

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Μάθημα Προπτυχιακών Σπουδών:
Υπηρεσιοστρεφές Λογισμικό
Ακαδημαϊκό έτος: 2023 - 2024
Εξάμηνο: 7ο
Απαλλακτική Εργασία

Ομάδα Εργασίας:

Θεόδωρος Κοζάνογλου Π20094
Απόστολος Σιαμπάνης Π20173

Υπεύθυνος Καθηγητής:

Ευάγγελος Σακκόπουλος

Εκφώνηση.....	3
Δημιουργία Βάσης Δεδομένων:.....	5
Παράδειγμα επιτυχούς εκτέλεσης του προγράμματος:.....	5
Παράδειγμα επιτυχούς εκτέλεσης του swagger:.....	6
Συνοδευτικό Εγχειρίδιο.....	7
Signup.....	7
Login.....	9
Logout.....	11
List all todos and todo items.....	11
Create a new todo.....	13
Get a todo.....	15
Update a todo.....	17
Delete a todo and its items.....	20
Get all todo items.....	22
Create a new todo item.....	24
Get a todo item.....	27
Update a todo item.....	29
Delete a todo item.....	32
Πηγές.....	34

Εκφώνηση

Δημιουργία REST API.

Δημιουργήστε ένα REST API με χρήση RAILS με τις παρακάτω λειτουργίες για μία [λίστα εκκρεμοτήτων](#) (todo) η οποία έχει λεπτομέρειες για κάθε todo (list of todo items).

Προσοχή: Η ανάπτυξή σας θα πρέπει να περιλαμβάνει τα κατάλληλα test (test driven development) καθώς και δοκιμή των test με httpie:

Θα ενεργοποιήσετε τα ακόλουθα endpoint για τις λειτουργικότητες αντίστοιχα.

<u>Endpoint</u>	<u>Functionality</u>
POST /signup	Signup
POST /auth/login	Login
GET /auth/logout	Logout
GET /todos	List all todos and todo items
POST /todos	Create a new todo
GET /todos/:id	Get a todo
PUT /todos/:id	Update a todo
DELETE /todos/:id	Delete a todo and its items
GET /todos/:id/items/:iid	Get a todo item
POST /todos/:id/items	Create a new todo item
PUT /todos/:id/items/:iid	Update a todo item
DELETE /todos/:id/items/:iid	Delete a todo item

- Η εφαρμογή θα συνοδεύεται από Εγχειρίδιο χρήσης (user manual) τύπου openapi [swagger](#)

- Η εργασία και τα δύο θέματα και θα πρέπει να γίνει από ομάδες των 1-2 ατόμων.
- Η υλοποίηση της εργασίας να γίνει με χρήση Ruby on Rails
- Η εργασία σας να εγκατασταθεί στο fly.io ή στο gitpod ή render η άλλο cloud της επιλογής σας

- Η εργασία μπορεί εναλλακτική να υλοποιηθεί με nodejs ή .net core
- Η ομάδα θα πρέπει να καταχωρήσει τον κώδικά της στο σύστημα github ή gitlab και να παρουσιάσει τον σχετικό κώδικα κατά την εξέταση. (δείτε σύντομο οδηγό github & <http://git-class.gr/>)
- Η ισχύς της παρούσας εργασίας είναι για το ακαδημαϊκό έτος 2023-2024

Η παράδοση της εργασίας θα πραγματοποιηθεί στο gunet2 στην ημερομηνία που αναφέρεται στο σύστημα. Θα περιλάβετε όλο το project σας καθώς και μία σύντομη αναφορά σε μορφή DOC/ODT/PDF με το εγχειρίδιο χρήσης ή το ορεπαρί. Θα περιλάβετε το ονοματεπώνυμο και το ΑΜ όλων των μελών της ομάδας σας. Η εξέταση της εργασίας θα πραγματοποιηθεί με οδηγίες που θα σας ανακοινωθούν.

Καλή Επιτυχία!

Τεχνολογίες που χρησιμοποιήθηκαν:

Development: Ruby (v3.3.0) on Rails (v7.1.3)

