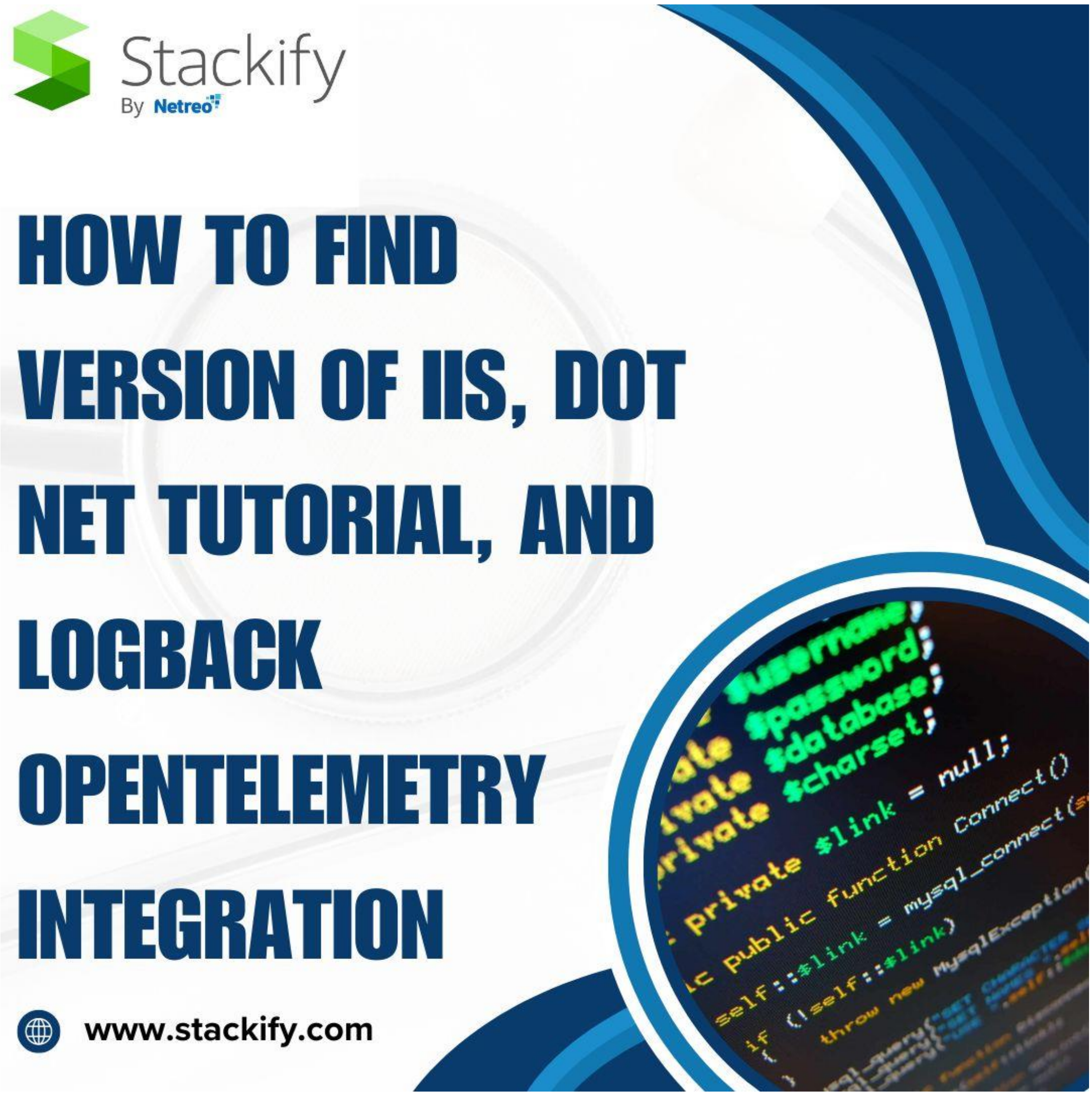
Stackify
By Netreo

How to Find Version of IIS, Dot Net Tutorial, and Logback OpenTelemetry Integration


Stackify
By Netreo
HOW TO FIND VERSION OF IIS, DOT NET TUTORIAL, AND LOGBACK OPENTELEMETRY INTEGRATION
www.stackify.com

Web development and application monitoring have evolved dramatically, requiring developers to stay up-to-date with tools and technologies such as Internet Information Services (IIS), .NET, and logging frameworks like Logback combined with modern observability tools such as OpenTelemetry. In this blog, we will walk through three core topics: how to find the version of IIS, a beginner-friendly .NET tutorial, and integrating Logback with OpenTelemetry.

1. How to Find Version of IIS

Internet Information Services (IIS) is Microsoft's web server that plays a vital role in hosting websites and web applications. Knowing the version of IIS running on your system can be crucial for troubleshooting, updating, or ensuring compatibility with web applications.

Methods to Find IIS Version

There are several ways to find the version of IIS installed on your system. Here are the most straightforward methods:

a. Using the IIS Manager

1. **Step 1: Open IIS Manager**
    - Click on the Start menu and search for "IIS Manager."
    - Alternatively, you can press Windows + R, type inetmgr, and hit Enter.
2. **Step 2: Locate the IIS Version**
    - Once the IIS Manager is open, click on the Help menu.
    - Select About Internet Information Services from the dropdown.
    - A pop-up window will appear showing the version number.

b. Using Command Prompt or PowerShell

1. **Step 1: Open Command Prompt/PowerShell**
    - Press Windows + R, type cmd (for Command Prompt) or powershell (for PowerShell), and hit Enter.
2. **Step 2: Run the Command**

For Command Prompt, type:
shell

```
%SystemRoot%\system32\inetsrv\InetMgr.exe
```

For PowerShell, use:
shell

```
Get-ItemProperty -Path 'HKLM:\Software\Microsoft\InetStp\' | Select-Object ReleaseID
```

3. **Step 3: Read the Version**
    - The system will display the IIS version running on your machine.

c. Checking Installed Programs

1. **Step 1: Open Control Panel**
    - Go to Control Panel > Programs > Programs and Features.
2. **Step 2: Locate IIS**
    - Scroll down and look for Internet Information Services.
    - The version number may be displayed next to it.

