

## File server API

This document gives an overview of how the file-server of the POSEIDON infrastructure works, and a description of how to use it.

The file server may be deployed at any server, but for the POSEIDON prototypes it is deployed on the SmartTracker research and innovation server where the POSEIDON web application is deployed. This may change in the future. The term `<BASE_URL>` in the documentation will therefore be referring to <https://ri.smarttracker.no>.

### Logging in

To be able to use the file server the client has to authenticate with the authentication token identifying the user in SmartTracker. This token can be received either by logging in using the SmartTracker API directly or by logging in using the file server API.

### Request

Method	POST	<code>&lt;BASE_URL&gt;/files/login.php</code>
Headers	Content-Type	application/json
Payload	JSON-Object	<pre>{   "username": "&lt;USERNAME&gt;",   "password": "&lt;PASSWORD&gt;" }</pre>

`<USERNAME>` and `<PASSWORD>` is you username and password in SmartTracker.

### Returns

On success the request above will return a JSON-object containing the authentication token that is used for interactions with the file server.

```
✓ {
  token: "36346d0f-6d08-423e-8059-491b1144ab6f"
}
```

### Errors

CODE	Error	Description
402	Payment Required	If authentication data is wrong
400	Bad Request	If body is missing or malformed

### Adding a resource

A file resource is added by posting a binary file and a category to the file server. The file will be given an UUID, and will be stored in a category.

## Request

Method	POST	<BASE_URL>/files/resource.php
Headers	Content-Type	multipart/form-data
	X-Auth-Token	<TOKEN>
Payload	type (TEXT)	String value containing the category of the resource.
	file (FILE)	The binary file that should be uploaded
	assetID (TEXT)	(OPTIONAL) indicating which asset the resource should be associated with. This only applies to users that control multiple assets.

## Returns

If successful, the request should return a JSON object containing the UUID of the resource.

```
{
  success: true,
  message: "File successfully uploaded",
  resourceID: "9db2c2fe-da91-429d-bb50-63e512a93328"
}
```

## Errors

Code	Error	Cause
400	Bad request	If the authentication token is not provided
401	Unauthorized	If the authentication token is not valid
400	Bad Request	If <i>type</i> is not provided
400	Bad Request	If <i>file</i> is not provided
500	Internal Server Error	Should not happen, but still...
400	Bad Request	If the user controls multiple assets and <i>assetID</i> is not provided
401	Unauthorized	If the <i>assetID</i> provided is not controlled by the user authenticated.

## Changing a resource

There are two things that can be changed for a resource: Either the category of the file or the binary content of the file resource. These operations cannot be executed simultaneously. If you want to change them both, you have to do it in two operations. If both the file content and the category are provided in the request, only the file content will be changed.

## Request

METHOD	POST	<BASE_URL>/files/resource.php
HEADERS	Content-Type	multipart/form-data
	X-Auth-Token	<TOKEN>
Payload	resourceID (TEXT)	The UUID of the resource that should be altered.
	file (FILE)	(OPTIONAL) Binary file with the new content. Type will not be changed if <i>file</i> is provided.
	type (TEXT)	(OPTIONAL) Category of the file. Will be ignored if <i>file</i> is provided.

	assetID (TEXT)	(OPTIONAL) indicating which asset the resource should be associated with. This only applies to users that control multiple assets.
--	----------------	--

## Returns

If successful the request will return a json object describing the change. Either:

```
{
  success: true,
  message: "New content for resource: '9db2c2fe-da91-429d-bb50-63e512a93328'"
}
```

when changing the file content or when changing the category:

```
{
  success: true,
  message: "Type has been changed for the resource: '9db2c2fe-da91-429d-bb50-63e512a93328'"
}
```

## Errors

Code	Error	Cause
400	Bad request	If the authentication token is not provided
401	Unauthorized	If the authentication token is not valid
404	Not found	If the no file with the provided <i>resourceID</i> can be found.
500	Internal Server Error	Should not happen, but still...
400	Bad Request	If the user controls multiple assets and <i>assetID</i> is not provided
401	Unauthorized	If the <i>assetID</i> provided is not controlled by the user authenticated.

## Listing resources

Resources can be listed by category, or as a complete list of resources for an asset. The entries in the list will contain the original name of the file resource, the UUID of the resource, the file's mime type and a md5 hash checksum of the file.

## Request

METHOD	GET	<BASE_URL>/files/resource.php
Headers	X-Auth-Token	<TOKEN>
Parameters	type (TEXT)	(OPTIONAL) Category of the file. Will list only resources in this category if provided.
	assetID (TEXT)	(OPTIONAL) indicating which asset the resource should be associated with. This only applies to users that control multiple assets.

