



API do trabalho de DAW 1.0.0 OAS 3.0

Documentação da API

Servers

http://localhost:8080

Appointments Operações relacionadas a agendamentos

GET

/api/daw/requests/appointments

Retorna todos os agendamentos

Obtém a lista completa de agendamentos

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>Agendamentos encontrados com sucesso</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>[{ "id": "string",</div>	No links

Code	Description	Links
	<pre>"specialty": "string", "comments": "string", "date": "2024-10-08", "student": "string", "professional": "string" }]</pre>	
500	Erro ao buscar agendamentos	No links

POST

/api/daw/requests/appointments

Cria um novo agendamento

Adiciona um novo agendamento ao sistema

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "specialty": "string",
  "comments": "string",
  "date": "2024-10-08",
  "student": "string",
  "professional": "string"
}
```

Responses

Code	Description	Links
200	Agendamento criado com sucesso	No links

Media type

application/json

Controls Accept header.

Example Value

Schema

```
{
  "id": "string",
```

Code	Description	Links
	<pre>"specialty": "string", "comments": "string", "date": "2024-10-08", "student": "string", "professional": "string" }</pre>	
500	Erro ao criar agendamento	No links

GET

/api/daw/requests/appointments/{id}

Retorna um agendamento por ID

^

Obtém um agendamento específico pelo ID

Parameters

Try it out

Name	Description
id * required string (path)	ID do agendamento
	<div>id</div>

Responses

Code	Description	Links
200	<div>Agendamento encontrado com sucesso</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": "string", "specialty": "string", "comments": "string", "date": "2024-10-08", "student": "string", "professional": "string"}</pre></div>	No links
500	Erro ao buscar agendamento por ID	No links

PUT

/api/daw/requests/appointments/{id}

Atualiza um agendamento

Atualiza os dados de um agendamento existente

Parameters

Try it out

Name	Description
<div><div>id * required</div><div>string</div><div>(path)</div></div>	<div>ID do agendamento a ser atualizado</div> <div>id</div>

Request body required

application/json

Example Value Schema

```
{  "specialty": "string",  "comments": "string",  "date": "2024-10-08",  "student": "string",  "professional": "string"}
```

Responses

Code	Description	Links
200	<div>Agendamento atualizado com sucesso</div> <div>Media type</div> <div><div>application/json</div></div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "id": "string", "specialty": "string", "comments": "string", "date": "2024-10-08", "student": "string", "professional": "string"}</pre></div>	No links
500	Erro ao atualizar agendamento	No links

DELETE /api/daw/requests/appointments/{id} Deleta um agendamento

Remove um agendamento do sistema

Parameters

Try it out

Name	Description
<div>id * required</div> <div>string</div> <div>(path)</div>	<div>ID do agendamento a ser deletado</div> <div><div>id</div></div>

Responses

Code	Description	Links
200	Agendamento deletado com sucesso	No links
500	Erro ao deletar agendamento	No links

Events

Operações relacionadas a eventos

GET

/api/daw/requests/events

Retorna todos os eventos

Obtém a lista completa de eventos

Parameters

Cancel

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \
 'http://localhost:8080/api/daw/requests/events' \
 -H 'accept: application/json'

Request URL

http://localhost:8080/api/daw/requests/events

Server response

Code	Details
200	<div><div>Response body</div><div><pre>{ "message": "Eventos encontrados com sucesso", "professionals": [{ "_id": "6705b3bf5cc4581cb0044983", "description": "ojihi", "date": "2024-10-08T00:00:00.000Z", "createdAt": "2024-10-08T22:35:43.409Z",</pre></div></div>

Download

Code

Details

