

Teddit

[Link](#)

Docker-compose method (production)

Create `/srv/path/Teddit/config.js`

Paste this, editing the subs you want to follow:

To find this, `/bin/bash` into the teddit container, and `cat config.js` while running the stack without the `-volumes` variable.

```
const config = {
  domain: process.env.DOMAIN || '127.0.0.1', // Or for example 'teddit.net'
  use_reddit_oauth: process.env.USE_REDDIT_OAUTH === 'true' || false, // If false, teddit
  uses Reddit's public API. If true, you need to have your own Reddit app ID (enter the app
  ID to the "reddit_app_id" config key).
  cert_dir: process.env.CERT_DIR || '', // For example
  '/home/teddit/letsencrypt/live/teddit.net', if you are using https. No trailing slash.
  theme: process.env.THEME || 'auto', // One of: 'dark', 'sepia', 'auto', ''. Auto theme
  uses browser's theme detection (Dark or White theme). White theme is set by the empty the
  option ('').
  clean_homepage: !('CLEAN_HOMEPAGE' in process.env) || process.env.CLEAN_HOMEPAGE ===
  'true', // Allows the clean homepage to be used (similar to invidious), instead of the
  usual reddit-like frontpage
  flairs_enabled: !('FLAIRS_ENABLED' in process.env) || process.env.FLAIRS_ENABLED ===
  'true', // Enables the rendering of user and link flairs on teddit
  highlight_controversial: !('HIGHLIGHT_CONTROVERSIAL' in process.env) ||
  process.env.HIGHLIGHT_CONTROVERSIAL === 'true', // Enables controversial comments to be
  indicated by a typographical dagger (†)
  api_enabled: !('API_ENABLED' in process.env) || process.env.API_ENABLED === 'true', //
  Teddit API feature. Might increase loads significantly on your instance.
  api_force_https: process.env.API_FORCE_HTTPS === 'true' || false, // Force HTTPS to
  Teddit API permalinks (see #285).
  video_enabled: !('VIDEO_ENABLED' in process.env) || process.env.VIDEO_ENABLED ===
```

```
'true',
    redis_enabled: !('REDIS_ENABLED' in process.env) || process.env.REDIS_ENABLED ===
'true', // If disabled, does not cache Reddit API calls
    redis_db: process.env.REDIS_DB,
    redis_host: process.env.REDIS_HOST || '127.0.0.1',
    redis_password: process.env.REDIS_PASSWORD,
    redis_port: process.env.REDIS_PORT || 6379,
    ssl_port: process.env.SSL_PORT || 8088,
    nonssl_port: process.env.NONSSL_PORT || 8080,
    listen_address: process.env.LISTEN_ADDRESS || '0.0.0.0', // '0.0.0.0' will accept
connections only from IPv4 addresses. If you want to also accept IPv6 addresses use ':::'.
    https_enabled: process.env.HTTPS_ENABLED === 'true' || false,
    redirect_http_to_https: process.env.REDIRECT_HTTP_TO_HTTPS === 'true' || false,
    redirect_www: process.env.REDIRECT_WWW === 'true' || false,
    use_compression: !('USE_COMPRESSION' in process.env) || process.env.USE_COMPRESSION ===
'true',
    use_view_cache: process.env.USE_VIEW_CACHE === 'true' || false,
    use_helmet: process.env.USE_HELMET === 'true' || false, // Recommended to be true when
using https
    use_helmet_hsts: process.env.USE_HELMET_HSTS === 'true' || false, // Recommended to be
true when using https
    trust_proxy: process.env.TRUST_PROXY === 'true' || false, // Enable trust_proxy if you
are using reverse proxy like nginx
    trust_proxy_address: process.env.TRUST_PROXY_ADDRESS || '127.0.0.1',
    http_proxy: process.env.HTTP_PROXY,
    nsfw_enabled: !('NSFW_ENABLED' in process.env) || process.env.NSFW_ENABLED === 'true',
// Enable NSFW (over 18) content. If false, a warning is shown to the user before opening
any NSFW post. When the NSFW content is disabled, NSFW posts are hidden from subreddits
and from user page feeds. Note: Users can set this to true or false from their
preferences.
    videos_muted: !('VIDEOS_MUTED' in process.env) || process.env.VIDEOS_MUTED === 'true',
// Automatically mute all videos in posts
    post_comments_sort: process.env.POST_COMMENTS_SORT || 'confidence', // "confidence" is
the default sorting in Reddit. Must be one of: confidence, top, new, controversial, old,
random, qa, live.
    reddit_app_id: process.env.REDDIT_APP_ID || 'ABfYqdDc9qPh1w', // If "use_reddit_oauth"
config key is set to true, you have to obtain your Reddit app ID. For testing purposes
it's okay to use this project's default app ID. Create your Reddit app here:
https://old.reddit.com/prefs/apps/. Make sure to create an "installed app" type of app.
```

```

domain_replacements: process.env.DOMAIN_REPLACEMENTS
  ? (JSON.parse(process.env.DOMAIN_REPLACEMENTS).map(([p, r]) => [new RegExp(p, 'gm'),
r]))
  : [], // Replacements for domains in outgoing links. Tuples with regular expressions
to match, and replacement values. This is in addition to user-level configuration of
privacyDomains.

cache_control: !('CACHE_CONTROL' in process.env) || process.env.CACHE_CONTROL ===
'true', // If true, teddit will automatically remove all cached static files. By default
this is set to true.

cache_control_interval: process.env.CACHE_CONTROL_INTERVAL || 24, // How often the cache
directory for static files is emptied (in hours). Requires cache_control to be true.
Default is every 24 hours.

show_upvoted_percentage: !('SHOW_UPVOTED_PERCENTAGE' in process.env) ||
process.env.SHOW_UPVOTED_PERCENTAGE === 'true',

show_upvotes: !('SHOW_UPVOTES' in process.env) || process.env.SHOW_UPVOTES === 'true',
// If true, teddit will show number of upvotes in posts and points in comments.

post_media_max_heights: {
  /**
   * Sets the max-height value for images and videos in posts.
   * Default is 'medium'.
   */
  'extra-small': 300,
  'small': 415,
  'medium': 600,
  'large': 850,
  'extra-large': 1200
},

setexs: {
  /**
   * Redis cache expiration values (in seconds).
   * When the cache expires, new content is fetched from Reddit's API (when
   * the given URL is revisited).
   */
  frontpage: 600,
  subreddit: 600,
  posts: 600,
  user: 600,
  searches: 600,
  sidebar: 60 * 60 * 24 * 7, // 7 days