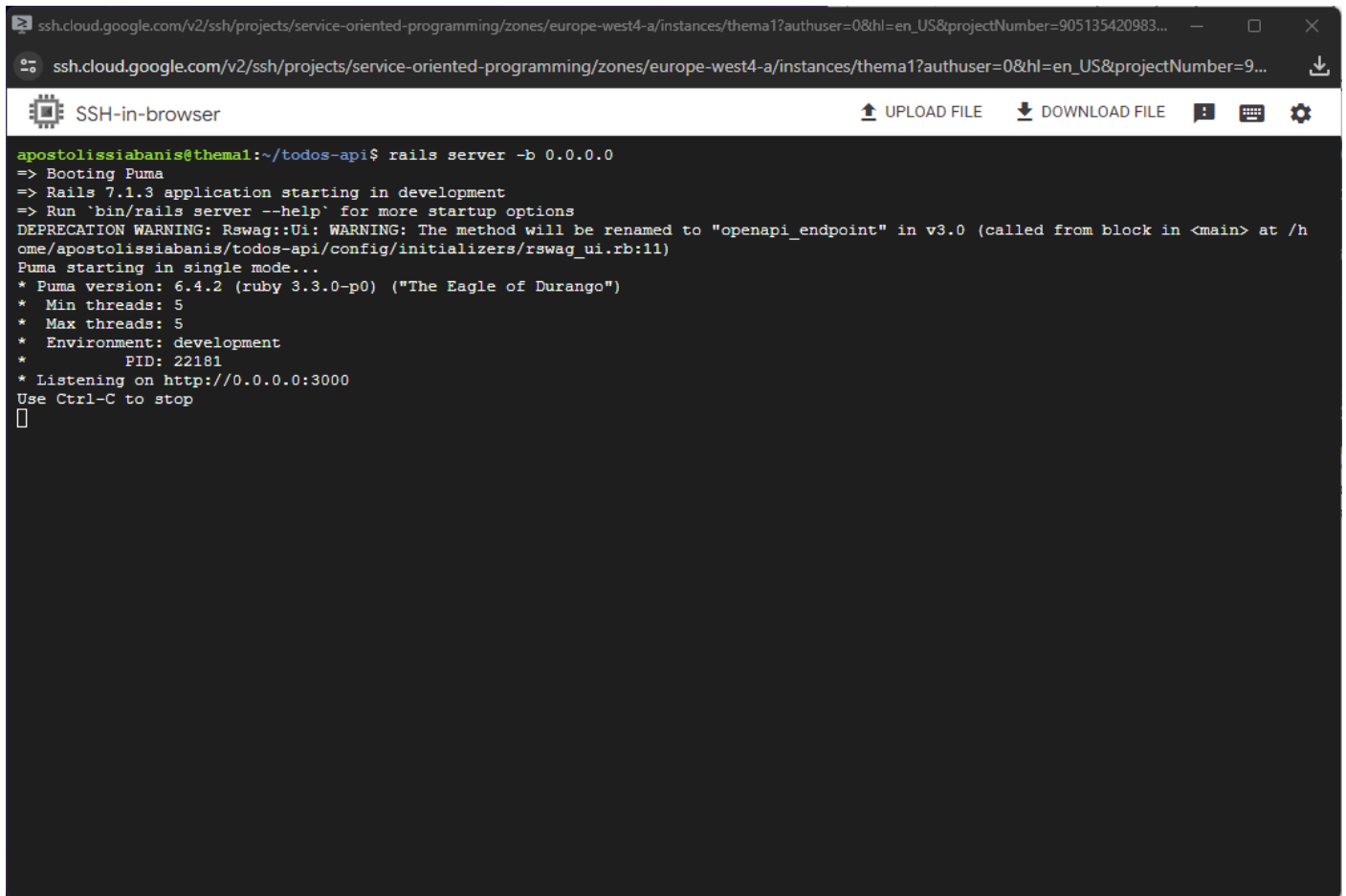
Development Tools: Visual Studio code

Database: SQLite

Δημιουργία Βάσης Δεδομένων:

- Εκτελούμε την εντολή rails db:create, ώστε να δημιουργηθεί η βάση δεδομένων μας.
- Ύστερα εκτελούμε την εντολή rails db:migrate ώστε να δημιουργηθούν οι πίνακες της βάσης δεδομένων μας.
- Προαιρετικά, μπορούμε να εκτελέσουμε την εντολή rails db:seed για την προσθήκη dummy δεδομένων για την demo εκτέλεση του προγράμματος

