

Docs » Data Handling Category » Certificate Project » Certificate Class

# Certificate Class

Certificate Class

Anonymous · 09/09/2026

---

## Certificate Class

The `Certificate` class is the core component of the Derafu Certificate library, representing a digital certificate with its public and private keys and providing methods to access certificate details and properties.

### Overview

---

The `Certificate` class implements the `CertificateInterface` and provides functionality to:

- Access public and private keys.
- Extract certificate metadata (ID, name, email, issuer, etc.).
- Check certificate validity.
- Get certificate validity dates.
- Retrieve cryptographic components (modulus, exponent).
- Generate PKCS#12 data.

### Usage

---

#### Creating a Certificate

To create a `Certificate` instance, you need a public key (certificate) and a private key:

```
use Derafu\Certificate\Certificate;  
  
$certificate = new Certificate($publicKey, $privateKey);
```

The constructor automatically normalizes the keys using `AsymmetricKeyHelper`.

#### Getting Keys

```
// Get both keys as an array.  
$keys = $certificate->getKeys();  
// Result: ['cert' => '...public key...', 'pkey' => '...private key...']
```

```
// Get both keys without headers/footers.
$cleanKeys = $certificate->getKeys(clean: true);

// Get just the public key.
$publicKey = $certificate->getPublicKey();
// Alternatively.
$publicKey = $certificate->getCertificate();

// Get just the private key.
$privateKey = $certificate->getPrivateKey();

// Get clean keys (without headers/footers).
$cleanPublicKey = $certificate->getPublicKey(clean: true);
$cleanPrivateKey = $certificate->getPrivateKey(clean: true);
```

## Certificate Metadata

```
// Get certificate ID (usually a document/tax number).
$id = $certificate->getId(); // e.g., "12345678-9"
$id = $certificate->getId(forceUpper:false); // with original case, e.g., "12345678-k"

// Get certificate owner's name.
$name = $certificate->getName();

// Get certificate owner's email.
$email = $certificate->getEmail();

// Get certificate issuer.
$issuer = $certificate->getIssuer();
```

## Certificate Validity

```
// Check if certificate is valid (not expired).
$isActive = $certificate->isActive();

// Check if certificate is valid at a specific date.
$isActiveOn = $certificate->isActive('2025-06-01');

// Get certificate validity period.
$validFrom = $certificate->getFrom(); // e.g., "2025-01-01T00:00:00"
$validTo = $certificate->getTo(); // e.g., "2026-01-01T00:00:00"

// Get total validity period in days.
$totalDays = $certificate->getTotalDays();

// Get days remaining until expiration.
$daysRemaining = $certificate->getExpirationDays();

// Get days remaining from a specific date.
```

```
$daysRemainingFrom = $certificate->getExpirationDays('2025-06-01');
```

## Cryptographic Components

```
// Get certificate raw data.
$data = $certificate->getData();

// Get private key details.
$privateKeyDetails = $certificate->getPrivateKeyDetails();

// Get modulus and exponent.
$modulus = $certificate->getModulus();
$exponent = $certificate->getExponent();

// With custom word wrapping.
$modulus = $certificate->getModulus(wordwrap: 75);
$exponent = $certificate->getExponent(wordwrap: 75);
```

## Exporting a Certificate

```
// Export certificate as PKCS#12 data.
$pkcs12 = $certificate->getPkcs12('password');

// Save to a file.
file_put_contents('certificate.p12', $pkcs12);
```

## Error Handling

The Certificate class throws CertificateException when it encounters problems such as:

- Unable to extract the certificate ID.
- Unable to find the certificate name.
- Unable to find the certificate email.
- Unable to access the modulus or exponent.

Example:

```
use Derafu\Certificate\Exception\CertificateException;

try {
    $id = $certificate->getId();
    $name = $certificate->getName();
    $email = $certificate->getEmail();
} catch (CertificateException $e) {
    echo "Certificate error: " . $e->getMessage();
}
```

## Implementation Details

---

### ID Extraction

The `getId()` method attempts to extract the certificate ID through multiple methods:

1. First, it checks the `serialNumber` field in the subject.
2. If not found, it looks for the ID in the Subject Alternative Name (SAN) extension.
3. If still not found, it throws a `CertificateException`.

### Certificate Validation

The `isActive()` method checks if the certificate is valid at the current date (or a specified date) by comparing it against the certificate's validity period.

### Cryptographic Components

The `getModulus()` and `getExponent()` methods extract these values from the private key details and return them as base64-encoded strings, which can be used for various cryptographic operations.

---

Last updated on 09/09/2026