
Derpeddit Documentation

Release 1.0

Viktor Miroshnikov

August 04, 2014

CONTENTS

1	Your assignment	3
1.1	About assignment	3
1.2	Prerequisites	3
1.3	Workflow	4
2	API Reference	7
2.1	General information	7
2.2	Authentication and registration	7
2.3	Posts	10
3	Glossary	17
	HTTP Routing Table	19
	Index	21

Contents:

YOUR ASSIGNMENT

1.1 About assignment

This assignment is a simulation of how a web ui team works in our company.

1.1.1 Goals

This assignment goal is to get know about your:

- technical skills in angular js and html overall
 - angular routing
 - angular directives
 - angular templating
 - twitter bootstrap framework
- familiarity with vcs, testing tools
- communication
- discipline

We also want you to get familiar with our workflow and toolkit.

1.2 Prerequisites

Before you start working on a test assignment you'll need to set up accounts. Asana and HipChat requires an email invite, thus you need to request it from us by sending an email to hi@stateful.eu with your request.

1.2.1 Github

- You'll need to create a github account if you dont have one
- Create a repository for ui project

We use Asana for a project management. You'll need to set up an account first.

1.2.2 Project management

We use Asana for a project management. You'll need to set up an account first.

1.2.3 Communication

We use HipChat for communication. Claim yourself at <http://www.hipchat.com/gCAkj0tyr> and receive an invitation.

1.3 Workflow

This is a replica of our company's workflow. You are expected to follow it during the project.

1.3.1 Tools you should use

- As you may guessed: AngularJS (routing, directives, templating..)
- Selenium and protractor.js for acceptance tests
- Github as repo hosting for your project

1.3.2 Before you start working

Make a day plan:

- Create yourself a task for a dayplan in your personal asana project
- Place a list of tasks you're going to do
- Assign yourself a task
 - Make sure you understand it
 - Add estimate tag

1.3.3 Work on tasks

Before you start on task

- Tag it with *#wip* (work in progress)
- Do you work in a separate branch

When you finish the task

- Make sure tests run correctly
- Check browser/platform compat
- Push your code to github
- Make a pull request to a feature branch
 - Mention asana task url in PR message
 - If there's no feature branch - create one
- Remove *#wip* tag

If you're in doubt or stuck shout about it into the hipchat as soon as possible. Ask if you dont understand something. Communication is the key to success.

API REFERENCE

2.1 General information

2.1.1 API Endpoint

API Server URL: <https://derpeddit.herokuapp.com>

2.1.2 Validation Errors

Invalid field value In cases when an invalid value specified for a field, or field is missing in supplied payload

Example:

```
HTTP/1.1 400 BAD REQUEST
```

```
{"password": ["This field is required."]}
```

Non-field errors In cases when payload was correct, but some constraint has been violated

Example:

```
HTTP/1.1 401 UNAUTHORIZED
```

```
{"non_field_errors": "Login and password do not match"}
```

Invalid payload In cases when an invalid JSON payload supplied

Example:

```
HTTP/1.1 400 BAD REQUEST
```

```
{"detail": "JSON parse error - Expecting ',' delimiter: line 1 column 19 (char 18)"}
```

2.2 Authentication and registration

Derpeddit uses **cookie based** authentication. In order to get auth cookie, you have to log in first.

2.2.1 Registration

POST /api/v1/core/auth/signup
Registers a new user

JSON Parameters

- **username** (*string*) – password
- **password** (*string*) – password

Request Headers

- **Content-type** – mandatory header, must be *application/json*

Status Codes

- **201** – user created

Example request:

```
POST /api/v1/core/auth/signup HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate
Content-Length: 55
Content-Type: application/json; charset=utf-8
Host: derpeddit.herokuapp.com
```

```
{
  "password": "derpedditftw",
  "username": "derpedditor"
}
```

Example response:

```
HTTP/1.1 201 CREATED
Allow: POST, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 05:57:32 GMT
Server: gunicorn/19.1.0
Set-Cookie: sessionid=38l4ehqzjq6cz1tbtus22dowdkqge753; expires=Mon, 18-Aug-2014 05:57:32 GMT; h
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur
```

```
{
  "id": 2,
  "username": "derpedditor"
}
```

2.2.2 Logging in

POST /api/v1/core/auth/login

Authenticates user against *login* and *password*.

Before you can log in you'll have to sign up.