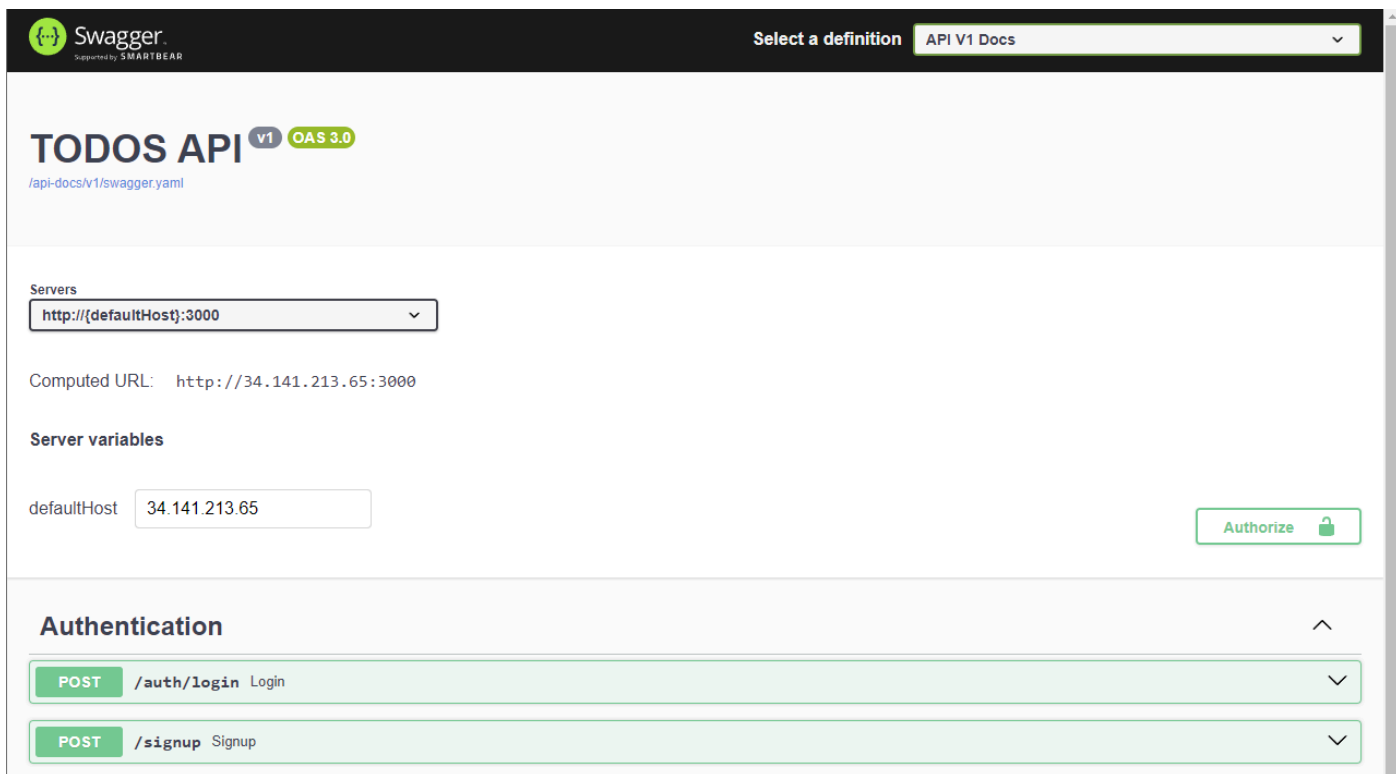
Παράδειγμα επιτυχούς εκτέλεσης του προγράμματος:



```
ssh.cloud.google.com/v2/ssh/projects/service-oriented-programming/zones/europe-west4-a/instances/thema1?authuser=0&hl=en_US&projectNumber=905135420983...
ssh.cloud.google.com/v2/ssh/projects/service-oriented-programming/zones/europe-west4-a/instances/thema1?authuser=0&hl=en_US&projectNumber=9...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
apostolissiabanis@thema1:~/todos-api$ rails server -b 0.0.0.0
=> Booting Puma
=> Rails 7.1.3 application starting in development
=> Run 'bin/rails server --help' for more startup options
DEPRECATION WARNING: Rswag::Ui: WARNING: The method will be renamed to "openapi_endpoint" in v3.0 (called from block in <main> at /home/apostolissiabanis/todos-api/config/initializers/rswag_ui.rb:11)
Puma starting in single mode...
* Puma version: 6.4.2 (ruby 3.3.0-p0) ("The Eagle of Durango")
* Min threads: 5
* Max threads: 5
* Environment: development
* PID: 22181
* Listening on http://0.0.0.0:3000
Use Ctrl-C to stop
[]
```

Ανοίγετε τον φυλλομετρητή της επιλογής σας και πληκτρολογείτε την διεύθυνση: <http://34.141.213.65:3000> ο server στεγάζεται από το google cloud, όμως επειδή έχουμε ένα REST API και δεν υπάρχει κάποιο user interface χρησιμοποιούμε swagger για να μπορούμε να διαχειριστούμε τα αιτήματα στα ανάλογα endpoints.

Παράδειγμα επιτυχούς εκτέλεσης του swagger:



The screenshot shows the Swagger UI for the 'TODO API'. The top bar includes the Swagger logo and a dropdown menu for 'Select a definition' with 'API V1 Docs' selected. The main header displays 'TODO API' with 'v1' and 'OAS 3.0' tags, and a link to '/api-docs/v1/swagger.yaml'. Below this, the 'Servers' section shows a dropdown menu with 'http://{defaultHost}:3000' selected. The 'Computed URL' is 'http://34.141.213.65:3000'. The 'Server variables' section shows 'defaultHost' with the value '34.141.213.65' and an 'Authorize' button. The 'Authentication' section is expanded, showing two endpoints: 'POST /auth/login Login' and 'POST /signup Signup', each with a dropdown arrow.

Swagger. Supported by SMARTBEAR

Select a definition API V1 Docs

TODO API v1 OAS 3.0

/api-docs/v1/swagger.yaml

Servers

http://{defaultHost}:3000

Computed URL: http://34.141.213.65:3000

Server variables

defaultHost 34.141.213.65

Authorize

Authentication

POST /auth/login Login

POST /signup Signup

Ανοίγεται τον φυλλομετρητή της επιλογής σας και πληκτρολογείτε την διεύθυνση:

<http://34.141.213.65:3000/api-docs>

Συνοδευτικό Εγχειρίδιο

Signup

POST

/signup Signup

Signup to generate a token

Parameters

No parameters

Request body

application/json

```
{
  "name": "",
  "email": "",
  "password": ""
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
'http://127.0.0.1:3000/signup' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "name": "",
  "email": "",
  "password": ""
}'
```

Request URL

```
http://127.0.0.1:3000/signup
```

Server response

Code	Details
422	<div>Error: Unprocessable Content</div> <div><div>Response body</div><div><pre>{ "message": "Validation failed: Password can't be blank, Name can't be blank, Email can't be blank, Password digest can't be blank" }</pre></div></div>

Αν τα στοιχεία που έδωσε ο χρήστης κατά την εγγραφή δεν είναι της σωστής μορφής εμφανίζεται το παραπάνω μήνυμα σφάλματος.

