

API for 3R-kontor

Table of Contents

Table of Contents.....	1
Requests in general.....	2
Authentication.....	2
Responses.....	2
Successful operations.....	2
Errors.....	2
API calls for user.....	4
Create user.....	4
Update user.....	4
API calls for case.....	5
Create case.....	5
Update case.....	5
Complete case.....	5
Add/update environment.....	6
Add image for environment.....	6
Remove image for environment.....	6
API calls for products.....	7
Add product.....	7
Add product (batch).....	8
Update product.....	11
Remove product.....	12
Add product image.....	12
Remove product image.....	12
Changelog.....	13

Requests in general

The API is built as a REST-like API. It is designed to store information from you. All requests need to be authenticated.

Information should generally be sent as JSON unless otherwise specified.

All API call should go to <https://3r-api.mdevelopment.dk>.

Authentication

Authentication is done using a token provided by MDevelopment ApS. This information is sent as headers for the request:

HTTP-Header field	Type	Required	Value
Authentication-Type	String	Yes	Token
Authentication-Token	String	Yes	%TOKEN%

Responses

Successful operations

All successful responses will have a HTTP status code in the range of 200-299. If no information is to be returned, they will respond only:

HTTP Status: 204

If data is to be returned, it will be returned as an JSON-object or an array .

HTTP Status: 200

```
{
    "key1": true,
    "data": 44
}
```

Errors

All errors will follow the format below. Note that the HTTP status code will be in the range [400-499] in case of input errors or [500-510] in case of server errors.

HTTP Status: XXX

```
{
    "success": false,
    "error": "Some error message"
}
```

Status code 404 means the information could not be found. This could be parent records as well.

HTTP Status: 404

```
{  
  "success": false,  
  "error": "Could not find case"  
}
```

Status code 422 means that validation failed.

HTTP Status: 422

```
{  
  "success": false,  
  "error": "One or more validation errors occurred",  
  "details": {  
    "zip": [  
      "The zip must be an integer."  
    ]  
  }  
}
```

Status code 423 means that the case has been completed, and therefore cannot be updated (environment, products, etc.)

HTTP Status: 423

```
{  
  "success": false,  
  "error": "Cannot update completed cases"  
}
```

Status code 500 means an internal error has happened saving records, etc.

API calls for user

Create user

POST /user

Field	Type	Required	Value
companyName	String	Yes	
attention	String	Yes	
phone	Int	YES	
email	String	YES	

Sample request:

```
POST /user HTTP/1.1
Authentication-Type: Token
Authentication-Token: Ad93dioik1909moggDkd4qw4adp24DO87
Content-Type: application/json
{
    "companyName": "Johns Company",
    "attention": "Jane Doe",
    "phone": 26183301,
    "email": "jane@john.com"
}
```

On success an UserID is returned. An error will be returned in case of invalid e-mail or missing fields.

```
HTTP Status: 201
{
    "id": 4
}
```

The provided e-mail will be used as an identifier for the user when creating cases and providing further information.

Update user

PUT /user/%USER ID%

Field	Type	Required	Value
companyName	String	No	
attention	String	No	
phone	Int	No	
email	String	No	

Please note that it is possible to update the user's e-mail. On success nothing is returned.

API calls for case

Create case

POST /case

Field	Type	Required	Value
user	INT	Yes	The UserID from /user
address	String	Yes	
zip	INT	Yes	
city	String	Yes	
startDate	DATE (YYYY-MM-DD)	Yes	
endDate	DATE (YYYY-MM-DD)	Yes	

The UserID is used to identify the user to whom the case belongs.

On success the new CaseID is returned.

```
HTTP Status: 201
{
    "id": 42
}
```

Update case

PUT /case/%CASE ID%

Field	Type	Required	Value
address	String	Yes	
zip	INT	Yes	
city	String	Yes	
startDate	DATE (YYYY-MM-DD)	Yes	
endDate	DATE (YYYY-MM-DD)	Yes	

On success nothing is returned.

Complete case

Important: When a case is completed (submitted) it will no longer be possible to edit information already sent. Neither will it be possible to add more information. All image uploads should be completed before calling this method. This step is irreversible.

POST /case/%CASE ID%/complete

On success nothing is returned.

Add/update environment

PUT /case/%CASE ID%/environment

Field	Type	Required	Value
stairs	BOOLEAN	Yes	
lift	BOOLEAN	Yes	
comment	String	No	

On success nothing is returned. To overwrite the comment, you will have to provide it as a parameter for any subsequent calls to this endpoint.

Add image for environment

POST /case/%CASE ID%/environment/image

Please be aware that this should be sent as a regular file and not as JSON.

Sample request:

```
POST /case/42/environment/image HTTP/1.1  
Authentication-Type: Token  
Authentication-Token: Ad93dioik1909moggDkd4qw4adp24DO87  
Content-Type: image/png  
Content-Length: 284  
  
raw image data
```

On success the new ImageID is returned:

```
HTTP Status: 201  
{  
    "id": 428  
}
```

Remove image for environment

DELETE /case/%CASE ID%/environment/image/%IMAGE ID%

On success nothing is returned.

