

# **Laravel - Encryption**

**Topics:** Laravel

Written on December 25, 2023

Laravel provides a simple and convenient way to encrypt and decrypt data using the Illuminate Encryption package. This package utilizes OpenSSL for encryption and provides a clean and expressive syntax.

# 1. Encrypting Data:

To encrypt data in Laravel, you can use the encrypt function or the encryptString method provided by the Illuminate\Encryption\Encrypter class.

#### Using encrypt function:

```
$encrypted = encrypt('Sensitive data');
```

#### Using encryptString method:

```
use Illuminate\Encryption\Encrypter;
$encrypter = new Encrypter($key, $cipher);
```

\$encrypted = \$encrypter->encryptString('Sensitive data');

#### 2. Decrypting Data:

To decrypt data, you can use the decrypt function or the decryptString method.

#### Using decrypt function:

```
$decrypted = decrypt($encrypted);
```

#### Using decryptString method:

```
use Illuminate\Encryption\Encrypter;

$encrypter = new Encrypter($key, $cipher);
$decrypted = $encrypter->decryptString($encrypted);
```

### 3. Using the Crypt Facade:

Laravel provides a Crypt facade for convenient access to the encryption methods.

```
use Illuminate\Support\Facades\Crypt;

$encrypted = Crypt::encrypt('Sensitive data');
$decrypted = Crypt::decrypt($encrypted);
```

### 4. Custom Encryption Configuration:

Laravel's encryption settings can be configured in the config/app.php file. The key and cipher configuration options are particularly important for encryption.

```
'key' => env('APP_KEY'),

'cipher' => 'AES-256-CBC',
```

### 5. Generating Encryption Keys:

Laravel requires a secret key for encryption. You can generate a new key using the php artisan key: generate command. This command will set the APP KEY in your .env file.

```
php artisan key:generate
```

## 6. Encrypting and Decrypting Cookies:

Laravel provides methods for encrypting and decrypting cookies.

```
use Illuminate\Support\Facades\Cookie;

// Encrypt a cookie
Cookie::queue('cookie_name', encrypt('cookie_value'), $minutes);

// Decrypt a cookie
$decryptedValue = decrypt(Cookie::get('cookie name'));
```

# 7. Encrypting Model Attributes:

You can use the \$casts property in your Eloquent models to automatically encrypt and decrypt model attributes.

```
class User extends Model
{
    protected $casts = [
         'secret_attribute' => 'encrypt',
    ];
}
```

# 8. Encrypting Configuration Values:

You can encrypt sensitive configuration values using the config:cache Artisan command.

php artisan config:cache

# 9. Choosing a Cipher:

Laravel supports various encryption ciphers. The default is AES-256-CBC, but you can choose a different one in your configuration.

# 10. Encryption in Jobs and Queues:

When working with jobs and queues, encrypted data should be serialized properly. Laravel automatically handles serialization and describilization of encrypted data within jobs.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by <u>Aryatechno</u>