

VALIDATION

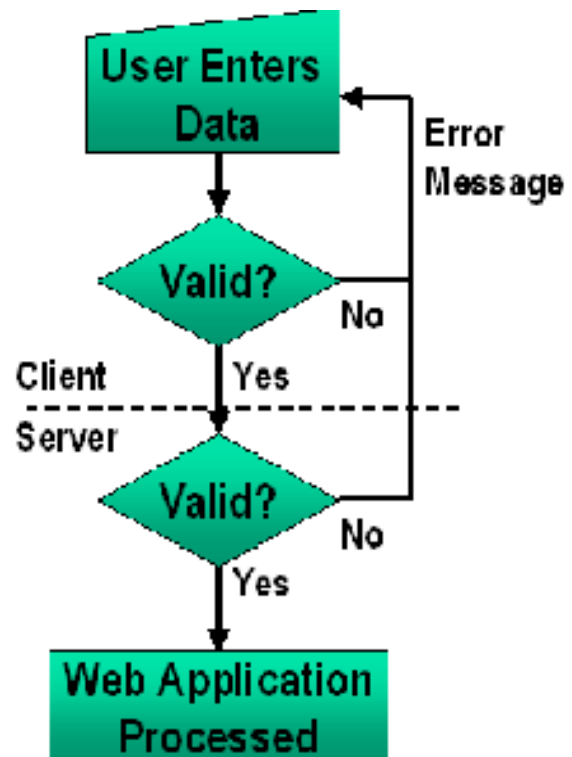
בכדי לאפשר ניהול יעיל וקל יותר של המידע המגיע מהמשתמש לשרת ישנן מספר מחלקות בשפה לשרותינו. אבל לפני שנראה אותן קודם כל ננסה להבין את השיקול בבניית אימות בצד השרת וגם בצד הלקוח.

קודם נראה דוגמה למה אנחנו רוצים להגיע, דהיינו מהם validators.

[03 Validators]

למה צריך אימות שרת ואימות בצד הלקוח גם כן?

- ASP.NET can create both client-side and server-side validation
- Client-side validation
 - Dependent on browser version
 - Instant feedback
 - Reduces postback cycles
- Server-side validation
 - Repeats all client-side validation
 - Can validate against stored data



עכשיו נראה דוגמה לתהליך של אימות פשוט לו היינו צריכים לכתוב אותו לבד. נתחיל בצד הלקוח.

[01 Manual Validation Client Side]

בכדי לדאבג את ה JS צריך להשתמש במילה השמורה `debugger` בתוך הפונקציה הרצויה. כמו כן צריך להשתמש בדפדפן האינטרנט אקספלורר לשם כך. מקש ימני על הדף ב solution explorer ואז Browse with...
הדוגמה הבאה מראה אימות בצד השרת

[02 Manual Validation Server Side]

הדוגמאות הללו הראו כמה עבודה צריך לעשות רק בכדי לקבל אימות לגבי המצאות תוכן בשדות או לא. כמובן שהאימות שיעשה בדרך כלל הוא הרבה יותר מתוחכם ולכן ישנם מאמטים בשפה, וזו הרשימה שלהם:

[03 Validators]

Control Name	Explanation	Properties
RequiredFieldValidator	Empty field	InitialValue – checks if the value is <u>not</u> 'InitialValue'
CompareValidator	Compare to constant, another field or database field	ValueToCompare or ControlToCompare Type (Integer, String, Double, Date, Currency) Operator (DataTypeCheck, Equal, GreaterThanEqual, LessThan, NotEqual...)
RangeValidator	Range between numbers, dates, letters. Compare to constant or another field	MinimumValue MaximumValue Type
RegularExpressionValidator	Check if it fits a specific expression	. Single char \$x ends with x {x} x amount of chars [] any of a group of chars * 0 or more + one or more
CustomValidator	Your own validate roles (run time)	OnServerValidate() ClientValidationFunction()

* <http://www.regexplanet.com/simple/>

* <http://www.zytrax.com/tech/web/regex.htm>

הפקדים ותפקידים:

