

מדריך לשליחת אימייל מאתר

הערות:

- המדריך מצריך ידע בשפת C#, בעצמים, מחלקות ומאפיינים, וכמו כן ידע ב HTML ו XML, ובשימוש בתגיות.
- המדריך מלמד איך ליצור פעולה (בתוך מחלקת עזר) ששולחת אימייל. הוא לא מסביר איך לעשות שימוש פרקטי בנושא, כמו איך ליישם אותו על דף של שחזור סיסמה, או דוגמאות כאלה ואחרות. דברים כאלה נבנים על מה שהמדריך הזה מלמד.

לפני שנבין איך שולחים אימייל מאתר, נעבור על התהליך בו אנחנו שולחים אימייל בצורה רגילה.

כשאנו רוצים לשלוח אימייל אנו עוברים דרך כמה שלבים:

1. נכנסים לתוכנת אימייל מסוימת, או לאתר שממנו אנו שולחים אימייל (גם הדף באתר בו אנו שולחים אימייל נקרא תוכנת האימייל).
2. מכניסים את שם המשתמש והסיסמה שלנו, על מנת שתוכנת האימייל תדע מי השולח של ההודעה.
3. פותחים הודעה חדשה, מכניסים לתוכה את הנמענים, נושא ההודעה וגוף ההודעה, ושולחים.

אלו הם השלבים שקורים כשאנו רוצים לשלוח אימייל באופן ידני. אבל אם אנחנו רוצים שהאתר שלנו ישלח אימייל בעצמו?
קודם כל נעבור על כל המושגים שתיארנו בשלבים, על מנת שנבין את הכל.

- **תוכנת אימייל** – תוכנה שיוצרת הודעה חדשה, לקבל מאיתנו את הנמענים, הגוף, והנושא, לשים הכל בהודעה אחת, ולתת אותה לשרת אימייל שיעשה את שאר העבודה. תוכנת אימייל לא חייבת להיות בהכרח תוכנה שפועלת על שולחן העבודה שלנו, אלא יכולה להיות גם תוכנה שעובדת באתר, לדוגמה G-Mail, Windows Live Mail, Walla, וכדומה. באתרים אלו יש דפים שמתפקדים כתוכנות אימייל.
- **משתמש** – לכל משתמש יש שם משתמש וסיסמה. כל משתמש מזוהה ע"י כתובת האימייל הייחודית לו.
- **שרת אימייל** – שרת האימייל מקבל הודעות אימייל מתוכנת אימייל מסוימת, נותן את ההודעה לשרת אחר, והשרת האחר מעביר את ההודעה לנמען. כל שרת אימייל מופקד על מספר משתמשים. לדוגמה, שרת האימייל של Walla, מופקד על כל המשתמשים שכתובות האימייל שלהם מסתיימות ב @walla.com, ולכן, אם משתמש שכתובתו מסתיימת כך שולח הודעה חדשה, הוא למעשה שולח אותה לשרת של Walla, ואם מישהו אחר שולח הודעה לנמען שכתובתו היא של Walla, אז השרת של Walla ידאג למצוא את הנמען ולהעביר לו את ההודעה.

למדנו מהם השלבים כדי לשלוח הודעת אימייל, עכשיו רק צריך ליישם אותם באתר שלנו.

בשלב הראשון אנו נכנסים באופן רגיל לתוכנת האימייל שלנו. כדי לשלוח אימייל מאתר, אנו בונים את האתר כך שהוא יתפקד גם כתוכנת אימייל, כלומר, הוא יידע ליצור הודעה חדשה, לשים בה את כל הנתונים הדרושים, ולמסור את ההודעה לשרת האימייל.

בשלב השני אנו מתחברים לאתר. האתר ישמש גם כתוכנת האימייל, וגם כמשתמש שנעזר בתוכנת האימייל, ולכן, כמו כל משתמש רגיל, הוא צריך שם משתמש וסיסמה, וחשבון אימייל באחד ממיליון שרתי האימייל הקיימים.

כעת עליכם ליצור חשבון אימייל חדש, בכל תיבת אימייל שתבחרו (Walla, G-Mail, וכו'...), שתשמש את האתר על מנת לשלוח אימיילים. זכרו שהאתר שלנו לא מאובטח, ולכן אפשר לראות את השם משתמש והסיסמה, אז אל תשתמשו בתיבה הזו לצרכים אישיים או חשובים מאוד.

