Authentication {host}/api/auth/

**login**

pass username and password (POST) get user info + JWT

**signup**

pass username, password and password2 (POST) get user info + JWT

**refresh**

pass users refresh token to get (GET) new access token

**profile**

pass authorization header to get all user info (GET)

pass avatar or author\_pseudo to change either (PATCH)

if you simply pass query param ?user=<int> along with authorization token it will return someones profile info (it works only if request user is authenticated, so basically this shit is useless and probably should not be used)

**writer**

pass author\_pseudo to create writer profile (POST)

**subscribe**

pass author\_id to subscribe to author (POST)

**subscriptions**

pass authorization token to get list of writers you’re subscribed to (GET)

**notify**

pass author\_id to either receive notifications from writer (about him creating new story) or not (POST)

Stories {host}/api/stories/

**all\*\***

get all stories (GET), you can also pass query\_params such as: search\_prompt, genres (id’s), tag (id), sort\_by (likes\_count, -likes\_count, -created, created, -views\_counter, views\_counter, comments\_count, -comments\_count - use any if needed) those with minus sign return sorted in manner from greater to lesser (if I’m not mistaken)

***{host}/api/stories/all/?search\_prompt=<string>&genres=<list of int’s> e.g. 1, or 1,2,3, &tag=<int>&sort\_by=<string>***

**manipulate**

pass genre\* (string), title\* (string), body\* (string), image, tags (string with comma e.g.: tag,tag,tag, - valid; tag,tag,tag – invalid, each tag name should end with comma as I’m splitting by comma on backend) (POST) – to create story

pass the exact same shit to update story (PATCH)

pass story\_id to delete it (DELETE)

\* - must-have

**get\_genres\*\***

fetches all genres, sorted by popularity (most used come first) (GET)

**get\_tags\*\***

fetches all tags, sorted by popularity (most used come first) (GET)

**single**

fetch single story query\_param: story\_id (GET)

**get\_writer\_stories\*\***

fetch stories made by user (if he has writer profile) pass Authorization header (GET)

**get\_viewed\*\***

fetch stories viewed by user, pass Authorization header (GET)

**get\_liked\*\***

fetch stories liked by user, pass Authorization header (GET)

**react**

react to story, like or dislike (POST), request body should contain story\_id and type (type can be either like or dislike – idk why I did that, just use it as is)