```

```

shorts: 60 * 60 * 24 * 31,
wikis: 60 * 60 * 24 * 7,
subreddits_explore: {
    front: 60 * 60 * 24 * 1,
    new_page: 60
},
},
rate_limiting: {
    enabled: false,
    initial_limit: 100, // This is the amount of page loads one IP address can make in one
minute without getting limited.
    limit_after_limited: 30 // When an IP is limited, this is the amount of page loads the
IP can make in one minute.
},
valid_media_domains: process.env.VALID_MEDIA_DOMAINS
    ? JSON.parse(process.env.VALID_MEDIA_DOMAINS)
    : ['preview.redd.it', 'external-preview.redd.it', 'i.redd.it', 'v.redd.it',
'a.thumbs.redditmedia.com', 'b.thumbs.redditmedia.com', 'emoji.redditmedia.com',
'styles.redditmedia.com', 'www.redditstatic.com', 'thumbs.gfycat.com', 'i.ytimg.com',
'i.imgur.com'],
valid_embed_video_domains: ['gfycat.com', 'youtube.com'],
reddit_api_error_text: `Seems like your instance is either blocked (e.g. due to API rate
limiting), reddit is currently down, or your API key is expired and not renewed properly.
This can also happen for other reasons.`,
/**
 * Here you can configure the suggested subreddits which are visible in the
 * cleaned homepage, and in the top bar.
 * You should keep at least 'All', and 'Saved'.
 *
 * If you set your configs with an environment variables for example with
 * docker-compose.yml, your suggested_subreddits config could be something
 * like this (note the quotes):
 * - SUGGESTED_SUBREDDITS=["Popular", "All", "Saved", "selfhosted", "linux",
"datahoarder", "Monero"]
 * or
 * - 'SUGGESTED_SUBREDDITS=["Popular", "All", "Saved", "selfhosted", "linux",
"datahoarder", "Monero"]'
 */
suggested_subreddits: process.env.SUGGESTED_SUBREDDITS

```

```

? JSON.parse(process.env.SUGGESTED_SUBREDDITS)
:
['Popular', 'All', 'Saved', 'AskReddit', 'pics', 'news',
'worldnews', 'funny', 'tifu', 'videos', 'gaming', 'aww',
'todayilearned', 'gifs', 'Art', 'explainlikeimfive',
'movies', 'Jokes', 'TwoXChromosomes',
'mildlyinteresting', 'LifeProTips', 'askscience',
'IAmA', 'dataisbeautiful', 'books', 'science',
'Showerthoughts', 'gadgets', 'Futurology',
'nottheonion', 'history', 'sports', 'OldSchoolCool',
'GetMotivated', 'DIY', 'photoshopbattles', 'nosleep',
'Music', 'space', 'food', 'UpliftingNews', 'EarthPorn',
'Documentaries', 'InternetIsBeautiful',
'WritingPrompts', 'creepy', 'philosophy',
'announcements', 'listentothis', 'blog'],

};

module.exports = config;

```

Docker-compose:

```

version: "3.8"

services:

  teddit:
    container_name: teddit
    image: teddit/teddit:latest
    environment:
      # - DOMAIN=teddit.net
      # - USE_HELMET=true
      # - USE_HELMET_HSTS=true
      # - TRUST_PROXY=true
      - REDIS_HOST=teddit-redis
    ports:
      - "8056:8080" #original 8080:8080
    networks:
      - teddit_net
    healthcheck:

```

```
    test: ["CMD", "wget", "--no-verbose", "--tries=1", "--spider",
"http://localhost:8080/about"]
    interval: 1m
    timeout: 3s
    depends_on:
      - teddit-redis
    volumes:
      - /srv/path/Teddit/config.js:/teddit/config.js

teddit-redis:
  container_name: teddit-redis
  image: redis:6.2.5-alpine
  command: redis-server
  environment:
    - REDIS_REPLICATION_MODE=master
  networks:
    - teddit_net

networks:
  teddit_net:
```

Note: This compose is made for a true "production" setup, and is made to be used to have teddit behind a reverse proxy, if you don't want that and prefer to directly access teddit via its port:

- Change `ports: - "127.0.0.1:8080:8080"` to `ports: - "8080:8080"`
- Remove `DOMAIN=teddit.net`, `USE_HELMET=true`, `USE_HELMET_HSTS=true`, `TRUST_PROXY=true`

Revision #6

Created 4 July 2023 20:14:46 by Admin

Updated 21 November 2024 22:16:21 by Admin