```
"updatedAt": "2024-10-08T22:35:43.409Z",
  "__v": 0
},
{
  "_id": "6705b4287e3930ce43f2fc29",
  "description": "srvfr",
  "comments": "rvedrf",
  "date": "2024-10-20T00:00:00.000Z",
  "createdAt": "2024-10-08T22:37:28.201Z",
  "updatedAt": "2024-10-08T22:37:28.201Z",
  "__v": 0
},
{
  "_id": "6705b8eb078e6b407fb1703a",
  "description": "bdfvbkhbdv",
  "comments": "ubdrvkb rd",
  "date": "2024-12-08T00:00:00.000Z",
  "createdAt": "2024-10-08T22:57:47.485Z",
  "updatedAt": "2024-10-08T22:57:47.485Z"
```

Response headers

access-control-allow-headers: Origin,X-Requested-With,Content,Accept,Content-Type,Authorization
access-control-allow-methods: GET,POST,PUT,DELETE,PATCH,OPTIONS
access-control-allow-origin: *
connection: keep-alive
content-length: 642
content-type: application/json; charset=utf-8
date: Tue,08 Oct 2024 22:57:59 GMT
etag: W/"282-g/NkgKRiBo/q5uTemQB7AGz7P0I"
keep-alive: timeout=5
x-powered-by: Express

Responses

Code	Description	Links
200	Eventos encontrados com sucesso	No links
<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>[{ "id": "string", "description": "string", "comments": "string", "date": "2024-10-08" }]</div>		
500	Erro ao buscar eventos	No links

POST

/api/daw/requests/events

Cria um novo evento



Adiciona um novo evento ao sistema

Parameters

Cancel

Reset

No parameters

Request body required

application/json

```
{
  "description": "bdfvbkhbdv",
  "comments": "ubdrvkbd",
  "date": "2024-12-08"
}string
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'http://localhost:8080/api/daw/requests/events' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "description": "bdfvbkhbdv",
    "comments": "ubdrvkbd",
    "date": "2024-12-08"
  }'
```

Request URL

http://localhost:8080/api/daw/requests/events

Server response

Code

Details

Code

Details

200

Response body

```
{  "message": "Evento criado com sucesso",  "professional": {    "description": "bdfvbkhbdv",    "comments": "ubdrvkbrd",    "date": "2024-12-08T00:00:00.000Z",    "_id": "6705b8eb078e6b407fb1703a",    "createdAt": "2024-10-08T22:57:47.485Z",    "updatedAt": "2024-10-08T22:57:47.485Z",    "__v": 0  } }
```

Download

Response headers

```
access-control-allow-headers: Origin,X-Requested-With,Content,Accept,Content-Type,Authorization access-control-allow-methods: GET,POST,PUT,DELETE,PATCH,OPTIONS access-control-allow-origin: * connection: keep-alive content-length: 259 content-type: application/json; charset=utf-8 date: Tue,08 Oct 2024 22:57:47 GMT etag: W/"103-0U/700bIPr2mIfIGwrvXbNegez0" keep-alive: timeout=5 x-powered-by: Express
```

Responses

Code	Description	Links
200	Evento criado com sucesso	No links
<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>{ "id": "string", "description": "string", "comments": "string", "date": "2024-10-08" }</pre></div>		
500	Erro ao criar evento	No links

GET	/api/daw/requests/events/{id}	Retorna um evento por ID	⤴
Obtém um evento específico pelo ID			

Parameters

Try it out

Name	Description
id * required string (path)	ID do evento <div>id</div>

Responses

Code	Description	Links
200	Evento encontrado com sucesso <div>Media type application/json Controls Accept header. Example Value Schema <pre>{ "id": "string", "description": "string", "comments": "string", "date": "2024-10-08"}</pre></div>	No links
500	Erro ao buscar evento por ID	No links

PUT /api/daw/requests/events/{id} Atualiza um evento

^

Atualiza os dados de um evento existente

Parameters

Try it out

Name	Description
id * required string (path)	ID do evento a ser atualizado <div>id</div>

Request body required

application/json

Example Value

Schema

