**עקרונות וטיפים ב-ASP.NET**

**רשימת נושאים**

1. מבוא
2. WebSite, WebApplication
3. PostBack
4. IIS
5. Web.config
6. Tracing
7. Validation
8. Web User Control
9. Master Page
10. מבוא ל- State Management
11. View State
12. Query String
13. Cookies
14. Gloabal.asax
15. Session
16. Application
17. Caching
18. Authentication, Authorization
19. מושגים

**מבוא**

ASP.NET היא סביבת עבודה המאפשרת בניית דפי אינטרנט דינאמיים (בשונה מדפי HTML סטאטיים).

עובדת במודל מונחה אירועים- השרת כל הזמן מאזין ללקוח ומגיב לכל קריאה לאירוע .

סיומת הקבצים היא ASPX.

אופן הפעולה של בקשת דף אינטרנט דינאמי:

1. לקוח-> מבקש דף
2. שרת-> בדיקה של הקובץ המבוקש+ ביצוע קוד של הדף
3. שרת-> מתרגם את הדף ל- HTML+JS
4. שרת-> שולח את הדף המעובד ללקוח
5. לקוח-> מקבל את הדף

אופן הפעולה של המודל מונחה האירועים:

בכל פעם שקורה אירוע – השרת מקבל מידע מהצד לקוח על האירוע.

השרת מקבל את האירוע ומבצע קוד שנכתב בהתאם לאירוע.

ניתן להשתמש בכל שפת תכנות שבסביבת .NET

**WebSite, WebApplication**

**PostBack**

isPostBack==true - כאשר הפנייה נעשית בעקבות אירוע שקרה בדף.

isPostBack==false - כאשר הפנייה נעשית בעת טעינת הדף

עבור controls מסויימים יש צורך בהגדרת :  
AutoPostBack==true על מנת לפנות לשרת בעת שינוי ב-control

**IIS**

**Web.config**

* קובץ המאכסן הגדרות של האתר
* היתרון של הקובץ זה שלא צריך לקמפל את האתר בכדי לבצע שינוי
* ניתן להגדיר לאתר קובץ אחד כזה אבל לא חובה

<configuration>  
 <system.web>  
  
 </system.web>  
</configuration>

* הגדרות האתר יבואו בתוך תגיות ה-system.web:

**Tracing**

מנגנון  [מעקב שמאפשר לקבל מידע על בקשות מגיעות לדף לתהליכים שקורים בדף](https://www.google.co.il/search?q=%D7%9E%D7%A2%D7%A7%D7%91+%D7%A9%D7%9E%D7%90%D7%A4%D7%A9%D7%A8+%D7%9C%D7%A7%D7%91%D7%9C+%D7%9E%D7%99%D7%93%D7%A2+%D7%A2%D7%9C+%D7%91%D7%A7%D7%A9%D7%95%D7%AA+%D7%9E%D7%92%D7%99%D7%A2%D7%95%D7%AA+%D7%9C%D7%93%D7%A3+%D7%9C%D7%AA%D7%94%D7%9C%D7%99%D7%9B%D7%99%D7%9D+%D7%A9%D7%A7%D7%95%D7%A8%D7%99%D7%9D+%D7%91%D7%93%D7%A3&spell=1&sa=X&ved=0ahUKEwjvv8zI-53ZAhXJ16QKHVRRA5kQkeECCCMoAA)

מטרות המנגנון:

* זיהוי צווארי בקבוק
* זיהוי בעיות בבקשות שמגיעות
* מציאת טעויות בקוד בצד שרת

להמשיך...

