9 פרויקט: שרת אינטרנט משופר

הגיע הזמן לשפר את שרת האינטרנט כך שהוא ישרת קבצים אמיתיים!

אנחנו הולכים לגרום לכך שכאשר לקוח אינטרנט (במקרה הזה נשתמש בדפדפן) מבקש קובץ ספציפי, שרת האינטרנט יחזיר את הקובץ הזה.

יש כמה פרטים מעניינים שנגלה בדרך.

9.1 מגבלות

כדי להבין טוב יותר את ממשק ה-sockets ברמה נמוכה יותר, בפרויקט זה אסור להשתמש באף אחת מפונקציות העזר הבאות:

הפונקציה socket.create_connection()
הפונקציה socket.create_server()
כל דבר במודולים urllib
לאחר כתיבת הקוד לפרויקט, יהיה יותר ברור איך פונקציות העזר האלה מיושמות.

9.2 התהליך

אם תלך לדפדפן שלך ותזין URL כמו זה (תחליף את מספר הפורט של השרת הפועל שלך):

http://localhost:33490/file1.txt
הלקוח ישלח בקשה לשרת שלך שנראית כך:

```
GET /file1.txt HTTP/1.1
Host: localhost
Connection: close
```

שים לב ששם הקובץ נמצא ממש שם בבקשת ה-GET בשורה הראשונה!

השרת שלך:

ינתח את כותרת הבקשה הזו כדי לקבל את שם הקובץ.
יסיר את הנתיב מסיבות אבטחה.
יקרא את הנתונים מהקובץ המבוקש.
יקבע את סוג הנתונים בקובץ, HTML או טקסט.
יבנה חבילת תגובת HTTP עם נתוני הקובץ במטען.
ישלח את תגובת ה-HTTP בחזרה ללקוח.
התגובה תיראה כמו קובץ הדוגמה הזה:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 357
Connection: close

<DOCtype html!>

<html>
```

<head>

[שאר קובץ ה-HTML קוצר בדוגמה זו.]

בנקודה זו, הדפדפן אמור להציג את הקובץ.

שים לב לכמה דברים בכותרת שצריך לחשב: ה-Content-Type ייקבע בהתאם לסוג הנתונים בקובץ המשורת, וה-Content-Length ייקבע לאורך בבתים של אותם נתונים.

אנחנו רוצים להיות מסוגלים להציג לפחות שני סוגים שונים של קבצים: קבצי HTML וקבצי טקסט.

9.3 ניתוח כותרת הבקשה
תרצה לקרוא את כל כותרת הבקשה, אז כנראה תעשה משהו כמו צבירת נתונים מכל ה-recv() שלך במשתנה יחיד וחיפוש בו (עם משהו כמו מתודת .find() של מחרוזת כדי למצוא את "\r\n\r\n" שמסמן את סוף הכותרת.

בנקודה זו, אתה יכול להשתמש ב-.split() על נתוני הכותרת על "\r\n" כדי לקבל שורות בודדות.

השורה הראשונה היא שורת ה-GET.

אתה יכול להשתמש ב-.split() על אותה שורה בודדת לשלושת חלקיה: שיטת הבקשה (GET), הנתיב (למשל /file1.txt), והפרוטוקול (HTTP/1.1).

אל תשכח להשתמש ב-.decode("ISO-8859-1") על השורה הראשונה של הבקשה כדי שתוכל להשתמש בה כמחרוזת.

אנחנו צריכים באמת רק את הנתיב.

9.4 הסרת הנתיב עד לשם הקובץ
סיכון אבטחה! אם לא נסיר את הנתיב, תוקף זדוני יכול להשתמש בו כדי לגשת לקבצים שרירותיים במערכת שלך. האם אתה יכול לחשוב איך הם יכולים לבנות URL שקורא את /etc/password?

שרתי אינטרנט אמיתיים פשוט בודקים כדי לוודא שהנתיב מוגבל להיררכיית ספריות מסוימת, אבל אנחנו ניקח את הדרך הקלה ופשוט נסיר את כל מידע הנתיב ונשרת קבצים רק מהספרייה שבה שרת האינטרנט רץ.

הנתיב יהיה מורכב משמות ספריות מופרדים בלוכסן (/), אז הדבר הכי קל לעשות בנקודה זו הוא להשתמש ב-.split('/') על הנתיב ושם הקובץ שלך, ואז להסתכל על האלמנט האחרון.

```
fullpath = "/foo/bar/baz.txt"

file_name = fullpath.split("/")[-1]
```

דרך יותר ניידת היא להשתמש בפונקציית הספרייה הסטנדרטית os.path.split. הערך שמוחזר על ידי os.path.split יהיה טאפל עם שני אלמנטים, כשהשני הוא שם הקובץ:

