

AKTUALNOSTI S PODROČJA IT STORITEV

XPIKA

NOVEMBER 2022

EXCEL – POIŠČI ZADNJO INSTANCO DOLOČENEGA ZNAKA

stran 10

.NET MAUI delavnica 2. del – Nekaj osnovnih informacij o sami platformi

stran 18

TRIKI IN NASVETI ZA INTRANET V SHAREPOINT ONLINE

stran 12

JAVASCRIPT NOVOSTI V LETU 2022

stran 22

ZA VEČJO STOPNJO KONVERZIJE JE POTREBNO IZBOLJŠATI UPORABNIŠKO IZKUŠNJO VAŠEGA SPLETNEGA MESTA

stran 27



ORGANIZATOR



THRIVE
CONFERENCE

Zagotovite si
Super Early Bird Prijavo!

popust velja samo še do 15. decembra 2022

Polna kotizacija	580,00€	406,00€
Enodnevna - 1. dan	390,00€	273,00€
Enodnevna - 2. dan	300,00€	210,00€
CxO	320,00€	224,00€
Študentska kotizacija		200,00€



30. do 31. maj 2023



Lipica, Hotel Maestoso

thriveconf.com

Spoštovane in spoštovani, ozrem se skozi okno in uživam v pogledu na zlate in škrlatne barve te čudovite jeseni. Narava nas razvaja s toplimi sončnimi žarki in obdaruje s slastnimi plodovi. Privoščim si sproščujoč sprehod, ki zbistri misli in rodi ideje.



BRANKA SLINKAR
Direktorica

Nezadržno se približuje konec leta, ki zahteva obračun z iztekajočim se letom, soočenje z rezultati, ki smo jih dosegli. Obenem pa se spogledujemo z letom 2023, kaj nam bo prineslo, kako se nam bo godilo.

Ker je »**vsak svoje sreče kovač**«, pripravljamo plane za prihodnje leto, ki morajo biti razvojno naravnani, skladni z dolgoročno vizijo uspešnega poslovanja in rasti podjetja.

Prilagodljivost in učinkovitost sta ključni v času nenehnih sprememb in hitrega razvoja tehnologije. Da bi naglemu tempu sploh lahko sledili in bili uspešni v procesu digitalizacije, je nenehno izpopolnjevanje izjemno pomembno. Znanje je neprecenljivo in moramo ga skrbno negovati in spoštovati. Možnost, da svoje znanje nadgradimo, moramo razumeti kot privilegij in vsako priložnost pograbititi z obema rokama.

Naši predavatelji so nadvse predani, svoje bogato znanje in izkušnje z veseljem delijo, tako na projektih kot na slehernem izobraževanju, naj bo to

kratko predavanje na konferenci ali pa poglobljen in zahteven bootcamp. Imamo že 30 letne izkušnje z izobraževanji, zato lahko z vso odgovornostjo trdim, da boste pri nas dejansko pridobili največ novega znanja in napotkov za delo.

	Administrator Administrators implement, monitor,		Developer Developers design, build, test, and maintain cloud solutions.
	Business User Use Microsoft applications to move your business forward.		Data Analyst Data analysts make data easy to understand through designing an...
	DevOps Engineer DevOps engineers combine people, process, and...		Security Engineer Security engineers implement security controls and threat...

Microsoft je v zadnjem letu korenito spremenil učne in certifikacijske poti, veliko programov, ki so še kako aktualni, je celo umaknil s programa. Nekatere podrobnosti boste lahko prebrali v nadaljevanju, v članku, ki ga je za vas pripravila **Petra**. Za pomoč in pojasnila pa smo vam vedno na voljo.

Zavedamo se, da vam nenehne spremembe predstavljajo nemalo težav pri izbiri optimalnega šolanja. Veseli nas, ker se vedno več podjetij obrača na nas, da pri tem pomagamo,

skupaj tudi pripravljamo plane za naprej. In prav učni plani so zelo pomembni, saj se vam na podlagi letih odpirajo tudi možnosti sofinanciranja, kar lahko občutno zmanjša vašo investicijo v usposabljanja zaposlenih.

