Payden

Quick Start Guide

To use Payden, three services need to be configured and run:

- A Raiden node
- The Payden proxy
- A web server serving the content to be paywalled

These three services are defined in a docker compose file located in `docker/compose/docker-compose.yaml.`

This guide will provide step by step instructions for setting up Payden on mainnet using a basic configuration. A README with more in-depth instructions and advanced use-cases is included with the product.

Raiden Node

The Raiden node configuration can be found in `docker/compose/raiden/config/config.toml.` For a basic configuration the following adjustments are needed:

- Add your Ethereum keystore file in `docker/compose/raiden/config/keystore.`
- Add your keystore password as a plain text file in `docker/compose/raiden/config/password.`

In the config file:

- Change `address` to the address of your keystore file.
- Change `network-id` to mainnet.
- Change `environment-type` to production.
- Delete the `development-environment` line.

Payden Proxy

The Payden proxy configuration can be found in `docker/compose/payden/config/config.conf.` For a basic configuration the following adjustments are needed:

In the config file:

- Change `TOKEN_ADDRESS` to the address for either WETH or DAI.
- Change `TOKEN_SYMBOL` to either WETH or DAI depending on which token is used.
- Change `DEFAULT_AMOUNT` to the amount (as a decimal number) that you want to charge for an article, e.g. `0.5` as in `0.5 DAI` which = ~`$0.5.`
- Change `DEFAULT_CONTENT_PROVIDER_ID` to a unique name, e.g. `my_unique_content_website.`
- For `CONTENT_PROTECTED_REGEX`, define a Python regex for your protected content. The example in the configuration protects all content once that comes

after articles in a URL, e.g. `articles/title-of-article`. The structure of the regex is highly individual and a tool like regex101 can help for visualizing which parts get protected.

- For `CONTENT_SCOPE_REGEX`, define a Python regex for individual content. It works together with the `CONTENT_PROTECTED_REGEX` and needs to be a matching group that includes `content_scope`. The example in the configuration protects individual content once that comes after `articles`. In the URLs `articles/one.html` and `articles/two.html` both `one.html` and `two.html` will be protected. The structure of the regex is highly individual and a tool like regex101 can help for visualizing which parts get protected.
- Change `PAYWALL_STORE_URL` to the URL where you will be serving the Payden payment page.

Content Web-Server

This is highly dependant on the individual setup and infrastructure used for serving the content and can be tailored for any specific needs.

For the most basic approach however, it is enough to replace the example content in `docker/compose/content/data` with your own content.

Routing

When the three services have been configured and are up and running the Payden proxy will be exposed on port `8080` and the Payden payment page will be exposed on port `8081`.

Your webserver needs to be accessible via a publicly registered domain and route requests to the Payden proxy and the Payden payment page over HTTPS. The configurations for this are highly dependant on your individual setup and infrastructure.