URL Shortener API Assignment

By G Rushivardhan
email: rushivardhan18@gmail.com
Ph:+91 9603366515

Overview

The URL Shortener API allows users to shorten long URLs and retrieve the original URLs using a short code. It is built using FastAPI and PostgreSQL. This documentation provides details on how to set up, run, and use the API. Database is deployed in a liver server using render and FastAPI application is deployed using vercel app, details are mentioned below.

Features

- **Shorten URL**: Accept a long URL and return a shortened version.
- **Redirect to Original URL**: Retrieve the original URL using the shortened code and redirect to it.
- **Retrieve All URL Details**: Fetch all stored URLs and their corresponding short codes.

Technology Stack

- **FastAPI**: For building the web API.
- **PostgreSQL**: For storing the original URLs and their corresponding short codes.
- **SQLAlchemy**: For database ORM.
- **Pydantic**: For data validation and serialization.
- **Render**: For Database deployment
- **Vercel**: For Backend deployment

Requirements

- Python 3.8+
- PostgreSQL
- FastAPI
- SQLAlchemy
- Uvicorn (for running the FastAPI application)

Setup Instructions

1. Clone the Repository

bash

```
git clone https://github.com/Rushi1820/Url_Shortner.git
```

2. Create a Virtual Environment

bash

```
python -m venv venv
```

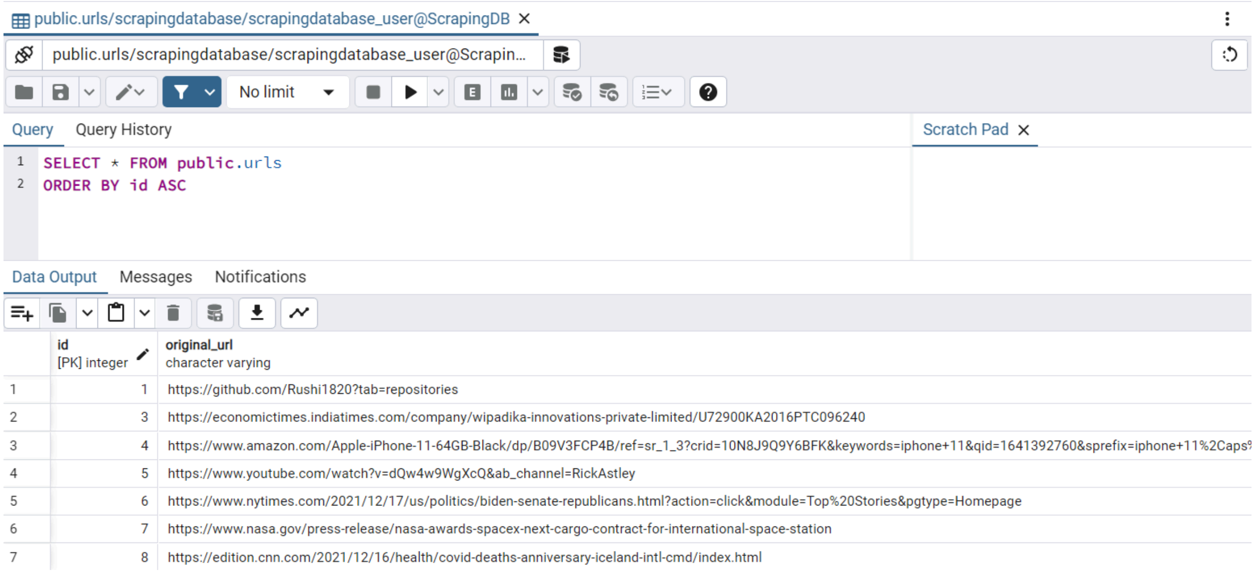
3. Install Dependencies

bash

```
pip install -r requirements.txt
```

4. Configure the Database

The database is deployed on a Render server. All tables will be automatically created by FastAPI when the application is run. You do not need to manually set up the tables or configure additional settings beyond specifying the correct database connection string in your FastAPI application.


