



Modular App API

1.0.0

OAS 3.0

This is the API documentation for the Modular APP

[Authorize](#)

Auth

**POST**[/api/auth/login](#) Authenticate a user

Authenticate a user with a username and a password and return a JWT token

Parameters[Try it out](#)

No parameters

Request body required[application/json](#) ▾**Example Value** | Schema

```
{  
  "username": "string",  
  "password": "string"  
}
```

Responses

Code	Description	Links
200	Authentication success	<i>No links</i>

Media type

Code	Description	Links
------	-------------	-------

application/json ▾

Controls Accept header.

[Example Value](#) | [Schema](#)

```
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9"
```

400

Username or password missing

No links

Media type

application/json ▾

[Example Value](#) | [Schema](#)

```
{
  "errors": [
    {
      "type": "field",
      "value": "",
      "msg": "Password is required",
      "path": "password",
      "location": "body"
    }
  ]
}
```

401

Authentication failed

No links

Media type

application/json ▾

[Example Value](#) | [Schema](#)

```
{
  "message": "string"
}
```

GET

/api/auth/me Get the connected user information



Decode the JWT token and return the user information

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Valid token, return the user information	No links
401	Invalid or expired token	No links

Modules



GET

/api/modules Get the imported modules



Get the list of modules imported in the application

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	<p>List of modules</p> <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>[{ "id": "1344ca73-5bd2-472e-a75d-45d2c6c5f7a0", "name": "ZIP-Module", "description": "A module used for debug", "author": "Test", "version": "1.0.0", "icon": "debug.png", "nickname": null, "enabled": false, "importedAt": "2023-06-22 08:08:47" }]</pre>	No links

POST

/api/modules Import a module



Import the archive file of a module into the application

Parameters

[Try it out](#)

Name	Description
file (<i>formData</i>)	The module archive file

Responses

Code	Description	Links
200	Module imported successfully	No links

Code	Description	Links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "message": "Module uploaded and registered successfully", "moduleId": "1344ca73-5bd2-472e-a75d-45d2c6c5f7a0" }</pre>	
400	No file provided, invalid file format or invalid archive structure. Each error is returned in the response message	<i>No links</i>
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Example Value Schema</p> <pre>{ "message": "No files were uploaded" }</pre>	

GET

/api/modules/{id} Get a module



Get a module information and its configuration

Parameters

[Try it out](#)

Name	Description
id * required	The module id

id * required

string

(path)

Responses

Code	Description	Links
------	-------------	-------

200

Module exists

No links

Code	Description	Links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>{ "id": "1344ca73-5bd2-472e-a75d-45d2c6c5f7a0", "name": "ZIP-Module", "description": "A module used for debug", "author": "Nicolas Crausaz", "version": "1.0.0", "icon": "debug.png", "nickname": null, "enabled": false, "importedAt": "2023-06-22 08:08:47", "currentConfig": [] }</pre>	

404	The module does not exist	No links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Example Value Schema</p> <pre>{ "message": "Module not found" }</pre>	

DELETE /api/modules/{id} Delete a module ▲ 

Delete a module from the application

Parameters

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

Responses

Code	Description	Links
204	Module deleted	No links
404	The module does not exist	No links

Media type

application/json ▾

Example Value | Schema

```
{  
  "message": "Module not found"  
}
```

PATCH

/api/modules/{id} Update a module



Update a module information

Parameters

Try it out

Name Description

id * required

string

(path)

id

Request body required

application/json ▾

Example Value | Schema

```
{  
  "nickname": "string"  
}
```

Responses

Code	Description	Links
204	Module updated	No links
404	The module does not exist	No links

Media type

application/json ▾

Example Value | Schema

```
{  
  "message": "Module not found"  
}
```

POST

/api/modules/{id}/events Send an event to a module



Send an event to a module

Parameters

Try it out

Name Description

id * required

string

(path)

id

Request body required

application/json ▾

Example Value | Schema

```
{  
  "data": {}  
}
```

Responses

Code	Description	Links
204	Event sent	No links
400	Invalid body structure (missing data)	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div>	
	Example Value Schema	
	<pre>{ "errors": [{ "type": "field", "value": "", "msg": "Data must be an object", "path": "data", "location": "body" }] }</pre>	
403	The module is not enabled or not allowed to receive events	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div>	
	Example Value Schema	
	<pre>{ "message": "Module with id 5f68ba88-df21-4fbe-98b8-f39a6b6e469a is not allowed to receive HTTP data" }</pre>	
404	The module does not exist	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div>	
	Example Value Schema	
	<pre>{ "message": "Module not found" }</pre>	