POST

/signup Signup

Signup to generate a token

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{
  "name": "apostolis",
  "email": "apo@email.com",
  "password": "password"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
'http://127.0.0.1:3000/signup' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "name": "apostolis",
  "email": "apo@email.com",
  "password": "password"
}'
```

Request URL

```
http://127.0.0.1:3000/signup
```

Server response

Code	Details
201	<div>Response body<pre>{ "message": "Account created successfully", "auth_token": "eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJleHAiOjE3MDc5ODk2MjF9.ztK3L6pN-xTTSQbPJ4m7YBmrIs4UicgI0YqNsJsDjzo" }</pre><div>Download</div></div>

Αν τα στοιχεία που έδωσε ο χρήστης κατά την εγγραφή είναι της σωστής μορφής εμφανίζεται το παραπάνω μήνυμα, μαζί με το token του χρήστη για την συγκεκριμένη σύνδεση.

Login

POST /auth/login Login

Login to generate a token

Parameters Cancel Reset

No parameters

Request body application/json

```
{
  "email": "",
  "password": ""
}
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \
'http://127.0.0.1:3000/auth/login' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "email": "",
  "password": ""
}'
```

Request URL

```
http://127.0.0.1:3000/auth/login
```

Server response

Code	Details
401	Error: Unauthorized

Response body

```
{
  "message": "Invalid credentials"
}
```

Download

Αν τα στοιχεία που έδωσε ο χρήστης κατά την είσοδο του δεν είναι σωστά εμφανίζεται το παραπάνω μήνυμα σφάλματος.

POST

/auth/login

Login

Login to generate a token

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{
  "email": "apo@email.com",
  "password": "password"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:3000/auth/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "apo@email.com",
    "password": "password"
  }'
```

Request URL

http://127.0.0.1:3000/auth/login

Server response

Code	Details
200	<div><div>Response body</div><pre>{ "auth_token": "eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJleHAiOjE3MDc5OTYyMTR9.w2yWnP-oHCfZ9nfck2ucW8gbfhEdZl2UiYprUxvRGA" }</pre><div><div>Download</div></div></div>

Αν τα στοιχεία που έδωσε ο χρήστης κατά την σύνδεση είναι σωστά εμφανίζεται το παραπάνω token του χρήστη για την συγκεκριμένη σύνδεση.

Logout

Καθώς η σύνδεση του χρήστη σχετίζεται με την χρήση του token, το οποίο λήγει μετά από ένα συγκεκριμένο χρονικό διάστημα (24ωρων ωρών) ο χρήστης αποσυνδέεται αυτόματα.

List all todos and todo items

GET

/todos

List all todos and todo items

🔒 ⤴

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \
'http://127.0.0.1:3000/todos' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZXQzLkIjLCJleHAiOiE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvfYvBKzVAqFBRVDRGLYs'

Request URL

http://127.0.0.1:3000/todos

Server response

Code

Details

200

Response body

[
{
 "id": 1,
 "title": "quae",
 "created_by": "1",
 "created_at": "2024-02-13T15:15:17.445Z",
 "updated_at": "2024-02-13T15:15:17.445Z",
 "items": [
 {
 "id": 1,
 "name": "molestiae",
 "done": false,
 "todo_id": 1,
 "created_at": "2024-02-13T15:15:17.483Z",
 "updated_at": "2024-02-13T15:15:17.483Z"
 },
 {
 "id": 51,
 "name": "THOMAS",
 "done": false,
 "todo_id": 1,
 "created_at": "2024-02-13T15:17:47.773Z",
 "updated_at": "2024-02-13T15:17:47.773Z"
 }
]
}
],

Download

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς, τότε θα του επιστραφούν όλα τα todos και όλα τα items για το κάθε todo αντίστοιχα.