Na podlagi predloženih planov je Microsoft v preteklih dveh letih odobril **sofinanciranje v višini kar 50%** celotne vrednosti programa, hkrati pa dodal še brezplačne vouchere za izpite. Tako so nekatera podjetja v Sloveniji prejela znatna sredstva za dvig IT kompetenc svojih zaposlenih.

Zato bi vas rada spodbudila, da skupaj pripravimo plane tudi za vaš kolektiv in naredili bomo vse, da bo šolanje tudi za vas bolj ugodno.

Na voljo so še različni razpisi, pretežno iz EU skladov.

Do **31.03.2023 je še odprt razpis** Slovenskega podjetniškega sklada - Vavčer za dvig digitalnih kompetenc. Namen vavčerja je spodbuditi ciljne skupine k zagotovitvi ustreznih znanj zaposlenih vodstvenega kadra za ključna področja digitalizacije.

Kakšne spodbude digitalizaciji lahko pričakujemo v letu 2023?



Do **15. aprila 2024 je odprt razpis** SPIRIT Slovenija - Podpora zagonskim, mikro, malim in srednjim podjetjem pri strateški trajnostni in krožni

transformaciji poslovanja v letih 2022-2025 (kratica JR STKTP NOO 2022-2025)



Rezervirajte si čas že danes za 12. Thrive konferenco in si zagotovite še eno izjemno izkušnjo učenja in izmenjave izkušenj. Konec maja se bomo družili na še eni izjemni lokaciji, v Lipici.

Kupite vstopnice s kar 30 % popustom za zgodnje prijave, ki velja do 15. decembra 2022.

Ko boste iskali partnerja za **posodobitev intraneta ali izvedbo nove spletne strani; izdelavo mobilne aplikacije, nadgradnje IT infrastrukture, morda migracijo v oblak? Želite izboljšati varnost svojega IT okolja, ...**

... naj bo **Xnet** vaša prva izbira, ko gre za IT rešitve in storitve. Microsoft tehnologije so naša strast.

ČUVAJTE SE IN OStanITE ZDRAVI!
Branka Slinkar





Zaposlujemo



Pridruži se nam in postani del vrhunske ekipe predavateljev

Xnet išče Office predavatelja, ki bo navdušeno delil znanje.



KAZA

MICROSOFT OFFICE

8

Excel – Poišči zadnjo instanco določenega znaka

Klemen Vončina

MCT, Microsoft Office Specialist Master

SHAREPOINT

10

Triki in nasveti za intranet v SharePoint Online

Robi Vončina

MVP, MCT, MCITP, MCSA, MCTS

SQL

13

Optimizing Analytical Queries Part 7: Introducing Columnar Storage

Dejan Sarka

MVP, MCT

RAZVOJ

16

.NET MAUI delavnica 2. del – Nekaj osnovnih informacij o sami platformi

Gašper Rupnik

MCT, MS, MCSA, MCPS

RAZVOJ

20

JavaScript novosti v letu 2022

Domen Gričar

MCSA, MCSA, MCT

RAZVOJ

23

Metoda »lean developmenta« v praksi

Alen Dedić

Front-end programer

Xnet ekipa

Gašper	MAUI je čista zmaga
Alen	Dobrodošel v ekipi
Anja	Alfe ni več, odpeljali so jo
Urška	Nesreča nikoli ne počiva
Klemen	Ne bo več tema
Domen	Še veliko ga je ostalo
Aida	Od Poncija do Pilata, pa nič
Robi	Vedno bolje je
Urška	Bolj / manj svetlo, pixel več/manj
Dijana	Ufff, koliko je vsega...
Aleš	Da bo vse po pravilih
Petra	Konec dober, vse dobro
Aleks Manca	Čestitke za pol tortice
Miha	Prva plata – DONE!
Dejan	Spet mora biti odlično
Jože	Še kar online
Mojca, Dragica	Nimata časa, sta v pokoju

UX / UI DESIGN

25

Za večjo stopnjo konverzije je potrebno izboljšati uporabniško izkušnjo vašega spletnega mesta

Premzl Urška

ADMINISTRACIJA

28

Powershell kotiček

Aleš Lipušček
MCP, MCTS, MCITP

DRUGO

30

Zaupanje, prepoznavnost, predanost, sposobnost...

