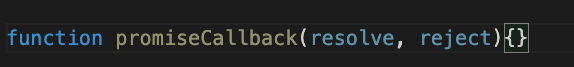
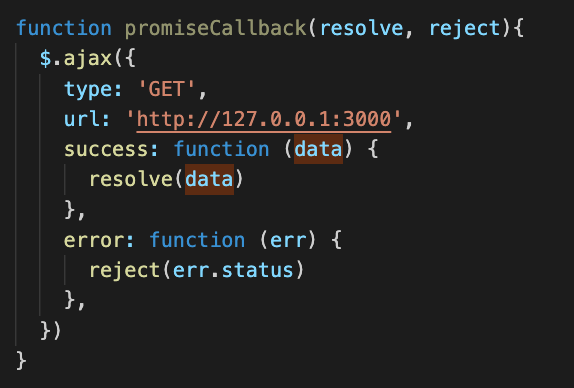
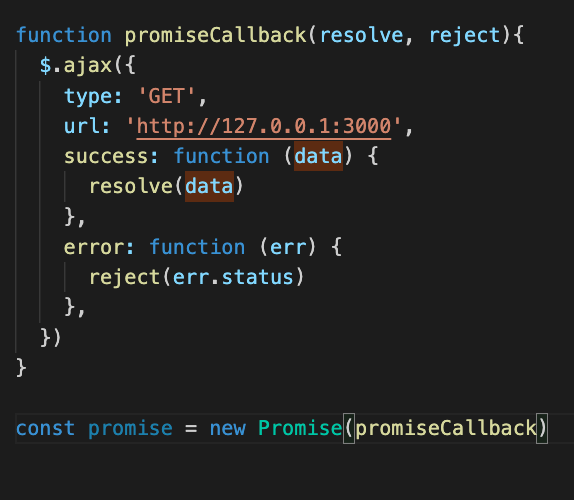
כאשר נרצה לבצע פעולה אסינכרונית ב- js על ידי promise נצטרך לבצע את הפעולות הבאות:

* ניצור פונקציית callback, נקרא לה promise callback. לפונקציה שני פרמטרים ששניהם יקבלו פונקציות callback. לפרמטר הראשון נהוג לקרוא resolve ולפרמטר השני נהוג לקרוא reject.   
  כך נראית חתימת ה – promise callback:  
    
   
* תפקידה של ה – promise callback היא להריץ את הפעולה האסינכרונית. מאחר שלפעולות אסינכרוניות יש פונקציית callback )נקרא לה ,(task callback שמורצת עם חזרת המידע, ה- promise callback תתערב בפעולת ה- task callback באופן הבא:  
    
  במקרה של successful response, ה- promise callback תקרא מה- task callback ל- resolve כך שה- successful response ישלח כארגומנט לפונקציית .resolve  
    
  במקרה של error response, ה- promise callback תקרא מה- task callback ל- reject כך שה- error response ישלח כארגומנט לפונקציית .reject  
    
  

אך מי הן resolve ו- reject ומה הן מבצעות?

* האובייקט Promise הוא זה שאחראי להגדיר את reject ו- resolve ולהריץ את ה- promise callback. לשם כך נצטרך ליצור מופע של האובייקט שיקבל לפונקציית ה- constructor שלו את ה- promise callback כך שעם יצירת המופע הוא יריץ את ה- promise callback.  
  
* לאובייקט promise יש שתי מתודות חשובות, then ו- catch.
* then מקבלת פונקציית callback, נקרא לה then callback, ל- then callback יש פרמטר שמקבל את ה- response מה- task callback במקרה של successful response. על הפרמטר הזה ה- then callback תבצע פעולות.
* catch מקבלת פונקציית callback, נקרא לה catch callback, ל- catch callback יש פרמטר שמקבל את ה- response מה- task callback במקרה של error response. על הפרמטר הזה ה- catch callback תבצע פעולות.
* נריץ את then ו- catch אחרי יצירת ה- promise. בהרצתן, המתודה then מגדירה את ה- then callback להיות זו שתתבצע בהגיע successful response, והמתודה catch מגדירה את ה- catch callback להיות זו שתתבצע בהגיע error response.  
    
  

אך מי אחראי לשלוח את ה- responses ל- then callback ול- catch callback?

* ה- resolve שנקראת מתוך ה- task callback במקרה של successful responseהיא זו שאחראית להעביר את ה- successful response ל- then callback ולאחר מכן להריץ את ה- then callback.
* ה- reject שנקראת מתוך ה- task callback במקרה של error responseהיא זו שאחראית להעביר את ה- error response ל- catch callback ולאחר מכן להריץ את ה- catch callback.

שאלה:

הרי עם יצירת המופע של ה- promise, האובייקט מריץ את ה- promise callback (שורה 14) שכזכור מריץ את הפעולה האסינכרונית.   
אם כן, איך אנו לא חוששים שפעולת ה- task callback תסתיים לפני שהרצנו את ה- then וה- catch (שורות 26- 27)?

תשובה:  
ה- task callback לא תתבצע רק אחרי שה- call stack יתרוקן וכל עוד יש עוד פעולות בקובץ וביניהן הרצת ה- then וה- catch, ה- task callback לא תרוץ.

שיפורים:

נהוג ליצור פונקציה שתחזיר אובייקט promise ולשרשר לקריאה שלה (למופע ה- promise שהיא מחזירה) את הרצות ה- then וה- catch.

מכיוון ש- then ו- catch מחזירות את מופע ה- promise הן ניתנות לשרשור.  
  