**Validation**

סוגי קונטרולס:

|  |  |
| --- | --- |
| תפקיד | Control |
| ואלידציה לשדה חובה למלא | RequiredFieldValidator |
| ואלידציה לטווח ערכים | RangeValidator |
| ואלידציה להשוואה | CompareValidator |
| ואלידציה לבדיקה מול תבנית רגולר אקספריישן | RegularExpressionValidator |
| מאפשר ליצור ואלידציה לפי בחירה שלנו | CustomValidator |
| מאפשר להציג את כל השגיאות בדף במקום מרוכז | ValidationSummary |

אותה טבלה ,רק עם הערות

|  |  |  |
| --- | --- | --- |
| תפקיד | Control | הערות |
| ואלידציה לשדה חובה למלא | RequiredFieldValidator | אם רוצים לעשות ואלידציה שגם הקונטרול לא ריק- הכרחי |
| ואלידציה לטווח ערכים | RangeValidator | צריך להגדיר ערך עליון וערך תחתון וסוג הטווח(מספרי, תאריך, מחרוזות וכו') |
| ואלידציה להשוואה | CompareValidator | מאפיינים שצריך להגדיר:  1. ControlToCompare או ValueToCompare  2. Operator(compare operator)  3. Type(of compare) |
| ואלידציה לבדיקה מול תבנית רגולר אקספריישן | RegularExpressionValidator | צריך להשתמש במאפיין ValidationExpression |
| מאפשר ליצור ואלידציה לפי בחירה שלנו | CustomValidator | על מנת להשתמש בקונטרול יש להירשם לאירוע ServerValidate.  ניתן הגדיר בקוד אם הבדיקה הצליחה או לא באמצעות המאפיין IsValid |
| מאפשר להציג את כל השגיאות בדף במקום מרוכז | ValidationSummary | ההודעה שלו מוצגת בכל מקרה דרך ההודעה ErrorMessage |

**Web User Control**

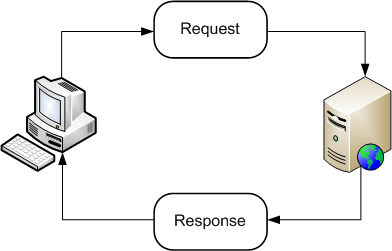
להוסיף...

**Master Page**

להוסיף..

**מבוא ל- State Management**

דיאגרמת Request, Response.



הבקשה והתגובה הן הודעות בפרוטוקול HTTP.

פרוטוקל HTTP הוא stateless- כלומר לאחר הגעת התגובה לדפדפן , הקשר בינו ובין השרת מתנתק. ניתן יהיה לראות את הדף שנמצאים בו גם אם ננתק את החיבור לאינטרנט כל זמן שלא שלחנו בקשה חדשה מהשרת.

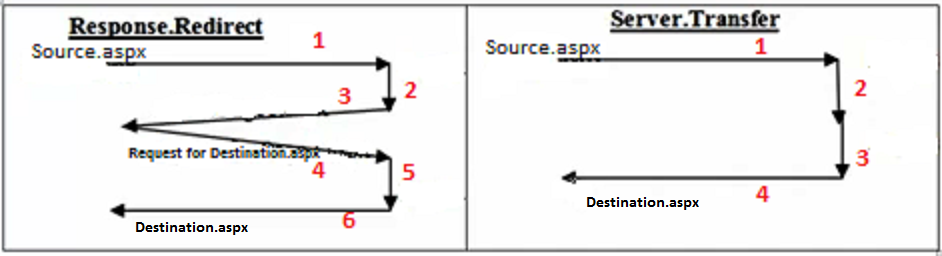
האובייקט Request:

מאפשר לשרת לקבל מידע על בקשה שהגיעה או טיפול בבקשה שהגיעה. דוגמאות: סוג הדפדפן, האם מאפשר JS, האם מאפשר קוקיס.

האובייקט Response:

מאפשר לשרת לשלוט בתגובה שתשלח לקליינט. דוגמא: העברת הקליינט לדפדפן אחר.

