Git – עבודה מול שרת חיצוני של גיטהאב

במאמר הקודם למדנו על קומיטים מקומיים. למדנו איך יוצרים repo מאפס, איך מוסיפים קבצים שונים ל-commit, איך סוקרים את ה-commit עם git status, איך מבצעים את ה-commit עצמו ואיך בוחנים את השינויים באמצעות git log.

בעוד שאפשר לכאורה להסתפק בשינויים המקומיים, ברוב הפעמים אנו עובדים עם שרת אחד (או יותר) המכילים את הגרסה המרוחקת של ניהול הגרסאות שלנו. למרות שיחסית קל להתקין (על לינוקס) שרת שתומך בגיט. ברוב המקומות משתמשים בשירותי ענן לאחסן את נתוני הגיט. אחת המערכות הפופולריות ביותר היא GitHub שמאפשרת לכל משתמש ליצור מספר רב של מאגרים ולהתחבר אליהם וכן למשתמשים אחרים לראות את השינויים, לבצע העתקה של המאגר ולתרום קוד. אני אפרט בהרחבה על גיטהאב בהמשך. ישנם עוד שירותים שדומים לגיטהאב והמפורסם מביניהם הוא ביטבאקט של Atlassian שגם הוא מצויין.

על מנת להמשיך את הלימוד, אנו חייבים להתאמן על שרת מרוחק. אם אין לכם חשבון בגיטהאב, צרו אחד. היצירה היא פשוטה וקלה ונעשית דרך האתר. אחרי היצירה המוצלחת של החשבון ואישור המייל, אנחנו ניצור repo חדש – שהוא בעצם המאגר הראשון. זה די פשוט:

אחרי הלוגין הראשון, אנו נלחץ על הוספת new repository.

כפתור ירוק של הוספת new repository

כפתור ירוק של הוספת new repository

מיד נעבור למסך שבו אנו יכולים לבחור שם. לשם התרגול והפשטות אבחר שם שנקרא temp:

בחירת שם ל-repository החדש

בחירת שם ל-repository החדש

אחרי היצירה המוצלחת, אנו מקבלים הוראות שעליהם אעבור מייד:

קבלת ההוראות ל-repo החדש

קבלת ההוראות ל-repo החדש

הממשק של Github, כמו כל ממשק, עשוי להשתנות עם הזמן. אבל העקרון צריך להיות ברור לכם – שימוש באתר על מנת ליצור repository. אחרי היצירה, אנו יכולים לדחוף את הקוד שלנו ל-repo החדש שנוצר. במאמר הקודם יצרנו כבר קוד ועשינו קומיט. אבל הקומיט הוא מקומי. אנחנו יכולים להמשיך ולעבוד על המקומי אבל אם אנחנו עובדים בצוות, אנו רוצים ששאר חברי הצוות יוכלו לראות את הקוד ויש להעביר אותו מהקומיטים המקומיים שנעשו אל השרת. התהליך הזה נקרא push. הדבר הראשון שאנחנו צריכים לעשות זה לחבר את המאגר המקומי שלנו אל השרת המרוחק של github. עושים זאת באמצעות פקודה שנקראת git remote add שמקבלת שני פרמטרים – הראשון הוא שם השרת המרוחק (בדרך כלל origin אבל אנחנו יכולים לקרוא לו בכל שם) והשני הוא כתובת שרת הגיט – במקרה של הדוגמה שלנו הוא https://github.com/sometestuser2/temp.git. הכתובת הזו מורכבת משם המשתמש שלי ושם הפרויקט שלי. כל פרויקט בגיטהאב מכיל את הקונבנציה הזו.

כשאני מקליד את הפקודה git remote add origin https://github.com/sometestuser2/temp.git בתיקית git מקומית, אני בעצם אומר ל-git שמעכשיו origin הוא השרת המרוחק שיש להתיחס אליו.

איך אנחנו יודעים איזה שרתים מרוחקים מחוברים ל-repo שלנו? כותבים את הפקודה git remote -v

$ git remote -v

origin https://github.com/sometestuser2/temp.git (fetch)

origin https://github.com/sometestuser2/temp.git (push)

אנו רואים שיש שרת מרוחק שמוגדר כ-origin עבור ה-push שזו הדחיפה ויש שרת מרוחק שמוגדר כ-origin עבור ה-fetch שזה העדכון. כרגע מה שמעניין אותנו זו הדחיפה – כלומר העדכון.

איך עושים את הדחיפה? באמצעות push. פקודת push מקבלת גם שני פרמטרים. הראשון הוא שם השרת שבחרנו (במקרה הזה הגדרנו אותו כ-origin) והשני הוא שם הענף. שם הענף שאני עובד איתו בהתחלה הוא master. לפיכך הפקודה שלנו תהיה git push origin master. גיטהאב החביב יבקש מאיתנו את שם המשתמש והסיסמה שלנו. שאותם הגדרנו בעת יצירת החשבון. אם הכל יהיה תקין, כל התהליך יראה ככה:

$ git remote add origin https://github.com/sometestuser2/temp.git

$ git push origin master

Username for 'https://github.com': XXXX@XXXX.com

Password for 'https://XXXX@XXXX.com@github.com':

Counting objects: 6, done.

Delta compression using up to 2 threads.

Compressing objects: 100% (2/2), done.

Writing objects: 100% (6/6), 488 bytes | 0 bytes/s, done.

Total 6 (delta 0), reused 0 (delta 0)

To https://github.com/sometestuser2/temp.git

\* [new branch] master -> master

אם נכנס לממשק של github נראה שהוא התעדכן ומכיל את השינויים. אם עבדתם לפי המדריך הקודם, יהיה שם רק קובץ אחד של PHP. עכשיו כל אחד, בכל העולם, יכול למשוך את השינויים (לא לדחוף, את זה רק אתם יכולים). איך מושכים את השינויים? פתחו תיקיה זמנית במחשב. את המשיכה עושים באמצעות clone:

$ git clone https://github.com/sometestuser2/temp.gitCloning into 'temp'...

remote: Counting objects: 6, done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0

Unpacking objects: 100% (6/6), done.

Checking connectivity... done.

כל מה שפקודת clone מקבלת הוא ה-URL של השרת המרוחק. לכל פרויקט בגיטהאב יש אחד כזה וגם לכם. מה הוא? קיבלנו אותו בעת היצירה ואפילו הגדרנו אותו כבר. ה-URL הוא https://github.com/sometestuser2/temp.git וכאשר אנחנו עושים clone, גיט מושך את הקבצים, פורס אותם וגם מקשר עבורנו את ה-remote origin לאותו URL. אם יש לנו את ההרשאות אנחנו יכולים להתחיל לעבוד, לעשות שינויים מקומיים כ-commits ורק כשאנחנו מוכנים או יכולים לדחוף אותם למקור.

זה יכול קצת לבלבל בתחילה – הפיצול הזה בין המקומי למרוחק. בניגוד ל-SVN או למערכות אחרות שבהן יש עבודה רק מול השרת. פה יש לנו עבודה שהיא בעצם מול המקומי (ה-commits) ומול השרת המרוחק (push). זה נראה קצת ביג’רס לנהל את זה, אבל מהר מאוד זה הופך לטבע שני. במאמר הבא נדבר על branches ונמשיך לחקור את הכוח הגדול בעבודה המפוצלת הזו.