```
{  "description": "string",  "comments": "string",  "date": "2024-10-08"}  
```

Responses

Code	Description	Links
200	Evento atualizado com sucesso <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div><div><div>Example Value</div><div>Schema</div><div><pre>{ "id": "string", "description": "string", "comments": "string", "date": "2024-10-08"} </pre></div></div></div>	No links
500	Erro ao atualizar evento	No links

DELETE

/api/daw/requests/events/{id}

Deleta um evento

Remove um evento do sistema

Parameters

Try it out

Name	Description
<div><div>id * required</div><div>string</div><div>(path)</div></div>	<div>ID do evento a ser deletado</div> <div><div>id</div></div>

Responses

Code	Description	Links
200	Evento deletado com sucesso	No links
500	Erro ao deletar evento	No links

Professionals

Operações relacionadas a Profissionais

**GET** /api/daw/requests/professionals Retorna todos os profissionais

Obtém a lista completa de profissionais

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Profissionais encontrados com sucesso	No links
500	Erro ao buscar profissionais	No links

POST /api/daw/requests/professionals Cria um novo profissional

Adiciona um novo profissional ao sistema

Parameters

[Try it out](#)

No parameters

Request body required

application/json

Example Value

Schema

```
{  "name": "string",  "specialty": "string",  "contact": "string",  "phone_number": "string",  "status": true}
```

Responses

Code	Description	Links
200	Profissional criado com sucesso	No links
500	Erro ao criar profissional	No links

GET

/api/daw/requests/professionals/{id}

Retorna um profissional por ID

Obtém um profissional específico pelo ID

Parameters

Try it out

Name	Description
<div><div>id * required</div><div>string</div><div>(path)</div></div>	<div>ID do profissional</div> <div><div>id</div></div>

Responses

Code	Description	Links
200	Profissional encontrado com sucesso	No links

Code	Description	Links
500	Erro ao buscar profissional por ID	No links

PUT

/api/daw/requests/professionals/{id}

Atualiza um profissional

Atualiza os dados de um profissional existente

Parameters

Try it out

Name	Description
id * required string (path)	ID do profissional a ser atualizado

id

Request body required

application/json

Example Value

Schema

```
{  "name": "string",  "specialty": "string",  "contact": "string",  "phone_number": "string",  "status": true}
```

Responses

Code	Description	Links
200	Profissional atualizado com sucesso	No links
500	Erro ao atualizar profissional	No links

DELETE

/api/daw/requests/professionals/{id}

Deleta um profissional

Remove um profissional do sistema

Parameters

Try it out

Name	Description
------	-------------

id * required string (path)	ID do profissional a ser deletado
--	-----------------------------------

Responses

Code	Description	Links
200	Profissional deletado com sucesso	No links
500	Erro ao deletar profissional	No links

Students

Operações relacionadas a Estudantes



GET /api/daw/requests/students Retorna todos os estudantes



Obtém a lista completa de estudantes

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Estudantes encontrados com sucesso	No links

Code	Description	Links
500	Erro ao buscar estudantes	No links

POST

/api/daw/requests/students

Cria um novo estudante

Adiciona um novo estudante ao sistema

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{
  "name": "string",
  "age": 0,
  "parents": "string",
  "phone_number": "string",
  "special_needs": "string",
  "status": true
}
```

Responses

Code	Description	Links
200	Estudante criado com sucesso	No links
500	Erro ao criar estudante	No links

GET

/api/daw/requests/students/{id}

Retorna um estudante por ID

Obtém um estudante específico pelo ID

Parameters

Try it out

id

Responses

Code	Description	Links
200	Estudante encontrado com sucesso	<i>No links</i>
500	Erro ao buscar estudante por ID	<i>No links</i>

PUT `/api/daw/requests/students/{id}` Atualiza um estudante

Atualiza os dados de um estudante existente

Parameters

Try it out

Name	Description
------	-------------

id * required ID do estudante a ser atualizado

string
(path)

id

Request body required

application/json

Example Value	Schema
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

```
{
  "name": "string",
  "age": 0,
  "parents": "string",
  "phone_number": "string",
  "special_needs": "string",
  "status": true
}
```

Responses

Code	Description	Links
200	Estudante atualizado com sucesso	No links
500	Erro ao atualizar estudante	No links

DELETE /api/daw/requests/students/{id} Deleta um estudante ^

Remove um estudante do sistema

Parameters

Try it out

Name	Description
id * required string (path)	ID do estudante a ser deletado

Responses

Code	Description	Links
200	Estudante deletado com sucesso	No links
500	Erro ao deletar estudante	No links