GET

/todos List all todos and todo items

Cancel

No parameters

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:3000/todos' \
-H 'accept: */*' \
-H 'Authorization: aasda'
```

Request URL

```
http://127.0.0.1:3000/todos
```

Server response

Code	Details
422	Error: Unprocessable Content

Response body

```
{
  "message": "Not enough or too many segments"
}
```

Download

Αν ο χρήστης δεν χρησιμοποιήσει ή εισάγει λανθασμένα το token του, τότε θα επιστραφεί το παραπάνω σφάλμα με το ανάλογο μήνυμα κάθε φορά.

Create a new todo

POST

/todos Create a new todo

Parameters

Cancel

Reset

No parameters

Request body

application/json

{
 "title": "todo2"
}

Execute

Clear

Responses

Curl

curl -X 'POST' \
 'http://127.0.0.1:3000/todos' \
 -H 'accept: */*' \
 -H 'Authorization:eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOjE3MDgwMzU0OTZ9.xPXj7GnRW9J-n92nA5WZw0VvfYvBKzVAqFBRVDRGLYs' \
 -H 'Content-Type: application/json' \
 -d '{
 "title": "todo2"
 }'

Request URL

http://127.0.0.1:3000/todos

Server response

Code

Details

201

Response body

{
 "id": 52,
 "title": "todo2",
 "created_by": "1",
 "created_at": "2024-02-14T22:57:01.092Z",
 "updated_at": "2024-02-14T22:57:01.092Z",
 "items": []
}

Download

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς, τότε θα μπορέσει με επιτυχία να δημιουργήσει ένα νέο todo.

POST

/todos Create a new todo

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{
  "title": "todo2"
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
'http://127.0.0.1:3000/todos' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "title": "todo2"
}'
```

Request URL

```
http://127.0.0.1:3000/todos
```

Server response

Code	Details
422	Error: Unprocessable Content

Response body

```
{
  "message": "Missing token"
}
```

Download



Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Get a todo

GET

/todos/{id}

Get a todo



Parameters

Cancel

Name	Description
id * required	id
string (path)	<input type="text" value="2"/>

Execute

Clear

Responses


Curl

```
curl -X 'GET' \
'http://127.0.0.1:3000/todos/2' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJleHAiOiE3MDgwMzU0OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs'
```

Request URL

```
http://127.0.0.1:3000/todos/2
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": 2, "title": "blanditiis", "created_by": "1", "created_at": "2024-02-13T15:15:17.486Z", "updated_at": "2024-02-13T15:15:17.486Z", "items": [{ "id": 2, "name": "officiis", "done": false, "todo_id": 2, "created_at": "2024-02-13T15:15:17.489Z", "updated_at": "2024-02-13T15:15:17.489Z" }] }</pre></div><div>Download</div></div>

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα έγκυρο todo id τότε θα του επιστραφεί συγκεκριμένο todo που αναζητεί.

GET /todos/{id} Get a todo

Parameters

Name	Description
id * required	id
string (path)	1000

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:3000/todos/1000' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOiJlE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs'
```

Request URL

http://127.0.0.1:3000/todos/1000

Server response

Code	Details
404	Error: Not Found

Response body

```
{
  "message": "Couldn't find Todo with 'id'=1000"
}
```

Download

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα μη έγκυρο todo id τότε θα του επιστραφεί το παραπάνω μήνυμα.

GET /todos/{id} Get a todo

Parameters

Name	Description
id * required	id
string (path)	2

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:3000/todos/2' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOiJlE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs'
```

Request URL

http://127.0.0.1:3000/todos/2

Server response

Code	Details
422	Error: Unprocessable Content

Response body

```
{
  "message": "Missing token"
}
```

Download

Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Update a todo

PUT

/todos/{id} Update a todo

Parameters

Cancel

Reset

Name	Description
id * required	id
string	
(path)	1

Request body

application/json

{

"title": "test"

}

Execute

Clear

Responses

Curl

curl -X 'PUT' \

'http://127.0.0.1:3000/todos/1' \

-H 'accept: */*' \

-H 'Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJleHAiOjE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvfYv8KzVAqFBRVDRGLys' \

-H 'Content-Type: application/json' \

-d '{

"title": "test"

}'

Request URL

http://127.0.0.1:3000/todos/1

Server response

Code	Details
204	<div><div>Response headers</div><div>cache-control: no-cache</div><div>referrer-policy: strict-origin-when-cross-origin</div><div>server-timing: start_processing.action_controller;dur=0.01,sql.active_record;dur=69.67,instantiation.active_record;dur=0.07,unpermitted_parameters.action_controller;dur=0.00,transaction.active_record;dur=77.46,process_action.action_controller;dur=82.40</div><div>x-content-type-options: nosniff</div><div>x-frame-options: SAMEORIGIN</div><div>x-permitted-cross-domain-policies: none</div><div>x-request-id: 089fd602-f888-4d1e-baaf-d81c8fb4c2b7</div><div>x-runtime: 0.092021</div><div>x-xss-protection: 0</div></div>

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα έγκυρο todo id τότε θα μπορέσει με επιτυχία να ανανεώσει ένα todo.

PUT

/todos/{id} Update a todo

Cancel

Reset

Name	Description
id * required string (path)	<div>id</div> <div>1000</div>

Request body

application/json

```
{
  "title": "test"
}
```

ExecuteClear

Responses

Curl

```
curl -X 'PUT' \
  'http://127.0.0.1:3000/todos/1000' \
  -H 'accept: */*' \
  -H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOjE3MDgwMzU0OTZ9.xPKj7GmRW9J-n92nA5wZw0VvYvBKzVAqFBRVDRGLYs' \
  -H 'Content-Type: application/json' \
  -d '{
    "title": "test"
  }'
```

Request URL

http://127.0.0.1:3000/todos/1000

Server response

Code	Details
404	<div>Error: Not Found</div> <div>Response body<div><pre>{ "message": "Couldn't find Todo with 'id'=1000" }</pre></div><div>Download</div></div>

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα μη έγκυρο todo id τότε θα του επιστραφεί το παραπάνω μήνυμα.

PUT

/todos/{id} Update a todo

Parameters

Cancel

Reset

Name	Description
id * required string (path)	id <input type="text" value="1"/>

Request body

application/json

