

Short urls

Сервис для создания коротких ссылок

Описание API

В проекте находятся несколько хендлеров для генерации, поиска, удаления, изменения и получения статистики по ссылкам, также добавлена возможность авторизации. Ссылки могут добавлять все пользователи, однако менять и удалять только зарегистрированные владельцы. Посмотрим на хендлеры поближе.

Links

POST	/links/shorten	Create Shorten	🔒	▼
GET	/links/{short_code}	Get Orig Url		▼
DELETE	/links/{short_code}	Delete Url	🔒	▼
PUT	/links/change	Change Url	🔒	▼
GET	/links/{short_code}/stats	Get Stats		▼
GET	/links/search/	Search Link		▼
GET	/links/all_user_urls/	Get All User Urls	🔒	▼

auth

POST	/auth/jwt/login	AuthJwt.Login		▼
POST	/auth/jwt/logout	AuthJwt.Logout	🔒	▼
POST	/auth/register	Register.Register		▼

post app/links/shorten - создание ссылки

По данному хендлеру можно создать ссылку либо со своим алиасом, либо с рандомно сгенерированным хешем, состоящем из 8 символов. В базу данных заносится ссылка с основными параметрами, а также с указанием автора создателя. Для незарегистрированных пользователей в поле автора указывается None. Ссылки проверяются на уникальность и на валидность.

```
{  
  "orig_url": "http://yandex.ru"  
}
```

```
{  
  "orig_url": "http://google.com",  
  "alias": "goo"  
}
```

```
{  
  "orig_url": "randomtxt"  
}
```

200

Response body

```
{  
  "status": "success",  
  "short_url": "http://ce93719d"  
}
```

200

Response body

```
{  
  "status": "success",  
  "short_url": "http://goo"  
}
```

400

Undocumented

Error: Bad Request

Response body

```
{  
  "detail": {  
    "status": "Bad Request",  
    "data": "invalid link"  
  }  
}
```

get app/links/{short_code} - переадресация по короткой ссылке

Используя этот запрос, можно перейти на сайт оригинальной ссылки используя созданную пару шагом ранее по короткой ссылке. В случае, если запись в бд не найдена, выкидывается ошибка с соответствующим описанием.

Name	Description	Code	Description
short_code <small>required</small> string (path)	<input type="text" value="goo"/>	200	Successful Response
		Media type	<input type="text" value="application/json"/>
		Controls Accept header.	

Name	Description
short_code <small>required</small> string (path)	<input type="text" value="not_existed_link"/>

400	Error: Bad Request
Undocumented	Response body
<pre>{ "detail": { "status": "Bad Request", "data": "data not found" } }</pre>	

delete app/links/{short_code} - удаление ссылки

Этим хендлером могут пользоваться только авторизованные пользователи. Тут можно удалить запись с ссылкой из бд. Ссылку могут удалять ее создатели. Если ссылку попытается удалить не владелец, то выкинет ошибку.

```
401 Error: Unauthorized
Undocumented
Response body
{
  "detail": {
    "status": "error",
    "data": "Unauthorized"
  }
}
```

```
403 Error: Forbidden
Undocumented
Response body
{
  "detail": {
    "status": "error",
    "data": "Forbidden"
  }
}
```

Code	Details
200	<p>Response body</p> <pre>{ "status": "success", "data": "url has been deleted" }</pre> <p>Response headers</p>

put app/links/change - изменить ссылку

У созданных ссылок можно обновлять их короткую форму на новый случайно сгенерированный хеш, либо же на алиас. По аналогии с предыдущим запросом каждый пользователь может менять только созданную им ссылку.

```
{  
  "short_code": "7120cf4d"  
}
```

200

Response body

```
{  
  "status": "success",  
  "data": "new short_url is http://a91cebbe"  
}
```

```
{  
  "short_code": "a91cebbe",  
  "alias": "new_url"  
}
```

200

Response body

```
{  
  "status": "success",  
  "data": "new short_url is http://new_url"  
}
```

get app/links/{short_code}/stats - получение информации по ссылке

По каждой созданной ссылке можно получить сводку основной информации о ней: оригинальная ссылка, дата истечения ссылки, создания, последнего использования и количество переходов по ссылке. Последнее использование и количество переходов обновляется с каждым разом, когда срабатывает перенаправление по вышеописанному хендлеру

200

Response body

```
{
  "status": "success",
  "orig_link": "http://google.com",
  "expires_at": "2025-04-01T07:35:27.977084+00:00",
  "date_of_create": "2025-04-01T07:25:27.977096+00:00",
  "last_usage": "2025-04-01T07:25:27.977097+00:00",
  "count_ref": 0
}
```

get app/links/search/ - получение короткой ссылки по оригиналу

Здесь можно найти короткую запись ссылки по ее оригиналу. В случае, если ссылки нет в бд, срабатывает ошибка.

