

DNS

Everything You Need to Know About DNS

If you are reading this article, chances are you have used the Domain Name System (DNS) without even knowing it. DNS is a fundamental part of the internet, is a fundamental part of the internet that allows us to access websites and online services by using human-friendly names instead of numerical addresses. But how does it work exactly? In this article, I will explain the basics of DNS and how it help us navigate the web

What is DNS?

DNS is like the internet's phone book, it translates domain names (like `www.example.com`) into IP addresses (like `192.0.2.1`), which are the actual locations of web servers. This allows users to access websites using easy-to-remember names instead of numeric IP addresses.

How DNS Works

DNS doesn't know all domain-to-ip mappings. Instead, it uses a network of DNS resolvers to find the correct IP address.

Step-by-step process:

- 1 Local Cache: Your computer checks its own DNS cache.
- 2 DNS Resolvers: If not found, it queries a DNS resolver (usually provided by your ISP).
- 3 Recursive Search: The resolver checks its cache or queries other DNS servers:

- Root DNS Servers

- Top-Level Domain (TLD) Servers (e.g., `.com`, `.org`)

- Authoritative DNS Servers (which hold the actual domain records)

Once the IP address is found, it's returned to your computer so the browser can load the website.

Step 1: Local Caches

Before querying external servers, your computer checks its local caches to find the IP address for a domain name. If found, it uses that IP directly, skipping further DNS steps.

Types of Local Caches:

Browser Cache: Store IP of previously visited websites.

DNS Cache: Temporarily store DNS records based on their TTL (Time to live)

Hosts File: A manual list of domain-to-ip mappings set by the user.

Step 2: Recursive DNS Server

Your computer or router is set to use a recursive DNS server (usually from your ISP). When the local cache doesn't have the IP address:

The recursive server checks its own cache.

If it finds the IP, it returns it to your device.

If not, it forwards the query to the Root DNS Servers to continue the search.

Step 3: Root DNS Servers

Root DNS servers are at the top of the DNS hierarchy. They don't store IP addresses for specific websites but instead direct queries to the appropriate Top-Level Domain (TLD) DNS servers (like .com, .org, .net).

For example, if you're looking for www.example.com, the root server will point you to DNS servers responsible for .com domains.

```
dig +short NS com
dig +short NS org
dig +short NS ai
dig +short NS fyi
dig +short NS io
```

Step 4: TLD DNS Servers

After the root DNS servers direct the query, it reaches the Top-Level Domain (TLD) DNS servers, which manage domains ending in extensions like .com, .org, .ai, etc.

What They Do:

TLD servers don't store the final IP address of websites.

Instead, they **delegate** the query to the **authoritative DNS servers** responsible for the specific second-level domain (e.g., example.com).

These authoritative servers hold the actual DNS records (like A, AAAA, MX, CNAME) that map domain names to IP addresses.

You can see the nameservers for the second-level domain by running:

```
dig +short NS cs.fyi
dig +short NS github.com
dig +short NS medium.com
```

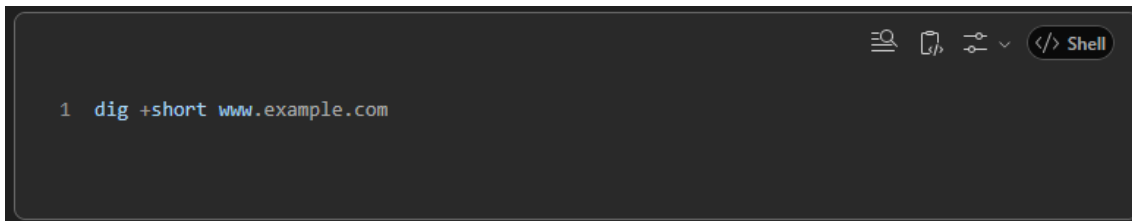
Step 5: Authoritative DNS Servers

This is the place where the actual DNS records are stored. At this stage, the Authoritative DNS server will be asked for the A record of the domain name. The A record is the DNS record that maps a domain name to an IP address. The Authoritative DNS server will then send the IP address back to the recursive DNS server. The recursive DNS server will then send the IP address back to your computer.

How does DNS work in practice?

To see DNS in action, you can use the dig command, a tool available on Linux and macOS (and installable on Windows via Chocolatey).

Basic Usage:

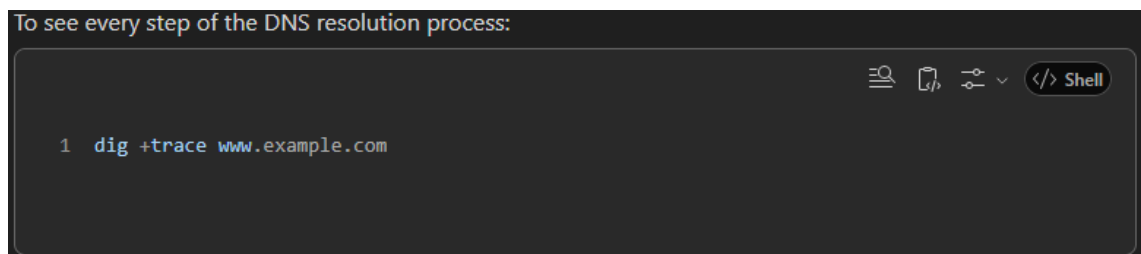
A terminal window with a dark background. The command '1 dig +short www.example.com' is entered at the prompt. The terminal has icons for search, copy, paste, and a dropdown menu in the top right corner, along with a 'Shell' button.

