API

הגדרה: אוסף של פונקציות שעוזרות לנו לתקשר בין מערכות תוכנה למיניהם.

API בעולם הHTTP זה מה שמעניין אותנו. באינטרנט- Web API. הגירסה הכי נפוצה של HTTP נקראת 1.1

יש כמה HTTP verbs: get, post, put, patch, delete, options.

כיום, הרוב עובד על REST.

PUT מחליף את כל האובייקט. PATCH מתקן רק שדה קטן.

גם אנחנו כנראה נעשה את הפרויקט שלנו בתור API. לכן צריך שזה יהיה הכי ברור וקל שיש.

צריך לשמור על קונסיסטנטיות: אם החלטנו להכניס query parameters, נעשה כך תמיד בפרויקט. בד"כ את הישויות נעשה סתם עם סלשים, והפרמטרים נעשה ב-query parameters.

פרוטוקול HTTP הוא לא מאובטח. עדיף לא להשתמש בו. HTTPS הוא טוב.

כך זה נראה: <https://www.mysite.com/api/v1/ball?color=green,blue&size=big>.

משפחות של סטטוסים:

2XX- הצליח. 3XX- היה משהו, כבר לא פה, הלך למקום אחר. 4XX- שגיאה של הקליינט. 5XX- שגיאה של הסרבר. של הAPI. לא אשמתנו.

אם יש בAPI שאנחנו יוצרות שגיאה של 500 והלאה, נחזיר את השגיאה, אבל לא את המסלול של השגיאה ((stack trace. זו בעיה חמורה באבטחה.

צריך הגנה לAPI שלנו. הגישה הכי קלילה אבל יחסית לא מאד בטוחה, פשוט להוסיף ל- query parameters בסוף את ה- apiKey. עוד שיטה שקיימת: JWT. זה איזשהו טוקן, קומבינציה של אותיות וספרות מעורבבות, במקום שבכל קריאה צריך להוסיף את הAPI-KEY , יש למשתמש טוקן של מחרוזת מאד ארוכה, ואז בheader מדביקים את כל הtoken .

דוקומנטציה, הסברים לכל דבר. מאד קל לעשות את זה גם אנחנו. C# swagger

איך בודקים API שלנו: דרך אחת, לבדוק דרך הקוד שלנו. עוד דרך אפשרית זה CURL. והדרך הכי נוחה זה Postman.

Git

מתקינים.

עומדים על שולחן העבודה ולוחצים לחצן ימני, gitbush here.

כותבים שם: git config --global user.name 'malki'

git config --global user.email 'malka789223@gmail.com'

כל פרוייקט מתחילים repository חדש.

הולכים לאיפה שהפרוייקט נמצא וכותבים git init.

עושים איזשהו שינוי, git add, git commit

יש דבר כזה branches. הראשי זה master, ואפשר לפתוח branch ואח"כ למזג עם הראשי- merge.

בסוף עושים push

לבדוק גירסה: git --version

להכנס לתיקייה של הפרוייקט שלי: עם cd או לפתוח מחדש את הcmd בתיקייה הרצויה.

הפקודה git init יוצרת repository בתוך התיקייה שבה אנחנו נמצאים כרגע.

בשביל לראות מה המצב שלנו git status

בכל פעם בהוספת קובץ, כדי שה- git יכיר גם אותו (ברור שבאותה תיקייה), עושים git add ..

Git commit –m "message" בודק מה השתנה, ונותן לזה שם.

Git log –oneline מביא איזה קומיטים עשינו.

Branch חדש: git checkout -b myFirstBranch

Git branch מביא לנו איפה אנחנו נמצאים בדיוק.

כדי לחזור ל-master: git checkout master.

מה-master נמזג: git merge myFirstBranch.

אחרי מיזוג בדרך כלל אנחנו לא חוזרים לבראנץ' שמיזגנו.

לחזור לcommit קודם: נציג את כל הקומיטים, ואז נראה בתחילת כל שורה קוד,

Git checkout 'code'

כדאי לעשות את הקובץ ReadMe

Git ignore

License זה תנאי השימוש.

כדי להשתמש בספריה אחרת בגיטהב, נלחץ fork

להעתיק פרוייקט בגיט למחשב שלי:

Clone, מעתיקים את הכתובת, git clone ואז להדביק את הכתובת.

אבל זה רק בפעם הראשונה. כל יום לפני שמתחילים לעבוד עושים git pull.

Clarifai