
cyl.me.api.docs Documentation

Release latest

May 10, 2016

1	Contents	3
1.1	Users	3
1.2	Points	5
1.3	Point types	8
1.4	Tasks	9
1.5	News	12
1.6	Instant messaging	14
1.7	Skills	15
1.8	Categories	16
1.9	Tokens	16
1.10	Invitations	17
1.11	Votes	18
1.12	Mobius ideas	19
1.13	Scopes	20
1.14	Storage	20
1.15	Subscriber	21
2	Indices and tables	23

- All responses emitted in JSON format
- API implements HATEOAS
- In all requests when authorization needed token must be provided
- Authorization between applications must be implemented using tokens
- Tokens have different ttl and scope. We have tokens for admins and tokens for users.

Answer type	Response
successful answer	status: ok response: data
conditionally successful answer	status: ok response: { error: "error description", code: "error code" }
unsuccessful response answer	status: notOk error: "error description" code: "error code"

All answers are successful except following cases: invalid token, error occurred internally in API (for example:database error)

Table of error codes

Code	Description
1	Token not specified
2	Token invalid
3	User not found
4	Authorization failed
5	Required fields are missing
6	Authorization by token failed
7	Cannot insert new element
8	Access denied
9	Element not found
10	Fields are invalid
11	Duplicated record

1.1 Users

User authorization handled by **tokens**

1.1.1 User information

/users/{user_id}/

method	GET
Auth	NO

Parameters

Parameter	Type	Description
user_id	integer	User's ID

Response

Parameter	Type	Description
id	integer	User Id
name	string	Use name
family_name	string	Family name
about	string	About yourself
phone	string	Phone
points_total	integer	Total amount of points
tasks_total	integer	Total amount of tasks
votes_total	integer	Total amount of votes
date	timestamp	Registration date
birthday	timestamp	User's birthday date
sex	string	male/female
avatar	string	Link to user avatar
skills	array	User skills
places	array	User places

```
{
  "status": "ok",
  "response":
  {
    "id": 4,
    "name": "Ivan",
    "family_name": "Sidorov"
  }
}
```

```

    "places": [
      {
        "lat": 55.33,
        "lng": 33.3,
        "address": "Moscow, Lenina 1"
      }
    ],
    "skills": [ 1, 6, 18 ]
  }
}

```

HATEOAS

/points/?user_id={user_id}	User's points
/skills/?user_id={user_id}	User's skills

1.1.2 New user

/users/{user_id}/

method	POST
Auth	YES

Input user

Parameter	Required	Type	Description
name	•	string	Use name
email	•	string	Email
family_name		string	Family name
about		string	About yourself
phone		string	Phone
birthday		timestamp	User's birthday date
sex		string	male/female
avatar		string	Link to user avatar
invitation		string	Invitation

Parameters

Parameter	Type	Description
user_id	integer	User's ID

Response

Parameter	Type	Description
id	integer	Id newly added user

1.1.3 Edit user

/users/{user_id}/

method	PUT
Auth	YES

Parameters

Parameter	Type	Description
user_id	integer	User's ID

Input vars

Parameter	Type	Description
Look	<i>user-information</i>	

Response

Parameter	Type	Description
result	1	

1.1.4 Remove user

/users/{user_id}/

method	DELETE
Auth	YES

Parameters

Parameter	Type	Description
user_id	integer	User's ID

Response

Parameter	Type	Description
result	1	

1.1.5 User favourite places

Coming soon...

1.1.6 User adds favourite place

Coming soon...

1.2 Points

1.2.1 Points list

/points/

Response

Parameter	Type	Description
points	integer	Id
Nested in points(look point's info)		

HATEOAS

/point/?page=1..10	Link to next data set
--------------------	-----------------------

1.2.2 Point's information

/points/{point_id}/

method	GET
Auth	NO

Parameters

Parameter	Type	Description
point_id	integer	Point's ID

Response

Parameter	Type	Description
id	integer	Id
user_id	integer	User Id
lat	float	Latitude
lng	float	Longitude
address	string	Address

1.2.3 New point

/points/

method	POST
Auth	YES

Input vars

Parameter	Required	Type	Description
lat	•	float	Latitude
lng	•	float	Longitude
address	•	string	Address

