

NurseryBack

Nursery Checker Backend project

CONTEXTO

Gestionar el servicio de recogida y entrega de los alumnos a los padres registrando los datos digitalmente.

- Gestión del Staff de la empresa:
 - Creación de empleados,
 - Consultar lista de empleados,
 - Consultas específicas del Staff
- Gestión de datos de los alumnos:
 - Creación de alumnos y edición de sus datos,
 - Consultas de datos de los alumnos,
 - Control digital de las llegadas y salidas de los alumnos

DEVELOPER

[Joaquín Gázquez](#)

REPOSITORY

<https://github.com/Joaquingse/nurseryback>

DB SCHEMAS

USER

KEY	TYPE	REFERENCE	REQUIRED	VALIDATION
name	string		YES	
surname	string		YES	
dni	string		YES	unique, maxLength, minLenght
phone	number		YES	
email	Mongoose Schema Type email		YES	Mongoose, Unique
password	string		YES	
role	string		default = worker	enum: admin, owner, chief, worker

nursery	ObjectId	nursery	{role !== admin}	
---------	----------	---------	------------------	--

...

TUTOR

KEY	TYPE	REFERENCE	REQUIRED	VALIDATION
name	string		YES	
surname	string		YES	
dni	string		YES	unique, maxLength, minLenght
phone	number		YES	
email	Mongoose Schema Type email		YES	Mongoose, Unique
password	string		YES	
relation	string		YES	enum: parents, family, others
...				

CHILD

KEY	TYPE	REFERENCE	REQUIRED	VALIDATION
name	string		YES	
surname	string		YES	
tutors	ObjectId	tutor	YES	
nursery	ObjectId	nursery	YES	
dinner	Boolean		YES	
early	Boolean		YES	
alergies	Array			
activities	Array			
...				

NURSERY

KEY	TYPE	REFERENCE	REQUIRED	VALIDATION
name	string		YES	
phone	number		YES	
email	Mongoose Schema Type email		YES	Mongoose, Unique
address	string		YES	Unique
...				

EVENT

KEY	TYPE	REFERENCE	REQUIRED	VALIDATION
title	string		YES	
date	Date		YES	
attendance	Number			
description	string			
...				

DROP

KEY	TYPE	REFERENCE	REQUIRED	VALIDATION
child	ObjectId		YES	
date	Date		YES	
who	ObjectId		YES	
...				

PICK

KEY	TYPE	REFERENCE	REQUIRED	VALIDATION
child	ObjectId		YES	
date	Date		YES	
who	ObjectId		YES	
...				

API ROUTES

Enviroment:

Route `{{baseUrl}}` used in my test, here in docs appears localhost route, you can use it to your local test, remember change it if you deploy it in an external DB.

Collection authentication settings:

NurseryBack

Authorization ● Pre-request Script Tests Variables Runs

Type

API Key

The authorization header will be automatically generated when you send the request. Learn more about [authorization](#) ↗

Key

token

Value

{{token}}

Add to

Header

Params Authorization Headers (9) Body ● Pre-request Script Tests ● Settings

1 const result = pm.response.json()

2

3 pm.environment.set('token', result.token)

AUTHENTICATION ENDPOINTS

METHOD	URL	AUTH	FUNCTION
POST	/auth/signup	NO	register a user
POST	/auth/login	NO	Authenticate a user

USERS ENDPOINTS

METHOD	URL	AUTH	FUNCTION
GET	/user/	admin, owner	List all users
GET	/user/:id	Authenticate user	Show user info[^1]
POST	/user/new	admin, owner	Create a user account[^2]
PUT	/user/:id	Authenticate user	Update a user account [^3]
DELETE	/user/:id	admin	Delete a user account
...			

-
-
-
- [^1]: Workers and chief can see own info only, admin and owner can see all.
- [^2]: Admin can create all type of account, owners only chief and worker types.
- [^3]: Only Admin can update data from other account, rest of users can update their own data only.

TUTORS ENDPOINTS

METHOD	URL	AUTH	FUNCTION
GET	'/tutor/'	admin, owner	List all tutors
GET	'/tutor/:id'	Authenticate user	Show tutor info
POST	'/tutor/add'	admin, owner	Create a tutor
PUT	'/tutor/:id'	admin, owner, chief	Update tutor data
DELETE	'/tutor/:id'	admin, owner	Delete a tutor profile
...			

