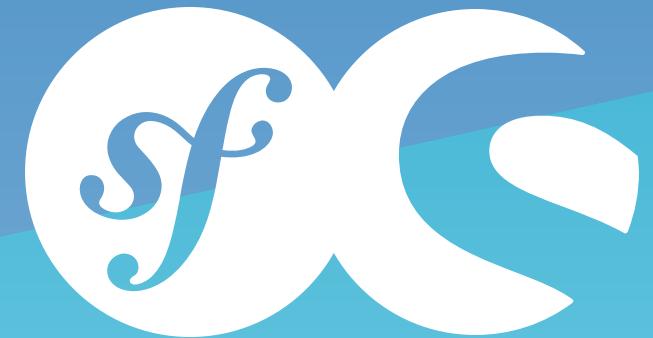


Symfony CLI for Shopware development





Why?

- ⚡ light-weight on
 - diskspace
 - performance penalty
 - filesystem performance
- ✅ official symfony way
 - Symfony Profiler integration
 - Platform.sh integration → Shopware PaaS 😊
- 💻 platform agnostic



What?

- single binary built with golang
- uses natively installed binaries for php and extensions
(easily done with homebrew on mac 😊)
- global proxy service for local SSL
- env-injecting proxy commands



How?

Prerequisites

- Docker (e.g. using OrbStack on Mac)
- System PHP

 quick install:

```
brew install shivammathur/php/php@8.4  
brew install shivammathur/extensions/xdebug@8.4
```

```
brew install symfony-cli/tap/symfony-cli
```

```
docker compose up -d
```

```
symfony serve
```

```
$ symfony serve

[WARNING] The local web server is optimized for local development and MUST never be used in a production setup.

[WARNING] Please note that the Symfony CLI only listens on 127.0.0.1 by default since version 5.10.3.
          You can use the --allow-all-ip or --listen-ip flags to change this behavior.

[OK] Web server listening
      The Web server is using PHP FPM 8.4.3
      https://127.0.0.1:8078

Stream the logs via symfony server:log
```



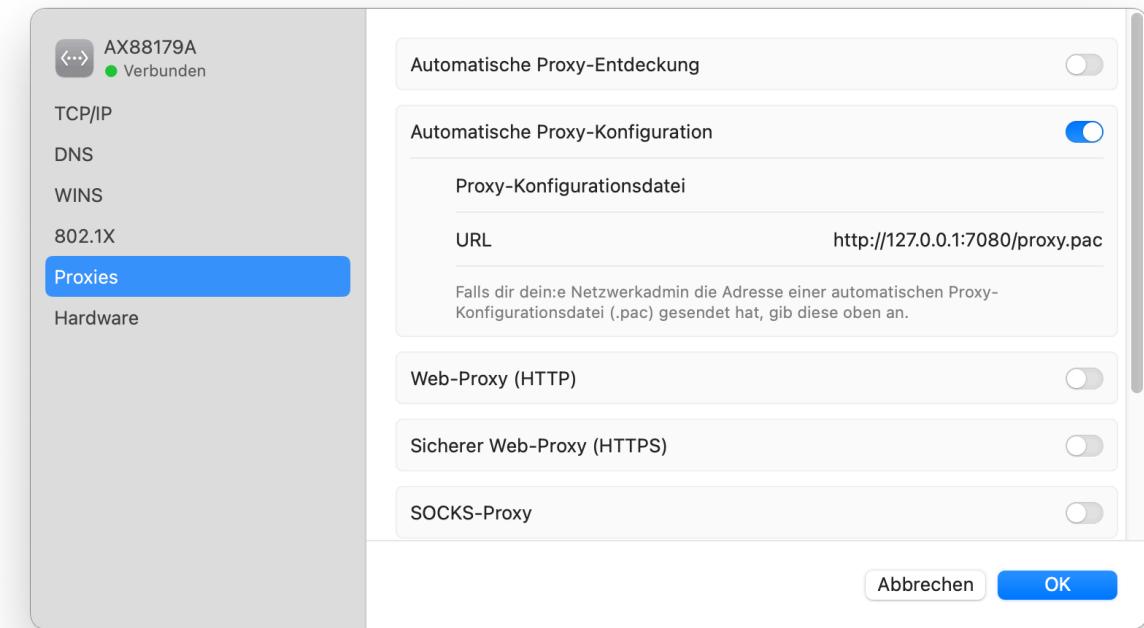
Stream server logs

```
symfony server:log
```

```
brew services symfony-cli start
```

```
symfony local:server:ca:install
```

→ setup proxy!



Configuration

Config file: `.symfony.local.yaml`

```
proxy:  
    domains:  
        - shop  
http:  
    daemon: true  
    use_gzip: true  
    port: 8078
```

To set up a different proxy TLD than `.wip`

```
symfony local:proxy:tld <tld>
```

Custom PHP ini settings

Create a `.php.ini`-file in your project and set options

```
upload_max_filesize = 256M  
memory_limit = 2G
```

Different PHP version

Create a `.php-version` file with the required PHP version

Running background tasks

```
...  
workers:  
    messenger_consume_async: ~
```

Just runs the `messenger:consume` command in the background
(restarting when failed)

Additional features

- Wildcard-Domains supported

```
proxy:  
  domains:  
    - *.shop
```

Configures `*.shop.wip` to be reachable - useful for multiple sales-channels

Debugging

- running xDebug

```
.php.ini
```

```
xdebug.mode = debug
```

- Docker env variables

```
symfony var:export --debug
```

Tips and tricks

- fixating `docker-compose` services to local ports using a `compose.override.yaml`
- using command line HTTP tools with
`https_proxy=$(symfony proxy:url) curl abc.wip/`

Sources

- <https://pixabay.com/photos/rocket-launch-rocket-space-shuttle-67649/>