Petra Militarev
Vodja izobraževanja

KOLOFON

ISSN: 1408-7863

Kompas Xnet d.o.o.

Stegne 7

1000 Ljubljana

Telefon: 01 5136 990

Fax: 01 5136 999

Email: info@kompas-xnet.si

Web: <https://www.kompas-xnet.si>

Urednica in oblikovalka

Urška Premzl

Člani uredništva

Aleš Lipušček, Aida Kalender Avdič, Gašper Rupnik, Miha Pihler, Jože Markič, Klemen Vončina, Robert Vončina, Anja Rupnik, Petra Militarev, Dejan Sarka, Manca Gruden.



Klemen Vončina
Microsoft Office Specialist Master, MCT
klemen.voncina@kompas-xnet.si



EXCEL – POIŠČI ZADNJO INSTANCO DOLOČENEGA ZNAKA

Ko v Excelu delamo z besedilom, hitro naletimo na težavo iskanja prve, druge, tretje... pojavitve nekega znaka v določeni celici. V našem primeru bomo iskali zadnji presledek v besedilu "Kje je zadnji presledek?". V mojem primeru sem to besedilo zapisal v celico D1.

Prva stvar, ki jo bomo naredili, je poiskali razliko med dolžino izvornega besedila in dolžino izvornega besedila, če mu odstranimo presledke. To bomo naredili s kombinacijo funkcij LEN ter SUBSTITUTE.

```
=LEN(D1)-LEN(SUBSTITUTE(D1;" ";""))
```

Prvi LEN vrne dolžino izvornega besedila, v drugi LEN funkciji pa najprej s pomočjo funkcije SUBSTITUTE v izvornem besedilu presledke nadomestimo z »ničemer« (beri: odstranimo presledke) in nato ugotovimo dolžino novega besedila. S tem posredno izvemo, koliko presledkov se nahaja v celici D1.

D	E
Kje je zadnji presledek?	3

V naslednjem koraku bomo izključno to zadnjo pojavitev presledka v izvornem besedilu, torej tretjo, nadomestili z nekim drugim znakom. Smiselno je uporabiti znak, ki ga v izvornem besedilu nimamo, v našem primeru denimo poševnico. Rezultat formule, ki smo jo spisali do sedaj, bomo uporabili kot argument nove SUBSTITUTE funkcije.

```
=SUBSTITUTE(D1;" ";""/";LEN(D1)-LEN(SUBSTITUTE(D1;" ";"")))
```

S tem bomo funkciji SUBSTITUTE povedali, da želimo le en točno določen presledek, v tem primeru tretji, nadomestiti z nekim drugim znakom.

D	E	F
Kje je zadnji presledek?	Kje je zadnji/presledek?	

Na tej točki smo večino dela že opravili. Vse kar nam še ostane je, da poiščemo mesto poševnice, kar je bilo v izvornem besedilu mesto zadnjega presledka. Točno tej nalogi je namenjena funkcija FIND.

```
=FIND("/";SUBSTITUTE(D1;"
";"/";LEN(D1)-LEN(SUBSTITUTE(D1;"
";""))))
```

FIND nam vrne številko, ki predstavlja mesto iskanega znaka.

D	E	F	G
Kje je zadnji presledek?	14		

Ta informacija je lahko silno koristna pri nadaljnjih akcijah. Na podlagi te informacije lahko na primer s kombinacijo funkcij RIGHT(LEN-FIND) (o kateri sem pisal v eni izmed

prejšnjih izdaj Pike) iz celice izvlečemo vse, kar je desno od tega znaka, torej vse kar je za zadnjim presledkom.

```
=RIGHT(D1;LEN(D1)-
FIND("/";SUBSTITUTE(D1;"
";"/";LEN(D1)-LEN(SUBSTITUTE(D1;"
";""))))
```

D	E	F	G	H	I
Kje je zadnji presledek?	presledek?				