CHILD ENDPOINTS

METHOD	URL	AUTH	FUNCTION
GET	'/child/'	admin, owner	List all childs
GET	'/child/:id'	Authenticate user	Show child info
POST	'/child/add'	admin, owner	Create a child
PUT	'/child/:id'	admin, owner, chief	Update child data
DELETE	'/child/:id'	admin, owner	Delete a child profile
...			

NURSERY ENDPOINTS

METHOD	URL	AUTH	FUNCTION
GET	'/nursery/'	admin, owner	List all nurseries
GET	'/nursery/:id'	admin, owner	Show nursery info
POST	'/nursery/add'	admin, owner	Create a nursery
PUT	'/nursery/:id'	admin, owner	Update nursery data
DELETE	'/nursery/:id'	admin, owner	Delete a nursery profile
...			

EVENT ENDPOINTS

METHOD	URL	AUTH	FUNCTION
GET	'/event/'	Authenticate user	List all events
GET	'/event/:id'	Authenticate user	Show event info
POST	'/event/add'	admin, owner, chief	Create an event
PUT	'/event/:id'	admin, owner, chief	Update event data
DELETE	'/event/:id'	admin, owner, chief	Delete an event
...			

INOUT ENDPOINTS (drop and pick child register)

METHOD	URL	AUTH	FUNCTION
GET	'/inout/drops'	Authenticate user	List all drops
GET	'/inout/picks'	Authenticate user	List all picks
POST	'/inout/drop'	Authenticate user	Create a drop
POST	'/inout/pick'	Authenticate user	Create a pick
PUT	'/inout/:id'	admin	Update drop or pick data[^4]
DELETE	'/inout/:id'	admin	Delete drop or pick data[^4]
...			

[^4]: Depends on action selected (drops or picks)

AUTHORIZATION API Key

Key	token
Value	{{token}}

AUTH

Signup and Login routes, This signup is used by the admin to create new owners accounts, or other admins accounts.

The login route is for all the users, to get the token and stay authenticated in the app.

AUTHORIZATION API Key

This folder is using API Key from collection NurseryBack

POST signup route



{{baseUrl}}/auth/signup

Signup for admin does not need nursery data, but is mandatory for the rest of users (owner, chief, worker)

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "name": "otro",
  "surname": "apellido",
  "dni": "14785236B",
  "password": "1234",
  "email": "admin2@staff.com",
  "phone": "698745214",
  "role": "admin"
}
```

POST Login route



{{baseUrl}}/auth/login

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "email": "owner@staff.com",
  "password": "1234"
}
```

USERS

Users folder contains all the actions to be used with the users data.

AUTHORIZATION API Key

This folder is using API Key from collection [NurseryBack](#)

GET GET user info



```
{{baseUrl}}/user/63e9f570ab94dc0f1a2680b8
```

You can get user's info by user ID, remember the level of authentication you need.

If you want to get a list of all users or users with specific params, you can try:

- user/ get all users
- user?param=query get specific users

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

POST New user



```
{{baseUrl}}/user/new
```

In this request, you can create a new user, depends on the type (admin, owner, chief or worker) you have to introduce the nursery param or not, remember only admins can create admin and owner users.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "name": "worker2",
  "surname": "surname",
  "dni": "48567953D",
  "password": "1234",
  "email": "worker2@staff.com",
  "phone": "625325899",
  "role": "worker",
  "nursery": "63ea05b8f1aa7d1bb6b19023"
}
```


PUT Update user



```
{{baseUrl}}/user/63e9f570ab94dc0f1a2680b8
```

This request updates the field you choose to modify, you can update all or a specific field.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

```
json

{
  "surname": "apellido3"
}
```

DELETE Delete user



```
{{baseUrl}}/user/63e9f570ab94dc0f1a2680b8
```

This request to delete users from User collection, check authorizations to do it.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

NURSERY

AUTHORIZATION API Key

This folder is using API Key from collection [NurseryBack](#)

POST add/create nursery



```
{{baseUrl}}/nursery/add
```

This action add a new nursery into the Nursery collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "name": "Micky mouse",
  "phone": "928987654",
  "email": "mickymouse@mail.com",
  "address": "c/micky n° 1"
}
```

GET get all nursery



{{baseUrl}}/nursery/

This request show you a list of all nurseries in the collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "name": "Micky mouse",
  "phone": "928987654",
  "email": "mickymouse@mail.com",
  "address": "c/micky n° 1"
}
```

GET Get a nursery



{{baseUrl}}/nursery/63ea05b8f1aa7d1bb6b19023

This request show you an specific nursery in the collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "name": "Micky mouse",
  "phone": "928987654",
  "email": "mickymouse@mail.com",
  "address": "c/micky nº 1"
}
```