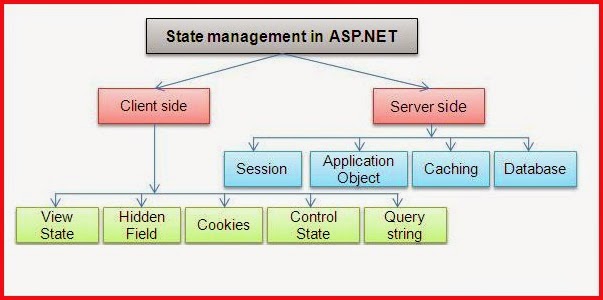
Response.Redirect- הסרבר שולח לקליינט הודעה שיש לעבור לדף אחר ולאחר מכן הקליינט עובר לדף אחר



**Response.Redirect execution**  
  
1 When the user clicks on the button a request will be sent to the server for the  
execution of button event code .  
2 Button click event code (containing Response.Redirect)  
will be executed by the server and the current page processing will  
be stopped  
3 Response will be rendered to the browser asking the browser  
to submit the request for Destination.aspx page  
4 The browser will request the Destination.aspx page  
5 The server processes the request for Destination.aspx page  
6 The server sends a response to the browser with the Destination.aspx page

1. בלחיצת CLICK, נשלחת ריקוסט לשרת
2. השרת מבצע את הטיפול באיבנט והטיפול בדף הנוכחי יעצר
3. השרת שולח רספונס לבראוסר עם בקשה לריקוסט עבור דף היעד שאליו צריך לעבור
4. הדפדפן שולח ריקוסט לדף היעד
5. השרת מעבד את הבקשה
6. השרת שולח רספונס עם דף היעד

**Server.Transfer execution**  
  
1 When the user clicks on the button a request will be sent to the server for the  
execution of button event code.  
2 Button click event code (containing Server.Transfer)  
will be executed by the server.  
3 The server will processes the Destination.aspx  
4 The output of the Destination.aspx will be rendered to the browser  
  
 **Other differences**  
  
**RR** -  Form variables of the source page are not accessible in destination page  
**ST** -  Form variables of the source page are accessible in destination page  
(by using the Server.Transfer overloaded method  containing   preserveForm property )  
  
**RR** - Execution is slow since there are two roundtrips and hence there load on server  
**ST** - Execution is fast since there is only one roundtrip and hence there is less load on the server  
  
**RR** - Refreshing of page does not cause any error.  
**ST** - Refreshing of page causes error.  
  
**RR** - RR changes the URL in the browser since browser is involved.  
**ST** - ST doesnt change the URL in the browser since browser is not involved.  
  
 **Uses**  
  
**RR** is to be used in operations like data insertion so that even  if the user refreshes the page then only confirmation will be rendered back ie executing steps 4,5 and 6.  
**ST** is to be used when the URL is to be hidden from the user.



**State Management in ASP.Net**

המטרה של ניהול הסטייט – היא לנהל את המידע בעובר רשת, איך לאכסן אותו ואיך להעביר אותו בתוך האפליקציה/האתר

**Hidden Fields**

מאפיינים:  
1. HTML קונטרול

2. ניתן לשמור בהם כל מידע שרוצים ולהחזיר אותו בכל שלב

3. ניתן לקבל גם את המידע ב-CODE BEHIND

4. חסרון אחד הוא שלא ניתן לנהל אובייקטים או קונטרולס

5. חסרון נוסף הוא שאין לזה אבטחה וניתן לקרוא את הערך שלו ב-סורס קוד של הדף

**View State(ברמת הדף)**

מאפיינים:

1. נשמר כל עוד המשתמש לא עבר לדף אחר ולא סגר את הדפדפן
2. נשמר בצד הלקוח ולכן חשוף

## Control State

מאפיינים:

בדיוק כמו VIEW STATE רק עבור קונטרולס

**Query String**

מאפיינים:

1. חשוף לשינויים מצד הלקוח
2. נשמר רק בין שני דפים (באופן אוטמטי)
3. מאפשר למשתמש לשמור את הכתובת כולל ה-QueryString כדי להגיע לדף באותו מצב בו המשתמש היה

**Cookies(ברמת הדיסקונקטד)**

קבצי טקסט הנשמרים על המחשב של הקליינט.