Name	Description
original_url <small>★ required</small> string <small>(query)</small>	<input type="text" value="google.com"/>

200

Response body

```
{  
  "status": "success",  
  "short_url": "http://new_url"  
}
```


get app/links/all_user_urls/ - получение всех созданных пользователем ссылок

Используя этот хендлер, можно вывести все созданные пользователем ссылки.

200

Response body

```
{  
  "status": "success",  
  "data": [  
    "http://google.com",  
    "http://yandex.ru"  
  ]  
}
```

Response headers

Кеширование данных и фоновые задачи

В проекте я кеширую все `get` хендлеры, а именно редирект по ссылкам, получение статистики, получение всех ссылок, созданных пользователем, и получение оригинала ссылок по коротким формам. Кеш сохраняется при первом запросе и обновляется при срабатывании таких хендлеров, как удаление, изменение, создание ссылок. Срок действия кеша равен 60 секундам.

Фоновые задачи

В проекте действует система удаления ссылок, у которых истекает срок их действия. Срок действия для каждой ссылки устанавливается при создании и равен 10 минутам.

Дополнительно был реализован механизм удаления ссылки, если с момента их последнего использования прошло 5 минут. Дата последнего использования устанавливается впервые при создании и далее обновляется каждый раз, когда происходит редирект по ссылкам.

