**Introduksjonskurs i ASP.NET Web API**

**Mål**

*Målet med kurset er å gi en oversikt over hva ASP.NET Web API er og gi deg mulighet til å lage et enkelt Web API som kan eksponeres over HTTP.*

**Forkunnskaper**

Det er en fordel med grunnleggende kunnskaper i .NET, C#, og Visual Studio. Det er også en fordel å være litt kjent med ASP.NET MVC3/MVC4.

**Varighet**

**Krav**

Visual Studio 2010 eller nyere må være installert på maskinen. Dersom dette ikke er installert sjekk MSDN siden i Confluence med instruks om hvordan du kan laste ned og installere det. I tillegg må man ha installert ASP.NET MVC4 som finnes her <http://www.asp.net/mvc/mvc4>. Det finnes mange verktøy for å se på trafikk over f.eks HTTP, Fiddler er et veldig godt et og det anbefales of installere dette også: <http://www.fiddler2.com/fiddler2/>

**Ressurser**

Eksempelkode og presentasjon ligger under <http://github.com/bekk/dotnetkurs>.

Dette er et privat repo under BEKK på Github og krever derfor at du blir lagt til i Team dotnetkurs for å få tilgang. Brukernavn og passord til bekkadmin på github finnes på Confluence.

**Introduksjon**

Med ASP.NET Web API så kan man lage tjenester som blir eksponert over ren HTTP i motsetning til over SOAP, som er mer vanlig med f.eks .NET Windows Communication Foundation (WCF). Motivasjonen for å kunne gjøre dette er blant annet for å kunne tilgjengeliggjør det samme API’et for flere typer enheter og platformer. I dette kurset ser vi nærmere på det å opprette et lite Web API samt å konsumere det fra en annen applikasjon. Med MVC4 er det en egen prosjekt mal som er tenkt brukt mot utvikling av Web API. Når man åpner Visual Studio og velger File->New Project->Web kan man velge ASP.NET MVC4 Web Application og videre velger man Web API. På den måten får man et relativt blankt prosjekt som man kan starte å gjøre oppgavene i.

Det oppfordres til å prøve å gjennomføre oppgavene test-først.

**Løse avhengigheter med Ninject**

I dokumentet *Ninject\_ASP\_NET\_Web\_API.pdf* som ligger på i repoet på github er det en beskrivelse av hvordan man kommer i gang med å bruke Ninject som IoC rammeverk innenfor ASP.NET Web API – det anbefales å gå gjennom dette før man starter på selve oppgaven.

**Komme i gang**

Når man har fått opp et ”tomt” prosjekt i ASP.NET Web API vil mye av mappestrukturen ha likheter med den strukturen som opprettes for MVC prosjekter. Blant annet med *Controllers*, *Models* og *Views* mapper som vil ha respektive deler i seg. Controllere mtp ASP.NET Web API arver fra *ApiController* i stedet for *Controller* – det er en av hovedforskjellene. For å opprette en ny Controller er det en fordel å definere en model først, og deretter bruke funksjonalitet i Visual Studio for å generere selve Controlleren. Dette gjøres ved å høyreklikke på Controllers mappen og velge Add Controller. Alle Controllere skal ha et navn som slutter på Controller. For enkelhetsskyld kan det være greit å starte med en *Empty API Controller*.

**Tips og triks**

*Minimèr logikk i controller*

Controllere brukt i både Web API og MVC bør være så ”dumme” som mulig og bør benyttes seg av andre komponenter/klasser for å få utført oppgavene de skal gjøre. Dette er for å minimere nødvendig testing direkte mot kontrollerne og for å få separert ut logikk i klasser hvor det faktisk hører hjemme.

*Dependency Injection*

For å få Controllerne så tynne som mulig kan man ved hjelp av dependency injection sende inn instanser av f.eks et repository til en kontroller. Kontrolleren i sin tur bruker dette repository for å hente ut data i stedet for at den sitter på all logikk selv for å hente ut data. Dette gjør det også enklere å teste funksjonaliteten i repositoryet og forventet oppførsel i kontrolleren.