לאחר שיצרתם משתמש, ויש לכם שם משתמש וסיסמה על תיבת אימייל מסוימת, צריך לתת את כל הנתונים האלה לאתר שלנו, שידע לעבוד איתם. למעשה, הנתונים שאנחנו צריכים לתת לו הם שם המשתמש והסיסמה שבהם הוא ישתמש, וגם את השרת שאליו הוא צריך לפנות, כי הרי אמרנו שהאתר ישמש גם כתוכנת האימייל, שתפקידה לתקשר עם שרת האימייל ולהעביר לו את ההודעה.

לכן, אנחנו צריכים את שם המשתמש, הסיסמה, ופרטי השרת. אבל מה הם בדיוק פרטי השרת? רוב השרתים בעולם משתמשים בשיטה אחת מאוד מקובלת על מנת לשלוח אימיילים. מה הכוונה שיטה? כמו שבני אדם יכולים לדבר עברית, אנגלית, ועוד הרבה שפות שונות, כך גם שרתים יכולים לדבר הרבה שפות. שרת מדבר שפה אחת בשביל קבלת אימיילים, ושפה אחרת בשביל שליחת אימיילים. אותנו לא מעניינת השפה שבה הוא משתמש בשביל לקבל אימיילים, אלא השפה שבה הוא משתמש בשביל לשלוח אימיילים. השפה שבה השרתים משתמשים בשביל לשלוח אימיילים נקראת SMTP, שזה קיצור של Simple Mail Transfer Protocol – שפת העברת מיילים פשוטה. אנחנו רק צריכים לדעת את שם השפה, לא יותר מזה.

לכל שרת בעולם יש 2 תכונות שלפיהם ניתן לזהות אותו ולהבדיל אותו משאר השרתים בעולם. הראשון זה הכתובת (host), והשני זה הפורט (port). הכתובת בנויה כמו כתובת של אתר אינטרנט רגיל, אין צורך להרחיב. אבל מה זה פורט? פורט הוא למעשה כמו הכביש שדרכו מתקשר השרת. לדוגמה, אם ידוע לי שהפורט של שרת מסוים הוא 576, אני צריך לתכנת את האתר שלי שיתקשר רק דרך פורט מספר 576, אם ניתן לו מספר אחר, הוא בחיים לא יגיע לשרת שאנחנו רוצים.

עכשיו שאנחנו יודעים איזה פרטים אנחנו צריכים, ניתן אותם לאתר. אבל רגע, מאיפה אנחנו יודעים את הכתובת והפורט של השרת אימייל שלנו? פה נכנסת קצת עבודת שיטוט באינטרנט. לכל שרת אימייל (Walla, G-Mail, Hotmail) יש HOST ו PORT שונים. על מנת למצוא אותם יש לחפש בגוגל, לדוגמה: Walla mail server details, או Walla mail server host/port. כאן אשים את הפרטים של שרתי המייל הגדולים, אם ברצונכם פרטים של עוד שרת, בקשו ואני אחפש, ואם אני אמצא אני אוסיף.

אני ממליץ לפתוח תיבה ב G-Mail, היא עושה הכי פחות בעיות, אבל כל תיבה אחרת תעבוד גם כן.

שם השרת	כתובת (HOST)	פורט (PORT)	דרוש SSL?
G-Mail (gmail.com)	smtp.gmail.com	587/465	כן
Hotmail/Windows Live	smtp.live.com	25	כן
Yahoo! Mail (yahoo.com)	smtp.mail.yahoo.com	25	לא
Yahoo! Mail Plus	plus.smtp.mail.yahoo.com	465	כן
MSN (msn.com)	smtp.email.msn.com	*	כן

* לא הצלחתי למצוא את הפורט של MSN, אם תצליחו למצוא את הפורט והוא עובד, בבקשה תגידו לי שאני אוסיף.

** אם ברצונכם מידע על תיבות נוספות, בקשו ואני אחפש ואוסיף.

בטח שמתם לב שהוספתי עמודה אחת שלא דיברנו עליה. במרוצת השנים, האקרים נהיו יותר ויותר מתוחכמים, וכעת יש את הטכנולוגיה לפרוץ הודעות מייל ולתפוס אותם בזמן שהם מועברות, ולקרוז את התוכן שלהם. למטרה זו פותחה טכנולוגיית ה-SSL, על מנת להצפין את ההודעות שלנו, כך שגם אם הם ייתפסו, לא יוכלו לקרוא אותם. SSL הם ראשי תיבות של Secure Sockets Layer, שכבת מידע בטוח. לא כל שרת משתמש ב-SSL, אבל רובם כן. כרגע לא הסברנו מה לעשות עם המידע הזה (האם השרת שבו בחרנו משתמש ב-SSL), אבל נלמד בהמשך.