V nedavnem primeru iz prakse sem takšno formulo uporabil, da sem iz velikega števila celic izluščil datum, ki se je nahajal na koncu besedila v celici (oz. za zadnjim presledkom).

D	E	F	G	H	I
Ta članek je bil napisan 7.11.2022	7.11.2022				

Šolanje za vaše napredovanje

AKTUALNI TEČAJI

Analize podatkov v Excelu

Kdaj: 12. 1. 2023



Uvod v Excel BI

Kdaj: 16. - 17. 1. 2023



Vrtilne tabele

Kdaj: 20. 1. 2023



Microsoft Excel nadaljevalni

Kdaj: 25. - 27. 1. 2023





Robi Vončina
MVP, MCT, MCITP, MCSA, MCTS
robi.voncina@kompas-xnet.si



TRIKI IN NASVETI ZA INTRANET V SHAREPOINT ONLINE

SharePoint Online je zelo kompleksen in obsežen produkt, zato se ničkolikokrat v pogovorih s stranko, srečamo z vprašanjem, kako sploh začeti. Tudi priprava želja in specifikacij za uvedbo portala zna delati preglavice, zato bom v nadaljevanju poskusil navesti in opisati nekaj nasvetov za lažji začetek.

Preden pa začnem z opisovanje in navedbo, bi na tem mestu rad poudaril, da ni enega recepta, za implementacijo intraneta v organizacijo, saj se vsaka organizacija srečuje z drugačnimi zahtevami, drugačno organizacijo, tako da se je potrebno pri vsaki uvedbi portala posebej posvetiti posebnostim.

1. Razdelitev intraneta

Preden se lotimo vzpostavitve in razdelitve intraneta, je potrebno vedeti, da morajo za uspešno uvedbo sodelovati različni deležniki v organizaciji. Nekatere izmed nastavitvev, ki jih lahko uvedemo v SharePoint Online, so lahko izvedene samo s strani Globalnega administratorja, druge lahko izvede SharePoint administrator, največ dela

pade pa na »Lastnike mest«. Lastniki mest so seveda druge osebe za vsako oddelčno mesto, mesto delovne skupine,... Naloga lastnika mesta je, da definira čemu točno bo mesto namenjeno in kateri so elementi oz. funkcionalnosti, ki morajo biti prisotni.

Na spodnji sliki, katere avtor je Marc Anderson, je zelo priročen kontrolni seznam opravil in oseb, ki jih morajo izvesti.

Intranet Checklist

- Tenant Level
 - [Create new Intranet Home Site]
 - Set up Home Site
 - Apply theme
 - Make Home Site a Hub Site
 - Set up Hub Site navigation
 - Enable Global Navigation
- Department Level
 - Create department sites
 - Set up wireframed home page
 - Set up pages, lists, and libraries
 - Set up permissions
- Optional
 - Set up Organizational Assets
 - Set up Organizational News
 - [Swap in the new Intranet site]
 - Set up Viva Connections
 - Deploy mobile apps

Typically done by:
• Global Admin
• SharePoint Admin
• Site Owner



V grobem so intranet portalu razdeljeni na tri dele:

- Organizacijski portal
- Oddelčni portali
- Ad-hoc oz. delovne skupine.

1.1 Organizacijski portal

Organizacijski portal lahko rečemo, da je tisti, do katerega imajo dostop vsi zaposleni, ne glede na to, kakšno vrsto dela opravljajo ali v katerem oddelku so zaposleni. Značilnosti tega dela portala so, da v tem sklopu imajo zaposleni večinoma pravice branja in, da se v tem sklopu nahaja mešanica formalnih kot tudi neformalnih informacij

Primeri neformalnih informacij:

- Vreme
- Jedilniki
- Delnice
- Odštevalniki časa
- Hitre povezave

Formalne informacije:

- Obrazci
- Povezave do internih aplikacij
- Poročila/grafi o poslovnih rezultatih
- Navodila za delo

Organizacijski portali so po navadi grafični z veliko slikovnega ali multimedijskega materiala, da je stran vizualno čim bolj privlačna.

