Configuring Web API, MVC, SignalR and Platform Features

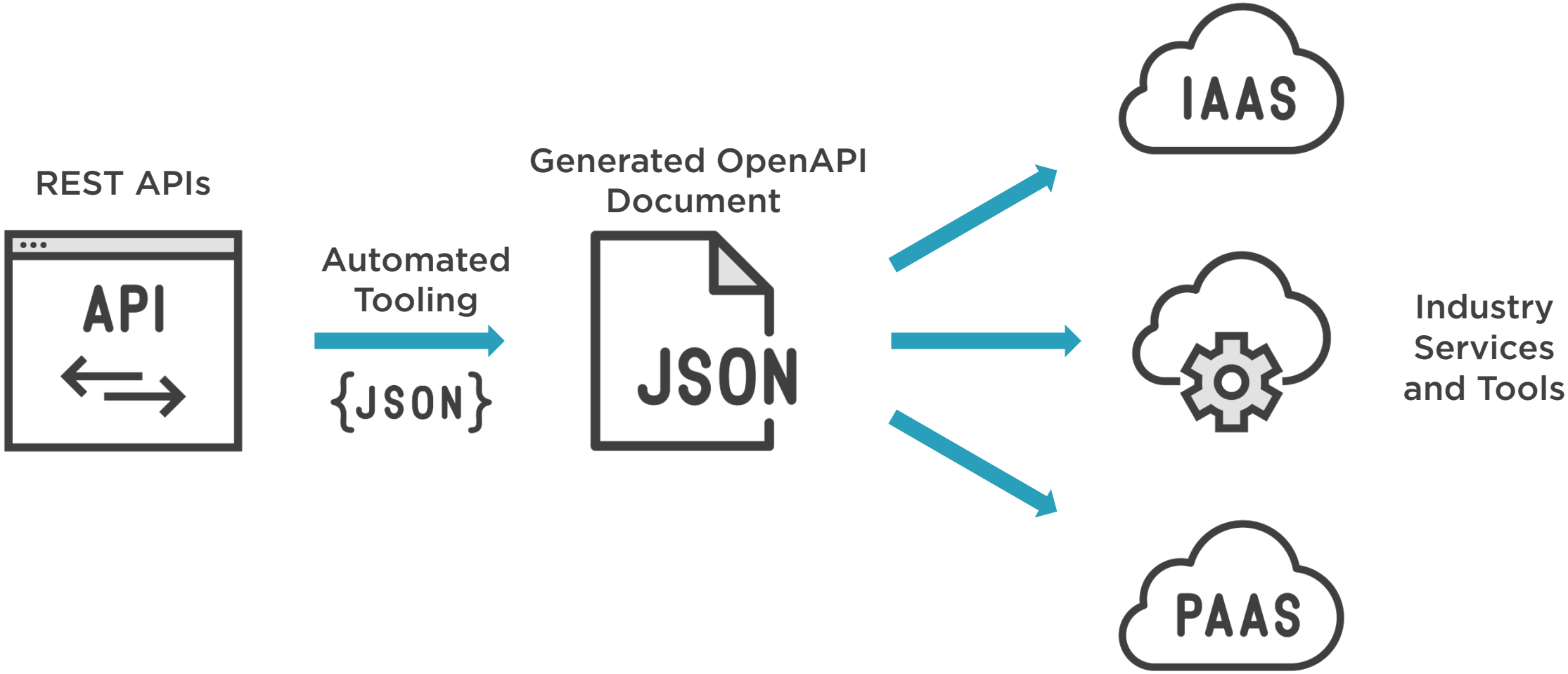
Alex Wolf

www.crywolfcode.com

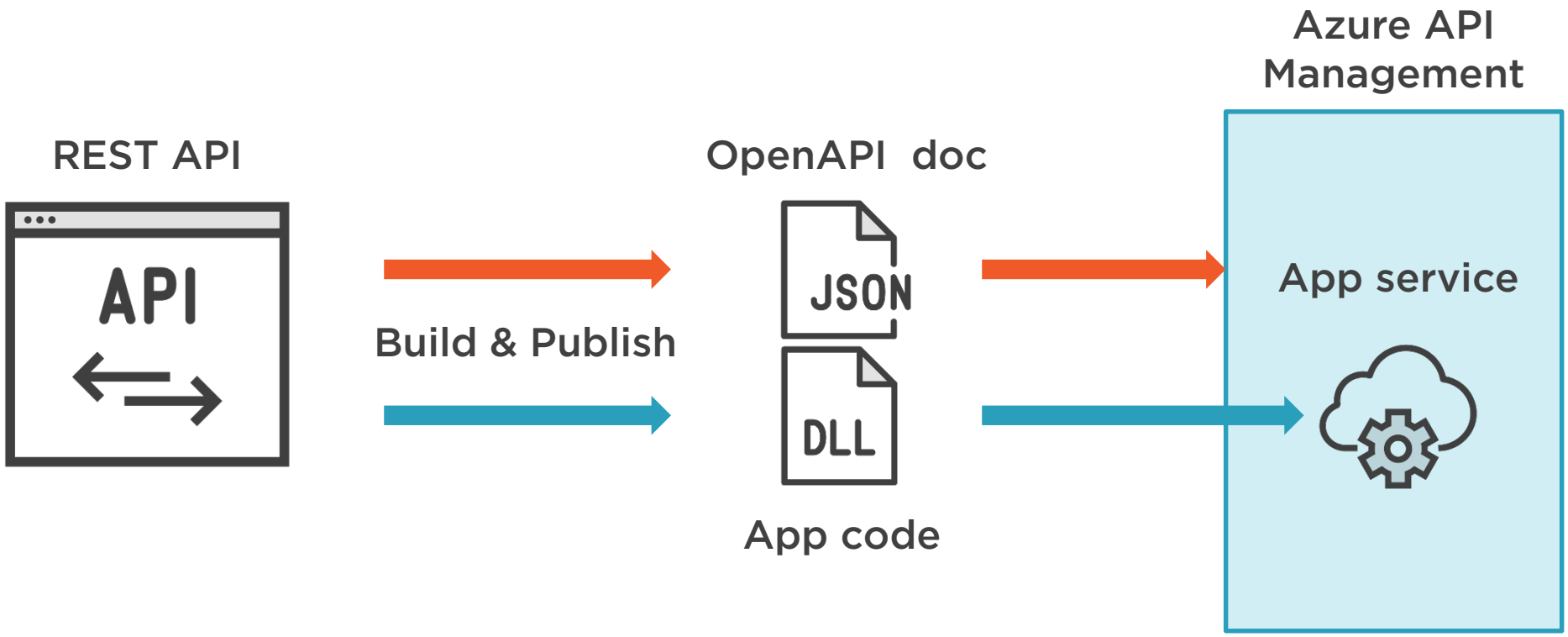
OpenAPI

An industry standard specification for describing the capabilities of REST APIs.

Understanding OpenAPI Documents



Using OpenAPI with Cloud Providers



(Other vendors offer similar features)

Understanding Swashbuckle

A NuGet package for implementing OpenAPI support

Included by default in .NET 5.0 Web API project templates

Utilizes Swagger tooling to improve the developer experience

OpenAPI vs Swagger

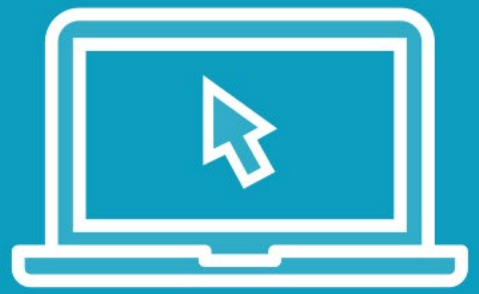
OpenAPI

A formal specification for describing REST APIs

Swagger

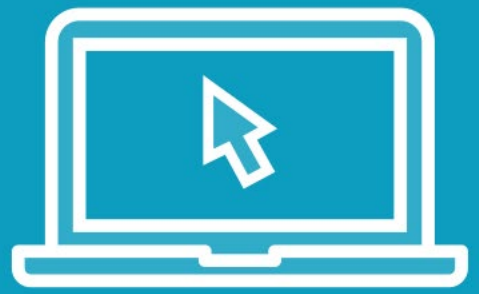
Tooling to help implement the OpenAPI specification

Demo



Working with OpenAPI

Demo



Leveraging OpenAPI with Azure

Exploring MVC and Razor Pages Improvements

```
public record Contact(string Name,
string Email);


public class ContactController

{

    [HttpPost]

    public IActionResult Index(Contact
    contact)

    {

            // Action Method code

    }

}
```

- Inline C# 9.0 record type

- Model Binding the record type

C# 9.0 Features

- Records
- Top-level statements
- New pattern matching techniques
- Syntactical improvements
- Partial class enhancements

Other Model Binding Considerations

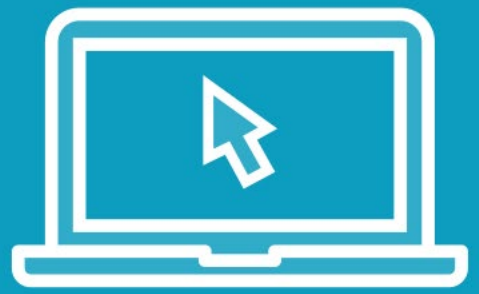
TimeController.cs

```
[HttpPost]

public IActionResult
Index(DateTime utcTime)

{

    // utcTime will be correct

}
```