מאפיינים:

1. נשמרים גם לאחר שהמשתמש סגר את הדפדפן
2. חשופים למשתמש
3. המשתמש יכול למחוק אותם
4. המשתמש יכול לחסום אותם

לשמירת קוקיס משתמשים באובייקט רספונס

לטעינה קוקיס משתמשים באובייקט רקוסט

למחיקת קוקיס משתמשים בטריק של שמירה בזמן שכבר עבר(תוך שימוש באובייקט רספונס)

**Gloabal.asax(ברמת האפליקציה)**

קובץ המאפשר לכתוב קוד לכל מיני אירועים כללים הקורים באתר.

יכול להיות רק קובץ אחד כזה

אירועים אפשריים:

1. עליית האתר
2. הורדת האתר/עדכון האתר
3. שגיאה באתר
4. תחילת סשן
5. סיום סשן (בתאם להגדרת- sessionstate mode==InProc בקובץ web.config)

**Session(ברמת הדפדפן)**

אובייקט שנוצר בשרת עבור כל קליינט(מופע של דפדפן) שפונה לשרת.

SessionID:

* מפתח ייחודי שנועד לזהות את הדפדפן שפונה לשרת
* נוצר בפניה הראשונה של הדפדפן לשרת
* נשמר בקוקי זמני עד לסגירת הדפדפן
* מטרתו – זיהוי לאיזה דפדפן ה- Session שייך
* נחסם במידה ונחסמו קוקיס בדפדפן

מאפיינים של Session:

1. נשמר בשרת ולכן יותר מאובטח
2. נוצר עבור כל משתמש ולכן יכול לשמור מידע הזמין לכל הדפים פר משתמש
3. יכול לשמור כל סוג של אובייקט
4. נשמר בזיכרון השרת ולכן עלול להעמיס

**Application**

להוסיף..

קישור   
[קישור CODE PROJECT CLIENT SIDE](https://www.codeproject.com/Articles/23919/Client-Side-State-Management-Objects)