1.2 Oddelčna mesta

Značilnosti oddelčnih mest so, da so mesta namenjena predvsem sodelovanju zaposlenih v določenem oddelku. S tem povezane so seveda tudi pravice, saj zaposleni iz drugih oddelkov pravic tu nimajo. Oddelčna delovna mesta so tako lahko

opremljena z novicami oddelka, s predstavitevijo organizacijske strukture oddelka, predvsem pa so tu najpomembnejši dokumenti, ki jih oddelek ustvari.

1.3 Ad-hoc mesta oz. projektna mesta

Zadnji sklop mest, ki ga po navadi obravnavamo pri implementaciji portala, pa so mesta, ki so pripravljena za delovne skupine. Značilnost delovnih skupin je po navadi ta, da skupino sestavljajo zaposleni iz različnih oddelkov, ki na določeni nalogi sodelujejo. Glede na potrebe skupine, se nato tudi določi katere komponente bo določeno mesto vsebovalo. Pri nekaterih primerih gre lahko za čisto projektno mesto z gantogrami, opravili in elementi projektnega vodenja. Spet drugi, so lahko bolj koncentrirani na ustvarjanje dokumentov in npr. sodelovanju z zunanjimi deležniki, kar pomeni da ima skupina tudi zahtevo po možnosti dostopa zunanjih uporabnikov. Slednje je seveda treba upoštevati tudi pri vidiku varnosti.

1.4 MS Teams

Tako za oddelčna mesta, kot tudi za mesta delovnih skupin, je pomembno tudi to, da načrtujemo ali se bodo mesta uporabljala tudi v povezavi s MS Teams, ker to za sabo potegne tudi določene omejitve pri načrtovanju same strukture mest.



Dejan Sarka
MVP, MCT
dsarka@solidq.com



OPTIMIZING ANALYTICAL QUERIES PART 7: INTRODUCING COLUMNAR STORAGE

Various researchers started to think about columnar storage in the eighties. The main idea is that a relational database management system (RDBMS) does not need to store the data in exactly the same way we understand it and work with it. In a relational model, a tuple represents an entity and is stored as a row of a table, which is an entity set. Traditionally, database management systems store entities row by row. However, as long as we get rows back to the client application, we do not care how an RDBMS stores the data.

This is actually one of the main premises of the relational model—we work with data on the logical level, which is independent of the physical level of the physical storage. However, it was not until approximately 2000 when the first attempts to create columnar storage came to life. SQL Server added columnar storage first in version 2012.

Columnar Storage Compression

Storing the data column by column instead of row by row gives you the opportunity to store each column in a sorted way. Imagine that you have every column totally sorted. Then for every equijoin, the merge join algorithm could be used, which is, as you already know, a very efficient algorithm. In addition, with sorted data, you get one more type of compression for free—run-length encoding (RLE) compression.

The following figure explains the idea graphically:



Sorted columnar storage and RLE

An RDBMS in the first step reorders every single column. Then RLE compression is implemented. For example, if you look at the Color column, you don't need to repeat the value Red three times; you can store it only once and store either the frequency of the value or the index in the form from position to position, as shown in the figure.

Note that SQL Server does not implement total sorting. Total sorting of every single column would simply be too expensive. Creating such storage would take too much time and too many resources. SQL Server uses its own patented row-rearranging algorithm, which rearranges the rows in the most optimal way to order all columns as best as possible, with a single pass through the data. This means that SQL Server does not totally sort any column; however, all columns are at least partially sorted. Therefore, SQL Server does not target the merge join algorithm; the hash join algorithm is preferred. Partial sort still optimizes the use of bitmap filters because fewer false positives are returned from a bitmap filter when compared to randomly organized values. RLE compression can still reduce the size of the data substantially.

Recreating Rows from Columnar Storage

Of course, there is still the question of how to recreate rows from columnar storage. In 1999, Stephen Tarin patented the Tarin Transform Method, which uses a row reconstruction table to regenerate the rows. Mr. Tarin called columnar storage the Trans-Relational Model. This does not mean the model is beyond relational; this was more a marketing term, short for Transform Relational Model.

SQL Server documentation does not publish the row recreation algorithm it uses. I am presenting the Tarin's method here. It should still be good enough to give you a better understanding of the amount of work SQL Server has to do when it recreates rows from columns.