The +short option shows only the final IP address (e.g., 93.184.216.34).

What Happens Behind the Scenes:

1. **Local Cache Check:** dig first checks your system's DNS cache.
2. **Recursive DNS Query:** If not cached, it sends the query to a **recursive DNS server**.
3. **Resolver Chain:** The recursive server checks its own cache, and if needed, queries:
 - **Root DNS Servers**
 - **TLD DNS Servers** (e.g., .com)
 - **Authoritative DNS Servers** (which hold the actual A record)
4. **Final Response:** Once the IP is found, it's returned to dig, and then to your computer.

Trace the Full Path:

A terminal window with a dark background. The command '1 dig +trace www.example.com' is entered at the prompt. Above the command, the text 'To see every step of the DNS resolution process:' is displayed. The terminal has icons for search, copy, paste, and a dropdown menu in the top right corner, along with a 'Shell' button.

This command shows each DNS server involved, from root to authoritative, helping you understand how the domain name is resolved step-by-step.

Debugging DNS Issues

If you are having trouble accessing a website, you can use the 'dig' command to debug the issue. Let's see some practical examples of how we can use the 'dig' command.

Checking DNS resolution for a Domain

You can use 'dig' to check whether a domain name can be resolved to an IP address. Here's an example.

```
dig example.com +short
```

This command will return the IP address(es) associated with the domain name **google.com**. The **+short** option is used to display only the IP addresses, without any additional information.

Retrieving DNS records for a Domain

The dig command is a powerful tool for querying DNS servers and retrieving specific types of DNS records for a domain.

Common Use Cases:

A Record (IPv4 Address):

```
1 dig example.com A
```

Returns all **A records**, which map the domain to its IPv4 address(es).

MX Record (Mail Exchange):

MX Record (Mail Exchange):

```
1 dig example.com MX
```

Returns all **MX records**, which specify the mail servers responsible for handling email for the domain.

Other Useful Record Types:

- **AAAA** – IPv6 address
- **CNAME** – Canonical name (alias)
- **NS** – Nameservers for the domain
- **TXT** – Text records (often used for verification or SPF/DKIM settings)

Why It's Useful:

- Helps troubleshoot DNS issues
- Verifies domain configurations
- Checks email routing setups
- Confirms propagation of DNS changes

Checking DNS propagation

You can use dig to check whether DNS records have been propagated to all DNS servers. For example, to check whether a domains 'MX' records have been propagated, you can use the following command:

```
dig example.com MX +trace
```

The '+trace' option is used to show the path of the DNS resolution process, starting from the root DNS servers. This will help you identify whether there are any DNS servers that have not yet received the updated DNS records.

Checking DNSSEC validation

You can used dig to check whether 'DNSSEC' validation is working properly for a domain name. For example, to check whether a domain's 'DNSSEC' records are valid, you can use the following command:

```
dig example.com +dnssec
```

This will show the DNSSEC-related records for the domain, and whether they are valid.

DNSSEC is a security extension to the Domain Name System (DNS) that authenticates the source of the DNS data and the integrity of the data. It also provides a mechanism to prevent DNS data from being tampered with during transit.

You can read more about DNSSEC [here](#).

Querying a specific DNS server

You can use 'dig' to query a specific DNS server for DNS records.

For example, to query the Google Public DNS server ('8.8.8.8') for 'A' records for a domain, you can use the following command:

```
dig example.com A @8.8.8.8
```

This will send the DNS query to the specific DNS server (in this case, 8.8.8.8), instead of using the default DNS server configuration on the local machine.

Common DNS errors

There are a few common DNS errors that you you might encounter.

Here are some of them:

DNS_PROBE_FINISHED_NXDOMAIN

Meaning: The domain name does not exist.

Causes:

Typo in the domain name.

The domain has expired or is not registered.

Misconfigured DNS settings.

DNS_PROBE_FINISHED_NO_INTERNET

Meaning: No internet connection is available to reach the DNS server.

Causes:

Network connectivity issues.

Router or modem problems.

Firewall or antivirus blocking access.

DNS_PROBE_FINISHED_BAD_CONFIG

Meaning: DNS configuration is incorrect or corrupted.

Causes:

Wrong DNS server settings.

Issues with the local DNS cache.

VPN or proxy interference.

How to Flush DNS Cache

There might be times when you want to flush the DNS cache on your local machine. You can use the 'ipconfig' command to do this on windows, and the 'dscacheutil' command to do this on MacOS.

Here is the command to flush the DNS cache on Windows:

```
ipconfig /flushdns
```

Conclusion

In this article, we learned about DNS and how it works. We also learned how to use the **dig** command to query DNS servers for DNS records. I hope you found this article useful; feel free to share it with your friends and colleagues.

<https://cs.fyi/guide/everything-you-need-to-know-about-dns>