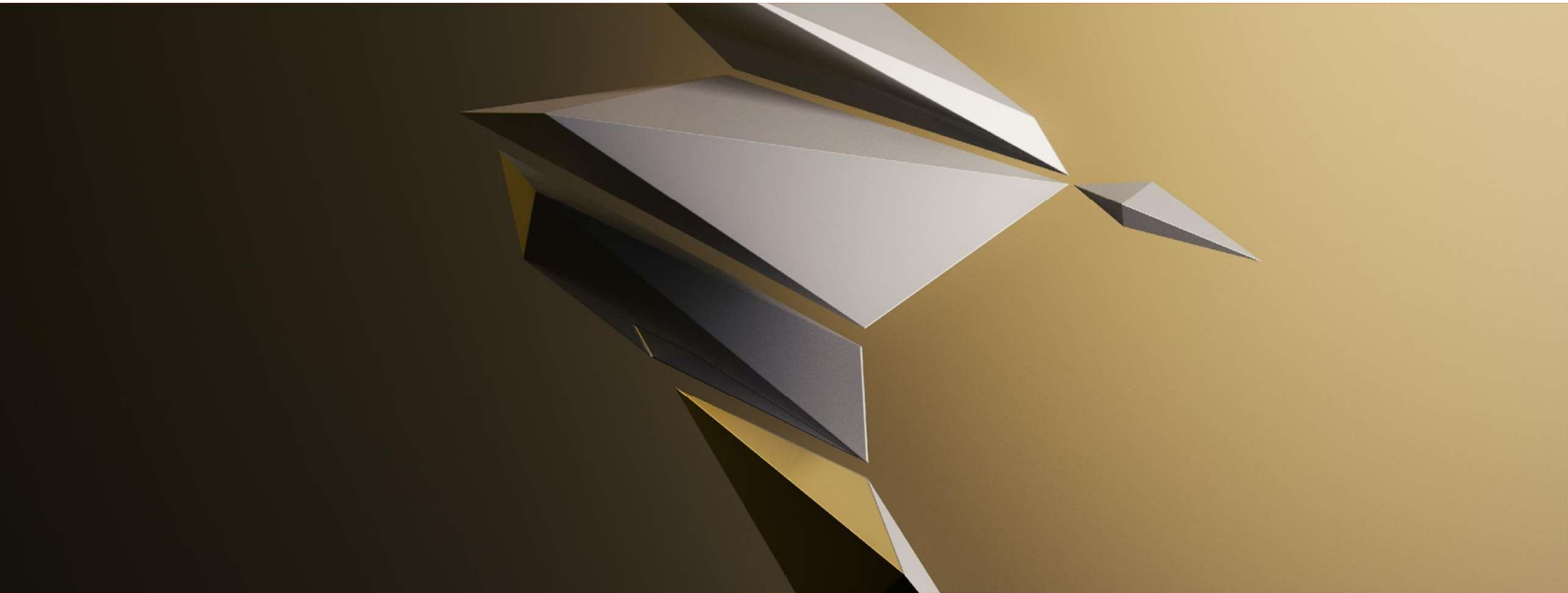




The first Hub for Developers

Asp.NET Core – Microservices
Observability and Monitoring



Observability and Monitoring

- Logging
- Centralized logging
- Application Insights
- Health Checks
- Metrics
- Diagnostics
- Distributed Tracing
- Alerting



Overview



Logging

- allows developers to log messages at various levels of severity.
- can be configured to output logs to various destinations such as the console, file system, or third-party logging services like
 - Serilog
 - Nlog
 - log4net



Centralized logging

aggregating logs from multiple instances of an application into a central location for easy monitoring, analysis, and troubleshooting.

- Install and configure the logging provider
- Configure the logging provider to send logs to a centralized location, often referred to as a "sink".
- Throughout the application code, use the logging framework to log events, errors, warnings, and other relevant information.

Log Aggregation Services

- ELK Stack (Elasticsearch, Logstash, Kibana)
- Splunk
- Seq, integrates well with Serilog
- Prometheus, Grafana, Loki



Application Insights

powerful tools for analyzing and visualizing this data to gain insights into application behavior and performance.



Health Checks

Enhancing user experience and reducing server load by validating user inputs on the client side.

- **Immediate Feedback:**
Providing real-time validation to users during data entry.
- **JavaScript Frameworks:**
Utilizing frameworks like React or Angular for client-side validation.
- **Consistency with Server-Side:**
Ensuring validation rules match those on the server to maintain data integrity.



Metrics

Metrics are quantitative measurements of various aspects of application behavior and performance. .NET Core applications can expose custom metrics using libraries like

- Prometheus.NET or
- Metrics.NET

Thank you!