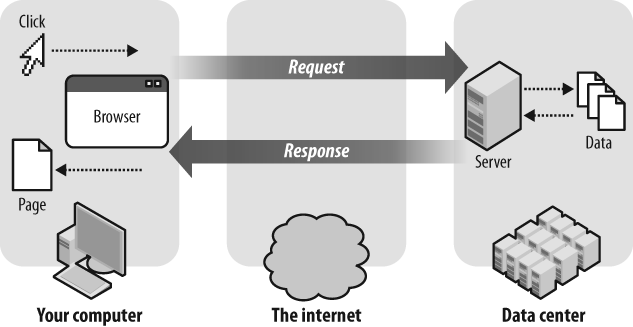
**מושגים**

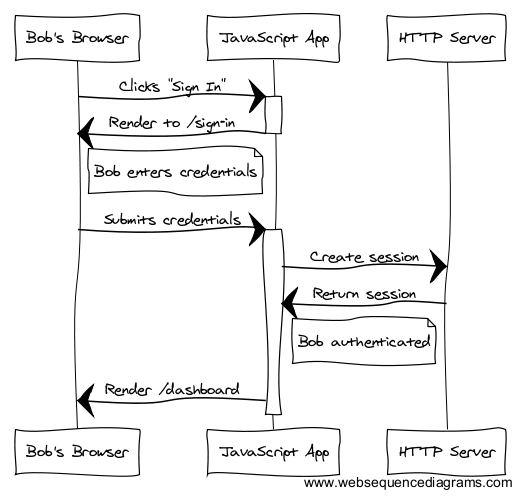
ASP- Active Server Pages

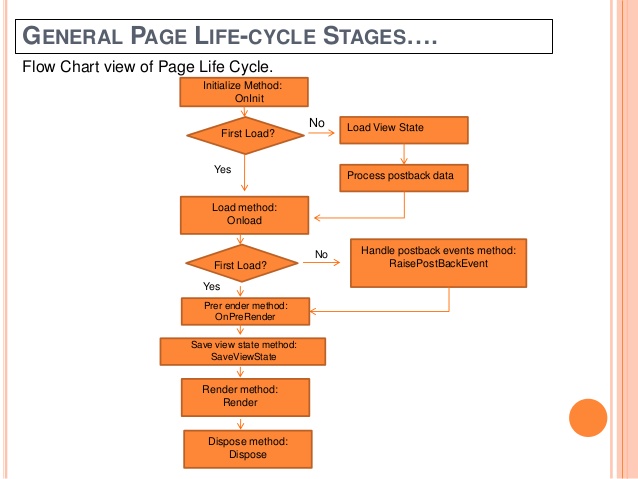
IIS- Internet Information Server

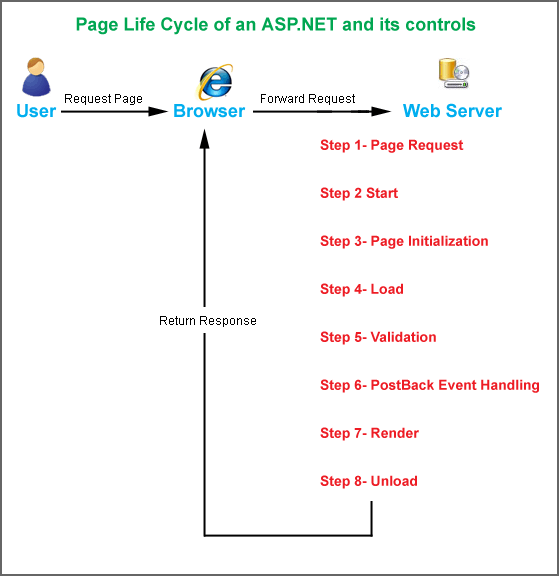
Rendering – Translate ASP.NET controls to HTML,JS

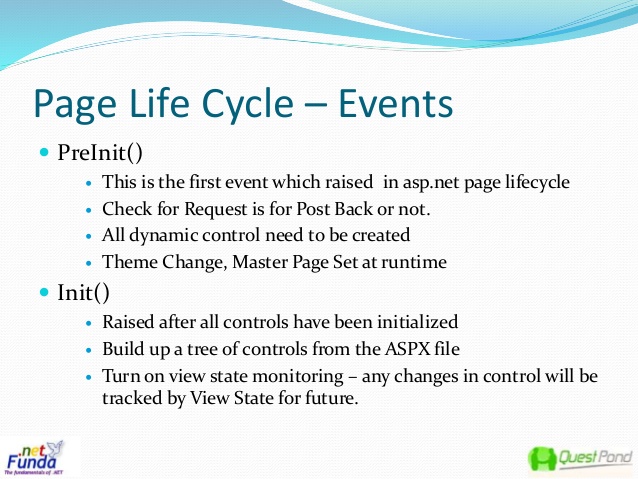
### HTTP- [Hypertext Transfer Protocol](https://he.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

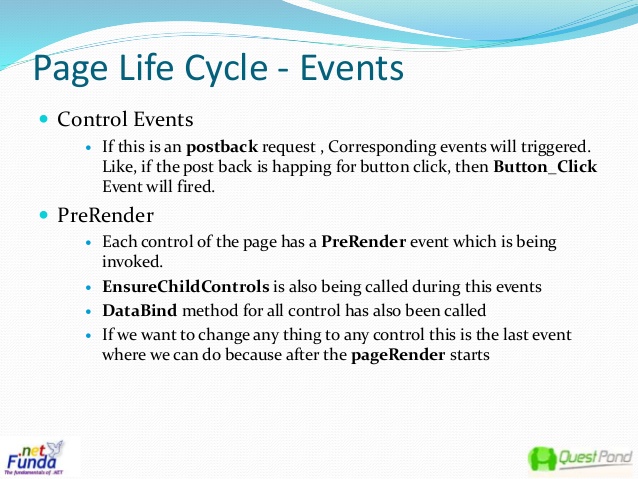
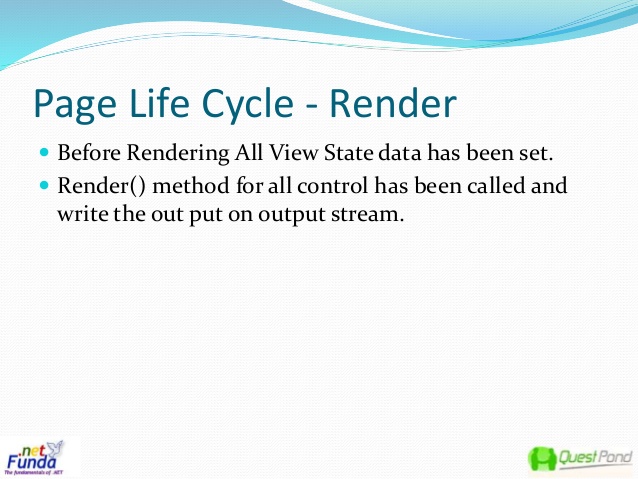


