**Vad är statemanagement?**

* Situation som kommer upp när man bygger applikationer, Används för applikationer, ex Pizzeria
* Hur man hanterar saker som ska skickas med hela tiden. Man ska hålla kvar saker i minnet som man har gjort tidigare.
* **Varför inte bara skapa en variabel som håller allt i minnet?**
* Bygger man en vanlig desptop applikation, windows app, ligger den på internminnet när man kör den. Gör man något på skärmen så sparas det i ett ”mellanlager”. Men om man kör en ***webapplikation*** så är det inte lika. Det kommuniceras mellan servern och webläsaren hela tiden. Respons o request, dialog mellan webbläsare och webbserver (alt hel frontend app).

När man kör med serverkod = webbserver och webbläsare kommunicerar

* Det man använder då är http-protokollet: http-prot i sig innehåller inte state. Om servern själv skulle hålla koll på allt så skulle det bli väldigt mycket. SÅ fort en webbserver hnterat en request från webläsare så har den inte kvar den informationen, utan den försvinner. När det sedan kommer en ny request så kollar man att det var från samma webbläsare, men man har inte kvar ngtg per automatik vad som hände i den förra.

Grundproblemet: Httppro håller inte state/det man gjort tidigare/historiken i en applikation utan det måste man hantera på eget sätt när man bygger applikationen.

**Vad är en session?**

* När en webbserver o webbläsare kommunicerar.
* När det används? Tillfällig datalagring av information (vad det används till, när man använder den… sessionen är också steget innan).
* Webbservern håller koll på att nu är det en webbläsare som skickat en förfrågan. Första gången som en webbläsare skickat en förfrågan till webbservern så noterar den att det kommer en ny request som vi inte haft kontakt med tidigare, registrear vi att en webläsare vill ha kontakt med mig. Då registrerar vi en session här uppe och så skickar man tillbaka ett sessionsid till webbläsaren. Sessionen hanteras med en vcookie.

Nästa gång webbläsare skickar en request till webbserver så är det egentligen sessionscookien med upp tills servern. Servern läser och ser att det är någon som pratat med mig tidigare. Det vet inte servern utan cookien, servern läser sessions-id:t.

Aktiv session: Tills man avslutar webbläsaren/stänger av datorn/studentportalen (finns en timeout (inställning)) sessioner som är inaktiva ska stängas ner. Vill inte ha inaktiva sessioner. Gör man ingenting så är default för sessioner att de lever 20min. Ex en webbshop vill man inte slänga ut någon p.g.a. den inte är aktiv. Internetbanker vill av säkerhetsskäl och resursskäl vara hårda.

**Sammanfattning session - Stegvis:**

1. Session är något som skapas när man skickar sin första request till en webbserver.
2. Det skickas tillbaka en sessionscookie
3. Den skickas mellan webbläsaren och webbservern
4. Sessionen avbryts när man stänger ner webbläsaren/sessions timeout
5. Kan använda sessioner för att hantera variabler, sessionsvariabler, ex varukorg i webbshop kan man lägga i en sessionsvarabel för att hålla koll på vad någon handlat.
6. Och så stoppar man ner det i sessionsvarabeln och så skcikar man tillbaka ner till klienten, de stoppar in något mer i varukorgen, produkt, och så skickar man upp det till webbservern och lägegr till det i sessionsvarabeln och när man går till kassan så plockar man ut det som finns i sessionsvarabeln.