By using any of these methods, you can easily find the version of IIS running on your system, which is helpful for maintaining compatibility with various web applications.

2. Dot Net Tutorial: Getting Started with .NET

.NET is one of the most powerful and widely used frameworks for building web applications, desktop applications, and services. It supports multiple programming languages like C#, F#, and Visual Basic, making it versatile for developers.

In this section, we'll dive into a basic .NET tutorial for beginners that will introduce you to setting up a .NET environment and building a simple application.

a. Setting Up the .NET Environment

Before you start building applications with .NET, you need to install the right tools.

Step 1: Install .NET SDK

To start developing in .NET, download the .NET SDK from the official Microsoft .NET website. Ensure you choose the appropriate version based on your operating system.

Step 2: Install an IDE (Visual Studio)

While you can use any code editor for .NET, Visual Studio provides the best development experience, especially for beginners. Download it from the Visual Studio website.

b. Creating Your First .NET Application

Step 1: Create a New Console Application

1. Open Visual Studio and click on Create a new project.
2. Select Console App (.NET Core) and click Next.
3. Choose a name for your project (e.g., "HelloWorld") and select the location to save it. Click Create.

Step 2: Write Your First C# Program

Once the project is created, Visual Studio will open the Program.cs file by default. Modify the Main method as follows:

csharp

```
using System;


namespace HelloWorld
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
        }
    }
}
```

This is a simple program that will display "Hello, World!" in the console.

Step 3: Run the Application

To run your application, simply click the Run button or press F5. The console window will open and display the output: Hello, World!

c. Exploring ASP.NET Core

If you want to move beyond a console application, ASP.NET Core is a great framework for building web applications. Here's a quick overview to get started:

1. **Create a New ASP.NET Core Application**: In Visual Studio, select ASP.NET Core Web Application from the project templates.
2. **Choose the Template**: Choose a project template based on your needs, like Web Application (Model-View-Controller).
3. **Run the Application**: Visual Studio will automatically generate the files, and you can run the web application using IIS Express.

With these basics, you've now started on your journey of .NET development, whether it's a simple console app or a more complex web application.

3. Logback and OpenTelemetry: Integrating Observability in Java Applications

Logback is a widely used logging framework in Java applications, and integrating it with OpenTelemetry can give you greater observability, including distributed tracing and metrics collection.

a. What is Logback?

Logback is part of the SLF4J (Simple Logging Facade for Java) ecosystem, offering flexible logging options. It provides high-performance logging features, including configuration via XML or Groovy, different logging levels, and appenders for various outputs (e.g., console, files, remote servers).

b. What is OpenTelemetry?

OpenTelemetry is an open-source observability framework for collecting telemetry data like logs, traces, and metrics from applications. It provides a standardized way of instrumenting your code, making it easier to monitor distributed applications.

c. Steps to Integrate Logback with OpenTelemetry

Step 1: Add Dependencies

First, you need to include the relevant dependencies in your pom.xml file if you are using Maven. These include Logback, OpenTelemetry, and the necessary exporter (e.g., Jaeger, Zipkin).

xml

```
<dependency>
   <groupId>io.opentelemetry</groupId>
   <artifactId>opentelemetry-api</artifactId>
   <version>1.9.0</version>
</dependency>

<dependency>
   <groupId>ch.qos.logback</groupId>
   <artifactId>logback-classic</artifactId>
   <version>1.2.3</version>
</dependency>
```

Step 2: Configure OpenTelemetry SDK

Next, you'll need to initialize OpenTelemetry in your Java application. Typically, this involves setting up the OpenTelemetry SDK and configuring an exporter for traces.

java

```
OpenTelemetry openTelemetry = OpenTelemetrySdk.builder()
    .setTracerProvider(SdkTracerProvider.builder().build())
    .build();
```

Step 3: Initialize Logback Appender

In your Logback configuration file (logback.xml), add an OpenTelemetry appender to export logs along with traces.

xml

```
<appender name="otel" class="io.opentelemetry.instrumentation.logback.OpenTelemetryAppender">
    <traceIdFieldName>traceId</traceIdFieldName>
    <spanIdFieldName>spanId</spanIdFieldName>
</appender>

<root level="INFO">
    <appender-ref ref="otel" />
</root>
```

This configuration allows Logback to append the OpenTelemetry trace IDs and span IDs to your logs, enabling better correlation between logs and traces.

Step 4: Run and Test the Integration

Once the configuration is complete, you can test your application to ensure that logs, traces, and metrics are being exported to the desired observability platform, such as Jaeger or Zipkin.

Conclusion

Whether you are managing web servers with IIS, learning .NET development, or integrating modern observability into Java applications with Logback and OpenTelemetry, staying informed on these technologies is essential. Knowing how to check your IIS version, getting started with .NET, and enhancing your logging framework with OpenTelemetry will greatly improve your workflow and application performance.