```
fullpath = "/foo/bar/baz.txt"
```

(os.path.split(fullpath
('foo/bar', 'baz.txt/') <=

בחר את האלמנט האחרון:

"fullpath = "/foo/bar/baz.txt

[file_name = os.path.split(fullpath)[-1

השתמש בזה כדי לקבל את שם הקובץ שאתה רוצה לשרת.

9.5 MIME וקבלת Content-Type
ב-HTTP, המטען יכול להיות כל דבר – כל אוסף של בתים. אז איך הדפדפן יודע איך להציג אותו?

התשובה נמצאת בכותרת Content-Type, שנותנת את סוג ה-MIME של הנתונים. זה מספיק כדי שהלקוח ידע איך להציג אותו.

כמה סוגי MIME לדוגמה:

סוג MIME	תיאור
text/plain	קובץ טקסט רגיל
text/html	קובץ HTML
application/pdf	קובץ PDF
image/jpeg	תמונת JPEG
image/gif	תמונת GIF
application/octet-stream	נתונים כלליים לא מסווגים

יש הרבה סוגי MIME לזיהוי כל סוג של נתונים.

אתה שם אותם ישר בתגובת ה-HTTP בכותרת Content-Type:

Content-Type: application/pdf
אבל איך אתה יודע איזה סוג של נתונים קובץ מכיל?

הדרך הקלאסית לעשות זאת היא על ידי הסתכלות על סיומת הקובץ, כל מה שאחרי הנקודה האחרונה בשם הקובץ.

למרבה המזל, os.path.splitext() נותן לנו דרך קלה למשוך את הסיומת משם קובץ:

('os.path.splitext('keyboardcat.gif

מחזיר טאפל שמכיל:

('keyboardcat', '.gif')

אתה יכול פשוט למפות את הסיומות הבאות למשימה זו:

סיומת	סוג MIME
txt	text/plain.
html	text/html.

אז אם לקובץ יש סיומת .txt, הקפד לשלוח בחזרה:

Content-Type: text/plain
בתגובה שלך.

אם אתה באמת רוצה להיות נכון, הוסף charset לכותרת שלך כדי לציין את קידוד התווים:

Content-Type: text/plain; charset=iso-8859-1
אבל זה לא הכרחי, מכיוון שדפדפנים בדרך כלל משתמשים בקידוד זה כברירת מחדל.

9.6 קריאת הקובץ, Content-Length, וטיפול במצב לא נמצא
הנה קצת קוד לקריאת קובץ שלם ובדיקת שגיאות:

```
try:
    with open(filename, "rb") as fp:
        data = fp.read()   # קרא קובץ שלם
        return data

except:
    הקובץ לא נמצא או שגיאה אחרת
    TODO שלח 404
```

הנתונים שאתה מקבל בחזרה מ-read() הם מה שיהיה המטען. השתמש ב-len() כדי לחשב את מספר הבתים.

מספר הבתים יישלח בחזרה בכותרת Content-Length, כך:

Content-Length: 357
(עם מספר הבתים של הקובץ שלך).

אתה אולי תוהה מה ה-"rb" הזה בקריאת open(). זה גורם לקובץ להיפתח לקריאה במצב בינארי. בפייתון, קובץ שנפתח לקריאה במצב בינארי יחזיר מחרוזת בתים שמייצגת את הקובץ שאתה יכול לשלוח ישירות על ה-socket.

מה לגבי ה-Not Found 404 הזה? זה מספיק נפוץ שכנראה ראית את זה בשימוש רגיל באינטרנט מדי פעם.

זה פשוט אומר שביקשת קובץ או משאב אחר שלא קיים.

במקרה שלנו, נזהה איזשהי שגיאת פתיחת קובץ (עם בלוק ה-except, למעלה) ונחזיר תגובת 404.

תגובת ה-404 היא תגובת HTTP, אבל במקום

HTTP/1.1 200 OK
התגובה שלנו תתחיל עם

HTTP/1.1 404 Not Found
אז כשאתה מנסה לפתוח את הקובץ וזה נכשל, אתה פשוט תחזיר את הדברים הבאים (מילה במילה) ותסגור את החיבור:

```
HTTP/1.1 404 Not Found
Content-Type: text/plain
Content-Length: 13
Connection: close

not found 404
```

(גם אורך התוכן וגם המטען יכולים פשוט להיות מקודדים קשיח במקרה הזה, אבל כמובן צריכים להיות encode.() לבתים.)

9.7 הרחבות
אלה נמצאות כאן אם יש לך זמן לתת לעצמך את האתגר הנוסף להבנה טובה יותר של החומר. דחוף את עצמך!

הוסף תמיכה ב-MIME לסוגי קבצים אחרים כך שתוכל לשרת JPEGs וקבצים אחרים.

הוסף תמיכה להצגת רשימת ספריות. אם המשתמש לא מציין קובץ ב-URL, הצג רשימת ספריות שבה כל שם קובץ הוא קישור לאותו קובץ.

רמז: os.listdir ו-os.path.join()

במקום פשוט להוריד את כל הנתיב, אפשר שירות מתוך ספריות משנה מספריית בסיס שאתה מציין בשרת.

סיכון אבטחה! וודא שהמשתמש לא יכול לפרוץ מספריית הבסיס על ידי שימוש בהמון .. בנתיב!

בדרך כלל היה לך איזשהו משתנה הגדרות שמציין את ספריית הבסיס של השרת כנתיב מוחלט. אבל אם אתה באחד מהקורסים שלי, זה היה הופך את חיי לקשים כשהייתי הולך לתת ציונים לפרויקטים. אז אם זה המקרה, אנא השתמש בנתיב יחסי לספריית הבסיס של השרת שלך וצור נתיב מלא עם הפונקציה os.path.abspath().

```
server_root = os.path.abspath('.')
server_root = os.path.abspath('./root')
```