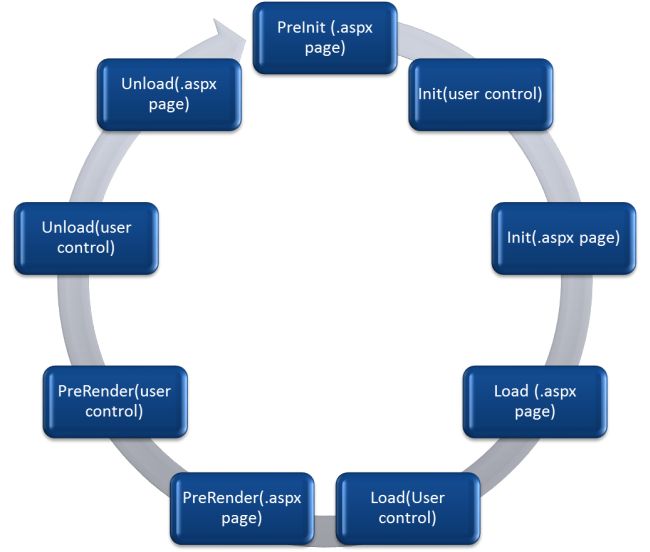


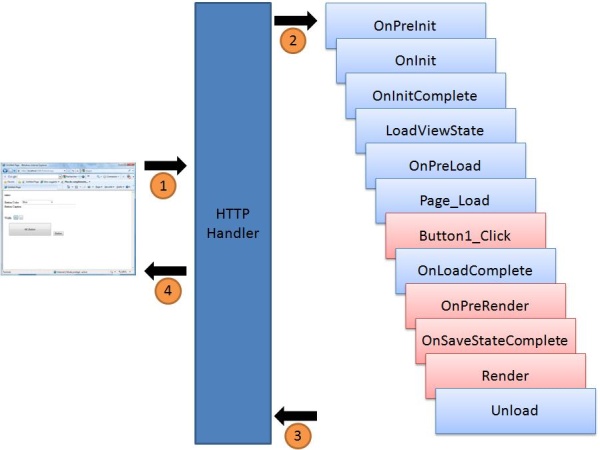


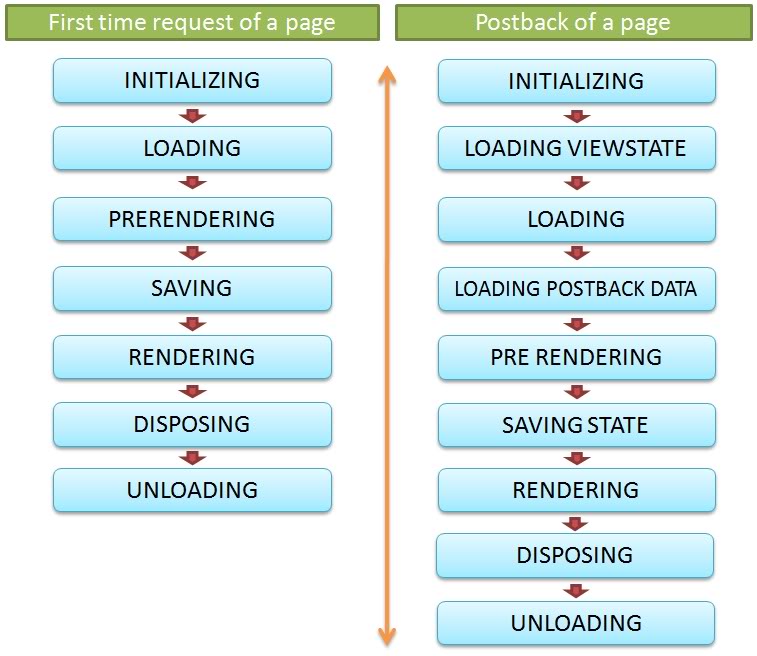




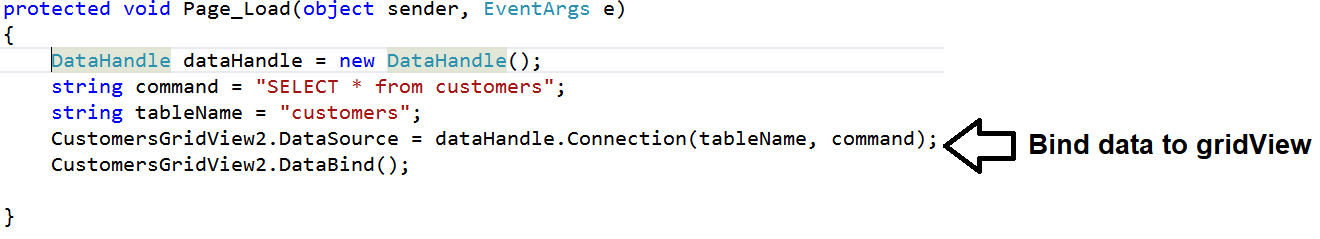