JSON Parameters

- **login** (*string*) – login
- **password** (*string*) – password

Request Headers

- **Content-type** – mandatory header, which must be *application/json*

Example request:

```
POST /api/v1/core/auth/login HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate
Content-Length: 55
Content-Type: application/json; charset=utf-8
Host: derpeddit.herokuapp.com
User-Agent: HTTPie/0.8.0
```

```
{
  "password": "derpedditftw",
  "username": "derpedditor"
}
```

Example response - incorrect login and password:

```
HTTP/1.1 401 UNAUTHORIZED
Content-Type: application/json

{"non_field_errors": "Login and password do not match"}
```

Example response - correct login and password:

```
HTTP/1.1 200 OK
Allow: POST, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 06:04:58 GMT
Server: gunicorn/19.1.0
Set-Cookie: sessionid=zjherkdqlx9byq1o4lvlh2hil05siafc; expires=Mon, 18-Aug-2014 06:04:58 GMT; h
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur

{
  "id": 2,
  "username": "derpedditor"
}
```

2.2.3 Logging out

POST /api/v1/core/auth/logout

De-authenticates current user

Example request:

```
POST /api/v1/core/auth/logout HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Content-Length: 0
Cookie: sessionid=qn010o0pgx75qbzlumvo15hxc425nep5
Host: derpeddit.herokuapp.com
User-Agent: HTTPie/0.8.0
```

Example response:

```
HTTP/1.1 200 OK
Allow: POST, OPTIONS
Connection: keep-alive
Date: Mon, 04 Aug 2014 06:22:07 GMT
Server: gunicorn/19.1.0
Set-Cookie: sessionId=uv9092xpk5g50aeuhqn5nfec0t3mvnxv; expires=Mon, 18-Aug-2014 06:22:07 GMT; h
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur
```

2.2.4 Current user profile

GET /api/v1/core/auth/profile

Returns current user profile

Example request:

```
GET /api/v1/core/auth/profile HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Cookie: sessionId=qn010o0pgx75qbz1umvo15hxc425nep5
Host: derpeddit.herokuapp.com
User-Agent: HTTPie/0.8.0
```

Example response:

```
HTTP/1.1 200 OK
Allow: GET, HEAD, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 06:03:50 GMT
Server: gunicorn/19.1.0
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur
```

```
{
  "id": 2,
  "username": "derpedditor"
}
```

2.3 Posts

2.3.1 Get all posts

GET /api/v1/posts

Returns all posts :param string mode: to sort posts by vote score - *top*, to sort by create time - *newest*

Example request:

```
GET /api/v1/posts HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Cookie: sessionId=kku4ymfeacxkknxbcm7yc0m9e1jz3bb
```

```
Host: derpedit.herokuapp.com
User-Agent: HTTPie/0.8.0
```

Example response:

```
HTTP/1.1 200 OK
Allow: GET, POST, HEAD, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 07:23:45 GMT
Server: gunicorn/19.1.0
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur

[
  {
    "comments_total": 0,
    "created": "2014-08-04T07:21:18.868Z",
    "id": 3,
    "link": "http://google.com",
    "owner": {
      "id": 2,
      "username": "derpedditor"
    },
    "score": 1,
    "text": null,
    "title": "yawn"
  },
  {
    "comments_total": 0,
    "created": "2014-08-04T07:20:55.254Z",
    "id": 2,
    "link": "http://bing.com",
    "owner": {
      "id": 2,
      "username": "derpedditor"
    },
    "score": 1,
    "text": null,
    "title": "ayawn"
  }
]
```

2.3.2 Create a new post

POST /api/v1/posts

Creates a news *post*

JSON Parameters

- **title** (*string*) – Post title
- **link** (*string*) – URL if its a “Link post”
- **text** (*string*) – text if its a “Text post”

Example request:

```
POST /api/v1/posts HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate
Content-Length: 46
Content-Type: application/json; charset=utf-8
Cookie: sessionid=kku4ymfeacxkknxbcma7yc0m9e1jz3bb
Host: derpeddit.herokuapp.com
User-Agent: HTTPie/0.8.0
```

```
{
  "link": "http://google.com",
  "title": "yawn"
}
```

Example response:

```
HTTP/1.1 201 CREATED
Allow: GET, POST, HEAD, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 07:21:18 GMT
Server: gunicorn/19.1.0
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur
```

```
{
  "link": "http://google.com",
  "text": null,
  "title": "yawn"
}
```

