

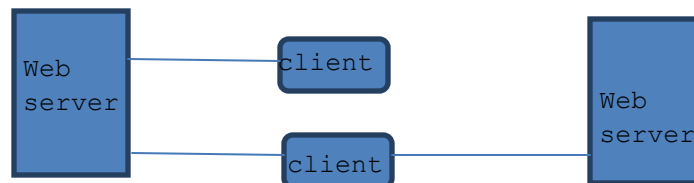
רשת האינטרנט החלה את דרכה בשנות ה-50. מטרת הרשת השיתופית הזו היתה בעיקר להקל על אנשי האקדמיה וכמו כן גם על המערכת הצבאית.

מערכת זו עברה שלושה שלבים עקריים בכדי להגיע למצבה הנוכחי, כאשר הקפיצה הגדולה ביותר נעשתה בתחילת שנות ה-90. באותה העת רשת האינטרנט הגדילה את יכולותיה, ולכן החלו אנשי עסקים רבים להשתמש בה. מאז ועד היום עברה הרשת שדרוגים רבים לגבי יכולתה להציג מידע מסוגים שונים כגון: תמונות, מוזיקה וכו'.

לאחר התחברות לאינטרנט, ניתן לגשת למידע הנמצא במחשבי השרתים. המידע מאורגן בצורה של אתר (Web Site). כל אתר מורכב מדפים (Pages) המקושרים ביניהם באמצעות קישורים (Links). בלחיצה על קישור ניתן לעבור בין דפים באותו אתר, ואף להגיע לדפים באתרים אחרים. לכל אתר יש כתובת (URL) שהיא למעשה הקישור לדף הראשי (דף הבית) של האתר.

מספר BUZZ WORDS:

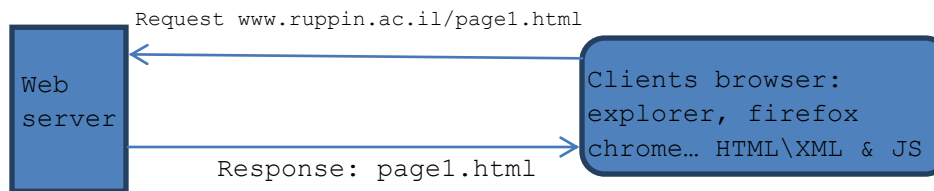
- Internet
- רשת תקשורת של העברת מידע בין nodes = צומת
- Web
- כל התוכן\מידע שניתן להגיע עליו דרך ה internet
- Html
- שפת הצגת מידע שאותה כל browser יודע להציג
- Web server
- מכיל מסמכי html, asp, pics, videos...



יש לו כתובת למשל www.ruppim.ac.il

- כתובת זו נקראת domain שיש לה התאמה חד-חד-ערכית למספר IP Address שבין 0.0.0.0 – 255.255.255.255 למשל 194.90.203.10
- DNS = Domain Name Server
- השרת שמתרגם בין ה domain לבין ה IP
- URL=Uniform Resource Locator=domain או URI= Uniform Resource Identifier
- URI מכיל ממש שם קובץ.
- http =hypertext transfer protocol
- פרוטוקול תקשורת של האינטרנט עבור דפי HTML
- ftp=file transfer protocol
- לשם העברת קבצים

שלב התפתחות ראשון – Document Retrieval System:



בשלב הזה ניתן רק להציג דפים סטטים שרק מציגים מידע וניתן ע"י LINK-ים לעבור בין הדפים.

התפקיד של HTML:

עיצוב הדפים ותוכנם נעשים באמצעות שפת HTML אותה למדנו בעבר. בהגדרתה HTML היא אוסף של פקודות/הוראות המשמשות להצגת מידע בדף אינטרנט. כל דפדפן אינטרנט יודע לפענח פקודות HTML ובכך להציג את המידע שנוצר ע"י יוצר הדף. השפה מורכבת מתגיות כאשר כל תגית הינה פקודה מסוימת. הסיומת של קבצים אלו הינם htm או html.

דפי HTML הינם סטאטיים!

להלן פירוט התהליך של בקשת דף אינטרנט סטאטי:

1. הלקוח (דפדפן) מבקש דף HTML כלשהו
2. השרת שולח את הדף המבוקש
3. הלקוח (דפדפן) מקבל את הדף ומציג אותו

מכאן ואילך השרת אינו יודע מה קורה במחשב הלקוח, עד שתתקבל בקשה חדשה. דף זה מוגבל מאוד מפני שהוא מציג משהו קבוע בכל קריאה.

שלב התפתחות שני - Web Forms:

<form>

<input>

בשלב הזה ניתן להזין נתונים ולעבד אותם בצד הלקוח וגם להגיב בהתאם ולבחור דף תגובה מתאים. שתי אפשרויות:

Get – מכיל את הפרמטרים בכתובת ה URI כאשר לפני הפרמטרים מופיע "?"

