provided.	
	404	SugestionID is incorrect	

POST Requests:

POST	**auth/tokens**		
User will login with his credentials and will create a Token			
Parameters:	**Name**	**Type**	**Description**

* required	User	body	The user data to be added. Must be JSON with email and password.
			
			Example: {
			"email": "user01@gmail.com",
			"password": "user1"
			}

Responses:	**Code**	**Description / example if successful**	
	200	User Login successfully, token created and given to user	
	404	Invalid input data - Email or Password not found	

POST	**auth**		
A new User will be registered			
Parameters:	**Name**	**Type**	**Description**

* required	User	body	The user data to be added. Must be JSON with email and password.
			
			Example: {
			"username": "user01 ",
			"email": "user01@gmail.com",
			"password": "user1",
			"birthday": "10/10/2000"
			}

Responses:	**Code**	**Description / example if successful**	
	200	A new User will be created	
	404	Invalid input data - Email or Password not found	
	500	Internal server error with the error message.	

POST	**tours** needs token from user and to be admin		
A new Tour will be registered			
Parameters:	**Name**	**Type**	**Description**

* required	Tour	body	The tour data to be added. Must be JSON with the tour data.
			
			Example: {
			"name": "big tour",
			"type": "historical",
			"highlights": ["<highlight1>","<highlight2>",...]
			}

Responses:	**Code**	**Description / example if successful**	
	200	A new tour will be created	
	404	Invalid input data – Name, Type or Highlights not found	
	500	Internal server error with the error message.	

POST	**tour/{tourID}/feedbacks** needs token from user		
A new Tour Feedback will be registered			
Parameters:	**Name**	**Type**	**Description**
* required	tourID	path	Needs the Tour ID to add the Feedback

	tour	body	The Feedback data to be added. Must be JSON with the tour data.
			
			Example: {
			"rate": "5",
			"message": "Test… Test… Test… Test… Test… Test… Test… Test… Test… Test… "
			}

Responses:	**Code**		**Description / example if successful**
	200		A new Tour Feedback will be created
	404		Invalid input data – Rate or Message not found
	500		Internal server error with the error message.

POST	**highlights** needs token from user and to be admin		
A new Highlight will be registered			
Parameters:	**Name**	**Type**	**Description**

** required*	highlight	body	The tour data to be added. Must be JSON with the tour data.
			
			Example: {
			"name": "place1",
			"type": "historical",
			"localization": "5.56522, 3.35855"
			}

Responses:	**Code**	**Description / example if successful**	
	200	A new Highlights will be created	
	404	Invalid input data – Name, Type, Location not found	
	500	Internal server error with the error message.	

POST	**highlights/{highlightID}/feedbacks** needs token from user		
A new Highlight will be registered			
Parameters:	**Name**	**Type**	**Description**
** required*	highlightID	path	Needs the highlight ID to be edit

	feedback	body	The Feedback data to be added. Must be JSON with the tour data.
			
			Example: {
			"rate": "5",
			"message": "Test… Test… Test… Test… Test… Test… Test… Test… Test… Test… "
			}

Responses:	**Code**		**Description / example if successful**
	200		A new Feedback will be created
	404		Invalid input data – Highlight Rate or Message not found
	500		Internal server error with the error message.

POST	**suggestions** needs token from user		
A new Suggestion will be registered			
Parameters:	**Name**	**Type**	**Description**

** required*	suggestion	body	The tour data to be added. Must be JSON with the tour data.
			
			Example: {
			"name": "place12",
			"type": "historical",
			"localization": "5.56522, 3.35855"
			}

Responses:	**Code**	**Description / example if successful**	
	200	A new Suggestion will be created	
	404	Invalid input data – Name, Type, Location not found	
	500	Internal server error with the error message.	

PATCH Requests:

PUT	**users/{userID}** needs token from user		
Update the selected Tour by its ID			
Parameters:	**Name**	**Type**	**Description**
** required*	userID	path	Needs the User ID to be edit

	user	body	The User data to be edited. Must be JSON with all card information.
			
