בס"ד

151026 הנדסת תכנה לאינטרנט – תשע"ט

תרגיל ראשון להגשה (הגשה בקובץ Word):

1. לקרוא את פרטי פרוטוקול http כמופיע בלינקים:  
   <http://www.httpwatch.com/httpgallery/>, <http://www.tutorialspoint.com/http/>

<http://www.w3schools.com/tags/ref_httpmethods.asp>

<https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol>  
ולענות על השאלות הבאות:

1. מהם Http Methods ? פרט

פירוט בקשת HTTP, ברמת מה מצפה הלקוח מהשרת להביא לו. ישנם 8 מתודות שונות כאשר כל אחת מהן תגיב אחרת. לפעמים מדובר בבקשת מידע קיים, ולפעמים מהעברת מידע מהלקוח ובקשת מענה מותאם למידע.

2) מה ההבדל בין Get ל- Post?

GET – בקשת מידע ישירות מהשרת.

POST – שליחה של מידע עם הבקשה שגורם לתופעות לוואי (למשל הודעה לפורום, פוסט בפייסבוק) שבתגובה השרת יחזיר את המידע ויטפל במידע המועבר לו על פי הדרוש.

3) מהם Headers? תן דוגמאות

צמדים של שם/ערך העוברים בבקשות/תשובות השונות.

גרסת HTTP – HTTP/1.1

שפה מועדפת על ידי הלקוח -

Accept-Language: en-gb

סוג החיבור (Persistent/non-persistent)

Connection: Keep-Alive

4) מהי Cookie ומדוע נחוץ מנגנון זה?

פרוטוקול HTTP הוא פרוטוקול שלא שומר מידע (Stateless) ולכן אנחנו צריכים מנגנון שישמור לנו מידע בזמן שאנחנו נמצאים באתר מסוים. הדרישה המינימלית היא למשל פרטי רישום, שללא ה"קוקית" (על פי האקדמיה ללשון) לא יישמר בכל רענון של הדף.

בזמן שאנחנו נכנסים לאתר שמשתמש בקוקיז הוא לוקח קובץ שכזה ושומר אותו על המחשב שלנו עם על הפרטים הרלוונטיים לזיהוי על ידי האתר, וכן מידע חשוב נוסף (סל קניות, רכישות קודמות וכו').

1. נא לקרוא את תיאור תבנית REST בלינק הבא:  
   http://www.infoq.com/articles/rest-introduction  
   וענה על השאלות :  
   1) מהי Rest?

סט של עקרונות וסטנדרטים לשימוש נכון ברשת, באופן שיביא את היכולות ליעילות מיטבית

2) תאר את מאפייני תבנית זו

תבנית REST מכילה 5 עקרונות עיקריים-

1. נתינת ID מסוים לכל "דבר"/אובייקט- איגוד של אובייקטים שונים ("קונים", "מוכרים", "משתמשים" וכו)והפניה דרך שם האובייקט עצמו.
2. קישור אובייקטים – ברגע שיש אובייקטים שמחוברים אחד לשני, כמו שם ותמונה, יש לקשר אותם אחד לשני באפן ממשי כךשהדבר יחסוך לנו בבקשות.
3. שימוש במתודות סטנדרטיות – קישור של GET,POST וכדו לכל אובייקט במערכת
4. מקורות רב-ייצוגיים – שימוש במקורות אחידים ומוכרים הן לשרת והן ללקוח
5. תקשורת Stateless – מצב כזה מונע תלות בין הלקוח לשרת, הן מצד השרת שצריך להתאים את עצמו לכל לקוח חדש, והן מצד הלקוח שעלול לגרור שינויים אם השרת פתאום ישנה צורת תקשורת.
6. נא לקרוא אודות חוקי המבנה של מסמכי XML בלינק הבא:

<http://www.w3schools.com/xml/xml_syntax.asp>

וענה על השאלות:

1. מהם חוקי המבנה למסמך XML?

היררכיה – ככל שיורדים בהיררכיית המידע, כך התגיות יתחילו בהזחה פנימה.

סגירת תגים – כל אלמנט חייב להופיע עם תג סוגר.

רגישות תוים – התגים חייבים להופיע באותו מבנהאותיות.

קינון תגים – צריך לעבוד בדומה לסוגריים.

מאפיינים בתג חייבים להופיע במרכאות

תוים מיוחדים – צריכים להופיע עם שיום אחר על מנת למנוע אי הבנות והתנגשויות.

1. מלבד חוקי המבנה, האם ישנם עוד חוקים שאפשר להחיל על מסמך XML?

???

1. בנה אתר הכולל דף ראשי ובו רשימה של 5 פרחים, כל פריט מהרשימה משמש כלינק לדף המתאר אותו בהרחבה. בכל דף תיאור תופיע טבלה ובא תמונת הפרח ופרטים אודות מקום הגידול המאפיין אותו וכן לינק המאפשר חזרה לדף הראשי
2. בנה Form הקולט שם וסיסמא של אחד משלושה אנשים ומחזיר תשובה האם הגישה מאושרת או שאין גישה
3. מה עושה הקוד הבא? הסבר את התנהגותו. נ.ב. כדי להיעזר בלינקים:

<http://www.w3schools.com/jsref/default.asp>

<https://en.wikipedia.org/wiki/List_of_HTTP_header_fields>

<http://www.w3.org/Protocols/HTTP/Object_Headers.html>

var http = require('http');

var port = 3000;

var current = new Date();

var counter = 1;

http.createServer(function (req, res) {

current.setSeconds(current.getSeconds() + 10);

res.writeHead( 200, "ok", { 'Content-Type': 'text/plain',

'Expires': current.toGMTString() });

res.write('Hello World\n');

res.write("count=" + counter++);

res.end();

}).listen(port);

בכל בקשה דשה של HTTP מרענן את הזמן מאז הבקשה הראשונה ומוסיף ל-count

1. לכתוב ב-JS קוד הפעלת שרת http ל-NodeJS שמקבל את כל בקשות ה-http ומחזיר תשובה "Hello wolrd\n". להוסיף מספר שדות ב-Header כמופיע בדוגמא ולראות ב-F12 את השינויים בדומה לדוגמא כדלקמן. יש להגיש גם את הקוד וגם את צילום המסך המציג את התוצאה.