<http://search.yahoo.com/bin/search?p=computers>

<http://weather.walla.co.il/?w=1/402/w>


ניתן לעשות לזה BOOKMARK , מוגבל עד 1024 תווים, מידע חשוף לכל אחד

Post - מעביר את המידע בתוך המסמך, לא מוגבל בגודלו, לא חשוף
בכל מקרה אנחנו במצב בו אין זכרון לשרת לגבי העבר - Stateless

Forms allow us to send data to the server for processing

Please enter your name:

Enter address here:



```
<form action= "/folder/page.aspx" method = "POST" >  
<input type="text" name="Name" />  
<input type="text" name="Address" />  
<input type="submit" value="OK" />  
</form>
```

Define:

- **Form** – action & method
- **Inputs** – name, type & value

Form data can be submitted by HTTP GET or POST

- **GET** places data in the QueryString portion of the URL
 - Can be bookmarked
 - Limited in length

```
GET /folder/page.aspx?Name=Paddington&Address=Peru HTTP/1.0  
User-agent: Mozilla/4.0
```

- **POST** places data in the body of the HTTP Request
 - 'Hidden from view'
 - Unlimited length

```
POST /folder/page.aspx HTTP/1.0  
User-agent: Mozilla/4.0  
Content-type: application/x-www-form-urlencoded  
Content-length: 28  
Name=Paddington&Address=Peru
```

בשיעור הבא נראה דוגמה ונרחיב בנושא.

התפקיד של JavaScript:

שפות תסריט (Script) פותחו עבור ביצוע עיבוד בצד הלקוח ע"פ אירועים שונים שקורים (לדוגמה: לחיצה על לחצן). בכך קיבלו דפי האינטרנט מימד חצי דינאמי. מימד זה לא מספיק מכיוון שרוב המידע עדיין נמצא בשרת ויכולות העיבוד מצד הלקוח מוגבל. עיבוד זה משמש בעיקר לבדיקת תקינות טפסים או לביצוע שינויים במראה הדף.

שלב התפתחות שלישי - Web Application :

ASP = Active Server Pages

שיטה שבה ניתן לתכנת בשפות מקופלות כמו C# והמטודולוגיה של ה ASP מחוללת מזה את הקוד המתאים בשפת HTML ו- JAVASCRIPT אותם ה BROWSER יודע לקרוא!
[2310B_01A001.HTM]

התפקיד של ASP קלאסי (spaghetti code):

ASP – Active Server Pages – סביבת עבודה לבניית דפי אינטרנט דינאמיים. הסיומת של קבצים אלו הינה asp.

דפי ASP הינם דינאמיים!

להלן פירוט תהליך של בקשת דף אינטרנט דינאמי:

1. הלקוח (דפדפן) מבקש דף ASP כלשהו
2. השרת בוחן את הקובץ המבוקש ומבצע את הקוד שדף זה מכיל
3. השרת מתרגם את תוצאות העיבוד ל - HTML + JS (במידת הצורך) ושולח את הדף המעובד ללקוח
4. הלקוח (דפדפן) מקבל את הדף ומציג אותו

- חשוב להבין שדפדפן האינטרנט יודע לקרוא רק שפות כמו JS, CSS, HTML ולכן התוצאה שמקבל הדפדפן חייבת להיות בפורמטים אלו.

על מנת לבצע עיבוד זה חובה להשתמש בשרת מסוג IIS (Internet Information Server). שרת IIS יודע לעבד דפי asp ולהמירם ל HTML, CSS, JS. עיבוד זה הינו כבר בצד השרת, וכך בכל קריאה לדף יתקבל מידע אחר, שנדרש בהתאם לבקשת הלקוח. גם כאן השרת אינו יודע מה קורה במחשב הלקוח, עד לבקשה הבאה.

יתרונות השימוש ב - ASP:

1. אינטראקטיביות
2. אבטחה
3. התאמה אישית
4. פשטות

התפקיד של ASP.NET:

ASP.NET דומה ל - ASP קלאסית (בדינאמיות), ויתרונה בכך שהיא פעולת במודל מונחה אירועים. במודל זה השרת כל הזמן מאזין ללקוח ומגיב לכל קריאה לאירוע. מודל זה קושר את השרת והלקוח

יחדיו, וגורם לאפליקציית ה- WEB להתנהג כמעט כמו אפליקציית WINDOWS. הסיומת של קבצים אלו הינה asp.x.

כיצד זה מתבצע?

פקדי ASP.NET בנויים כך שכאשר הם מתורגמים ל- HTML (תהליך התרגום נקרא Rendering) הם מכניסים תסריט (script) בצד הלקוח לכל אירוע שהם מכילים, תסריט זה שולח לשרת מידע (שרלוונטי לאותו סוג של אירוע) בכל פעם שקורה אירוע. השרת תופס את האירוע ומבצע קוד שנכתב כתגובה לאירוע. כתיבת הקוד יכולה להתבצע בכל שפת תכנות אשר קיימת בסביבת העבודה .NET.

היתרון העיקרי של השימוש ב- ASP.NET (מעבר ליתרונות השימוש ב- ASP) הינו היכולת לנצל כמעט את כל רכיבי סביבת העבודה .NET. ע"מ ליצור דפי אינטרנט, החל מפונקציות פשוטות ועד לגישה למסדי נתונים באמצעות ADO.NET. כמובן של- ASP.NET עוד יתרונות רבים, שבשורה התחתונה מקצרים את זמן הפיתוח של אתרי אינטרנט.