ContactController.cs

```
[HttpPost]

public IActionResult
Index([FromBody]Contact contact)

{

   // Contact properties now
   required by default

}
```

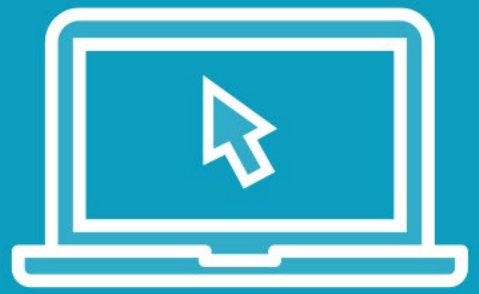
Demo



Handling New Model Binding Techniques

Understanding SignalR Improvements

Demo

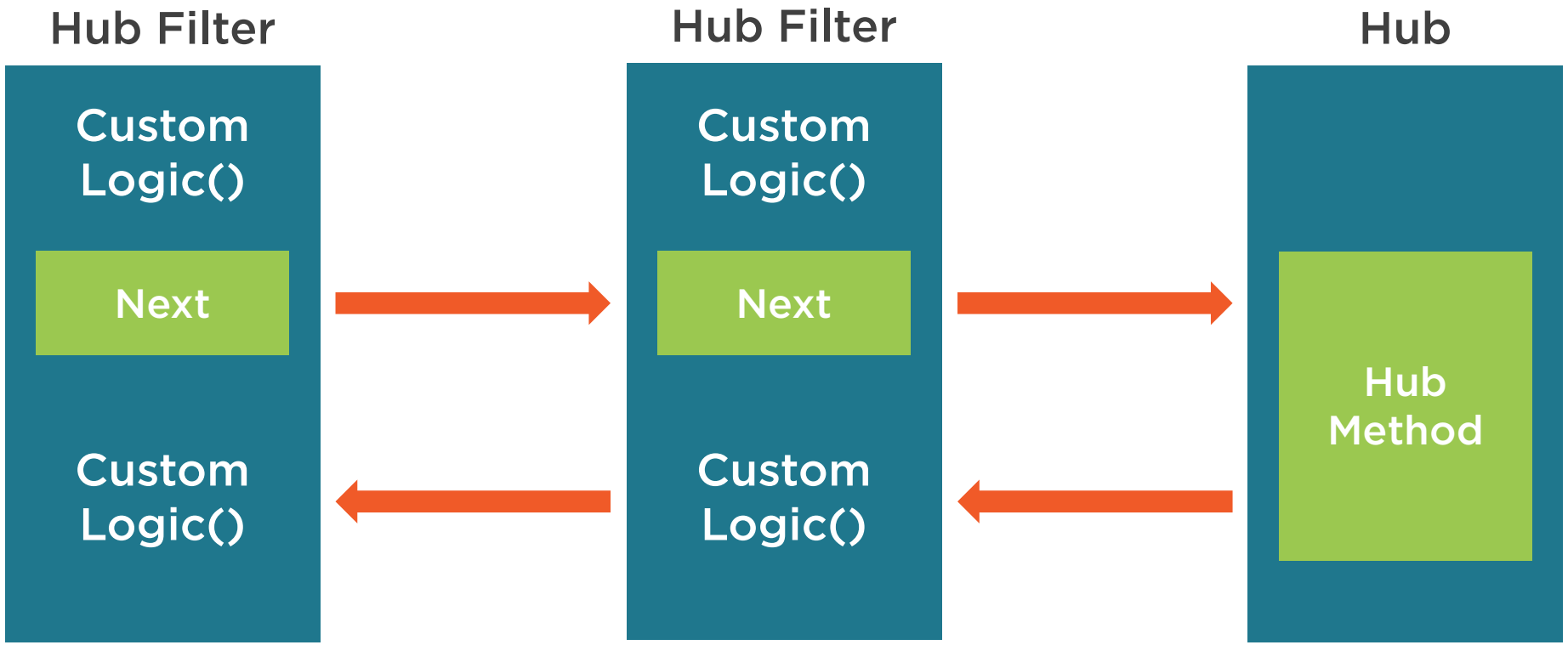


Creating SignalR Hub Filters

Hub Filters

Allow custom code to execute before and after Hub methods are invoked

Hub Filter Execution



```
// Simplified signatures for brevity
public interface IHubFilter {
    InvokeMethodAsync(next);
    OnConnectedAsync(next);
    OnDisconnectAsync(next);
}
```

