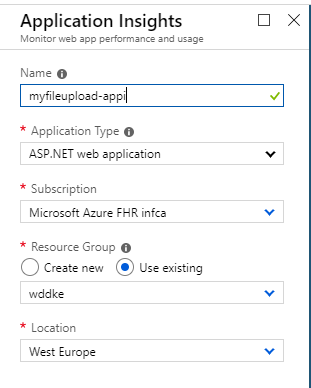
# Monitor Web App und Azure Function Hands On

Für den im Azure WebApps + Storage + Functions Hands On implementierten UseCase wird ein Monitoring implementiert.

* Hinzufügen von Application Insights zur Middleware
* Hinzufügen von Application Insights für Azure Functions
  + Auswerten eines fehlerhaften Uploads
* Custom Logging
  + Hinzufügen von Custom Monitoring
    - In der Middleware mit dem TelemetryClient
  + In der UI
    - Tracken von Page Views mit einem custom monitoring service
* Anzeigen/Auswerten im Portal

**Requirements:**

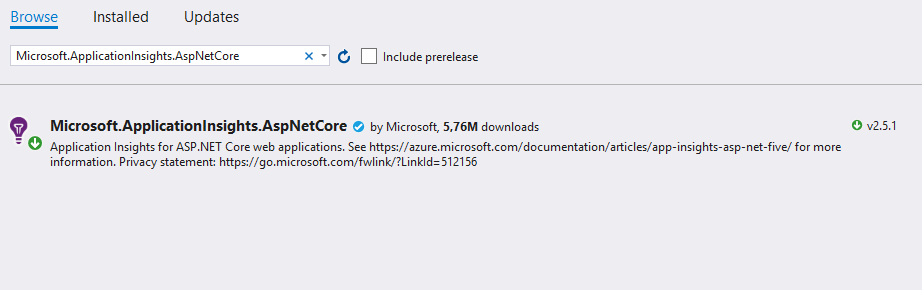
* Application Insights Resourcen (3 Stück) in eigener RG erstellen und sprechend benennen.
  + Middleware
    - Application Insights für die ASP.NET web application erstellen
  + Function
    - Application Insights für die ASP.NET web application erstellen
  + UI
    - Application Insights für Node.js application erstellen (UI)



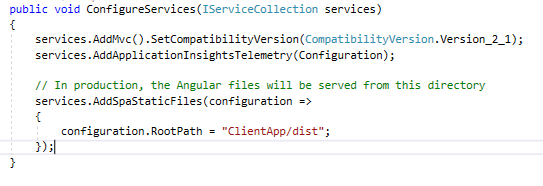
## 1 Implementieren von Application Insights in die Middleware

In Visual Studio in die File Upload App wechseln

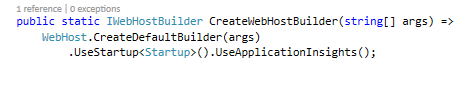
Unter Manage NuGet Packages App Insights hinzufügen und auf aktuelle Version (2.5.1) prüfen



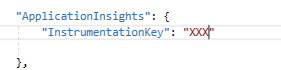
Startup.cs



Program.cs

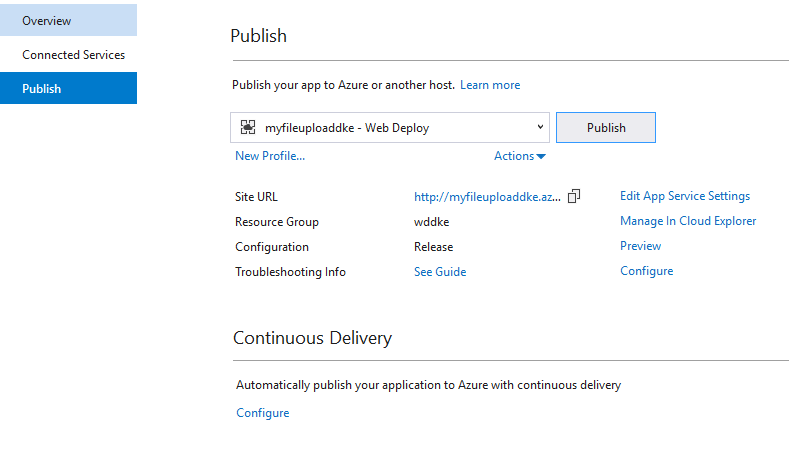


Appsettings.json (Instrumentation Key der richtigen App Insights Resource eintragen)



Ausführen

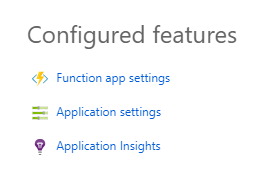
* Lokal ausführen und prüfen ob in der Cloud App Insights resource etwas ankommt (Kann ein wenig Zeit dauern)
* Wenn es funktioniert die App wieder Publishen
* Dann Online testen