```
{  
  "title": "test"  
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \  
  'http://127.0.0.1:3000/todos/1' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "title": "test"  
  }'
```

Request URL

```
http://127.0.0.1:3000/todos/1
```

Server response

Code	Details
422	Error: Unprocessable Content Response body <pre>{ "message": "Missing token" }</pre> <div>Download</div>

Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Delete a todo and its items

DELETE

/todos/{id}

Delete a todo and its items

Parameters

Cancel

Name	Description
id * required	id
string	
(path)	5

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://127.0.0.1:3000/todos/5' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOiJlE3MDgwMzU0OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs'
```

Request URL

```
http://127.0.0.1:3000/todos/5
```

Server response

Code	Details
204	Response headers

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα έγκυρο todo id τότε θα μπορέσει με επιτυχία να διαγράψει ένα todo.

DELETE

/todos/{id}

Delete a todo and its items

Parameters

Cancel

Name	Description
id * required	id
string	
(path)	1000

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://127.0.0.1:3000/todos/1000' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOiJlE3MDgwMzU0OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs'
```

Request URL

```
http://127.0.0.1:3000/todos/1000
```

Server response

Code	Details
404	Error: Not Found

Response body

```
{
  "message": "Couldn't find Todo with 'id'=1000"
}
```

Download

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα μη έγκυρο todo id τότε θα μπορέσει με επιτυχία να διαγράψει ένα todo.

DELETE /todos/{id} Delete a todo and its items

Parameters

Cancel

Name	Description
id * required string (path)	id <input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://127.0.0.1:3000/todos/1' \
-H 'accept: */*'
```

Request URL

```
http://127.0.0.1:3000/todos/1
```

Server response

Code	Details
422	Error: Unprocessable Content Response body <pre>{ "message": "Missing token" }</pre>

Download



Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Get all todo items

GET

/todos/{todo_id}/items

Get all todo items

Parameters

Try it out

Name	Description
todo_id * required	todo_id
string	
(path)	
	<input type="text" value="1"/>

Responses


Curl

```
curl -X 'GET' \
  'http://127.0.0.1:3000/todos/1/items' \
  -H 'accept: */*' \
  -H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOjE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvfYvBKzVAqFBRVDRGLYs'
```

Request URL

http://127.0.0.1:3000/todos/1/items

Server response

Code	Details
200	<div><div>Response body</div><pre>[{ "id": 1, "name": "molestiae", "done": false, "todo_id": 1, "created_at": "2024-02-13T15:15:17.483Z", "updated_at": "2024-02-13T15:15:17.483Z" }, { "id": 51, "name": "THOMAS", "done": false, "todo_id": 1, "created_at": "2024-02-13T15:17:47.773Z", "updated_at": "2024-02-13T15:17:47.773Z" }, { "id": 52, "name": "test", "done": false, "todo_id": 1, "created_at": "2024-02-14T22:35:21.433Z", "updated_at": "2024-02-14T22:35:21.433Z" }]</pre><div> Download</div></div>

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα έγκυρο todo id τότε θα του επιστραφούν όλα τα items για το συγκεκριμένο todo που αναζητεί.

GET

/todos/{todo_id}/items

Get all todo items

🔒 ⬆

Parameters

Cancel

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="1000"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:3000/todos/1000/items' \
  -H 'accept: */*' \
  -H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJleHAiOjE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvfYvBKzVAqFBRVDRGLYs'
```

Request URL

http://127.0.0.1:3000/todos/1000/items

Server response

Code	Details
404	Error: Not Found

Response body

```
{
  "message": "Couldn't find Todo with 'id'=1000"
}
```

Download

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα μη έγκυρο todo id τότε θα του επιστραφεί το παραπάνω μήνυμα.

GET

/todos/{todo_id}/items

Get all todo items

🔒 ⬆

Parameters

Cancel

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:3000/todos/1/items' \
  -H 'accept: */*' \
  -H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJleHAiOjE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvfYvBKzVAqFBRVDRGLYs'
```

Request URL

http://127.0.0.1:3000/todos/1/items

Server response

Code	Details
422	Error: Unprocessable Content

Response body

```
{
  "message": "Missing token"
}
```

Download

Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Create a new todo item

POST /todos/{todo_id}/items Create a new todo item

Parameters

Cancel

Reset

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="1"/>

Request body

application/json

```
{  "name": "test",  "done": false}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \  'http://127.0.0.1:3000/todos/1/items' \  -H 'accept: */*' \  -H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOiE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nASWZw0VvfYvBKzVAqFBRVDRGLYs' \  -H 'Content-Type: application/json' \  -d '{  "name": "test",  "done": false}'
```