Understanding the IHubFilter Interface

InvokeMethodAsync allows us to execute logic before and after the next item in the chain

OnConnectAsync and OnDisconnectAsync wrap the respective methods on the hub itself

Configuring Parallel Hub Invocations

startup.cs

```
services.AddSignalR(options =>
{
    options.MaximumParallelInvocationsPerClient = 3;
})
```

Configured in Startup.cs

Default is 1

Additional ASP.NET Core 5.0 Platform Features

Improved Performance in .NET 5.0

Containers

Smaller Container and SDK download sizes

HTTP/2

Improved performance throughout

gRPC

Improved performance throughout

ASP.NET Core 5.0 simply runs faster.

JSON Helper Improvements

```
// Slightly simplified code

endpoints.MapGet("/menu", async context =>
{

var menuService =
context.GetRequiredService<MenuService>();


var menuItems = await menuService.Get();


await context.Response

.WriteAsJsonAsync(response);

}
```

- Map route endpoint
- Get menu service using the context
- Get the menu items
- Write the menus directly to the response using JSON helpers

Improved Authentication and Authorization

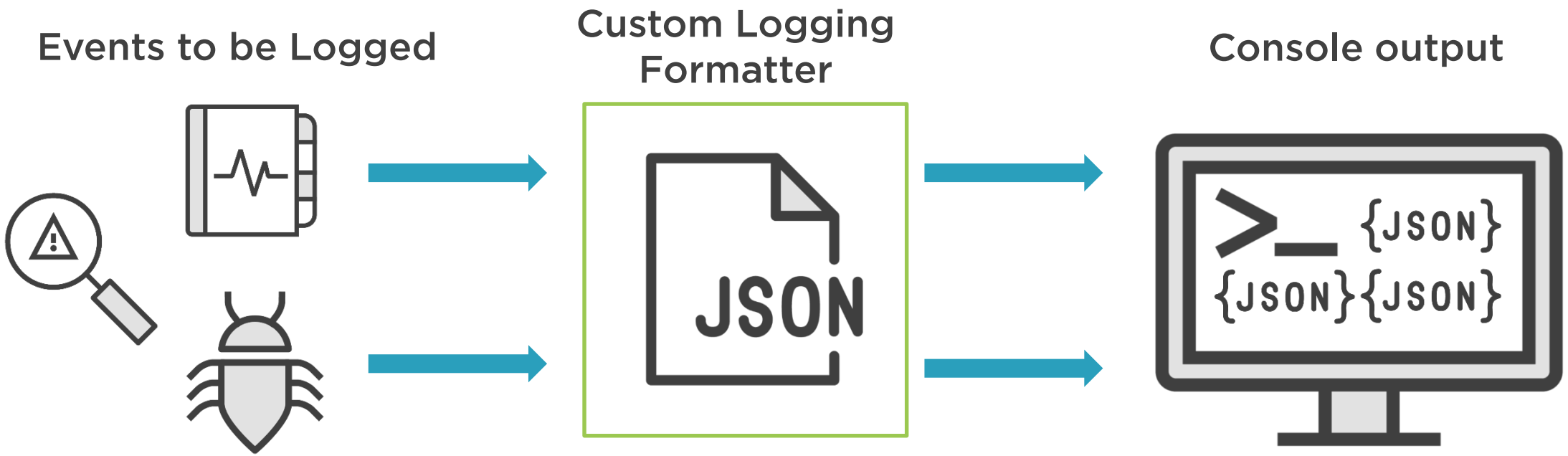
Azure Active Directory

An improved developer experience overall

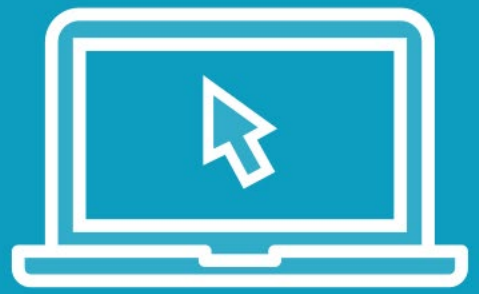
Microsoft Graph

An upgraded, powerful way to interface with Microsoft 365

Introducing Custom Logging Formatters



Demo



Improved Logging with Custom Formatters

Summary

Web API projects now include improved support for OpenAPI and related features

Model Binding now supports C# Records and improved DateTime handling

SignalR now includes Hub Filters, which can run custom logic before and after Hub Methods

ASP.NET benefits from the significant performance improves across all of .NET 5.0

Custom Logging Formatters can change the text format for logs printed to the console

Thank you, and good luck!