POST

/api/modules/events Send an event to multiple modules



Send an event to multiple modules

Parameters

[Try it out](#)

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

id * required

string

(path)

id

Request body required

application/json ▾

[Example Value](#) | [Schema](#)

```
{  
  "ids": [  
    "6881d200-9f96-4c9a-a290-226c42a9476f",  
    "cfc7c162-9700-493a-b8c8-5cf32e22d0ca"  
  ],  
  "data": {}  
}
```

Responses

Code	Description	Links
------	-------------	-------

204

Event sent

No links

400

Invalid body structure (missing ids or data)

No links

Media type

application/json ▾

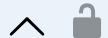
[Example Value](#) | [Schema](#)

```
{  
  "errors": [  
    {  
      "type": "field",  
      "value": "",  
      "msg": "Data must be an object",  
      "path": "data",  
      "location": "body"  
    }  
  ]  
}
```

Code	Description	Links
403	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Example Value Schema</p> <pre>{ "message": "Module with id 5f68ba88-df21-4fbe-98b8-f39a6b6e469a is not allowed to receive HTTP data" }</pre>	No links
404	<p>The module does not exist</p> <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Example Value Schema</p> <pre>{ "message": "Module not found" }</pre>	No links

GET

/api/modules/{id}/configuration Get the configuration of a module



Get the configuration of a module

Parameters

[Try it out](#)

Name Description

id * requiredstring
(path)

The module id

Responses

Code	Description	Links
200	Module exists	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Controls Accept header.</p> <p>Example Value Schema</p> <pre>[{ "name": "refreshRate", "type": "number", "label": "Refresh rate", "description": "The refresh rate in ms", "value": 1000 }]</pre>	

404	The module does not exist	No links
	<p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">application/json ▾</div> <p>Example Value Schema</p> <pre>{ "message": "Module not found" }</pre>	

PUT /api/modules/{id}/configuration Update the configuration of a module ▲ 

Update the configuration of a module

Parameters [Try it out](#)

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

Request body application/json ▾

[Example Value](#) | [Schema](#)

```
[  
  {  
    "name": "refreshRate",  
    "type": "number",  
    "label": "Refresh rate",  
    "description": "The refresh rate in ms",  
    "value": 1000  
  }  
]
```

Responses

Code	Description	Links
204	Module configuration updated	<i>No links</i>
400	Invalid configuration values	<i>No links</i>

Media type

application/json ▾

[Example Value](#) | [Schema](#)

```
{  
  "errors": [  
    {  
      "type": "field",  
      "value": "",  
      "msg": "Name is required",  
      "path": "fields[0].name",  
      "location": "body"  
    }  
  ]  
}
```

404	The module does not exist	<i>No links</i>
-----	---------------------------	-----------------

Media type

application/json ▾

[Example Value](#) | [Schema](#)

```
{  
  "message": "Module not found"  
}
```

POST

/api/modules/{id}/configuration/default Reset the configuration of a module to default



Reset the configuration of a module to default

Parameters

Try it out

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

id * required	
----------------------	--

string	
--------	--

(path)	
--------	--

<input type="text" value="id"/>

Responses

Code	Description	Links
------	-------------	-------

204		
-----	--	--

Module configuration updated	No links
------------------------------	----------

404		
-----	--	--

The module does not exist	No links
---------------------------	----------

Media type

application/json ▾

[Example Value](#) | [Schema](#)

```
{  
  "message": "Module not found"  
}
```

POST

/api/modules/{id}/status Update the status of a module



Enable or disable a module

Parameters

Try it out

Name	Description	
id * required	The module id	
string (path)	<input type="text" value="id"/>	
Request body required		
<input type="button" value="application/json ▾"/>		
Example Value Schema		
{ "enabled": true }		
Responses		
Code	Description	Links
204	Module status updated	No links
400	Invalid status value. The status must be a boolean	No links

Screens



GET

/api/screens Get all screens



Get all screens

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Return all screens	No links

Media type

application/json ▾

Controls Accept header.