Request URL

http://127.0.0.1:3000/todos/1/items

Server response

Code	Details
201	<div><div>Response body</div><pre>{ "id": 1, "title": "quae", "created_by": "1", "created_at": "2024-02-13T15:15:17.445Z", "updated_at": "2024-02-13T15:15:17.445Z", "items": [{ "id": 1, "name": "molestiae", "done": false, "todo_id": 1, "created_at": "2024-02-13T15:15:17.483Z", "updated_at": "2024-02-13T15:15:17.483Z" }, { "id": 51, "name": "THOMAS", "done": false, "todo_id": 1, "created_at": "2024-02-13T15:17:47.773Z", "updated_at": "2024-02-13T15:17:47.773Z" }, { "id": 52, "name": "test", "done": false, "todo_id": 1, "created_at": "2024-02-13T15:17:47.773Z", "updated_at": "2024-02-13T15:17:47.773Z" }]}</pre></div> <div>Download</div>

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα έγκυρο todo id, τότε θα εισάγει με επιτυχία ένα νέο item στο συγκεκριμένο todo.

POST

/todos/{todo_id}/items

Create a new todo item

Parameters

Cancel

Reset

Name	Description
todo_id * required	todo_id
string	
(path)	<input type="text" value="1000"/>

Request body

application/json

```
{
  "name": "test",
  "done": false
}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://127.0.0.1:3000/todos/1000/items' \
  -H 'accept: */*' \
  -H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOiE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvfYvBKzVAqFBRVDRGLYs' \
  -H 'Content-Type: application/json' \
  -d '{
    "name": "test",
    "done": false
  }'
```

Request URL

http://127.0.0.1:3000/todos/1000/items

Server response

Code	Details
404	Error: Not Found

Response body

```
{
  "message": "Couldn't find Todo with 'id'=1000"
}
```



Download

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς, αλλά προσπαθήσει να εισάγει κάποιο item σε ένα todo id το οποίο δεν υπάρχει θα εμφανιστεί το παραπάνω μήνυμα.

POST

/todos/{todo_id}/items

Create a new todo item



Parameters

Cancel

Reset

Name	Description
todo_id * required string <small>(path)</small>	todo_id <input type="text" value="1"/>

Request body

application/json

```
{  
  "name": "test",  
  "done": false  
}
```

Execute

Clear

Responses


Curl

```
curl -X 'POST' \  
  'http://127.0.0.1:3000/todos/1/items' \  
  -H 'accept: */*' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "name": "test",  
    "done": false  
  }'
```

Request URL

http://127.0.0.1:3000/todos/1/items

Server response

Code	Details
422	Error: Unprocessable Content Response body <pre>{ "message": "Missing token" }</pre> <div> Download</div>

Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Get a todo item

GET

/todos/{todo_id}/items/{id}

Get a todo item

Parameters

Cancel

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="1"/>
id * required	id
string (path)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:3000/todos/1/items/1' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOiE3MDgwMzU0OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs'
```

Request URL

```
http://127.0.0.1:3000/todos/1/items/1
```

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "id": 1, "name": "molestiae", "done": false, "todo_id": 1, "created_at": "2024-02-13T15:15:17.483Z", "updated_at": "2024-02-13T15:15:17.483Z" }</pre></div><div> Download</div></div>

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα έγκυρο todo id και item id τότε θα του επιστραφεί το συγκεκριμένο item που αναζητεί.

GET

/todos/{todo_id}/items/{id}

Get a todo item

🔒 ⬆

Parameters

Cancel

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="100"/>
id * required	id
string (path)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:3000/todos/100/items/1' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZXQlIjoxLCJleHAiOiE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs'
```

Request URL

http://127.0.0.1:3000/todos/100/items/1

Server response

Code	Details
404	Error: Not Found

Response body

```
{
  "message": "Couldn't find Todo with 'id'=100"
}
```

Download

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα μη έγκυρο todo id ή item id τότε θα του επιστραφεί το παραπάνω μήνυμα.

GET

/todos/{todo_id}/items/{id}

Get a todo item

🔒 ⬆

Parameters

Cancel

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="1"/>
id * required	id
string (path)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://127.0.0.1:3000/todos/1/items/1' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZXQlIjoxLCJleHAiOiE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs'
```

Request URL

http://127.0.0.1:3000/todos/1/items/1

Server response

Code	Details
422	Error: Unprocessable Content

Response body

```
{
  "message": "Missing token"
}
```

Download

Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Update a todo item

PUT /todos/{todo_id}/items/{id} Update a todo item

Parameters

Cancel

Reset

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="1"/>
id * required	id
string (path)	<input type="text" value="1"/>

Request body

application/json

```
{
  "name": "test",
  "done": true
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
'http://127.0.0.1:3000/todos/1/items/1' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLCJleHAiOiE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs' \
-H 'Content-Type: application/json' \
-d '{
  "name": "test",
  "done": true
}'
```

Request URL

