dnsimple Node API v2 Cheatsheet

Registering and managing your domains has never been easier. With the DNSimple node package you can easily interact our powerful API to administer domain names, configure DNS records, provision and install SSL certificates, and more!

GETTING STARTED

⚠ **ES6 ahead**. The examples shown are in ECMAScript 6, but our API is compatible with ES5.

Install via NPM

```
$ npm install dnsimple
```

API access token

To use our node client, you need an access token. We provide both Account tokens (with access to a single account), and User tokens (with access granted to all accounts associated with a user). For the purpose of this demonstration, we'll use Account tokens. To get an account token, login into DNSimple, select the account, and click on *Access tokens*.

Authentication

Use your access token to instantiate a client.

```
var client = require("dnsimple")({
  accessToken: "abc123",
});
```

Check authorizations

The majority of operations require the account associated with the token

```
client.identity.whoami()
.then(response => (
  console.log(response.data)
));
// { account:
//   { id: 1010,
//     email: 'your@email.com',
//... }
```

DOMAINS

Check domain availability

Check if a domain is available for registration

```
client.registrar.checkDomain(
  accountId = 1010,
  domainName = "dnsimple.com")
.then(response => (
  console.log(response.data)
));
// { domain: 'dnsimple.com',
//  available: false,
//  premium: false }
```

Register a domain

1 In order to register a domain you need to specify a `registrant_id`. This can be fetched via the Contacts API

```
client.contacts.listContacts(
  accountId = 1010)
.then(response => (
  console.log(response.data)
));
// [ { id: 12345,
//    first_name: 'John',
//    last_name: 'Appleseed',
//    email: 'john@john.com', ... }
// ]
```

2 With this information you can register the domain

```
client.registrar.registerDomain(
  accountId = 1010,
  domainName = "example.com",
  attributes = {
    registrant_id: 12345,
    whois_privacy: true,
    auto_renew: true
  })
.then(response => (
  console.log(response.data);
));
// { id: 34567,
//  domain_id: 67890,
//  registrant_id: 12345,
//  period: 1,
//  state: 'new',
//  auto_renew: true,
//  whois_privacy: true, ... }
```

Learn more about DNSimple API v2 at **https://dnsimple.com/api**

DNS

Create

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

```
client.zones.createZoneRecord(
  accountId = 1010,
  domainId = "foo.com",
  attributes = {name: "", type: "A", content: "1.2.3.4"})
.then(response => console.log(response.data));
```

```
// { id: 12345,
//  zone_id: 'foo.com',
//  name: 'test',
//  content: '1.2.3.4',
//  ttl: 3600,
//  type: 'A', ...}
```

Update

Update a previously created DNS record

```
client.zones.updateZoneRecord(
  accountId = 1010,
  domainId = "foo.com",
  recordId = 12345,
  attributes = { ttl: 60 })
.then(response => console.log(response.data));
```

```
// { id: 12345,
//  zone_id: 'foo.com',
//  name: 'test',
//  content: '1.2.3.4',
//  ttl: 60,
//  type: 'A', ... }
```

SSL CERTIFICATES

New SSL Certificate from Let's Encrypt

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

```
client.certificates.purchaseLetsencryptCertificate(
  accountId = 1010,
  domainId = "foo.com")
.then(response => console.log(response.data));
```

```
// { id: 12345,
//  common_name: 'www',
// }
```

Issue SSL Certificate from Let's Encrypt

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

```
client.certificates.issueLetsencryptCertificate(
  accountId = 1010,
  domainId = "foo.com",
  certificatePurchaseId = 12345)
.then(response => console.log(response.data));
```

```
// { id: 12345,
//  common_name: 'www',
//  state: 'requesting',
// }
```

Install a certificate

1 Download the certificate

```
fs = require("fs");
client.certificates.downloadCertificate(
  accountId = 1010,
  domainId = "foo.com",
  certificateId = 12345)
.then(response => (
  fs.writeFile("foo_com.pem",`${response.data.server}\n${response.data.chain}`)
));
```

2 ... and its private key

```
client.certificates.getCertificatePrivateKey(
  accountId = 25063,
  domainId = "foo.com",
  certificateId = 27059)
.then(response => (
  fs.writeFile("foo_com.key", response.data.private_key)
));
```

Learn more about DNSimple API v2 at **https://dnsimple.com/api**