

dnsimple



Java 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

Maven

Add this dependency to your project's POM:

```
<dependency>
  <groupId>com.dnsimple</groupId>
  <artifactId>dnsimple-java</artifactId>
  <version>X.X.X</version>
</dependency>
```

Gradle

Add this dependency to your `build.gradle` file:

```
compile 'com.dnsimple:dnsimple-java:X.X.X'
```

2. Authenticate

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

```
import com.dnsimple.Client;

public class MyApp {
    public static void main(String[] args) {
        Client client = new Client.Builder()
            .accessToken("your-access-token")
            .build();
    }
}
```

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.

```
WhoamiData identity = client.identity.Whoami().getData();  
  
System.out.println(identity.getAccount().getId());  
=> 1234 (your account ID)
```

Managing Domains

Check Domain Availability

Check if a domain is available for registration.

```
DomainCheck check = client.registrar.checkDomain(accountId, "foo.com")
    .getData();

System.out.println(check.isAvailable());
=> true
```

Register A Domain

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

```
List<Contact> contacts = client.contacts.listContacts(accountId).getData();

System.out.println(contacts.get(0).getId());
=> 123
```

2. You can register the domain with this information.

```
DomainRegistration registration = client.registrar.registerDomain(
    accountId, availableDomain.getDomainName(),
    RegistrationOptions.of(contact.getId()).getData());

System.out.println(String.format(
    "State: %s, AutoRenew: %s, WhoisPrivacy: %s, Period: %s, RegistrantId: %s",
    registration.getAutoRenew(), registration.getWhoisPrivacy(),
    registration.getPeriod(), registration.getRegistrantId()
));
=> 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.

```
ZoneRecordOptions zoneRecordOptions = ZoneRecordOptions.of(
    "www", "A", "127.0.0.1");

ZoneRecord record = client.zones.createZoneRecord(accountId, "foo.com",
    zoneRecordOptions).getData();

System.out.println(record.getId());
=> 123
```

Update a DNS record

Update a previously created DNS record.

```
ZoneRecordUpdateOptions zoneRecord = ZoneRecordUpdateOptions.ttl(60);

ZoneRecord updated = client.zones.updateZoneRecord(accountId, "foo.com",
    record.Id, zoneRecord).getData();

System.out.println(updated.getTtl());
=> 60
```

SSL Certificates

Order an SSL Certificate with Let's Encrypt

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

```
CertificatePurchaseOptions options = CertificatePurchaseOptions.of("www");

CertificatePurchase cert = client.certificates
    .purchaseLetsencryptCertificate(accountId, "foo.com", options)
    .getData();

System.out.println(String.format("Id: %s, CertificateId: %s",
    cert.getId(), cert.getCertificateId()));
=> 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 certificate = client.certificates.issueLetsencryptCertificate(
    accountId, "foo.com", cert.getId()).getData();

System.out.println(certificate.getState());
=> "requesting"
```

Install the certificate

Download the certificate.

```
CertificateBundle cert = client.certificates.downloadCertificate(
    accountId, "foo.com", certificate.getId()).getData();

FileWriter fileWriter = new FileWriter("www_foo_com.pem");
PrintWriter printWriter = new PrintWriter(fileWriter);

printWriter.print(cert.getServerCertificate());

for(String intermediateCert : cert.getIntermediateCertificates()) {
    printWriter.print(intermediateCert);
}

printWriter.close();
```

Download the certificate's private key.

```
CertificateBundle cert = client.certificates.downloadCertificate(
    accountId, "foo.com", certificate.getId()).getData();

FileWriter fileWriter = new FileWriter("www_foo_com.key");
PrintWriter printWriter = new PrintWriter(fileWriter);

printWriter.print(cert.getPrivateKey());
printWriter.close();
```

dnsimple

Get Help From Developers

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

Try us free for 30 days