

Docs » Data Handling Category » Certificate Project » Certificate Validator

# Certificate Validator

What the CertificateValidator checks, how to make it check the format of the ID with a regular expression and how to write your own validator

Anonymous · 17/10/2026

---

## CertificateValidator Class

`CertificateValidator::validate()` checks that a certificate can be used. It returns nothing if it can, and throws a `CertificateException` if it can not.

```
use Derafu\Certificate\Service\CertificateValidator;

$validator = new CertificateValidator();
$validator->validate($certificate);
```

## What It Checks

---

In this order:

1. The certificate has an ID (`getId()`; if it has none, the exception of `getId()` is the one that is thrown).
2. The ID matches the format that you gave to the validator, if you gave one.
3. The certificate is not expired.

Without a format, any ID is valid: the validator does not know what an ID should look like, because that depends on who issues the certificates.

## The Format of the ID

---

The format is a regular expression, with its delimiters, that the validator receives when it is created:

```
// The ID is digits, a dash and a check character that is a digit or an uppercase "K".
$validator = new CertificateValidator('/^\d+-[\dK]$/' );

$validator->validate($certificate); // "11222333-9" or "10-K": valid.
                                   // "10-k", "123" or "abc-9": it throws.
```

When the ID does not match, the message says the ID and the format:

The ID 10-k of the certificate does not match the format `/^\d+-[\dK]$/`.

The validator checks the ID as it is in the certificate (`getId(forceUpper: false)`), so a format that requires uppercase works.

If the regular expression is not valid, the validator throws a `CertificateException` when it is created, not when it validates.

## Using It with the Service

---

`CertificateService` receives the validator, so the format is set where the validator is created:

```
$service = new CertificateService(  
    $faker,  
    $loader,  
    new CertificateValidator('/^\d+-[\dK]$/')  
);  
  
$service->validate($certificate);
```

In an application with the Symfony container, the validator is registered in `resources/config/certificate-services.yaml` of the library without a format. To use one, declare the argument of the service in your own configuration.

## Your Own Validator

---

For other validations, implement `CertificateValidatorInterface` (see [Certificate Service](#)).

---

Last updated on 17/10/2026