

# Philip Jönsson SUT23 – Teorihandbok ASP.NET

## Vad är ASP.NET och MVC?

ASP.NET är ett ramverk utvecklat av Microsoft för att skapa webbapplikationer. Detta ramverk gör det möjligt att bygga dynamiska, robusta och skalbara webbplatser, webbtjänster och applikationer.

MVC, som står för Model-View-Controller, är ett designmönster som används för att strukturera applikationer på ett tydligt och enkelt sätt. Varje del av MVC representerar en specifik funktion, vilket gör det lättare att förstå och underhålla koden.

- **M = Model:** Modellen ansvarar för att hantera data och affärslogik. Den interagerar med databasen och andra datakällor för att hämta och manipulera information.
- **V = View:** Denna komponent står för användargränssnittet (UI). View visar data för användaren och tar emot input, som sedan skickas vidare för behandling.
- **C = Controller:** Controllern fungerar som en mellanhand mellan modellen och vyn. Den hanterar användarens inmatningar och beslutar vilken vy som ska visas baserat på dessa interaktioner.

Denna uppdelning i tre distinkta komponenter underlättar både utveckling och underhåll av applikationen, vilket bidrar till en mer effektiv kodstruktur.

## Vad är skillnaden mellan MVC och Web Forms?

**Web Forms** är en händelsedriven modell som möjliggör för utvecklare att bygga användargränssnitt genom att använda drag-and-drop-kontroller. Denna metod ger en liknande känsla som att utveckla skrivbordsapplikationer, vilket gör det enklare för nybörjare att komma igång. I Web Forms är mycket av koden kopplad till olika händelser, som när användaren klickar

på olika element. Trots sin användarvänlighet kan denna metod leda till en mer komplex och svårhanterlig kodbas när applikationen växer, vilket gör den mindre lämplig för större projekt.

**MVC** erbjuder en mer strukturerad och flexibel utvecklingsmetod. Den bygger på en tydlig separation av ansvar mellan komponenterna, vilket gör det enklare att skriva ren och testbar kod. I MVC interagerar man med webbapplikationen genom HTTP-förfrågningar, vilket ger utvecklaren större kontroll över koden och hur HTML genereras. Denna metod gör det också lättare att optimera för sökmotorer och att bygga responsiva gränssnitt.

## Vad är Webservices?

Webservices är tjänster som gör att olika program och applikationer kan kommunicera med varandra över internet. De använder vanliga protokoll och format, som HTTP, XML, JSON, SOAP och REST, för att skicka och ta emot information.

Genom att använda webservices kan olika system och plattformar dela data med varandra, oavsett vilket programmeringsspråk de är skrivna i. Detta gör det enklare för dem att samarbeta och fungera tillsammans.

Webservices används ofta för att koppla ihop externa tjänster, dela data mellan olika applikationer och bygga mikroservice-arkitekturer. Dessa egenskaper hjälper företag och utvecklare att skapa flexibla och skalbara lösningar.

## Hur publicerar vi webbapplikationer genom Azure?

För att lansera webbapplikationer i Azure följer man vanligtvis dessa steg:

1. **Utveckling och testning:** Börja med att utveckla och testa din applikation lokalt. Det är viktigt att allt fungerar som det ska innan du går vidare till publicering.
2. **Skapa en resurs i Azure:** Logga in på Azure-portalen och skapa en lämplig resurs för din applikation, som tex en App Service. Detta skapar den grundläggande plattformen som krävs för att köra applikationen.
3. **Publicering:** Använd verktyg som Azure DevOps, GitHub Actions eller andra CI/CD-pipelines för att automatisera publiceringsprocessen. Alternativt kan du använda Visual Studio för att publicera applikationen direkt till Azure.

4. **Övervakning och underhåll:** Efter publiceringen, använd Azure Monitor och andra övervakningsverktyg för att hålla koll på applikationens prestanda och eventuella fel. Se till att utföra regelbundna uppdateringar och underhåll för att hålla applikationen i gott skick.

### **Vad är Containers och vad används de till?**

Containers är ett lätt sätt att paketera och köra program. De skapar en isolerad miljö där applikationer kan köras med alla de delar och inställningar som behövs. Detta gör att applikationer fungerar på samma sätt oavsett var de körs – på din egen dator, i molnet, eller på en server.

Containers används för att förenkla utvecklingen och hanteringen av program. De gör det också lättare att skala upp applikationer och spara resurser. Containers är särskilt populära i system där olika delar av en applikation körs separat, som i mikrotjänstarkitektur. Detta gör det enklare att uppdatera och distribuera program utan att störa resten av systemet.

### **Vad är DevOps och Pipelines, och hur används de?**

**DevOps** är ett sätt att arbeta som främjar samarbete mellan utvecklingsteam och driftsteam, så att de kan jobba bättre tillsammans. Syftet är att snabbare och med högre kvalitet kunna utveckla, testa och lansera programvara. DevOps handlar också mycket om att automatisera olika steg i utvecklingsprocessen för att undvika misstag och göra arbetet mer effektivt.

**Pipelines** är en viktig del av DevOps och innebär ett automatiserat steg-för-steg-flöde för att bygga, testa och lansera programvara. En typisk pipeline kan inkludera:

- Granskning av kod
- Testning för att säkerställa att allt fungerar
- Integration för att se att koden fungerar ihop med resten av systemet
- Slutligen, lansering av programvaran

Genom att automatisera dessa steg sparar man tid, minskar risken för misstag, och kan snabbare få ut ny eller uppdaterad programvara till användarna.

## **Vad är AI och ML, och hur fungerar det?**

**AI (Artificiell Intelligens)** är ett område inom teknik där man försöker skapa datorer och system som kan utföra uppgifter som vanligtvis kräver mänsklig intelligens. Det kan vara saker som att förstå språk, fatta beslut eller känna igen bilder och ljud.

**ML (Maskininlärning)** är en del av AI som går ut på att få datorer att lära sig själva genom data. Istället för att programmera datorn för varje specifik uppgift, tränar man en ML-modell med stora mängder data så att den själv kan upptäcka mönster och göra förutsägelser.

Så här fungerar det i korthet: en ML-algoritm får data att "öva" på, och genom att analysera datan lär den sig att göra allt bättre förutsägelser med tiden. Vanliga exempel där AI och ML används är i bildigenkänning, röstassistenter, rekommendationssystem (som för filmer och musik) och att förutsäga trender.