			Example: {
			"username": "user01 ",
			"email": "user01@gmail.com",
			"password": "user1"
			}

Responses:	**Code**	**Description / example if successful**	
	200	User updated successfully	
	400	All errors 400 will be town here	
	404	Card not found	

PUT	**tours/{tourID}** needs token from user and to be admin		
Update the selected Tour by its ID			
Parameters:	**Name**	**Type**	**Description**
* *required*	tourID	path	Needs the Tour ID to be edit

	tour	body	The Tour data to be edited. Must be JSON with all card information.
			
			Example: {
			"name": "big tour",
			"type": "historical",
			
			"highlights":["<highlight1>","<highlight2>",...]
			}

Responses:	**Code**	**Description / example if successful**	
	200	Tour updated successfully	
	400	All errors 400 will be town here	
	404	Card not found	

PUT	**highlights/{highlightID}** needs token from user and to be admin		
Update the selected Highlight by its ID			
Parameters:	**Name**	**Type**	**Description**
* *required*	tourID	path	Needs the Highlight ID to be edit

	tour	body	The Highlight data to be edited. Must be JSON with all card information.
			
			Example: {
			"name": "place1",
			"type": "historical",
			"localization": "5.56522, 3.35855"
			}

Responses:	**Code**	**Description / example if successful**	
	200	Highlight updated successfully	
	400	All errors 400 will be town here	
	404	Card not found	

/routes/{routeID}/highlights/{highlightID}:

PUT	**highlights/{highlightID}/feedbacks/{feedbackID}** needs token from user and to be admin		
Update the selected Feedback from the selected Highlight by its ID			
Parameters:	**Name**	**Type**	**Description**
** required*	highlightID	path	Needs the HighlightID to select
	feedbackID	path	Needs the Feedback ID to be edit

	feedback	body	The Feedback data to be edited. Must be JSON with all card information.
			
			Example: {
			"rate": "5",
			"message": "Test... Test... Test... Test... Test... Test... Test... Test... Test... Test... "
			}

Responses:	**Code**	**Description / example if successful**	
	200	Feedback updated successfully	
	400	All errors 400 will be town here	
	404	Card not found	

DELETE Requests:

DELETE	**users/{userID}** needs token from user and to be admin		
Deletes the User by his ID			
Parameters:	**Name**	**Type**	**Description**
** required*	userID	path	Needs the User ID to be deleted
Responses:	**Code**	**Description / example if successful**	
	204	User deleted successfully	
	404	User not found	

DELETE	**users/{userID}/feedback/{feedbackID}** needs token from user and to be admin		
Deletes a feedback by his ID from the given User			
Parameters:	**Name**	**Type**	**Description**
** required*	feedbackID	path	Needs the Feedback ID to be deleted
	userID	path	Needs the User ID to be selected
Responses:	**Code**	**Description / example if successful**	
	204	Feedback deleted successfully	
	404	Feedback or User not found	

DELETE	**users/{userID}/suggestion/{suggestionID}** needs token from user and to be admin		
Deletes a suggestion by his ID from the given User			
Parameters:	**Name**	**Type**	**Description**
** required*	suggestionID	path	Needs the suggestion ID to be deleted
	userID	path	Needs the User ID to be selected
Responses:	**Code**	**Description / example if successful**	
	204	Suggestion deleted successfully	
	404	Suggestion or User not found	

DELETE	**highlights/{highlightID}** needs token from user and to be admin		
Deletes the highlights by his ID			
Parameters:	**Name**	**Type**	**Description**
** required*	highlightsID	path	Needs the Highlights ID to be deleted
Responses:	**Code**	**Description / example if successful**	
	204	Highlights deleted successfully	
	404	Highlights not found	

DELETE	**tours/{tourID}** needs token from user and to be admin		
Deletes the Tour by his ID			
Parameters:	**Name**	**Type**	**Description**
** required*	tourID	path	Needs the Tour ID to be deleted
Responses:	**Code**	**Description / example if successful**	
	204	Tour deleted successfully	
	404	Tour not found	

DELETE	**routes/{routeID}/highlights/{highlightID}** needs token from user and to be admin		
Deletes the highlights by his ID			
Parameters:	**Name**	**Type**	**Description**
** required*	highlightsID	path	Needs the Highlights ID to be deleted
	highlightID	path	Id of the selected Highlight
Responses:	**Code**	**Description / example if successful**	
	204	Highlight deleted successfully	
	404	Highlight or Route not found	