```
http://127.0.0.1:3000/todos/1/items/1
```

Server response

Code

Details

204

Response headers

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα έγκυρο todo id και item id τότε θα μπορέσει με επιτυχία να ανανεώσει ένα item το οποίο υπάρχει.

PUT

/todos/{todo_id}/items/{id} Update a todo item

Parameters

Cancel

Reset

Name	Description
todo_id * required string (path)	todo_id <input type="text" value="1000"/>
id * required string (path)	id <input type="text" value="1"/>

Request body

application/json

```
{
  "name": "test",
  "done": true
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
'http://127.0.0.1:3000/todos/1000/items/1' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjoxLjE3eHAiOiJlE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvFYvBKzVAqFBRVDRGLYs' \
-H 'Content-Type: application/json' \
-d '{
  "name": "test",
  "done": true
}'
```

Request URL

http://127.0.0.1:3000/todos/1000/items/1

Server response

Code	Details
404	Error: Not Found <div><div>Response body</div><pre>{ "message": "Couldn't find Todo with 'id'=1000" }</pre><div><div>Download</div></div></div>

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς, τότε δεν θα μπορέσει με επιτυχία να ανανεώσει ένα item το οποίο δεν υπάρχει και έτσι θα του εμφανιστεί το παρπάνω μήνυμα.

PUT

/todos/{todo_id}/items/{id} Update a todo item

Parameters

Cancel

Reset

Name	Description
todo_id <small>★ required</small>	todo_id
string (path)	<input type="text" value="1"/>
id <small>★ required</small>	id
string (path)	<input type="text" value="1"/>

Request body

application/json

```
{
  "name": "test",
  "done": true
}
```

Execute

Clear

Responses

Curl

```
curl -X 'PUT' \
'http://127.0.0.1:3000/todos/1/items/1' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "name": "test",
  "done": true
}'
```

Request URL

```
http://127.0.0.1:3000/todos/1/items/1
```

Server response

Code	Details
422	Error: Unprocessable Content

Response body

```
{
  "message": "Missing token"
}
```

Download

Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Delete a todo item

DELETE

/todos/{todo_id}/items/{id}

Delete a todo item

Parameters

Cancel

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="1"/>
id * required	id
string (path)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://127.0.0.1:3000/todos/1/items/1' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJleHAiOiE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvfYvBKzVAqFBRVDRGLYs'
```

Request URL

```
http://127.0.0.1:3000/todos/1/items/1
```

Server response

Code	Details
204	Response headers

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα έγκυρο todo id και item id τότε θα μπορέσει με επιτυχία να διαγράψει ένα item το οποίο υπάρχει.

DELETE

/todos/{todo_id}/items/{id}

Delete a todo item

Parameters

Cancel

Name	Description
todo_id * required	todo_id
string (path)	<input type="text" value="1000"/>
id * required	id
string (path)	<input type="text" value="1"/>

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
'http://127.0.0.1:3000/todos/1000/items/1' \
-H 'accept: */*' \
-H 'Authorization: eyJhbGciOiJIUzI1NiJ9.eyJ1c2VyX2lkIjozLCJleHAiOiE3MDgwMzU4OTZ9.xPXj7GmRW9J-n92nA5WZw0VvfYvBKzVAqFBRVDRGLYs'
```

Request URL

```
http://127.0.0.1:3000/todos/1000/items/1
```

Server response

Code	Details
404	Error: Not Found
	Response body

```
{
  "message": "Couldn't find Todo with 'id'=1000"
}
```

Download

Αν ο χρήστης χρησιμοποιήσει το token του επιτυχώς και εισάγει ένα μη έγκυρο todo id και item id τότε δεν θα μπορέσει με επιτυχία να διαγράψει ένα item το οποίο δεν υπάρχει και έτσι θα του εμφανιστεί το παραπάνω μήνυμα.

DELETE /todos/{todo_id}/items/{id} Delete a todo item

Cancel

Name	Description
todo_id * required string (path)	todo_id <input type="text" value="1"/>
id * required string (path)	id <input type="text" value="1"/>

ExecuteClear

Responses

Curl

```
curl -X 'DELETE' \
'http://127.0.0.1:3000/todos/1/items/1' \
-H 'accept: */*'
```

Request URL

```
http://127.0.0.1:3000/todos/1/items/1
```

Server response

Code	Details
422	Error: Unprocessable Content Response body <pre>{ "message": "Missing token" }</pre>

Download

Αν ο χρήστης δεν χρησιμοποιήσει το token του, τότε θα επιστραφεί το παραπάνω σφάλμα.

Πηγές

Kabiru A. Build a RESTful JSON API with rails 5 - Part one. DigitalOcean. 2020.

<https://www.digitalocean.com/community/tutorials/build-a-restful-json-api-with-rails-5-part-one>.

Kabiru A. Build a RESTful JSON API with rails 5 - Part two. DigitalOcean. 2020.

<https://www.digitalocean.com/community/tutorials/build-a-restful-json-api-with-rails-5-part-two>.

Kabiru A. Build a RESTful JSON API with rails 5 - Part three. DigitalOcean. 2020.

<https://www.digitalocean.com/community/tutorials/build-a-restful-json-api-with-rails-5-part-three>.