Response

Parameter	Type	Description
id	integer	Id newly added point

1.2.4 Edit point

/points/{point_id}/

method	PUT
Auth	YES

Input vars

Parameter	Type	Description
Look <i>point-information</i> input vars		

Parameters

Parameter	Type	Description
point_id	integer	Point's ID

Response

Parameter	Type	Description
result	1	

1.2.5 Remove point

/points/{point_id}/

method	DELETE
Auth	YES

Parameters

Parameter	Type	Description
point_id	integer	Point's ID

Response

Parameter	Type	Description
result	1	

1.2.6 Add comment

/points/{point_id}/comments/

method	POST
Auth	YES

Parameters

Parameter	Type	Description
point_id	integer	Point's ID

Input vars

Parameter	Type	Description
user_id	integer	User Id
comment	string	Comment

Response

Parameter	Type	Description
result	1	

1.2.7 Add like

/points/{point_id}/likes/

method	POST
Auth	YES

Parameters

Parameter	Type	Description
point_id	integer	Point's ID

Input vars

Parameter	Type	Description
user_id	integer	User Id

Response

Parameter	Type	Description
result	1	

1.2.8 Remove like

/points/{point_id}/likes/

method	DELETE
Auth	YES

Parameters

Parameter	Type	Description
point_id	integer	Point's ID

Input vars

Parameter	Type	Description
user_id	integer	User Id

1.2.9 Add point to favourite

Coming soon...

1.3 Point types

1.3.1 Point types list

/pointtypes/

method	GET
--------	-----

Response

Parameter	Type	Description
point_types	integer	Id
Nested in point types		
point_type_id	integer	Point type Id
name	string	Point type name

```
{
  "status": "ok",
  "response":
  {
    "point_types": [
      {
        "point_type_id": 1,
        "name": ""
      },
      {
        "point_type_id": 2,
        "name": ""
      }
    ]
  }
}
```

```
}
}
```

1.4 Tasks

1.4.1 Tasks list

/tasks/

Input vars

Parameter	Type	Description
skills	array	

Response

Parameter	Type	Description
tasks	array	
Nested in tasks(look <i>task-information</i>)		

HATEOAS

/tasks/?page=1..10	Link to next page
--------------------	-------------------

1.4.2 Task information

/tasks/{task_id}/

method	GET
Auth	NO

Parameters

Parameter	Type	Description
task_id	integer	Task's ID

Response

Parameter	Type	Description
id	integer	Task's ID
point_id	integer	Point ID
title	string	Title
description	string	Description
votes	integer	How many votes user ready to pay
required_solvers	integer	How many solvers required
lat	float	Latitude
lng	float	Longitude
address	string	Address
front_image	string	Link to front image
solvers_total	integer	Total amount of solvers
add_date	string	Insertion date
media	array	
skills	array	

```
{
  "status": "ok",
  "response":
  {
    "id": 1,
    "point_id": 10,
    "title": "I need your help",
    "description": "I need someone to conduct poll",
    "lat": 55.42,
    "lng": 33.10,
  }
}
```

HATEOAS

1.4.3 Show solvers

/tasks/{task_id}/solvers/

method	GET
Auth	NO

Parameters

Parameter	Type	Description
task_id	integer	Task's ID

Response

Parameter	Type	Description
solvers	array	
Nested in solvers		
user_id	integer	User ID

1.4.4 New task

/tasks/

method	POST
Auth	YES

Input vars

Parameter	Required	Type	Description
id	•	integer	Task's ID
point_id	•	integer	Point ID
title	•	string	Title
description	•	string	Description
votes	•	integer	How many votes user ready to pay
required_solvers		integer	How many solvers required
lat		float	Latitude
lng		float	Longitude
address		string	Address
front_image		string	Link to front image

1.4.5 Edit task

/tasks/{task_id}/

method	PUT
Auth	YES

Parameters

Parameter	Type	Description
task_id	integer	Task's ID

Input vars

Parameter	Type	Description
title	string	Title
description	string	Description
votes	integer	How many votes user ready to pay
required_solvers	integer	How many solvers required
lat	float	Latitude
lng	float	Longitude
address	string	Address
front_image	string	Link to front image

Response

Parameter	Type	Description
result	1	

1.4.6 Remove task

/tasks/{task_id}/

method	DELETE
Auth	YES

Parameters

Parameter	Type	Description
task_id	integer	Task's ID

Response

Parameter	Type	Description
result	1	

```
{
  "status": "ok",
  "response":
  {
    "result": 1,
    "error": null
  }
}
```

1.4.7 User takes task

Coming soon...