Teacher Operações relacionadas a professores ^

GET /api/daw/requests/teachers Retorna todos os professores ^

Obtém a lista completa de professores

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Professores encontrados com sucesso	No links
500	Erro ao buscar professores	No links

POST /api/daw/requests/teachers Cria um novo professor



Adiciona um novo professor ao sistema

Parameters

Try it out

No parameters

Request body **required**

application/json

Example Value Schema

```
{
  "name": "string",
  "school_disciplines": [
    "string"
  ],
  "contact": "string",
  "phone_number": "string",
  "status": true
}
```

Responses

Code	Description	Links
200	Professor criado com sucesso	No links
500	Erro ao criar professor	No links

GET /api/daw/requests/teachers/{id} Retorna um professor por ID

Obtém um professor específico pelo ID

Parameters

Try it out

Name	Description
id * required string (path)	ID do professor

Responses

Code	Description	Links
200	Professor encontrado com sucesso	No links
500	Erro ao buscar professor por ID	No links

PUT /api/daw/requests/teachers/{id} Atualiza um professor

Atualiza os dados de um professor existente

Parameters

Try it out

Name	Description
id * required	ID do professor a ser atualizado

Name

Description

string

(path)

id

Request body

required

application/json

Example Value

Schema

{

"name": "string",

"school_disciplines": [

"string"

],

"contact": "string",

"phone_number": "string",

"status": true

}

Responses

Code

Description

Links

200

Professor atualizado com sucesso

No links

500

Erro ao atualizar professor

No links

DELETE

/api/daw/requests/teachers/{id}

Deleta um professor

Remove um professor do sistema

Parameters

Try it out

Name

Description

id * required

string

(path)

ID do professor a ser deletado

id

Responses

Code	Description	Links
200	Professor deletado com sucesso	No links
500	Erro ao deletar professor	No links

Users

Operações relacionadas a usuários

**GET** /api/daw/requests/users Retorna todos os usuários

Obtém a lista completa de usuários

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Usuários retornados com sucesso	No links
500	Erro ao buscar usuários	No links

POST /api/daw/requests/users Cria um novo usuário

Adiciona um novo usuário ao sistema

Parameters

[Try it out](#)

No parameters

Request body required[application/json](#)

Example Value Schema

```
{
  "name": "string",
  "user": "string",
  "email": "string",
  "pwd": "string",
  "level": "string",
  "status": true
}
```

Responses

Code	Description	Links
200	Usuário criado com sucesso	No links
500	Erro ao processar dados do POST	No links

GET /api/daw/requests/users/{id} Retorna um usuário por ID



Obtém um usuário específico pelo ID

Parameters

Try it out

Name	Description
id * required string (path)	ID do usuário

Responses

Code	Description	Links
200	Usuário retornado com sucesso	No links
500	Erro ao processar dados do GET por id de usuário	No links

PUT

/api/daw/requests/users/{id}

Atualiza um usuário

Atualiza os dados de um usuário existente

Parameters

Try it out

Name	Description
id * required string (path)	ID do usuário a ser atualizado

id

Request body required

application/json

Example Value

Schema

```
{  "name": "string",  "user": "string",  "email": "string",  "pwd": "string",  "level": "string",  "status": true}
```

Responses

Code	Description	Links
200	Usuário atualizado com sucesso	No links
500	Erro ao processar dados do PUT	No links

DELETE

/api/daw/requests/users/{id} Deleta um usuário

Remove um usuário do sistema

Parameters

Try it out

Name	Description
id * required string (path)	ID do usuário a ser deletado

id

Responses

Code	Description	Links
200	Usuário deletado com sucesso	No links
500	Erro ao processar dados do DELETE	No links

POST

/api/daw/requests/users/login Realiza o login de um usuário

Autentica um usuário com base no email e senha

Parameters

Try it out

No parameters

Request body required

application/json

Example Value

Schema

```
{  "email": "string",  "pass": "string"}

```

Responses		
Code	Description	Links
200	Login realizado com sucesso	No links
500	Erro ao processar dados do POST	No links