Open in app ↗

Search

Write

K

Adding HTTPS to FastAPI

Mario van Rooij · Follow

2 min read · Mar 25, 2023

19 1

Here's a step-by-step guide on how to add HTTPS to your FastAPI server:

1. Obtain an SSL/TLS certificate You will need an SSL/TLS certificate for your domain. You can obtain a certificate from a certificate authority (CA), or you can use a free certificate from Let's Encrypt. Let's Encrypt offers free SSL/TLS certificates that are automatically renewed every 90 days.
2. Install dependencies You will need to install the necessary dependencies for SSL/TLS support in your FastAPI server. The following command installs the required dependencies:

```
pip install fastapi uvicorn[standard] cryptography
```

3. Create a self-signed certificate (optional) If you don't have an SSL/TLS certificate, you can create a self-signed certificate for testing purposes. You

can generate a self-signed certificate using OpenSSL:

```
openssl req -x509 -newkey rsa:4096 -nodes -out cert.pem -keyout key.pem -days 365
```

4. Configure your FastAPI app To enable HTTPS in your FastAPI app, you need to create an instance of the `SSLContext` class and configure it with your certificate and key. You can do this by creating a new `ssl.SSLContext` object and setting the `certfile` and `keyfile` attributes to the path of your certificate and key files:pythonCopy code

```
  import ssl
```

```
app = FastAPI()

ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLS_SERVER)
ssl_context.load_cert_chain('/path/to/cert.pem',
keyfile='/path/to/key.pem')
```

5. Start your server with SSL/TLS support You can start your FastAPI server with SSL/TLS support by using the `uvicorn` command with the `--ssl-keyfile` and `--ssl-certfile` options:

```
uvicorn main:app --ssl-keyfile /path/to/key.pem --ssl-certfile /path/to/cert.pem
```

Alternatively, you can pass the `ssl` argument to the `uvicorn.run()` function:

```
if __name__ == "__main__":
    uvicorn.run("main:app", host="0.0.0.0", port=8000, ssl=ssl_context)
```

And that's it! With these steps, your FastAPI app should now be running with HTTPS support.

after adding HTTPS support to your FastAPI server, you can use HTTPS instead of HTTP when making requests to the API.

To use HTTPS, simply change the URL in your client-side code from `http://` to `https://`. For example, if your FastAPI server is running on `https://example.com`, you would make requests to `https://example.com/api/endpoint` instead of `http://example.com/api/endpoint`.

It's important to note that using HTTPS provides an additional layer of security and helps protect sensitive data, such as access tokens and user credentials, from interception and snooping. Therefore, it's always recommended to use HTTPS when communicating with a web API over the internet.

You may need to replace the SSL/TLS certificate periodically, depending on the validity period of the certificate. SSL/TLS certificates typically have an expiration date, after which they are no longer considered valid.

If you obtained a certificate from a certificate authority (CA), the expiration date of the certificate will depend on the validity period set by the CA. Typically, SSL/TLS certificates issued by CAs have a validity period of one or two years.

If you obtained a free certificate from Let’s Encrypt, the certificate is valid for 90 days. However, Let’s Encrypt provides an automated process to renew certificates, so you don’t need to manually replace the certificate every 90 days.

In any case, it’s important to keep track of the expiration date of your SSL/TLS certificate and renew or replace it before it expires. Failure to do so can result in the web server becoming inaccessible or insecure.

API Python Encryption Programming Fastapi

Written by Mario van Rooij

9 Followers

Follow