1.4.8 User notifies that task fulfilled

Coming soon...

1.4.9 Task owner approves fulfilled task

Coming soon...

1.4.10 User refuses task

Coming soon...

1.4.11 User closes task

Coming soon...

User can close task as unfinished then task changes status to NEW There are “phase” and “status” for task. We need to discuss how it supposed to work.

1.5 News

1.5.1 News list

/news/

method	GET
--------	-----

Input vars

Parameter	Type	Description
tags	string	Filter news by keyword found in tags
text	string	Filter news by keyword found in the body

Response

Parameter	Type	Description
news	integer	Id
Nested in news(look <i>single-news-information</i>)		

1.5.2 Single news information

/news/{news_id}/

method	GET
--------	-----

Parameters

Parameter	Type	Description
news_id	integer	News ID

Response

Parameter	Type	Description
id	integer	News ID
title	integer	Point Id
brief	string	News title
text	string	News text
author	author	Company or author name
link	string	Link to original article
tags	array	Tags(array of strings)
image	string	Url for main image
pubdate	timestamp	Publication date

1.5.3 Add news

/news/

method	POST
Auth	YES

Input vars

Parameter	Type	Description
Look news's info response data		

Response

Parameter	Type	Description
id	integer	Id newly added news

1.5.4 Edit news

/news/{news_id}/

method	PUT
Auth	YES

Parameters

Parameter	Type	Description
news_id	integer	News ID

Input vars

Parameter	Type	Description
Look	<i>single-news-information</i>	

Response

Parameter	Type	Description
result	1	

1.5.5 Remove news

/news/{news_id}/

method	DELETE
Auth	YES

Parameters

Parameter	Type	Description
news_id	integer	News ID

Response

Parameter	Type	Description
result	1	

1.6 Instant messaging

1.6.1 Messages of user

/messages/{user_id}/

method	GET
Auth	YES

Parameters

Parameter	Type	Description
user_id	integer	User's ID

Response

Parameter	Type	Description
messages	array	
Nested in messages		
id	integer	Message Id
user_id	integer	Use Id
message	string	Message
date	timestamp	Date
read	1 or 0	Is message read

1.6.2 New message

/messages/{user_id}/

method	POST
Auth	YES

Parameters

Parameter	Type	Description
user_id	integer	User's ID

Input user

Parameter	Type	Description
message	string	Message

Response

Parameter	Type	Description
id	integer	Id newly added message
error	string	Not empty if error occurred

1.6.3 User read message

Coming soon...

1.7 Skills

1.7.1 Skills list

/skills/

method	GET
Auth	NO

Input vars

Parameter	Type	Description
user_id	integer	Filter skills by user

Response

Parameter	Type	Description
skills	array	
Nested in skills		
skill_id	integer	Skill Id
name	string	Skill name

```
{
  "status": "ok",
  "response":
  {
    "skills": [
      {
        "skill_id": 1,
        "name": ""
      }
    ]
  }
}
```

```
    {
      "skill_id": 2,
      "parent_id": 1
      "name": " "
    },
    {
      "skill_id": 3,
      "parent_id": 1
      "name": " "
    },
  ],
}
}
```

1.8 Categories

1.8.1 Categories list

/categories/

method	GET
--------	-----

Response

Parameter	Type	Description
categories	array	
Nested in <i>categories</i>		
category_id	integer	Category Id
name	string	Category name

```
{
  "status": "ok",
  "response":
  {
    "categories": [
      {
        "category_id": 1,
        "name": " "
      },
      {
        "category_id": 2,
        "name": " "
      },
    ],
  }
}
```

