

Tic-Tac-Toe Network Edition v.2024

Laboration 2. Webbserver v2.0, dubbelriktad kommunikation, websockets och timer

Webbserver v2.0

1. Importera http modulen (konstant med namnet http) och skapa samtidigt en http-server med app som indata (createServer(app)).
2. Kommentera bort Express-servern från laboration 1.
3. Ladda hem kodbiblioteket nedan samt importera det i app.js:
 - socket.io (konstant med namnet io med tillägg http).
4. Se till att din http-server lyssnar efter anrop på port 3000.

Observera att du bara ska och får skriva kod i app.js. Övriga filer får du inte ändra i eller göra tillägg till (gäller både statiskt och dynamiskt) och inte heller ändra i katalogstrukturen. Observera också att attribut och metoder i globalObject skall användas.

Dubbelriktad kommunikation och websockets

För io skall ni nu lyssna för "connection" och "newMove".

Vid "connection":

1. Om kakorna nickName och color inkommer till servern från klienten kontrollera om det är spelare ett eller spelare två som vill ansluta alternativt om det redan finns två anslutna spelare.
2. Om det redan finns två anslutna spelare gör disconnect med texten "Redan två spelare anslutna!".
3. Om det är spelare ett som ansluter se till att tilldela värden till lämpliga attribut i oGlobalObject.
4. Om det är spelare två som ansluter se också till att tilldela värden till lämpliga attribut i oGlobalObject, nollställ spelplanen i oGlobalObject samt skicka händelsen "newGame" till båda anslutna spelarna med den data som efterfrågas (se klientskriptet). Låt sedan spelare ett börja genom att tilldela ett värde till lämpligt attribut i oGlobalObject samt skicka händelsen "yourMove" till spelare ett (se klientskriptet för mer detaljer).
5. Om kakorna inte kommer till servern gör "disconnect" med texten "Kakorna saknas!".

Vid "newMove":

1. Uppdatera spelplanen i oGlobalObject med aktuell spelare.
2. Kontrollera vilken aktuell spelare är och ändra sedan till motståndaren genom att tilldela ett värde till lämpligt attribut i oGlobalObject samt skicka händelsen "yourMove" till spelaren (se klientskriptet för mer detaljer) som är på tur.
3. Kontrollera om spelet skall fortsätta, om det finns en vinnare eller om det är oavgjort. Om det är oavgjort eller om det finns en vinnare skicka händelsen "gameover" till båda spelarna och skicka med en lämplig text.

Timer

Innan ni börjar med Timer skall ni se till att er lösning fungerar felfritt utan timer-händelser.

1. Skapa funktionen "timeout" i vilken ni kontrollerar om aktuell spelare är spelare ett eller två.
2. Om aktuell spelare är ett skicka händelsen "timeout" till spelare ett och händelsen "yourMove" till spelare två. Ändra sedan lämpligt attribut i oGlobalObject för aktuell spelare.
3. Om aktuell spelare är två skicka händelsen "timeout" till spelare två och händelsen "yourMove" till spelare ett. Ändra sedan lämpligt attribut i oGlobalObject för aktuell spelare.
4. Stäng ner timer och starta sedan en ny timer som efter fem sekunder anropar timeout-funktionen.
5. Avslutningsvis skall ni på lämpliga ställen (ett tips är att stanna upp och fundera en stund om var lämpliga ställen är) i "connection" och "newMove" starta och avsluta timer-händelser. När ni startar en ny timer skall den efter fem sekunder anropa timeout-funktionen. Kom-ihåg att lagra undan timer-id i lämpligt attribut i oGlobalObject.

Lycka till!