
SimpleIdServer Documentation

SimpleIdServer

Nov 14, 2019

Getting started

1	How to setup an DNS server ?	3
2	DnsServerOptions	5

DnsServer is an open source library enabling any DOTNET CORE applications to host a DNS server. The project implements the RFC 1034 and 1035.

Github: <https://github.com/simpleidserver/DnsServer>

Nuget feed: <https://www.myget.org/F/advance-ict/api/v3/index.json>

How to setup an DNS server ?

A DNS server can be hosted in any DOTNET CORE project like ASP.NET CORE or Console application. Follow the steps below to deploy a DNS server into a console application :

- 1) Create an empty Console application.
- 2) Install the Nuget package **DnsServer**.
- 3) In the Program.cs file, insert the following code into the **Main** method.
- 4) Configure the network interface to use the local DNS server.
- 5) Open a command prompt and execute **ping www.example.com**.

```
var dnsServer = new DnsServerHostBuilder(o =>
{
    o.ExcludeForwardRequests.Add(new Regex("^.*example\\.com$"));
    o.ExcludeForwardRequests.Add(new Regex("^.*in-addr\\.arpa$"));
})
.AddDNSZones(new List<DNSZone>
{
    new DNSZone("example.com")
    {
        ResourceRecords = new List<ResourceRecord>
        {
            new AResourceRecord(3600)
            {
                Address = "127.0.0.1"
            },
            new AResourceRecord(3600, "www")
            {
                Address = "127.0.0.1"
            },
            new AResourceRecord(3600, "ns1")
            {
                Address = "127.0.0.1"
            },
        }
    }
})
```

(continues on next page)

(continued from previous page)

```
        new SOAResourceRecord(3600)
        {
            MName = "ns1.example.com",
            RName = "admin.example.com",
            Serial = 5,
            Refresh = 604800,
            Expire = 2419200,
            Minimum = 604800,
            Retry = 86400
        },
        new NSResourceRecord(3600)
        {
            NSDName = "ns1.example.com"
        }
    },
    new DNSZone("1.0.0.127.in-addr.arpa")
    {
        ResourceRecords = new List<ResourceRecord>
        {
            new PTRResourceRecord(100)
            {
                PTRDNAME = "localhost"
            }
        }
    }
})
.AddDNSRootServers(DnsServerConstants.DefaultRootServers)
.Build();
dnsServer.Run();
```

The **DnsServerHostBuilder** class accepts in its constructor a callback which can be used by developers to change the options of the DNS server. There are several operations exposed by the **DnsServerHostBuilder** class :

- **AddDNSZones** : Configure DNS zones.
- **AddDNSRootServers** : Configure the DNS root servers.

A WPF client also exists [here](#).

CHAPTER 2

DnsServerOptions

ExcludeForwardRequests List of Regular Expressions used to filter UDP requests and redirect them to the Authoritative server.

TimeoutInMilliseconds Configure the timeout in milliseconds, the default value is **400**.

DefaultCpu Central Processing Unit (CPU) of the machine, the default value is **intel**.

DefaultOS Operating System (OS) of the machine, the default value is **win**.

DefaultTtl Default time life of a DNS resource record.