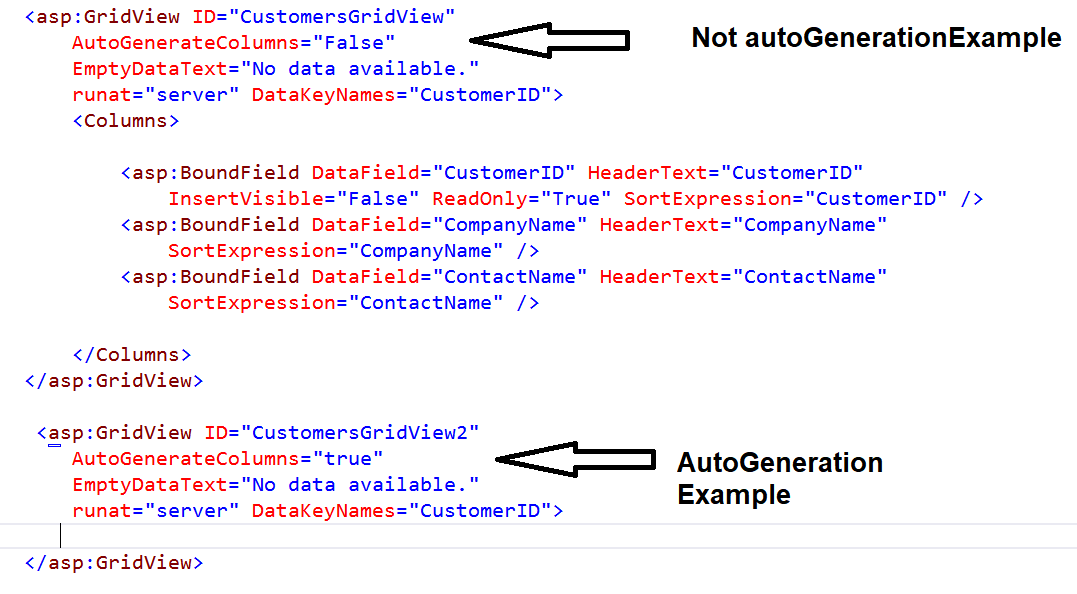




Controls

DataGridView





[קישור GridView באתר Codeproject](https://www.codeproject.com/Articles/36528/GridView-all-in-one)