[Example Value](#) | [Schema](#)

```
[  
  {  
    "id": 1,  
    "name": "screen 1",  
    "enabled": true,  
    "slots": []  
  }  
]
```

PUT

/api/screens Create or update a screen



Create or update a screen

Parameters

[Try it out](#)

Name

Description

id * required

string

(path)

The screen id

id

Request body required

application/json ▾

[Example Value](#) | [Schema](#)

```
{  
  "id": 0,  
  "name": "string",  
  "enabled": true  
}
```

Responses

Code	Description	Links
204	Screen updated	No links
400	Invalid screen body	No links
	<p>Media type</p> <div><p>application/json ▾</p></div>	
	<p>Example Value Schema</p> <div><pre>{\n \"errors\": [\n {\n \"type\": \"field\",\n \"value\": \"\",\n \"msg\": \"Name is required\",\n \"path\": \"name\",\n \"location\": \"body\"\n }\n]\n}</pre></div>	
404	The screen does not exist	No links
	<p>Media type</p> <div><p>application/json ▾</p></div>	
	<p>Example Value Schema</p> <div><pre>{\n \"message\": \"Screen not found\"\n}</pre></div>	

GET

/api/screens/{id} Get a screen



Get a screen

Parameters

Try it out

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

Responses

Code	Description	Links
200	Screen exists Media type application/json ▾ Controls Accept header. Example Value Schema <pre>{ "id": 1, "name": "screen 1", "enabled": true, "slots": [] }</pre>	<i>No links</i>
404	The screen does not exist Media type application/json ▾ Example Value Schema <pre>{ "message": "Screen not found" }</pre>	<i>No links</i>

DELETE	/api/screens/{id} Delete a screen	^	🔒
---------------	-----------------------------------	-------------------	-------------------

Delete a screen

Parameters	Try it out
<hr/>	

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

Responses

Code	Description	Links
204	Screen deleted	No links
404	The screen does not exist	No links

Media type

[Example Value](#) | [Schema](#)

```
{
  "message": "Screen not found"
}
```

Box



GET

/api/box Get box information



Get box information

Parameters

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Return box information	No links

Media type

application/json ▾

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{  
  "name": "Modular APP",  
  "icon": "logo.svg",  
  "version": "1.0.0"  
}
```

PUT

/api/box Update box information



Update box information

Parameters

[Try it out](#)

No parameters

Request body required

application/json ▾

[Example Value](#) | [Schema](#)

```
{  
  "name": "Modular APP"  
}
```

Responses

Code	Description	Links
204	Box information updated	No links
400	Invalid box body	No links

Media type

Code	Description	Links
------	-------------	-------

application/json ▾

[Example Value](#) | [Schema](#)

```
{\n  \"errors\": [\n    {\n      \"type\": \"field\",\n      \"value\": \"\",\n      \"msg\": \"Name is required\",\n      \"path\": \"name\",\n      \"location\": \"body\"\n    }\n  ]\n}
```

POST

/api/box/icon Update the box icon



Upload and update the box icon

Parameters

[Try it out](#)

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

file

(*formData*)

The icon to upload

file

Responses

Code	Description	Links
------	-------------	-------

200

Icon imported successfully

No links

Media type

application/json ▾

Controls Accept header.

[Example Value](#) | [Schema](#)

```
{\n  \"message\": \"Icon updated successfully\",
```

Code	Description	Links
400	<pre>"moduleId": "icon.svg" }</pre> <p>No file provided, invalid file format or invalid archive structure. Each error is returned in the response message</p> <p>Media type <input checked="" type="button"/> application/json ▾</p> <p>Example Value Schema</p> <pre>{ "message": "No files were uploaded" }</pre>	No links

GET /api/box/static/{filename} Get a static file (public) ^

Get a static file from the public directory

Parameters

[Try it out](#)

Name	Description
filename * required string (path)	The name of the file to get <input type="text" value="filename"/>

Responses

Code	Description	Links
200	Return the file	No links
404	The file does not exist	No links

GET /api/box/static/module/{moduleId}/{filename} Get a static file (module) ^

Get a static file from a module directory

Parameters

[Try it out](#)

Name	Description
moduleId * required string (path)	The module id <input type="text" value="moduleId"/>
filename * required string (path)	The name of the file to get <input type="text" value="filename"/>

Responses

Code	Description	Links
200	Return the file	No links
404	The file does not exist	No links

[GET](#)[/api/box/static/user/{filename}](#) Get a static file (user avatar)

Get a static file from the user avatar directory

Parameters

[Try it out](#)

Name	Description
filename * required string (path)	The name of the file to get <input type="text" value="filename"/>

Responses

Code	Description	Links
200	Return the file	<i>No links</i>
404	The file does not exist	<i>No links</i>

GET

/api/box/security/keys Get API keys



Get existing API keys, only the prefix of the keys are returned

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Return the API keys	<i>No links</i>

Media type

application/json ▾

Controls Accept header.

Example Value | Schema