לאחר שיש לנו את כל הנתונים: שם המשתמש, הסיסמה, כתובת השרת (HOST), ופורט השרת (PORT), נתכנת את האתר שלנו להשתמש בהם.

נכנס לדף ה-Web.config שבאתר שלנו. לא צריך ליצור אחד כזה, כשאתם יוצרים אתר ב-Visual Web Developer הוא יוצר אחד לבד. **אזהרה! אל תשנו שום דבר בדף הזה בלי לדעת מה אתם עושים, זה עלול לגרום לקריסה של האתר שיהיה קשה מאוד לתקן אותה!**

דף ה-Web.config הוא דף XML למעשה. דף XML הוא דף שמשתמש בתגיות, כמו HTML, רק שאת התגיות ב-XML אנחנו יוצרים, והתגיות של HTML הם משותפות לכל העולם, ואנחנו לא קובעים אותם. בדף Web.config, כבר קבעו לנו את תגיות ה-XML, ואנחנו רק משתמשים בהם.

בתוך דף ה-Web.config, נכתוב את התגיות הבאות, ונקפיד שהם נמצאות בתוך תגית ה-configuration, אך לא בתוך תגית ה-system.web:

```
<system.net>
  <mailSettings>
    <smtp>
      <network host="" port="" userName="" password="" />
    </smtp>
  </mailSettings>
</system.net>
```

התגית הראשונה, <system.net> היא תגית שבתוכה שמים את כל הגדרות האינטרנט של האתר.

התגית השנייה, `<mailSettings>` היא תגית שבתוכה שמים את כל הגדרות האימייל של האתר.

התגית השלישית, `<smtp>` היא תגית שבתוכה שמים את כל הגדרות האימייל שמתבצעות בשפת ה SMTP, וכמו שאמרנו קודם, אנחנו משתמשים רק בשפת SMTP, לכן כל הגדרות האימייל שנשתמש בהם יהיו בתוך התגית הזו.

התגית הרביעית, תגית ה `<network>` היא תגית שלמעשה מגדירה לאתר את כל הנתונים שדיברנו עליהם מקודם. שימו לב שבתוך התגית הזו שמנו את שם המשתמש, הסיסמה, כתובת השרת והפורט שלו. שימו לב גם שלא שמנו ערכים בתוך השדות, מכיוון שאת הערכים האלה תצטרכו להשלים בעצמכם, בהתאם לשרת שאותו בחרתם. לבסוף, התגית תראה כך, רק שתצטרכו להוסיף ערכים:

```
<network host="" port="" userName="" password="" />
```

לאחר שסיימנו לכתוב את השורות האלה, סיימנו להגדיר לאתר שלנו את המשתמש והסיסמה, ולאיזה שרת להתחבר ואיפה הוא נמצא. האתר שלנו כרגע מתפקד גם כמשתמש, וגם כתוכנת אימייל.

נחזור לשלבים שדיברנו עליהם בהתחלה, ונעבור לשלב הבא.

בשלב הבא, הוא השלב האחרון, אנו צריכים לתכנת את האתר לפתוח הודעה חדשה, להכניס בה נתונים, ולשלוח אותה. אנו רוצים ליצור פעולה, שתדע לשלוח אימייל, וכל פעם שנרצה לשלוח אימייל נפעיל את הפעולה הזו, כך שלא נצטרך לכתוב את הקוד של שליחת האימייל כל פעם מחדש. למטרה זו ניצור מחלקה סטטית חדשה, בשם MailUtility ובתוכה נשים את הפעולה שתשלח אימייל:

```
public static class MailUtility
{
}
```

מחלקה סטטית היא מחלקה שאין לה פעולה בונה, כלומר, היא לא מגדירה עצם מסוים, אלא רק מכילה פעולות סטטיות שניתן להשתמש בהם. אז למה לא מספיק למחוק את הפעולה בונה? כי אם אנחנו רק מוחקים את הפעולה הבונה, C# עדיין מתייחס למחלקה כמחלקה שמגדירה עצם מסוים, ועדיין נוכל לעשות ממנה עצם. אנו לא רוצים שמישהו יוכל ליצור עצמים מהמחלקה הזו, ולכן אנחנו מגדירים אותה `static`. כמובן שאחרי שהגדרנו אותה כסטטית, נמחק את הפעולה הבונה שלה, כי אם לא נמחק זו תהיה טעות.

