

dnsimple



## PHP API Cheatsheet

---

With the DNSimple package you can easily interact our powerful API to administer domain names, configure DNS records, provision and install SSL certificates, and more.

# Getting Started

---

1. Install the package

## Composer

Add this dependency to your project's `composer.json` :

```
{
  "require": {
    "dnsimple/dnsimple": "x.y.z"
  }
}
```

2. Authenticate

Obtain your API access token: <https://support.dnsimple.com/articles/api-access-token/>

```
$client = new Client("your-access-token", [
    "base_uri" => "https://api.sandbox.dnsimple.com"
]);
```

3. Check Authorization

If you want to know which account is associated with the current access token, you can use `#identity` . The account ID is required for the majority of API operations.

```
$whoami = $client->identity->whoami()->getData();

print_r($whoami->id);
=> 1234 (your account ID)
```

# Managing Domains

---

## Check Domain Availability

Check if a domain is available for registration.

```
$domain = $client->registrar->checkDomain($accountId, "foo.com")->
    getData();

print_r($domain->available);
=> true
```

## Register A Domain

1. To register a domain, you need to specify a registrantId. This can be fetched via the Contacts API.

```
$contacts = $client->contacts->listContacts($accountId)->getData();

print_r($contacts[0]->id);
=> 123
```

2. You can register the domain with this information.

```
$attributes = [
    "registrant_id" => 2,
    "whois_privacy" => false,
    "auto_renew" => false
];
$registration = $client->domains->registerDomain($accountId, $domain, $attr

print_r(
    "State: {$registration->state},".
    "AutoRenew: {$registration->autoRenew},".
    "WhoisPrivacy:{$registration->whoisPrivacy},".
    "Period:{$registration->period},".
    "RegistrantId: {$registration->registrantId}");
=> State: registered, AutoRenew: false, WhoisPrivacy: false,
    Period: 1, RegistrantId: 123
```

# DNS

---

## Create a DNS record

Create a DNS A record to map an IP address to a domain.

```
$attributes = [  
  "name" => "www",  
  "type" => "A",  
  "content" => "127.0.0.1"  
];  
$record = $client->zones->createRecord($accountId, $domain,  
  $attributes)->getData();  
  
print_r($record->id);  
=> 123
```

## Update a DNS record

Update a previously created DNS record.

```
$attributes = [  
  "ttl" => 60  
];  
$updated = $client->zones->updateRecord($accountId, $domain, 5,  
  $attributes);  
  
print_r($updated->ttl);  
=> 60
```

# SSL Certificates

---

## Order an SSL Certificate with Let's Encrypt

Creates the purchase order. Use the ID to issue the certificate.

```
$attributes = [];  
$certPurchase = $client->certificates->  
    purchaseLetsEncryptCertificate($accountId,  
        $domain, $attributes)->getData();  
  
print_r("Id: {$certPurchase->id}, CertificateId: {$certPurchase->certificateId}");  
=> Id: 123, CertificateId: 3323
```

## Issue an Let's Encrypt Certificate

Issues the pending order. This process is async. A successful response means that the response is queued.

```
$certificate = $client->certificates->issueLetsencryptCertificate(  
    $accountId, $domain, cert->id)->getData();  
  
print_r($certificate->state);  
=> "requesting"
```

# Install the certificate

Download the certificate.

```
$cert = $client->certificates->downloadCertificate($accountId, $domain,
    $certificate->id)->getData();

$pemFile = fopen("www_foo_com.pem", "w") or die("Unable to open file!");
fwrite($pemFile, $cert->server);

foreach($cert->chain as &$intermediateCert) {
    fwrite($pemFile, $intermediateCert);
}

fclose($pemFile);
```

Download the certificate's private key.

```
$privateKey = $client->certificates->getCertificatePrivateKey($accountId,
    $domain, $certificate->id)->getData();

$keyFile = fopen("www_foo_com.key", "w") or die("Unable to open file!");
fwrite($keyFile, $privateKey->privateKey);

fclose($keyFile);
```

**dnsimple**

**Get Help From Developers**

We provide worry-free DNS services to simplify your life.

Try us free for 30 days