```
[  
  {  
    "id": 1,  
    "name": "sdfg",  
    "display": "bcef9*****",  
    "createdAt": "2023-07-18 11:51:01"  
  },  
  {  
    "id": 2,  
    "name": "asdf",  
    "display": "6dd90*****",  
    "createdAt": "2023-07-18 12:27:56"  
  }]
```

Code	Description	Links
	<pre>}</pre>	
POST	/api/box/security/keys Generate a new API key	 
Generate a new API key and return it		
Parameters		Try it out
No parameters		
Request body		application/json 
Example Value Schema <pre>{ "name": "string" }</pre>		
Responses		
Code	Description	Links
200	Return the API key	<i>No links</i>
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json </div> <p>Controls Accept header.</p>	
	Example Value Schema <pre>{ "key": "string" }</pre>	
400	Invalid api key name	<i>No links</i>
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">application/json </div>	
	Example Value Schema	

Code	Description	Links
	<pre>{\n \"errors\": [\n {\n \"type\": \"field\",\n \"value\": \"\",\n \"msg\": \"Name is required\",\n \"path\": \"name\",\n \"location\": \"body\"\n }\n]\n}</pre>	

DELETE /api/box/security/keys/{id} Delete an API key ^ 

Delete an API key

Parameters

[Try it out](#)

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

Responses

Code	Description	Links
204	The API key has been deleted	No links
404	The API key does not exist	No links

Users

^

GET	/api/users Get all users ^ 
Get all users	
Parameters	

[Try it out](#)

No parameters

Responses

Code	Description	Links
200	Return the users	No links

Media type

application/json ▾

Controls Accept header.

[Example Value](#) | [Schema](#)

```
[  
  {  
    "id": 0,  
    "username": "string",  
    "avatar": "string",  
    "isDefault": true  
  }  
]
```

POST[/api/users/{id}](#) Create a new user

Create a new user

Parameters

[Try it out](#)

No parameters

Request body

application/json ▾

[Example Value](#) | [Schema](#)

```
{  
  "username": "string",  
  "password": "string"  
}
```

Responses

Code	Description	Links
204	The user has been created	No links
400	<p>Some fields are missing or invalid:</p> <ul style="list-style-type: none"> Missing username The username is already taken The password does not match the requirements (8 chars min, 1 uppercase, 1 lowercase, 1 number, 1 special char) 	No links

Media type

application/json ▾

Example Value | Schema

```
{
  "errors": [
    {
      "type": "field",
      "value": "",
      "msg": "Username is required",
      "path": "username",
      "location": "body"
    }
  ]
}
```

PATCH

/api/users/{id} Update a user



Update a user

Parameters**Try it out**

Name	Description
id integer (path)	<input type="text" value="id"/>

Request body

application/json ▾

Example Value | Schema

```
{
  "username": "string",
```

```
"password": "string"  
}
```

Responses

Code	Description	Links
204	The user has been updated	No links
400	Some fields are missing or invalid: <ul style="list-style-type: none">Missing usernameThe username is already takenThe password does not match the requirements (8 chars min, 1 uppercase, 1 lowercase, 1 number, 1 special char)	No links

Media type

application/json ▾

Example Value | Schema

```
{  
  "errors": [  
    {  
      "type": "field",  
      "value": "",  
      "msg": "Username is required",  
      "path": "username",  
      "location": "body"  
    }  
  ]  
}
```

404

The user does not exist

No links

DELETE

/api/users/{id} Delete a user



Delete a user if it is not the default one

Parameters

Try it out

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

id integer	id
---------------	----

Name	Description
(path)	

Responses

Code	Description	Links
204	The user has been deleted	No links
404	The user does not exist	No links

PUT /api/users/{id}/avatar Update a user avatar



Update a user avatar

Parameters

[Try it out](#)

Name	Description
id integer (path)	id
file * required (formData)	The icon to upload

Responses

Code	Description	Links
204	The user avatar has been updated	No links

Code	Description	Links
404	The user does not exist	<i>No links</i>