HTTP/1.1 200 OK

Server: MYNODE.JS

Date: Mon, 04 Jan 2015 12:04:43 GMT

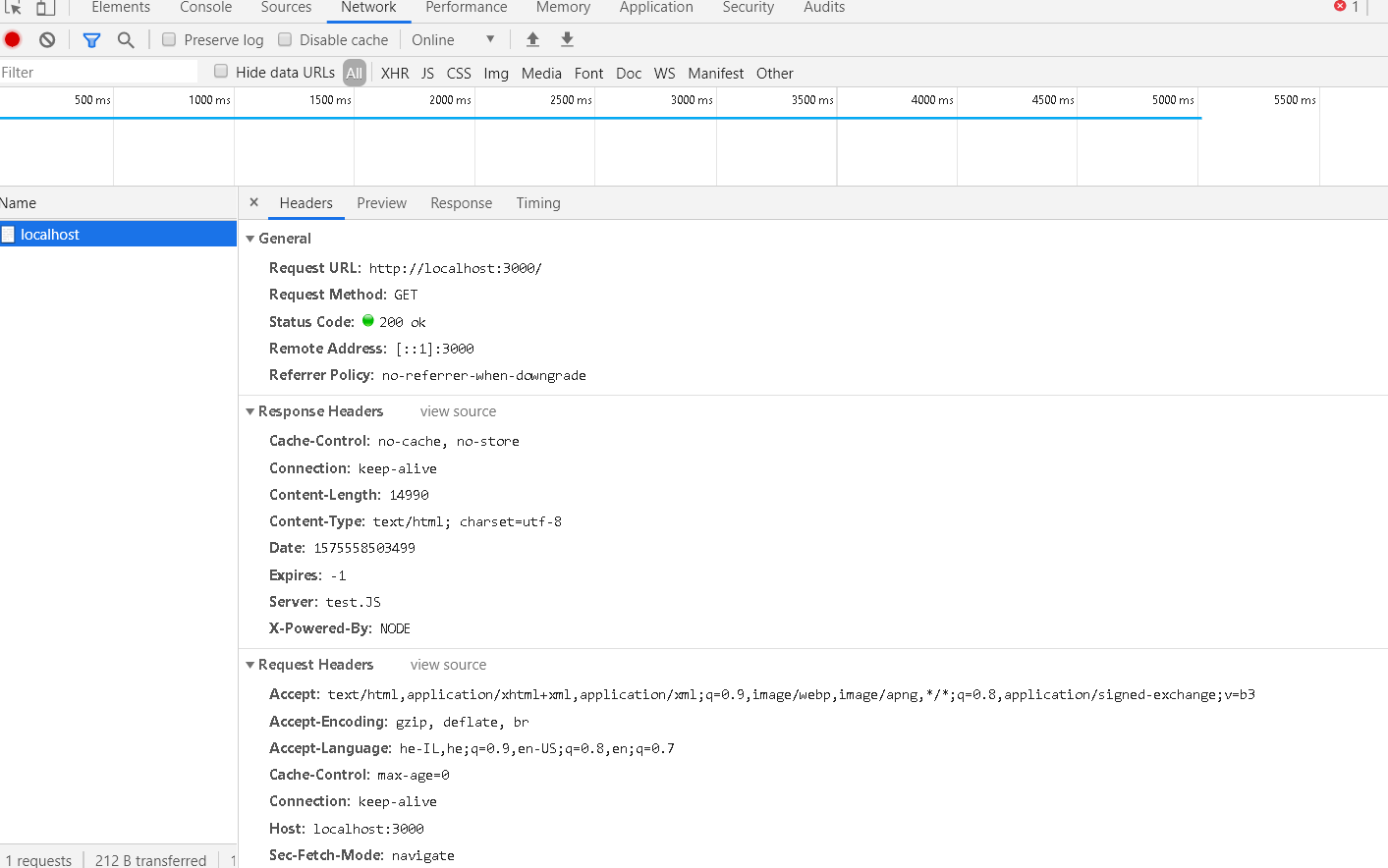
X-Powered-By: NODE

Cache-Control: no-cache, no-store

Expires: -1

Content-Type: text/html; charset=utf-8

Content-Length: 14990



1. להוסיף לקוד שנכתב לשאלה ג Cookie ולראות ב-F12 שהיא הגיעה ( הם מגישים Screen Capture של ה-tab המציג זאת). הדגמת הוספת cookie:

// To Write a Cookie

response.writeHead( 200, { 'Set-Cookie' : 'mycookie=test',

'Content-Type' : 'text/plain' });

כמה קישורים להכרת cookies:

<https://en.wikipedia.org/wiki/HTTP_cookie><https://www.nczonline.net/blog/2009/05/05/http-cookies-explained>