1.9 Tokens

1.9.1 Get new token

/token/

method	GET
Auth	NO

Input vars

Parameter	Type	Description
soc_id	string	If user trying to authorize via social network network soc_id must be provided
login	string	Login or email
password	string	Password

Response

Parameter	Type	Description
token	string	Token
ttl	timestamp	Date to token invalidation

```
{
  "status": "ok",
  "response":
  {
    "token": "sf6df5s7dfcsd6f5cs6d5fc6s5df6s5cdfs6f5",
    "ttl": 1443780969
  }
}
```

1.9.2 Invalidate token

Logging out user

/token/invalidate/

method	GET
Auth	YES

Input vars

Parameter	Type	Description
user_id	integer	User Id

Response

Parameter	Type	Description
result	1 or 0	
error	string	Not empty if error occurred

1.10 Invitations

1.10.1 User's invitations

/invitations/

method	GET
Auth	YES

Input vars

Parameter	Type	Description
user_id	integer	User's ID

Response

Parameter	Type	Description
invitations	array	
Nested in invitations		
id	integer	Invitation Id
user_id	integer	User Id
code	string	Invitation code
used	1 or 0	Was invitation used

```
{
  "status": "ok",
  "response":
  {
    "invitations": [
      {
        "id": 1,
        "user_id": 4,
        "code": "234y234t2udg2u3y4dg23u4vd234yg",
        "used": 1
      },
      {
        "id": 2,
        "user_id": 5,
        "code": "ertetetertertert44g23u4vd234yg",
        "used": 0
      }
    ]
  }
}
```

1.11 Votes

1.11.1 List of all votes

/votes/

method	GET
Auth	YES

Input vars

Parameter	Type	Description
user_id	integer	User's ID

Response

Parameter	Type	Description
votes	array	
Nested in <i>votes</i>		
user_id	integer	User's ID
votes_total	integer	Total amount of votes
expire_date	timestamp	Next expiration date
expire_votes	integer	How many votes expire

1.11.2 User gives votes to point

Coming soon...

1.11.3 User gives votes to user

Coming soon...

1.12 Mabijs ideas

1.12.1 List of ideas

/projects/mabijs/ideas/

Input vars

Parameter	Type	Description

Response

Parameter	Type	Description
points	integer	Id
Nested in points(look point's info)		

1.12.2 Project's information

/projects/mabijs/ideas/{idea_id}/

method	GET
Auth	NO

Response

Parameter	Type	Description
idea_id	integer	Id
name	integer	User Id
brief	string	Title
description	string	Description
point_entity	object	Object with entity info
preview	object	Object with preview image info
comments_count	integer	Total amount of comments
website	string	Website
votes_count	integer	Total amount of votes
created_at	string	Insertion date

1.12.3 Point's comments

/projects/mabijs/ideas/{idea_id}/comments/

method	GET
Auth	NO

Response

Parameter	Type	Description
comments	array	
Nested in comments		
user_id	integer	User Id
comment	string	Comment
date	timestamp	Insertion date

1.13 Scopes

1.13.1 Scopes list

/scopes/

method	GET
Auth	NO

Response

Parameter	Type	Description
scopes	array	
Nested in scopes		
scope_id	integer	Scope Id
name	string	Scope name

```
{
  "status": "ok",
  "response":
  {
    "scopes": [
      {
        "scope_id": 1,
        "name": ""
      },
      {
        "scope_id": 2,
        "name": ""
      }
    ]
  }
}
```

1.14 Storage

This endpoint provides you the way to upload images on cyl.me project

/storage/image

method	GET
Auth	YES

Input vars

Parameter	Type	Description
file	string	Image

Response

Parameter	Type	Description
id	integer	Image ID
preview	string	Relative path to image

1.15 Subscriber

1.15.1 Subscriber

/subscriber/

method	POST
Auth	NO

Input vars

Parameter	Type	Description
email*	string	Email
name	string	User's name
project	string	Project where email taken from(default: mabius)

Response

Parameter	Type	Description
id	integer	Id
email	string	E-mail

```
{
  "status": "ok",
  "response":
  {
    "id": 1,
    "email": "email@example.com"
  }
}
```

Indices and tables

- `genindex`
- `modindex`
- `search`