אין צורך לציין שהפעולה שניצור לוק חת הכל ממשתנים שהיא מקבלת כמו שתכף נראה, והיא לא מניחה שום הנחות בעצמה, כל זאת על מנת שהיא תהיה כללית ויהיה אפשר להשתמש בה כבסיס לכל דבר, מבלי לשכתב אותה.

לאחר שיש לנו את מחלקת העזר, ניצור בה פעולה חדשה:

```
public void SendMail(string sendTo, string subject, string body)
{
    MailMessage m = new MailMessage();
    m.From = new MailAddress("<הכתובת שיצרתם לאתר>",
        "<שם שיוצג לנמען, במקום הכתובת>");
    m.To.Add(new MailAddress(sendTo));
    m.Subject = subject;
    m.Body = body;
    SmtplibClient sc = new SmtplibClient();
    sc.EnableSsl = true;
    sc.Send(m);
}
```

למעשה מה שאנו עושים בפעולה זו, זה ליצור הודעה חדשה, לשים בה נתונים, ולשלוח אותה. יש לנו פה 2 מחלקות חדשות, שנעבור עליהם מיד. שימו לב שבשורות השנייה והשלישית שבתוך הפעולה ישנם מחרוזות שאתם צריכים לשים בהם ערכים, לפי מה שכתוב.

אם אינכם יודעים מאפיינים, עצרו בנקודה זו, למדו את הבסיס של מאפיינים, והמשיכו לקרוא.

אם בהמשך יהיה כתוב, לדוגמה "מאפיין שמכיל את המוען", הכוונה "מאפיין שדרכו מגיעים לתכונה שמכילה את המוען", אך אנו נכתוב בצורה הראשונה, לשם נוחות הקריאה.

הפעולה שיצרנו היא פעולה סטטית, שלא מחזירה כלום, ומקבלת את כתובת האימייל של מקבל ההודעה, נושא ההודעה, וגוף ההודעה.

המחלקה הראשונה, `MailMessage`, היא בעצם הודעת אימייל. כמה מהתכונות שלה הם נמען, מוען, גוף ונושא.

<code>MailMessage()</code>	פעולה בונה, שבעצם בונה הודעת אימייל חדשה. ניתן להעביר לפעולה הבונה את כל הנתונים הדרושים, אך אנחנו בחרנו לשים אותם צעד צעד, לשם נוחות ההבנה.
<code>MailMessage.From</code>	מאפיין שמכיל את המוען. למעשה המוען הוא עצם מסוג כתובת אימייל, שעליו נדבר בהמשך.
<code>MailMessage.To</code>	מאפיין המכיל אוסף של נמענים, כאשר כל נמען הוא עצם מסוג כתובת אימייל. למעשה זהו אוסף של כתובות אימייל. כל כתובות האימייל במערך זה יהיו הנמענים, ולכולם תשלח ההודעה.
<code>MailMessage.To.Add(MailAddress item)</code>	פעולה של המאפיין <code>MailMessage.To</code> , שמקבלת עצם מסוג כתובת אימייל, ומוסיפה אותה לרשימת הנמענים
<code>MailMessage.Subject</code> , <code>MailMessage.Body</code>	מאפיינים די ברורים, מכילים את נושא ההודעה ואת גוף ההודעה
<code>MailMessage.IsBodyHtml</code>	מאפיין שלא השתמשנו בו בדוגמה, אך תוכלו להשתמש בו אם תרצו. נניח ותרצו שההודעה שלכם תהיה מעוצבת, עם טבלה או תמונה או קישור או צבעים או כל דבר אחר, איך תעשו זאת? תכתבו את גוף ההודעה כאילו הוא דף HTML, כלומר, תכתבו את גוף ההודעה, וכשתרצו עיצוב מסוים תשתמשו בתגיות, לדוגמה, עבור קישור, השתמשו בתגית <code><a></code> . על מנת שהעצם שלנו, של הודעת האימייל, יידע שגוף ההודעה היא HTML, ויידע שהוא צריך לעצב אותו לפי התגיות, אנו אומרים לו בעזרת מאפיין זה שהוא צריך להתייחס לגוף ההודעה כ

	.HTML אם ערך המאפיין הוא true, אזי הוא יתייחס לגוף ההודעה כ HTML ויפענח את התגיות לעיצוב.
--	--

לאחר שהבנו את המחלקה של הודעת אימייל, נעבור למחלקה של כתובת אימייל `MailAddress`.