- RequiredFieldValidator – מאפשר לוודא שהמשתמש הזין ערך כלשהו בשדה. כל שאר פקדי האימות מבצעים את הבדיקה רק במידה ושדה אינו ריק, כך שאם נרצה גם לחייב הזנת ערך בשדה וגם בדיקה אחרת נצטרך לשים שני פקדים על אותו שדה.
- RangeValidator – מאפשר לוודא שהערך שהוזן תואם לטווח מסוים לדוגמא: בין שני מספרים. בכדי להשתמש בו יש להגדיר את המאפיינים MinimumValue, MaximumValue וכן את טיפוס הטווח (מספרים / מחרוזות / תאריכים וכד') באמצעות המאפיין Type.
- CompareValidator – מאפשר לבצע השוואה בין השדה לבין ערך מסוים (כגון גדול ממספר מסוים) או בין השדה לבין פקד אחר (כגון אימות סיסמא). יש להשתמש במאפיין ControlToCompare (לפקד אחר) או ValueToCompare (מול ערך מסוים) ובנוסף להגדיר את אופרטור ההשוואה (גדול / קטן / שווה וכד') באמצעות המאפיין Operator ואת טיפוס ההשוואה (מספרים / מחרוזות / תאריכים וכד') באמצעות המאפיין Type.
- RegularExpressionValidator – מאפשר לבצע בדיקה מול תבנית של Regular Expression שהיא שפה לייצוג מחרוזות. יש להשתמש במאפיין ValidationExpression כדי להגדיר את התבנית. ישנם גם מספר אפשרויות מוכנות כגון: אימייל, כתובת אינטרנט וכד'.
- CustomValidator – מאפשר לנו לקבוע איזה בדיקה בדיוק לבצע, משמש בד"כ לבדיקות בצד השרת כגון: בדיקת שם משתמש וסיסמא מול DB. בכדי להשתמש בו יש להירשם לאירוע ServerValidate אשר מעביר כפרמטר אובייקט מסוג ServerValidateEventArgs באובייקט זה ניתן להגדיר בקוד האם הבדיקה הצליחה או לא באמצעות המאפיין IsValid (נראה בהמשך)

לכל הקונטרולים הללו יש למלא את המאפיינים הבאים:

```
<asp:Type_of_Validator
  id="Validator_id"
  runat="server"
  ControlToValidate="txtName"
  ErrorMessage="Message_for_error_summary"
  Display="static/dynamic/none"
  Text="Text_to_display_by_input_control">
</asp:Type_of_Validator>
```

Display=dynamic will order the page in the proper way.
Text="Text_to_display_by_input_control"> will be displayed under the control. when summary is used the ErrorMessage is displayed in the summary itself.

Usually Text="*"

נבחן את המאפיינים העיקריים הקיימים בכל הפקדים:

- **ControlToValidate** – כל פקד אימות אמור לבדוק פקד אחר (בד"כ TextBox) מאפיין זה קובע על איזה פקד תתבצע הבדיקה
- **Text** – הודעת שגיאה שתוצג בפקד במידה והאימות נכשל
- **ErrorMessage** – הודעת שגיאה שתוצג בפקד במידה והאימות נכשל ובמידה והמאפיין Text ריק. הודעה זו תוצג בכל מקרה בפקד ValidationSummary (עליו נלמד בהמשך) ולא ההודעה במאפיין Text
- **EnableClientScript** – מייצג האם האימות יבוצע בצד הלקוח באמצעות JavaScript או רק בצד השרת. מטעמי אבטחה תמיד מתבצעת בדיקה בצד השרת
- **ValidationGroup** – מאפשר ליצור קבוצות של פקדים כך שעבור כל לחצן תופעל בדיקה של קבוצה אחרת. גם ללחצן יש מאפיין זה כדי להגדיר איזו קבוצה להפעיל (נראה בהמשך)

ValidationSummary – פקד זה אינו פקד אימות אלא תפקידו להציג את כל השגיאות שיש בדף במקום אחד מרוכז. שימו לב להבדל בין המאפיין Text למאפיין ErrorMessage שקשור לפקד זה (ההבדל מוסבר בסעיף הקודם בנושא מאפייני הפקדים).

- **DisplayMode="List/BulletList/SingleParagraph"**
- **ShowMessageBox = "true"**

```
<asp:ValidationSummary id="valSummary"
runat="server"
HeaderText="These errors were found:"
ShowSummary="True"
DisplayMode="List"/>
```

בשלב הזה אפשר לעשות את תרגיל חנות הדגים.

[04 Validate() LifeCycle and Display_Dynamic]

בדוגמה זו נראה מספר בעיות ופינויות עליהן צריך לשים לב בין היתר נראה גם את חשיבות הפונקציה של `Page.Validate()` ואת אימות הקלט כחלק ממחזור החיים של הדף. נראה גם שלפעמים דווקא מעדיפים את התצוגה הדינאמית עבור הפקד בכדי שהכוכביות לא יזוזו לא משנה אם הכוכבית מראה שגיאה על שדה ריק או על התוכן שכן הוכנס לשדה.

בעיות עבור `Default1.aspx`:

כאשר נמלא ערכים נכונים בעליית הדף (דהיינו למשל dududu ו 1980) הכל עובד טוב אבל עכשיו נכניס לשדה השנה את שנת 2000 וקיבלנו שהדף נתקע ולא חוזר לשרת גם כאשר נשנה את השנה לשנה 1980. (ניתן לצאת מה"תקיעה" ע"י back של הדפדפן).