API calls for products

Add product

POST /case/%CASE ID%/product

Field	Type	Required	Value
category	String	Yes	
quantity	INT	Yes	
condition	INT	No	1-5 for the condition. 5 is like new.
brand	String	No	
width	DECIMAL	No	All measures are in cm
height	DECIMAL	No	All measures are in cm
depth	DECIMAL	No	All measures are in cm
diameter	DECIMAL	No	All measures are in cm
comment	String	No	

Sample request:

```
POST /case/42/product HTTP/1.1  
Authentication-Type: Token  
Authentication-Token: Ad93dioik1909moggDkd4qw4adp24DO87  
Content-Type: application/json  
{  
    "category": "Chairs",  
    "quantity": 6,  
    "condition": 4  
    "brand": "Hoeg",  
    "width": 60.0,  
    "height": 82.5,  
    "depth": 60.0,  
    "comment": "Red and black"  
}
```

On success the product-object along with the ProductID is returned. Undefined attributes will have their values set to *null* or an empty string for fields of type String. This will be needed for future reference.

```
HTTP Status: 201
{
  "id": 241,
  "category": "Chairs",
  "quantity": 6,
  "condition": 4,
  "brand": "Hoeg",
  "width": 60.0,
  "height": 82.5,
  "depth": 60.0,
  "diameter": null,
  "comment": "Red and black"
}
```

Add product (batch)

POST /case/%CASE ID%/products

Field	Type	Required	Value
category	String	Yes	
quantity	INT	Yes	
condition	INT	No	1-5 for the condition. 5 is like new.
brand	String	No	
width	DECIMAL	No	
height	DECIMAL	No	
depth	DECIMAL	No	
diameter	DECIMAL	No	
comment	String	No	

Using this endpoint is basically the same as calling the /product endpoint. However using this endpoint it is possible to send up to 50 products at a time. Information should be sent as an array of product-objects.

Sample request:

POST /case/42/products HTTP/1.1

Authentication-Type: Token

Authentication-Token: Ad93dioik1909moggDkd4qw4adp24DO87

Content-Type: application/json

```
[
  {
    "category": "Chairs",
    "quantity": 6,
    "condition": 4,
    "brand": "Hoeg",
    "width": 60.0,
    "height": 82.5,
    "depth": 60.0,
    "comment": "Red and black"
  },
  {
    "category": "Tables",
    "quantity": 1,
    "condition": 4,
    "brand": "Lampala",
    "diameter": 200.0,
    "comment": "Large round table"
  }
]
```

On success the product-objects is returned along with an extra attribute for the product id:

HTTP Status: 201

```
[
  {
    "id": 241,
    "category": "Chairs",
    "quantity": 6,
    "condition": 4,
    "brand": "Hoeg",
    "width": 60.0,
    "height": 82.5,
    "depth": 60.0,
    "diameter": null,
    "comment": "Red and black"
  },
  {
    "id": 242,
    "category": "Tables",
    "quantity": 1,
    "condition": 4,
    "brand": "Lampala",
    "width": null,
    "height": null,
    "depth": null,
    "diameter": 200.0,
    "comment": "Large round table"
  }
]
```

Update product

PATCH /case/%CASE ID%/product/%PRODUCT ID%

Field	Type	Required	Value
category	String	No	
quantity	INT	No	
condition	INT	No	1-5 for the condition. 5 is like new.
brand	String	No	
width	DECIMAL	No	
height	DECIMAL	No	
depth	DECIMAL	No	
diameter	DECIMAL	No	
comment	String	No	

The server will return an error if no fields are provided.

Sample request for a successful operation:

```
PATCH /case/42/product/241 HTTP/1.1  
Authentication-Type: Token  
Authentication-Token: Ad93dioik1909moggDkd4qw4adp24DO87  
Content-Type: application/json  
{  
    "brand": "Mickens",  
    "comment": "Unsure of brand, could also be Hoeg"  
}
```

On success nothing is returned.

Remove product

DELETE /case/%CASE ID%/product/%PRODUCT ID%

On success nothing is returned. All associated images and resources will be permanently deleted.

Add product image

POST /case/%CASE ID%/product/%PRODUCT ID%/image

Please be aware that this should be sent as a regular file and not as JSON.

Sample request:

```
POST /case/42/product/241/image HTTP/1.1  
Authentication-Type: Token  
Authentication-Token: Ad93dioik1909moggDkd4qw4adp24DO87  
Content-Type: image/png  
Content-Length: 3004  
  
raw image data
```

On success the new ImageID is returned:

```
HTTP Status: 201  
{  
    "id": 2410  
}
```

Remove product image

DELETE /case/%CASE ID%/product/%PRODUCT ID%/image/%IMAGE ID%

On success nothing is returned.

Changelog

Version *Changes*

2.3	- Removed name parameter from case endpoint - Removed ramp, parkingUnder200 and parkingOver200 parameters from environment endpoint - Reusing a e-mail for a user will not throw a duplication error
2.2	- Uploading images is no longer handled as form/multipart - Minor improvements for documentation
2.1	- Removed separate endpoint for updating product quantity - Changed update product endpoint from PUT to PATCH