

Lathund för versionshantering med Git och IntelliJ

Initialisera Git

Alternativ 1: Bocka i rutan "Create Git repository" när ni skapar ett nytt projekt så gör programmet automatiskt en git init åt er, och ni kan därefter commita filer. (Tänk dock på att IntelliJ kommer ihåg den här inställningen framöver, vilket kan vara irriterande).

Alternativ 2: Skapa ett nytt projekt och gå sedan in under VCS i menyn, tryck på "Enable Version Control Integration" och klicka på OK.

Gör en commit

Alternativ 1: Klicka på "Commit" i vänstermarginalen på IntelliJ, oftast någonstans under projektvyn.

Alternativ 2: View > Tool Windows > Commit.

Alternativ 3: Alt+0 (om ni inte ändrat grundinställningarna för genvägar i programmet).

Förläng listan som säger "Unversioned files" och kryssa i vilka filer ni vill commita (alla era filer är unversioned till att börja med eftersom de saknar versionshistorik).

Jag rekommenderar att ni har definierat en .gitignore för projektet innan ni commitar någonting eftersom den gör alla filer och mappar som ska ignoreras osynliga för commit-fönstret, och det går därmed inte att lägga till dem av misstag.

Pusha kod

Alternativ 1: Skapa ett repository först på Github och klicka sedan på "Define remote" i rutan som kommer upp när ni försöker göra en push (Git > Push i menyn efter att ni gjort en commit), och klistra in länken i URL-fältet.

Alternativ 2: Skapa ett repository och pusha kod till det direkt inifrån IntelliJ. Det här alternativet fungerar inte alltid beroende på nätverksinställningar, brandväggar och liknande, men det är ett mycket enklare arbetsflöde. Klicka på Git (som ersätter VCS i menyn när man aktiverat versionshantering) och välj sedan antingen GitHub > Share Project on Github eller enbart "Share Project on Github", beroende på vilken version och vilket gränssnitt ni använder för IntelliJ.

Tänk på att filer måste commitas först, annars går det inte att pusha någonting.

Uppdatera projekt

Om ni klonat ett projekt från Github, eller om ni är flera som arbetar med ett projekt, kan det hända att koden i repositoryt online ändrats. För att vara säkra på att ni har den senaste versionen kan vi välja Git > Update Project... i menyn.

Tänk på: Om ni gjort ändringar i någon lokal fil kan det uppstå en "merge conflict" när in gör detta, och programmet kommer i så fall att be er välja vilken version av filen ni vill behålla.

Byta remote

En annan sak som vi tyvärr inte hann gå genom i fredags, men som är bra att känna till, är att man även kan byta remote för ett projekt. Det här kan vara bra att veta om man t.ex. anger en felaktig remote, eller om man klonar ner ett repository och sedan vill pusha det till sitt eget GitHub-konto, eller liknande.

På både Mac och Windows kan man trycka två gånger på Shift (↑), söka efter remotes i fältet som dyker upp, och sedan klicka på "Manage Remotes..." bland resultaten. Man har då möjlighet att ta bort en existerande remote och lägga till en ny i stället.

Klona ner ett projekt från Github

Kopiera länken till det repository ni vill klona. I stället för att skapa ett nytt projekt när ni öppnar IntelliJ trycker ni på knappen som antingen heter "Get from VCS" eller "Clone repository". (Om ni har ett projekt öppet redan i IntelliJ behöver ni stänga det för att den här dialogrutan ska dyka upp.)

Ta bort Git från ett projekt

Ibland kan det hända att man råkat aktivera versionshantering för ett projekt som inte ska ha det, eller att man pushat en specifik version för ett visst syfte och sedan inte längre vill ha Git aktiverat, eller kanske helt enkelt har klatrat till det med sina commits och vill göra om allting.

Om man öppnar projektmappen och raderar mappen som heter .git så försvinner all lokal versionshantering för projektet (ingenting som redan pushats försvinner från GitHub dock!). Den här mappen är egentligen osynlig, och det kan hända att man behöver bocka för någonstans i inställningarna för datorn på både Windows och Mac att osynliga mappar och filer ska visas för att kunna se den.

Åtkomstnivåer

Tänk på att alltid göra allting som har med skoluppgifter att göra "Private" så att ni inte gör era lösningar synliga för resten av GitHub.

Dela ett repository med andra

Om ni är flera som arbetar med ett repository är det fortfarande en dålig idé att göra det publikt, särskilt om ni arbetar med projekt som görs åt utomstående (t.ex. programvaruprojektet), eftersom alla kan granska och klona er kod i så fall.

Ett bättre sätt att arbeta är att en av er skapar själva repositoryt och sedan bjuder in de andra gruppmedlemmarna. Man kan göra detta genom att klicka på repositoryt och sedan på "⚙ Settings" i den övre menyraden. Detta öppnar upp inställningarna för repositoryt och genererar ännu en meny i vänsterspalten. Klicka på "Collaborators" (nästan längst upp) och sedan på den gröna knappen som säger "Add people" på sidan som laddas in. Sök efter namnen på de githubkonton ni vill lägga till.

Alla som läggs till på det här viset kommer att ha rättigheter att pusha filer till repositoryt.