בעיות עבור `Default2.aspx`:

השינוי פה הוא הפיכת תיבת הטקסט של השנה ל `autopostback` ואז פתרנו את בעיית ה"תקיעה" שהיתה לנו בדף הקודם. אבל ישנה תופעה מוזרה כאשר נמלא מספר שגוי בשדה השנה נקבל חישוב עבור הגיל למשל עבור 2000 נקבל 11 בלי שישנו חייו של הכוכבית האדומה אותה אנחנו רוצים לראות. היא תופיעה רק כאשר נלחץ על הכפתור. בכדי להמנע מזה אנחנו נוסיף `Page.Validate()` בסוף פונקציית החישוב של הגיל. ניתן לראות את התוצאה ב `Default3.aspx`. חשוב להבין שכאשר פקד מכיל `autopostback=true` אז הוא נשלח לשרת בלי בדיקת שאר הפקדים. ואפילו בלי בדיקה שלו עצמו. (למשל אם לא נמלא כלום בפקד עדין נגיע לשרת ושם נעוף) אם רוצים בכל זאת לבצע את הבדיקה אנחנו חייבים לעשות זאת מהקוד ע"י `Page.Validate()`

בעיות עבור `Default4.aspx`:

השימוש אפילו החשוב יותר של הפונקציה `Page.Validate()` בא לידי ביטוי בדפדפנים שלא מריצים SCRIPT בכלל. זו אפשרות שניתן להגדיר אצל הלקוח בדפדפן. ניתן לדמות זאת ע"י `ClientTarget = "DownLevel"` ואז הבדיקות נעשות רק בשרת. שימו לב שאם לא ניבדוק את `Page.IsValid` לא נדע בכלל אם הבדיקות עברו או לא למרות שהן נעשו בכל מקרה! שימו לב שלא השתמשנו פה בכלל ב `Page.Validate()` באופן מפורש כי ...

עבור `Default5.aspx`:

בדיקות השרת, `Page.Validate()`, רצות באופן אוטומטי ממש לפני שאירוע `click` (או האירוע שגרם ל SUBMIT) מתחיל לרוץ. זאת אומרת שבדיקת `Page.IsValid` באירוע `Page_Load` תגרום לשגיאת זמן ריצה! ולכן במקרה הזה חייבים לקרוא לפונקציה `Page.Validate()` קודם לפני הבדיקה של `Page.IsValid`.

עבור Default7.aspx:

ניתן לראות שאכן גם לאחר שכל המידע נכון - כבר בתוך הטקסט של השם אם נריץ בעצמנו את הבדיקות נראה שהן עברו לעומת זאת אם לא נריץ בעצמנו הן יורצו ע"י הסביבה בדיוק לפני הכניסה לפונקציית ה Check() שעלידיה נעשה ה SUBMIT. ניתן גם לראות ש validate() תרוץ אוטומטית רק לפני האירוע של ה submit \ postback ולא בין כל שני אירועים של change !

Custom validator

[05 Custom Validator]

אם אף אחד מהקונטרולים שראינו לא מתאים (למשל כאשר רוצים אימות על פי מסד נתונים) ניתן להשתמש ב custom validator. בנוסף לכל המאפיינים שמשותפים לכל המאמתיים מכיל הקונטרול הזה גם שתי קריאות לאירועים שאנחנו יכולים לממש את הפונקציה שלהם:

```
ClientValidationFunction="ClientValFunc"  
OnServerValidate="ServerValFunc"
```

```
<script type="text/javascript">  
function ClientValFunc(source, args) {  
    if (args.Value % 2 == 0)  
    {  
        args.IsValid = true;  
    }  
    else  
    {  
        args.IsValid = false;  
    }  
}
```

```
protected void ServerValFunc(object sender, ServerValidateEventArgs e)  
{  
    if(int.Parse( e.Value) % 2 ==0 )  
    {  
        e.IsValid = true;  
    }  
}
```

ניתן לשחק עם האופציה של ClientTarget="downlevel" בכדי לראות את ה not valid של צד השרת

Validation groups

[06 Validation Groups]

במקרה בו נרצה לעשות submit אבל לא לאמת את כל הפקדים עבור אותו ה submit, פשוט ניתן לקבץ את המאמטים שמיועדים לאותו submit. עבור submit אחר ניתן יהיה לקבץ מאמטים אחרים עבורו. כל זאת ניתן לעשות ע"י שימוש במאפיין

`ValidationGroup="NameGroup"`

שימו לב ש submit כללי לא יפעיל את האימותים שמקובצים ל group ספציפי!