

# Developers API / RESTful ([PDF](#))

---

## Signup

To login and obtain session token you need to send POST request to <http://docscamp.com/api/users/signup> with following parameteres:

Parameter	Description
email	Your e-mail
password	Password

In response you will get loginToken and set of user attributes. Please find example below:

```
{ "id": "2"
, "login": "jagermesh"
, "email": "jagermesh@gmail.com"
, "name": null
, "nickname": null
, "avatar": null
, "__permissions": {"canUpdate": true, "canRemove": false}
, "__loginToken": "3425ea54-6640-2e74-fd74-4ec2d34c8aca"
, "__expiredAt": "1366526928"
}
```

Some of attributes explained:

Parameter	Description
id	User unique identifier
__loginToken	Login token which will expire in regards to __expiredAt attribute, usually 2 hrs from current moment
__expiredAt	Login token expiration date/time as Unix timestamp (seconds since '1970-01-01 00:00:00' UTC)
__permissions	Set of user permissions. Contain two attributes - canUpdate and canRemove.

For invalid credentials you'll get error text and HTTP/1.0 403 Forbidden.

## Login

To login and obtain session token you need to send POST request to <http://docscamp.com/api/users/login> with following parameteres:

Parameter	Description
login	Your login
password	Password

In response you will get loginToken and set of user attributes. Please find example below:

```
{ "id":"2"
, "login":"jagermesh"
, "email":"jagermesh@gmail.com"
, "name":null
, "nickname":null
, "avatar":null
, "__permissions": {"canUpdate":true, "canRemove":false}
, "__loginToken":"3425ea54-6640-2e74-fd74-4ec2d34c8aca"
, "__expiredAt":"1366526928"
}
```

Some of attributes explained:

Parameter	Description
id	User unique identifier
__loginToken	Login token which will expire in regards to __expiredAt attribute, usually 2 hrs from current moment
__expiredAt	Login token expiration date/time as Unix timestamp (seconds since '1970-01-01 00:00:00' UTC)
permissions	Set of user permissions. Contain two attributes - canUpdate and canRemove.

For invalid credentials you'll get error text and HTTP/1.0 403 Forbidden.

## Retrieving projects list

To get projects list you need to send GET request to <http://docscamp.com/api/projects/> with following parameters:

Parameter	Description
__loginToken	Your login token
keyword	optional If passed - only project with name/description containing this keyword will be returned.

In response you will get list of projects with attributes. Please find example below:

```
[ { "id":"20"
, "name":"Blog"
, "description":"This project is published as my blog. I need to be careful what documents I'm adding here :)"
, "created_at":"2013-04-03 02:27:32"
, "__permissions":{"canUpdate":false,"canRemove":false}
, "display":{"authorName":"Jager Mesh!","createdAt":"21 April, 2013 at 02:05","documentsAmount":"3"}
, "flag":{"isPrivate":false}
, "author":{"id":"1","email":"admin","name":"Jager Mesh!","nickname":"jagermesh","avatar":null}
, "renderedDescription":"<p>This project is published as my blog. I need to be careful what documents I'm adding here :)</p>\n"
```

```
}  
]
```

Some of attributes explained:

Parameter	Description
id	Project unique identifier
name	Name of the project
description	Raw version of project description (usually Makrdown)
renderedDescription	HTML version of project description
display.authorName	Project author
display.createdAt	Formatted project creation date/time.
display.documentsAmount	Amount of documents in this project
flag.isPrivate	Means this is private project
__permissions	Set of project permissions. Contain two attributes - canUpdate and canRemove.

For invalid credentials you'll get error text and HTTP/1.0 401 Unathorized.

## Retrieving documents list

To get documents list you need to send GET request to <http://docscamp.com/api/documents/> with following parameteres:

Parameter	Description
__loginToken	Your login token
project_id	ID of project
keyword	optional If passed - only documents with specified keyword in name/body will be returned.

In response you will get list of documents with attributes. Please find example below:

```
[ { "id": "83"  
  , "project_id": "20"  
  , "name": "asd"  
  , "body": "# Quick Markdown Syntax Guide..."  
  , "created_at": "2013-04-05 00:04:06"  
  , "format_id": "1"  
  , "description": null  
  , "original_url": null  
  , "original_author": null  
  , "__permissions": {"canUpdate": false, "canRemove": false}  
  , "display": {"authorName": "Jager Mesh!", "createdAt": "21 April, 2013 at  
02:57"}  
  , "flag": {"isPrivate": false}  
  , "renderedBody": "<h1>Quick Markdown Syntax Guide</h1>  
  ,  
  "publicUrl": "http://localhost/breeze/docshub/document/c02d15ef466dc9499
```

```
6a99c9a65aa98ff.html "
  }
]
```

Some of attributes explained:

Parameter	Description
id	Document unique identifier
name	Name of the document
body	Raw version of project description (usually Makrdown)
renderedBody	HTML version of project description
display.authorName	Project author
display.createdAt	Formatted project creation date/time.
flag.isPrivate	Means this is private project
publicUrl	Url to this document on the web
__permissions	Set of document permissions. Contain two attributes - canUpdate and canRemove.

For invalid credentials you'll get error text and HTTP/1.0 401 Unathorized.

## Additional paramteres for GET methods

Both api/documents and api/projects can accept additional parameteres to minimize output.

Parameter	Description
__fields	Array of fields you want to get in result. For example id. Please make sure you are passing array of values. So it must looke like this __fields[]=id&__fields[]=description if converted to regular GET request.
__limit	Amount of records you want to get in result. Must be in pair with __skip.
__skip	Amount of records you want to skip. Must be in pair with __limit. So if you need 4 records starting from 3rd you need to pass __limit=4&__skip=2

## Getting only records count

Both api/documents and api/projects can return only records count. For this you need to pass additional parameteres \_\_result to your request with value count.

Parameter	Value
__result	count

## Getting only one record

Both api/documents and api/projects can return only one record. For this you need to send request like in example below:

GET /api/documents/DOCUMENTID

## Creating or updating document

To create document you need to send PUT request to `/api/documents/` with document fields as parameters.

To update document you need to send POST request to `/api/documents/ROWID` with document fields as parameters.

Parameter	Description
<code>_loginToken</code>	Your login token
<code>name</code>	Document name
<code>body</code>	Document body
<code>...</code>	Any other document field

Basically that's all document fields.