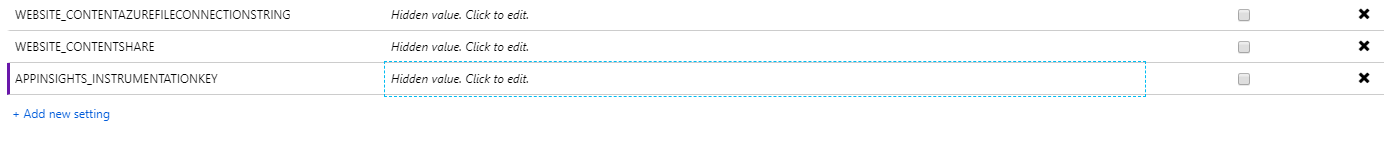
## 2 Implementieren von Application Insights für die Azure Function

Kopieren des Instrumentation Key der Azure Function Application Insights resource

In der Azure Function 🡪 Application Settings



Add new setting



**ÜBUNG**

Zu der Azure Function Application Insights resource wechseln

* Bilder hochladen und prüfen ob App Insights etwas registriert
* Eine Datei (Kein Bild!) hochladen z.B .pdf
* Die Ursache für einen Fehler in der function mithilfe von App Insights suchen (Wo steht wieso der Dateiupload fehlgeschlagen ist)

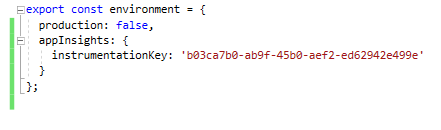
## 3 Implementieren von Custom Events mit dem Telemetry Client

FileUploadController.cs (Den Filenamen und die Filegröße mit einem Upload Event tracken)



## 4 Implementieren von Application Insights für die UI (Custom Logging)

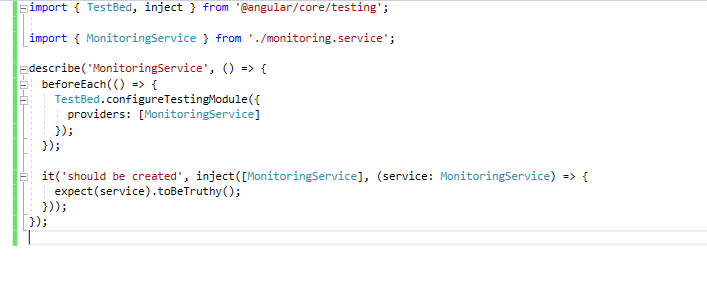
Environment.ts und environment.prod.ts (Key hinzufügen)



Under Ordner app 🡪 neuen Ordner services erstellen und monitoring service files anlegen



Monitoring.service.spec.ts



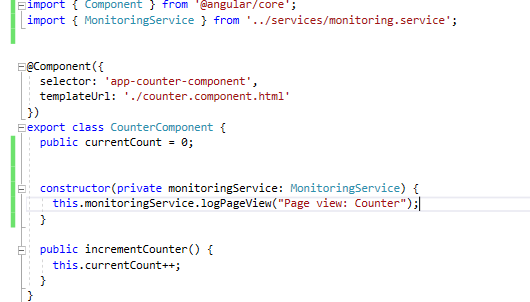
monitoring.service.ts



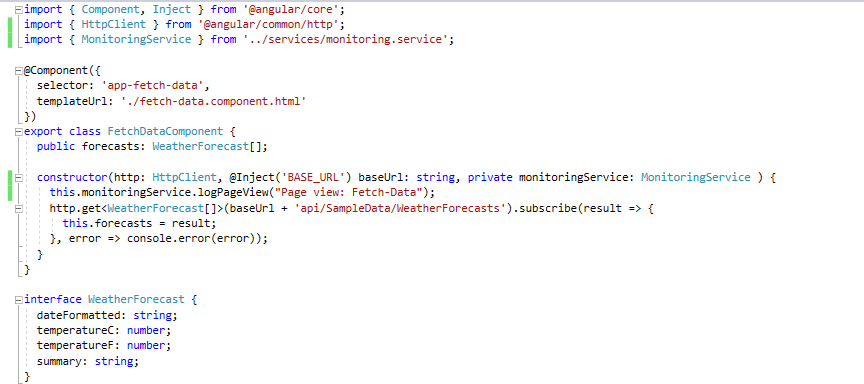
App.module.ts



Counter.components.ts



Fetch-data.component.ts



File-upload.conponents.ts



Ausführen

* Lokal und prüfen in App Insights was angezeigt wird
* Publishen und nochmals prüfen