2.3.3 Upvote a post

PUT /api/v1/posts/ (*post_id*) /vote
Votes +1 post

Parameters

- **post_id** (*number*) – Post id

JSON Parameters

- **positive** (*bool*) – must be true

Example request:

```
PUT /api/v1/posts/2/vote HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate
Content-Length: 18
Content-Type: application/json; charset=utf-8
Cookie: sessionid=kku4ymfeacxkknxbcma7yc0m9e1jz3bb
Host: derpeddit.herokuapp.com
User-Agent: HTTPie/0.8.0
```

```
{
  "positive": true
}
```


Example response:

```

HTTP/1.1 200 OK
Allow: PUT, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 07:30:19 GMT
Server: gunicorn/19.1.0
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur

{
  "new_post_score": 1
}

```

2.3.4 Downvote a post

PUT `/api/v1/posts/` (*post_id*) `/vote`

Votes -1 a *post*

Parameters

- **post_id** (*number*) – Post id

JSON Parameters

- **negative** (*bool*) – must be true

Example request:

```

PUT /api/v1/posts/2/vote HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate
Content-Length: 18
Content-Type: application/json; charset=utf-8
Cookie: sessionid=kku4ymfeacxkknxbcma7yc0m9e1jz3bb
Host: derpeddit.herokuapp.com
User-Agent: HTTPie/0.8.0

{
  "negative": true
}

```

Example response:

```

HTTP/1.1 200 OK
Allow: PUT, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 07:30:19 GMT
Server: gunicorn/19.1.0
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur

{
  "new_post_score": 0
}

```

2.3.5 Get post comments

GET `/api/v1/posts/ (post_id) /comment`

Returns all comments left for term:*post*

Parameters

- **post_id** (*number*) – Post id

Example request:

```
GET /api/v1/posts/2/comment HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
Host: derpeddit.herokuapp.com
User-Agent: HTTPie/0.8.0
```

Example response:

```
HTTP/1.1 200 OK
Allow: GET, POST, HEAD, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 08:01:32 GMT
Server: gunicorn/19.1.0
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur

[
  {
    "created": "2014-08-04T07:59:53.763Z",
    "id": 2,
    "owner": {
      "id": 2,
      "username": "derpedditor"
    },
    "post": 2,
    "text": "huh?"
  },
  {
    "created": "2014-08-04T07:59:43.340Z",
    "id": 1,
    "owner": {
      "id": 2,
      "username": "derpedditor"
    },
    "post": 2,
    "text": "meh"
  }
]
```

2.3.6 Make a new comment on a post

POST `/api/v1/posts/ (post_id) /comment`

Makes a new comments for a term:*post*

Parameters

- **post_id** (*number*) – Post id

JSON Parameters

- **text** (*string*) – comment text

Example request:

```
POST /api/v1/posts/2/comment HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate
Content-Length: 16
Content-Type: application/json; charset=utf-8
Cookie: sessionid=kku4ymfeacxkknxbcm7yc0m9e1jz3bb
Host: derpeddit.herokuapp.com
User-Agent: HTTPie/0.8.0
```

```
{
  "text": "huh?"
}
```

Example response:

```
HTTP/1.1 201 CREATED
Allow: GET, POST, HEAD, OPTIONS
Connection: keep-alive
Content-Type: application/json
Date: Mon, 04 Aug 2014 07:59:53 GMT
Server: gunicorn/19.1.0
Transfer-Encoding: chunked
Vary: Accept, Cookie
Via: 1.1 vegur
```

```
{
  "created": "2014-08-04T07:59:53.763Z",
  "id": 2,
  "owner": {
    "id": 2,
    "username": "derpedditor"
  },
  "post": 2,
  "text": "huh?"
}
```


GLOSSARY

derpeddit derpeddit is a *RaaS* API server that allows you to build your own content share interface

post, text post, link post Post is a text or URL that a derpeddit user can publish.

RaaS Reddit-as-a-Service

HTTP ROUTING TABLE

/api

POST /api/v1/core/auth/login, 8
POST /api/v1/core/auth/logout, 9
GET /api/v1/core/auth/profile, 10
POST /api/v1/core/auth/signup, 7
GET /api/v1/posts, 10
POST /api/v1/posts, 11
GET /api/v1/posts/(post_id)/comment, 14
POST /api/v1/posts/(post_id)/comment,
14
PUT /api/v1/posts/(post_id)/vote, 13

D

derpeddit, [17](#)

L

link post, [17](#)

P

post, [17](#)

R

RaaS, [17](#)

T

text post, [17](#)