Self-Signed Certificate

SelfSignedCertificate Class

Anonymous · 09/09/2026

SelfSignedCertificate Class

The `SelfSignedCertificate` class provides functionality to generate self-signed certificates for testing and development purposes. It allows customization of certificate properties such as subject, issuer, validity period, and password protection.

Overview

This class creates a digital certificate that includes:

1. A Certificate Authority (CA) certificate.
2. A user certificate signed by the CA.
3. Both packaged together in PKCS#12 format.

The generated certificate is suitable for testing electronic signature functionality without requiring a certificate from a commercial Certificate Authority.

Basic Usage

```
use Derafu\Certificate\SelfSignedCertificate;
use Derafu\Certificate\Service\CertificateLoader;

// Create with default settings.
$selfSigned = new SelfSignedCertificate();
$certificateArray = $selfSigned->toArray();

// Load as a Certificate object.
$loader = new CertificateLoader();
$certificate = $loader->loadFromArray($certificateArray);
```

Configuration Methods

Setting the Subject

The subject represents the owner of the certificate:

```
$selfSigned->setSubject(
    C: 'US',                              // Country code (2 letters).
    ST: 'California',                     // State/Province.
    L: 'San Francisco',                   // Locality/City.
    O: 'Example Corporation',             // Organization.
    OU: 'IT Department',                  // Organizational Unit.
    CN: 'John Doe',                       // Common Name (full name).
    emailAddress: 'john.doe@example.com', // Email address.
    serialNumber: '12345678-9',           // ID/Serial number (tax ID, etc.).
    title: 'System Administrator'         // Title.
);
```

Only `CN`, `emailAddress`, and `serialNumber` are strictly required. The method will throw a `CertificateException` if any of these are empty.

Setting the Issuer

The issuer represents the Certificate Authority that issues the certificate:

```
$selfSigned->setIssuer(
    C: 'US',                          // Country code
    ST: 'Washington',                 // State/Province
    L: 'Seattle',                     // Locality/City
    O: 'Example CA',                  // Organization
    OU: 'Certificate Authority',      // Organizational Unit
    CN: 'Example Root CA',            // CA Name
    emailAddress: 'ca@example.com',   // CA Email
    serialNumber: '87654321-0'        // CA ID
);
```

Setting the Validity Period

```
// Set validity to 730 days (2 years).
$selfSigned->setValidity(730);
```

The default validity period is 365 days (1 year).

Setting the Password

```
// Set password for the private key.
$selfSigned->setPassword('secure_password');
```

The default password is `'i_love_derafu'`.

Generating the Certificate

Getting the Certificate Array

```
// Get the certificate as an array with 'cert' and 'pkey' elements.
$certificateArray = $selfSigned->toArray();

// The array contains:
// [
//     'cert' => '...public key certificate...',
//     'pkey' => '...private key...'
// ]
```

Getting the Raw PKCS#12 Data

```
// Get the toPkcs12() method result (PKCS#12 binary data).
$pkcs12Data = $selfSigned->toPkcs12();

// Save to a file.
file_put_contents('certificate.p12', $pkcs12Data);
```

Certificate Structure

The generated certificate consists of:

1. An issuer (CA) certificate that is self-signed.
2. A subject certificate that is signed by the issuer.
3. Both certificates' private keys.

The PKCS#12 container includes both certificates and is password-protected using the specified password.

Default Values

If you create a `SelfSignedCertificate` without any customization, it will use the following default values:

Default Subject

- Country: `CL` (Chile).
- State: `Colchagua`.
- Locality: `Santa Cruz`.
- Organization: `Intergalactic Robots Organization`.

- Organizational Unit: `Technology`.
- Common Name: `Daniel Bot`.
- Email: `daniel.bot@example.com`.
- Serial Number: `11222333-9`.
- Title: `Bot`.

Default Issuer

- Country: `CL` (Chile).
- State: `Colchagua`.
- Locality: `Santa Cruz`.
- Organization: `Derafu`.
- Organizational Unit: `Technology`.
- Common Name: `Derafu Test Certificate Authority`.
- Email: `fakes-certificates@derafu.org`.
- Serial Number: `76192083-9`.

Other Defaults

- Validity: 365 days.
- Password: `i_love_derafu`.

Implementation Details

The `SelfSignedCertificate` class uses the PHP OpenSSL extension to:

1. Generate a key pair for the issuer.
2. Create a Certificate Signing Request (CSR) for the issuer.
3. Self-sign the issuer's CSR to create the issuer certificate.
4. Generate a key pair for the subject.
5. Create a CSR for the subject.
6. Sign the subject's CSR with the issuer's certificate.
7. Package everything into a PKCS#12 container.

The process mimics a real CA-issued certificate but is entirely self-contained and suitable for testing purposes.

Last updated on 09/09/2026