PUT Update a nursery



{{baseUrl}}/nursery/63ea05b8f1aa7d1bb6b19023

Update data from a nursery, you can update one or more fields at the same time.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "address": "c/Pepa nº 1"
}
```

DELETE Delete a nursery



{{baseUrl}}/nursery/63ea06e9f1aa7d1bb6b19027

Delete a nursery from the collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

TUTORS

AUTHORIZATION API Key

This folder is using API Key from collection [NurseryBack](#)

POST Add tutor



```
{{baseUrl}}/tutor/add
```

This request add a tutor document into Tutors collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

```
json

{
  "name": "Person3",
  "surname": "apellido3",
  "dni": "58965472H",
  "email": "person@tutor.com",
  "phone": "658952361",
  "relation": "family"
}
```

GET get all tutors



```
{{baseUrl}}/tutor/
```

Get a list of all tutors.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

GET Get a Tutor



```
{{baseUrl}}/tutor/63ea14c5f3ecf78f1ad37722
```

Get a specific tutor, you need the tutor ID to get it.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

PUT Update a tutor



{{baseUrl}}/tutor/63f348e53124f96ad48106ae

Update tutor document in the collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

```
json

{
  "surname": "apellidoTres"
}
```

DELETE Delete a Tutor



{{baseUrl}}/tutor/63f348e53124f96ad48106ae

Delete a tutor document from the Tutors collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

CHILDS

AUTHORIZATION API Key

This folder is using API Key from collection [NurseryBack](#)

POST Add childs



{{baseUrl}}/child/add

Add new childs to the DB.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "name": "niño 2",
  "surname": "apellido 2",
  "tutors": "63eb584590b0ab5791005121",
  "nursery": "63ea05b8f1aa7d1bb6b19023",
  "dinner": "false",
  "early": "true"
}
```

GET Get children



{{baseUrl}}/child

List all children.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

GET Get child info



{{baseUrl}}/child/63eb5ab45da0e733f41e2cef

Get specific child from the collection, You must have the child ID to get it.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

PUT Update child



{{baseUrl}}/child/63eb5ab45da0e733f41e2cef

Update child info.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

```
json

{
  "surname": "apellido 2"
}
```

DELETE Delete child



```
{{baseUrl}}/child/63eb5acc5da0e733f41e2cf1
```

Delete a child from the collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

EVENTS

AUTHORIZATION API Key

This folder is using API Key from collection [NurseryBack](#)

POST Add new event



```
{{baseUrl}}/event/new
```

Add an event to the collection.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

```
json

{
  "title": "Evento 2",
}
```

```
"date": "2023-02-18, 10:00",
"attendance": 15,

"description": "Manualidades con pintura"
}
```

GET Get all events



{{baseUrl}}/event/

List all events.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

GET Get an event



{{baseUrl}}/event/63eca2282bd35dbb506e89c4

Get a specific event.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

PUT Edit an event



{{baseUrl}}/event/63eca2282bd35dbb506e89c4

Update event data.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

```
json

{
  "date": "2023-02-19, 10:30"
}
```


DELETE Delete an event



{{baseUrl}}/event/63eca2962bd35dbb506e89c6

Delete an event.

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

CHECK IN-OUT

AUTHORIZATION API Key

This folder is using API Key from collection [NurseryBack](#)

POST Dropoff child



{{baseUrl}}/inout/drop

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

```
json

{
  "child": "63eb6fdb2fcfd5062996ebce",
  "date": "2023-02-14, 08:50",
  "who": "63ea14c5f3ecf78f1ad37722"
}
```

POST Pickup child



{{baseUrl}}/inout/pick

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

Body raw (json)

json

```
{
  "child": "63eb6fdb2fcfd5062996ebce",
  "date": "2023-02-14, 14:50",
  "who": "63ea14c5f3ecf78f1ad37722"
}
```

GET Get drops or picks (chg route to /drops or /picks)



```
{{baseUrl}}/inout/picks?child=63eb5ab45da0e733f41e2cef
```

This request can get picks list or drop list, You can try:

- inout/picks list all picks
- inout/drops list all drops
- inout/picks?query=params list specific picks
- inout/drops?query=params list specific drops

AUTHORIZATION API Key

This request is using API Key from collection [NurseryBack](#)

PARAMS

child	63eb5ab45da0e733f41e2cef
-------	--------------------------