

dnsimple

# Getting started with the DNSimple API in Elixir



# Getting started with the DNSimple API in Elixir

You can check out the DNSimple Elixir API client [on GitHub](#). It's fully featured and maintained by several members of the DNSimple team. In fact, we're using the client extensively in several internal projects. This is the library we'll be using in this tutorial. Let's get started.

## Getting ready

There are some initial steps you have to take before you can start making calls to the DNSimple API: subscribing to DNSimple and installing Elixir.

## Subscribing

To follow this tutorial, you'll need an active subscription with DNSimple. You can see the offered plans and the included features for each of them on our [pricing page](#). The subscription fee covers our various domain management automation services, which include API access.

However, you can **follow this tutorial without spending a dime** - we have a sandbox environment ready for you. Just go to [sandbox.dnsimple.com](#) and subscribe to any of our plans using the credit card number **1**. There, you'll be able to use our API as much as you want.

## Installing the client

With your account ready, you can start with the fun part: writing code!

In this tutorial we'll assume you have Elixir installed on your machine. If you don't, follow [the official documentation](#) to get ready.

The first step is to create a new Elixir app:

```
~/dev > mix new dnsimple-api-test
...[Creating files]
Your Mix project was created successfully.
```

Now you have to edit our `mix.exs` file, and add the DNSimple API client as a dependency. You'll have to look for the `deps` function, and update it so it looks like this:

```
defp deps do
  [{:dnsimple, "~> 1.0.0"}]
end
```

Install the newly added dependency:

```
~/dev/dnsimple-api-test > mix deps.get
Running dependency resolution
...
```

Finally, make sure to start the DNSimple OTP app when your application runs. Find the `application` function and add `Dnsimple` to the list:

```
def application do
  [applications: [:logger, :dnsimple]]
end
```

## Authenticating

With the Elixir application ready, you now need to learn how to make API calls. Looking at the [library README](#), you'll see that you need to provide a `Dnsimple.Client` struct for each call:

```
{:ok, response} = Dnsimple.Identity.whoami(client)
```

This struct contains the required authentication data to perform the API call and the URL of the environment against which you're performing the API call (`https://api.dnsimple.com` being the default). You can provide a custom `base_url` so your client [targets the sandbox environment](#):

```
client = %Dnsimple.Client{access_token: "TOKEN",
  base_url: "https://api.sandbox.dnsimple.com/"}
```

## Obtaining an access token

The next question is: what is an access token and how do you obtain it?

The access token is a particular key that allows you to authenticate an API request. This is the preferred authentication mechanism over using your username and password, as you can easily delete and regenerate tokens without affecting your primary credentials.

The [how to obtain an access token article](#) provides step by step instructions for getting started.

With your token ready, you can make your first API call from IEx:

```
~/dev/dnsimple-api-test > iex -S mix
iex(1)> client = %Dnsimple.Client{access_token: "TOKEN", base_url:
"https://api.sandbox.dnsimple.com/"}
iex(2)> {:ok, response} = Dnsimple.Identity.whoami(client)
[debug] [dnsimple] GET https://api.sandbox.dnsimple.com/v2/whoami
{:ok,
 %Dnsimple.Response{data: %Dnsimple.Whoami{account:
%Dnsimple.Account{email: "javier@dnsimple.com",
 id: 63, plan_identfier: "dnsimple-professional"}, user: nil},
 http_response: %HTTPoison.Response{body:
"{\"data\":{\"user\":null,\"account\":{\"id\":63,\"email\":\"javier
@dnsimple.com\",\"plan_identfier\":\"dnsimple-
professional\",\"created_at\":\"2014-07-
02T14:15:56Z\",\"updated_at\":\"2016-12-14T08:44:06Z\"}}}\",
 headers: [{"Server", "nginx"}, {"Date", "Fri, 30 Dec 2016
09:16:03 GMT"},
 {"Content-Type", "application/json; charset=utf-8"}]}
iex(3)> response.data
%Dnsimple.Whoami{account: %Dnsimple.Account{email:
"javier@dnsimple.com",id: 63, plan_identfier: "dnsimple-
professional"}, user: nil}
```

As you can see, we pattern matched the value returned by the call to `Dnsimple.Identity.whoami` to capture the response. Then, we extracted the data of the response by checking the `data` key of the returned struct. This is a pattern that you'll repeat all the time when using the DNSimple Elixir API client.

## Finding out your account ID

There is one important bit of information in that response: your `account_id`. In my case my `account_id` on the sandbox environment is `63`.

If you look through the different [endpoints of the API](#), you'll see that you have to provide this value in most of them. There are several reasons for this which have [been documented already](#), so we won't go into those right now.

If your app is always going to use the same account, you can hard-code this value (or put it in a configuration setting). However, when your application deals with different users, you'll have to provide a different `account_id` for each of them.

## Wrapping up

That's it! Here's what you've achieved so far:

- You have a DNSimple account ready on the sandbox environment.
- You have installed the DNSimple Elixir API client.
- You have created a new mix app and configured DNSimple as a dependency.
- You have generated an access\_token that you can use to make API calls.
- You have made your first call to the DNSimple API.

Now you're ready to grow your app as much as you want.

### Have Questions? Get Help From Developers

Everyone at DNSimple enjoys writing tutorials.  
We love answering your emails, too.

Try us free for 30 days

dnsimple