Example: GET <https://ri.smartracker.no/files/resource.php?type=images>

## Returns

Successful result will return a JSON array containing metadata for the requested resources.

```
{
  image: [
    {
      resourceID: "0408c04f-8ac9-421e-bf98-9fe595c404ef",
      name: "Trondheim.jpg",
      mime: "jpg",
      md5: "8876bc2339fd622cd99ca4bbf2651613"
    },
    {resourceID: "90db5641-c74c-4a4c-b13d-ad20623567b7", name: "Snøkanon.jpg", mime: "jpg",...},
    {resourceID: "b351a3be-1f22-4267-9bc7-57cb65d427d2", name: "Trondheim.jpg", mime: "jpg",...}
  ],
  route: [
    {resourceID: "9db2c2fe-da91-429d-bb50-63e512a93328", name: "directions.json", mime: "json",...},
    {resourceID: "ba19decd-ebbd-45b3-afdb-56806db96bcb", name: "directions.json", mime: "json",...},
    {resourceID: "c8e12b5d-0f76-463f-9470-50ca45d4b066", name: "directions.json", mime: "json",...}
  ],
  shoppinglist: [
    {resourceID: "150c6ccf-2ea8-4a6c-a2ea-173002374155", name: "shoppinglist.json", mime: "json",...},
    {resourceID: "1dc5b4ea-7cd6-449e-ac7d-527380063ac2", name: "directions.json", mime: "json",...},
    {resourceID: "1df79998-dc93-483e-8ffa-34d739b394c6", name: "shoppinglist.json", mime: "json",...},
    {resourceID: "bf034f0f-8333-435a-8342-40894c504b3c", name: "shoppinglist.json", mime: "json",...}
  ]
}
```

or by category (image):

```
[
  {
    resourceID: "0408c04f-8ac9-421e-bf98-9fe595c404ef",
```

```

    name: "Trondheim.jpg",
    mime: "jpg",
    md5: "8876bc2339fd622cd99ca4bbf2651613"
  },
  {
    resourceID: "90db5641-c74c-4a4c-b13d-ad20623567b7",
    name: "Snøkanon.jpg",
    mime: "jpg",
    md5: "1358c53f9572424ba30565d8147a2937"
  },
  {
    resourceID: "b351a3be-1f22-4267-9bc7-57cb65d427d2",
    name: "Trondheim.jpg",
    mime: "jpg",
    md5: "8876bc2339fd622cd99ca4bbf2651613"
  }
]

```

## Errors

Code	Error	Cause
400	Bad request	If the authentication token is not provided
401	Unauthorized	If the authentication token is not valid
500	Internal Server Error	Should not happen, but still...
400	Bad Request	If the user controls multiple assets and <i>assetID</i> is not provided
401	Unauthorized	If the <i>assetID</i> provided is not controlled by the user authenticated.

## Fetching a resource

The resource is fetched by a GET request that returns the binary file.

### Request

<b>METHOD</b>	GET	<BASE_URL>/files/resource.php
<b>Headers</b>	X-Auth-Token	<TOKEN>
<b>Parameters</b>	resourceID (TEXT)	The UUID of the resource to be fetched
	assetID (TEXT)	(OPTIONAL) indicating which asset the resource should be associated with. This only applies to users that control multiple assets.

### Returns

A successful result will return the file as binary data, with the file's original name.

## Errors

Code	Error	Cause
400	Bad request	If the authentication token is not provided
401	Unauthorized	If the authentication token is not valid
404	Not found	If the no file with the provided <i>resourceID</i> can be found.
500	Internal Server Error	Hopefully you won't experience this one. Notify developer if you do.
400	Bad Request	If the user controls multiple assets and <i>assetID</i> is not provided
401	Unauthorized	If the <i>assetID</i> provided is not controlled by the user authenticated.

## Deleting a resource

Deleting a resource will remove it permanently from the file server.

## Request

METHOD	DELETE	<BASE_URL>/files/resource.php
Headers	X-Auth-Token	<TOKEN>
Parameters	resourceID (TEXT)	The UUID of the resource to be deleted
	assetID (TEXT)	(OPTIONAL) indicating which asset the resource should be associated with. This only applies to users that control multiple assets.

## Returns

A successful result will return confirmation that the resource has been deleted.

```
{  
  success: true,  
  message: "The resource with id '0a542d5b-29a9-4568-b8fe-3a2b4b9a2ad3'  
  was successfully deleted"  
}
```

## Errors

Code	Error	Cause
400	Bad request	If the authentication token is not provided
401	Unauthorized	If the authentication token is not valid
404	Not found	If the no file with the provided <i>resourceID</i> can be found.
500	Internal Server Error	Hopefully you won't experience this one. Notify developer if you do.

400	Bad Request	If the user controls multiple assets and <i>assetID</i> is not provided
401	Unauthorized	If the <i>assetID</i> provided is not controlled by the user authenticated.