Session

Sessions are handled by the Symfony2 framework, specifically API and underlying session handlers provided by HTTP Foundation component. This is further enhanced in eZ Platform with support for siteaccess-aware session cookie configuration.

Use of Memcached (or experimentally using PDO) as session handler is a requirement in Cluster setup, for details see below. For an overview of clustering feature see Clustering.

Session handlers

In Symfony, a session handler is configured using `framework.session.handler_id.` Symfony can be configured to use custom handlers, or just fallback to what is configured in PHP by setting it to null (~).

Default configuration

eZ Platform uses the same default configuration as recent versions of Symfony standard distribution. This makes sure you can configure sessions purely in PHP by default, and allows Debian/Ubuntu session file cleanup cronjob to work as intended.

Default config.yml session configuration

```
framework:
    session:
        # handler_id set to null will use default session handler from php.ini
        handler_id:  ~
```

Recommendations for production setup

Single server setup

For single server, default handler should be preferred.

Cluster setup

For Cluster setup we need to configure Sessions to use a backend that is shared between web servers and supports locking. Only options out of the box supporting this in Symfony are the native PHP memcached session save handler provided by the php-memcached extension, and Symfony session handler for PDO (database).

Storing sessions in Memcached using php-memcached

For setting up eZ Platform using this memcached you'll need to configure the session save handler settings in php.ini as documented here, optionally tweak php-memcached session settings.

Alternative storing sessions in database using PDO

While not currently our recommendation from performance perspective, for setups where Database is preferred for storing Sessions, you may use Symfony's PdoSessionHandler.
Below is an configuration example for eZ Platform, but please refer to documented in Symfony Cookbook documentation for full documentation.

```
framework:
    session:
        # ...
        handler_id: session.handler.pdo

parameters:
    pdo.db_options:
        db_table:    session
        db_id_col:   session_id
        db_data_col: session_value
        db_time_col: session_time

services:
    pdo:
        class: PDO
        arguments:
            dsn:      "mysql:dbname=<mysql_database>"
            user:     <mysql_user>
            password: <mysql_password>

    session.handler.pdo:
        class:
Symfony\Component\HttpFoundation\Session\Storage\Handler\PdoSessionHandler
        arguments: ["@pdo", "%pdo.db_options%"]
```

Further Symfony references

1. Cookbook Session recipes (symfony.com)
2. HTTP Foundation Component documentation (symfony.com)
3. Source code of NativeFileSessionHandler (github.com),
4. Cookbook Configuration recipe for setting-up PdoSessionHandler (symfony.com), aka `session.handler.pdo` service