למעשה לא השתמשנו בפעולות של המחלקה הזו, אלא רק בשתי פעולות בונות. למחלקה זו יש פעולה בונה המקבלת פרמטר אחד, ופעולה בונה המקבלת שני פרמטרים.

הפעולה הבונה המקבלת 2 פרמטרים, מקבלת בפרמטר הראשון מחרוזת שהיא בעצם כתובת האימייל של השולח. שימו לב שכתובת האימייל הזו חייבת להיות נכונה ותואמת למה שכתבתם בדף ה `Web.config`. הפרמטר השני הוא שם שיוצג למקבל ההודעה, במקום כתובת האימייל, רק מפני שבד"כ בני אדם מעדיפים לראות שם ולא כתובת אימייל, כי זה יותר אסתטי. ניתן לוותר על הפרמטר הזה, אך עדיף לשים אותו, למען נוחות הקריאה.

בפעולה הבונה שמקבלת פרמטר אחד מכניסים רק את כתובת האימייל, בלי שם. אנו משתמשים בפעולה בונה זו על מנת להגדיר נמען, כי אנו לא צריכים את שם הנמען, הרי גם ככה הוא לא יראה את שמו, וחוץ מזה שהוא יודע את השם שלו.

המחלקה האחרונה שאנו משתמשים בה פה היא `SmtpClient`. עצם של המחלקה הזו הוא עצם שיועד לתקשר עם שרת מייל בשיטת SMTP, שהיא השיטה בה אנחנו משתמשים. אנו יוצרים עצם חדש מסוג `SmtpClient`, ולא נותנים לו כלום בפעולה הבונה.

השורה הלפני אחרונה, היא שורה שמשתנה בהתאם לשרת שבחרתם:

```
sc.EnableSsl = true;
```

המאפיין הזה שייך לעצם ה `SmtpClient`, והוא אומר לעצם שאמור למסור לשרת את ההודעה, האם להשתמש בטכנולוגיית ה SSL שדיברנו עליה מקודם, או לא. במקרה זה השתמשנו ב SSL. הביטוי בטבלה שלמעלה או מצאו באינטרנט האם השרת שבחרתם משתמש ב SSL, ולפי כך שנו את הערך של המאפיין הזה לפי הצורך.

לאחר שסיימנו את כל העברת הנתונים, ניתן לשלוח את ההודעה, וזה מתבצע באמצעות פעולה פשוטה של המחלקה `SmtpClient`:

```
sc.Send(m)
```

אנו מעבירים לפעולה את הודעת האימייל שלנו, והיא שולחת אותו לשרת האימייל, שמשם ישלח אותה לנמען.

לאחר שהפעולה הזו מסתיימת, האימייל נשלח.

זהו! סיימתם את המדריך, ויש לכם פעולה מוכנה ששולחת אימייל, גמישה, שיכולה לשמש לכל דבר הצורך שליחת אימייל, ולא צריך לשכתב אותה מחדש.

אם תרצו לעשות כל דבר הצורך שליחת אימייל, תוכלו להשתמש בפעולה הזו.

בהמשך אכתוב מדריכים על איך לעשות דף שחזור סיסמה ושם משתמש במקרה שהמשתמש שכח אותם, ואיך לעשות דף אימות אימייל, שכאשר משתמש נרשם לאתר, הוא מקבל קוד לכתובת

האימייל שלו, והוא צריך להכניס אותה באתר, על מנת שנדע בוודאות שהוא לא שיקר כשהוא כתב את כתובת האימייל שלו.

קרדיטים!!

דור כהן (כותב המדריך)

אורן קליינבוים

יורם לס

דור כהן, אורן קליינבוים, יורם לס

י"ב 5, 2009-2010, תיכון שז"ר בת-ים

גרסה: 1.2