Скриншоты локального запуска докер компоуз

```
[+] Running 4/4
✓ Container db app      Created
✓ Container redis app   Created
✓ Container celery_worker Created
✓ Container fastapi app  Created

attaching to celery_worker, db_app, fastapi_app, redis_app
redis app    1:C 31 Mar 2025 21:54:05.414 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
redis app    1:C 31 Mar 2025 21:54:05.415 * o08000000000000000 Redis is starting o08000000000000000
redis app    1:C 31 Mar 2025 21:54:05.415 * Redis version 7.4.2, bits=64, commit=00000000, modified=0, pid=1, just started
redis app    1:C 31 Mar 2025 21:54:05.415 # Warning: no config file specified, using the default config. In order to specify a config file use redis-server /path/to/redis.conf
redis app    1:M 31 Mar 2025 21:54:05.416 * monotonic clock: POSIX clock_gettime
redis app    1:M 31 Mar 2025 21:54:05.421 * Running mode=standalone, port=6379.
redis app    1:M 31 Mar 2025 21:54:05.424 * Server initialized
redis app    1:M 31 Mar 2025 21:54:05.425 * Loading RDB produced by version 7.4.2
redis app    1:M 31 Mar 2025 21:54:05.426 * RDB age 175486 seconds
redis app    1:M 31 Mar 2025 21:54:05.427 * RDB memory usage when created 1.28 Mb
db app       PostgreSQL Database directory appears to contain a database; Skipping initialization
db app       1:M 31 Mar 2025 21:54:05.427 * Done loading RDB, keys loaded: 3, keys expired: 0.
redis app    1:M 31 Mar 2025 21:54:05.427 * DB loaded from disk: 0.003 seconds
redis app    1:M 31 Mar 2025 21:54:05.427 * Ready to accept connections tcp
db app       2025-03-31 21:54:05.600 UTC [1] LOG: starting PostgreSQL 16.8 (Debian 16.8-1.pgdg12b1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
db app       2025-03-31 21:54:05.601 UTC [1] LOG: listening on IPv4 address '0.0.0.0', port 5432
db app       2025-03-31 21:54:05.601 UTC [1] LOG: listening on IPv6 address '::', port 5432
db app       2025-03-31 21:54:05.608 UTC [1] LOG: listening on Unix socket '/var/run/postgresql/.s.PGSQL.5432'
db app       2025-03-31 21:54:05.634 UTC [28] LOG: database system was shut down at 2025-03-29 21:09:19 UTC
db app       2025-03-31 21:54:05.605 UTC [1] LOG: database system is ready to accept connections
fastapi app  INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
fastapi app  INFO [alembic.runtime.migration] Will assume transactional DDL.
fastapi app  INFO [alembic.runtime.migration] Running upgrade >=250a385a61b, initial commit
fastapi app  [2025-03-31 21:54:09 +0000] [10] [INFO] Starting gunicorn 23.0.0
fastapi app  [2025-03-31 21:54:09 +0000] [10] [INFO] Listening at: http://0.0.0.0:8080 (10)
fastapi app  [2025-03-31 21:54:09 +0000] [10] [INFO] Using worker: uvicorn.workers.UvicornWorker
fastapi app  [2025-03-31 21:54:09 +0000] [11] [INFO] Booting worker with pid: 11
fastapi app  [2025-03-31 21:54:09 +0000] [12] [INFO] Booting worker with pid: 12
fastapi app  [2025-03-31 21:54:09 +0000] [13] [INFO] Booting worker with pid: 13
fastapi app  [2025-03-31 21:54:09 +0000] [14] [INFO] Booting worker with pid: 14
celery_worker /usr/local/lib/python3.9/site-packages/celery/platforms.py:829: SecurityWarning: You're running the worker with superuser privileges; this is absolutely not recommended!
```

```
celery_worker - ** ----- [config]
celery_worker - ** ----- -> app: tasks:0x73d194219340
celery_worker - ** ----- -> transport: redis://redis:6379/0
celery_worker - ** ----- -> result_backend: disabled//
celery_worker - *** ... * ... -> concurrency: 5 (prefork)
celery_worker - ** ----- -> task_events: OFF (enable -E to monitor tasks in this worker)
celery_worker - .. *****
celery_worker ----- [queues]
celery_worker -> celery exchange=celery(direct) key=celery
celery_worker
celery_worker [tasks]
celery_worker celery_folder.tasks.task_expire_url
celery_worker
celery_worker [2025-03-31 21:54:11.510: WARNING/MainProcess] /usr/local/lib/python3.9/site-packages/celery/worker/consumer/consumer.py:588: CPendingDeprecationWarning: The broker_
celery_worker connection_retr configuration setting will no longer determine
celery_worker whether broker connection retries are made during startup in Celery 6.0 and above.
celery_worker If you wish to retain the existing behavior for retrying connections on startup,
celery_worker you should set broker_connection_retr_ on_startup to True.
celery_worker warnings.warn(
celery_worker
celery_worker [2025-03-31 21:54:11.543: INFO/MainProcess] Connected to redis://redis:6379/0
celery_worker [2025-03-31 21:54:11.544: WARNING/MainProcess] /usr/local/lib/python3.9/site-packages/celery/worker/consumer/consumer.py:588: CPendingDeprecationWarning: The broker_
celery_worker connection_retr configuration setting will no longer determine
celery_worker whether broker connection retries are made during startup in Celery 6.0 and above.
celery_worker If you wish to retain the existing behavior for retrying connections on startup,
celery_worker you should set broker_connection_retr_ on_startup to True.
celery_worker warnings.warn(
celery_worker
celery_worker [2025-03-31 21:54:11.568: INFO/MainProcess] mingle: searching for neighbors
celery_worker [2025-03-31 21:54:12.593: INFO/MainProcess] mingle: all alone
celery_worker
celery_worker [2025-03-31 21:54:12.639: INFO/MainProcess] celery@9925bda3a2el ready.
fastapi app  [2025-03-31 21:54:12 +0000] [14] [INFO] Started server process [14]
fastapi app  [2025-03-31 21:54:12 +0000] [12] [INFO] Started server process [12]
fastapi app  [2025-03-31 21:54:12 +0000] [12] [INFO] Waiting for application startup.
fastapi app  [2025-03-31 21:54:12 +0000] [13] [INFO] Started server process [13]
fastapi app  [2025-03-31 21:54:12 +0000] [12] [INFO] Application startup complete.
fastapi app  [2025-03-31 21:54:12 +0000] [14] [INFO] Waiting for application startup.
fastapi app  [2025-03-31 21:54:12 +0000] [14] [INFO] Application startup complete.
fastapi app  [2025-03-31 21:54:12 +0000] [13] [INFO] Waiting for application startup.
fastapi app  [2025-03-31 21:54:12 +0000] [13] [INFO] Application startup complete.
fastapi app  [2025-03-31 21:54:13 +0000] [11] [INFO] Started server process [11]
fastapi app  [2025-03-31 21:54:13 +0000] [11] [INFO] Waiting for application startup.
fastapi app  [2025-03-31 21:54:13 +0000] [11] [INFO] Application startup complete.
```

Итоги

В проекте удалось реализовать все основные функции, использовать несколько сервисов, а также был добавлен дополнительный функционал: вывод всех ссылок, созданных пользователем, и механизм удаления ссылок по истечению даты последнего использования и изменение даты последнего использования по редиректу по ссылке.

Из того, что не получилось реализовать: при обновлении у хендлеров удаляется весь кеш этого хендлера, а не только тот, ссылка которого была обновлена. При срабатывании кеша по редиректу ссылки не удалось обновлять параметр `count_ref` у ссылок. Он обновляется только тогда, когда срабатывает редирект не по кешу.