public.urls/scrapingdatabase/scrapingdatabase_user@ScrapingDB
public.urls/scrapingdatabase/scrapingdatabase_user@Scrapin...
No limit
Query
Query History
Scratch Pad
Data Output
Messages
Notifications

```
SELECT * FROM public.urls
ORDER BY id ASC
```

	id [PK] integer	original_url character varying
1	1	https://github.com/Rushi1820?tab=repositories
2	3	https://economictimes.indiatimes.com/company/wipadika-innovations-private-limited/U72900KA2016PTC096240
3	4	https://www.amazon.com/Apple-iPhone-11-64GB-Black/dp/B09V3FCP4B/ref=sr_1_3?crid=10N8J9Q9Y6BFK&keywords=iphone+11&qid=1641392760&sprefix=iphone+11%2Caps
4	5	https://www.youtube.com/watch?v=dQw4w9WgXcQ&ab_channel=RickAstley
5	6	https://www.nytimes.com/2021/12/17/us/politics/biden-senate-republicans.html?action=click&module=Top%20Stories&pgtype=Homepage
6	7	https://www.nasa.gov/press-release/nasa-awards-spacex-next-cargo-contract-for-international-space-station
7	8	https://edition.cnn.com/2021/12/16/health/covid-deaths-anniversary-iceland-intl-cmd/index.html

6. Run the Application

Use Uvicorn to run the FastAPI application:

```
py -m uvicorn main:app --reload
```

The API will be available at `http://127.0.0.1:8000`.

API Endpoints

1. POST `/shorten`

URL: http://localhost:8000/shorten

Description: Accept a long URL and return a shortened URL.

Request Body:

json

```
{
  "original_url": "https://example.com/very/long/url"
}
```

Response:

json

```
{
  "original_url": "https://example.com/very/long/url",
  "short_code": "abc123"
}
```

2. GET `/redirect/{short_code}`

URI: http://localhost:8000/redirect/{short_code}

Description: Redirect to the original URL using the shortened code.

Response: Redirects to the original URL.

Example: If the short code is `abc123`, then `GET /redirect/abc123` will redirect to `https://example.com/very/long/url`.

3. GET `/getallurlsdetails`

URL: http://localhost:8000/getallurldetails

Description: Retrieve all stored URLs and their corresponding short codes.

Response:

json
Copy code

```
[
  {
    "original_url": "https://github.com/Rushi1820?tab=repositories",
    "short_code": "edb917"
  },
  {
    "original_url":
"https://economictimes.indiatimes.com/company/wipadika-innovations-pri
vate-limited/U72900KA2016PTC096240",
    "short_code": "17237d"
  },
  {
    "original_url":
"https://www.amazon.com/Apple-iPhone-11-64GB-Black/dp/B09V3FCP4B/ref=s
r_1_3?crid=10N8J9Q9Y6BFK&keywords=iphone+11&qid=1641392760&sprefix=iph
one+11%2Caps%2C280&sr=8-3",
    "short_code": "56492a"
  },
  {
    "original_url":
"https://www.youtube.com/watch?v=dQw4w9WgXcQ&ab_channel=RickAstley",
    "short_code": "84ded1"
  },
  {
]
```

Live Application Details

URL Shortener Application is deployed using Vercel. You can interact with the live application and test the endpoints directly.

- **Swagger UI**: Access the API documentation and test endpoints at FastAPI - Swagger UI (url-shortner-six-pi.vercel.app). This interface allows you to explore and test the API without needing to set up the project locally.
- **Redirecting URL**: To test URL redirection, use the localhost URL format with the short code. For example, if you have a short code `356a1b`, you can test redirection by visiting: `http://localhost:8000/redirect/356a1b`

  This will redirect you to the original URL associated with the short code.

Testing the API

If you want to test the API Locally, then use tools like Postman or directly from the Swagger UI.

1. **Swagger UI**: Navigate to `http://127.0.0.1:8000/docs` to access the auto-generated documentation and test the endpoints.
2. **Postman**: Send HTTP requests to the endpoints to verify functionality.

Troubleshooting

Common Issues

- **404 Not Found**: Ensure that the endpoint path is correct and matches what's defined in the routes.
- **Database Connection Errors**: Verify your PostgreSQL credentials and ensure the database is running.
- **Import Errors**: Ensure that your Python environment is activated and all dependencies are installed.

Logs

Check the terminal output where Uvicorn is running for any error messages or logs that can help diagnose issues.

Assignment submitted By : G Rushivardhan

Email: rushivardhan18@gmail.com

Ph: +91 9603366515

Thank you for considering my application. I look forward to discussing how I can contribute to your team.