The following figure explains the Tarin Transform Method:

Row / Col	1	2	3
	Name	Color	City
1	3	6	4
2	1	4	6
3	6	5	3
4	4	1	5
5	2	2	1
6	5	3	2

Row / Col	1	2	3
	Name	Color	City
1	Bolt [1:1]	Blue [1:2]	London [1:3]
2	Cam [2:2]	Green [3:3]	Oslo [4:4]
3	Coe [3:3]	Red [4:6]	Paris [5:6]
4	Nut [4:4]		
5	Screw [5:6]		
6			

In the figure, the top table is the row reconstruction table. You start reconstructing the rows in the top left corner. For the first column of the first

row value, take the top left value in the columnar storage table at the bottom, in this case the value Bolt. In the first cell of the row reconstruction table, there is a pointer to the row number of the second column in this table. In addition, it is an index for the value of the second column in the columnar storage table. In the figure, this value is 3. This means that you need to find the value in the second column of the columnar storage table with index 3, which is Green. In the row reconstruction table, you read the value 5 in the second column, third row. You use this value as an index for the value of the third column in the columnar storage table, which, in the example, is Paris. In the row reconstruction table, you read the value 1. Because this is the last column in the table, this value is used for a cyclic redundancy check, checking whether you can correctly get to the starting point of the row reconstruction process. The row Bolt, Green, and Paris, the second row from the original table from the previous figure, was successfully reconstructed.

As mentioned, how SQL Server reconstructs the rows has not been published. Nevertheless, you can appreciate that this is quite an intensive process. You can also imagine changes in the original data. Just a small update to the original values might cause the complete recreation of the row reconstruction table. This would simply be too expensive. This is why columnar storage, once it is created, is actually read-only. In SQL Server 2014 and later, columnstore indexes are updateable; however, SQL Server does not update the columnar storage online. SQL Server uses additional row storage for the updates. You will learn further details later in the following articles.

Conclusion

This article brought a theoretical introduction to columnar storage. I will describe the creation process of the columnar storage inside SQL Server in the next article. In addition, I will talk about the development of the columnar storage in SQL Server from version to version.

Šolanje za vaše napredovanje

AKTUALNI TEČAJI

BundleSQL1

Administracija in razvoj baz podatkov na strežniku SQL Server

Kdaj: 12. – 14. 12. 2022
Predava: Dejan Sarka



DP-300

Administering Relational Databases on Microsoft Azure

Kdaj: 30. 1. – 2. 2. 2023
Predava: Dejan Sarka





Gašper Rupnik
MCT, MS, MCSD, MCPS
gasper.rupnik@kompas-xnet.si



.NET MAUI DELAVNICA

2. DEL – NEKAJ OSNOVNIH INFORMACIJ O PLATFORMI

Kot že lahko veste, sem v prejšnji Piki začel s t.i. .NET MAUI delavnico, serijo člankov na temo najnovejše Microsoft platforme za razvoj ene in iste aplikacije, ki jo lahko po želji objaviš kot spletno aplikacijo, namizno aplikacijo ali pa mobilno aplikacijo za poljuben tip pametne naprave.

V prejšnji Piki sem tako govoril o postavitvi razvojnega okolja, danes pa si bomo lahko pogledali par osnovnih temeljev in posebnosti .NET MAUI platforme, katere je potrebno poznati oz. so ključne pri razumevanju platforme.

.NET MAUI XAML vs. .NET MAUI Blazor

Kot prvo se je potrebno zavedati, da obstajata dve poti, po kateri lahko spelješ oz. začrtaš razvoj svoje prve take aplikacije. Razlika je v temu, katero sintakso, kateri jezik uporabiš za razvoj UI – uporabniškega vmesnika oz. samega izgleda tvoje aplikacije.

.NET MAUI XAML predstavlja razvoj UI s pomočjo označevalnega jezika

(primer označevalnega jezika je tudi HTML jezik), ki ga je za nas ustvaril Microsoft že leta 2009 in sliši na ime XAML. Le ta se je pojavil prvotno v Windows Presentation Foundation (WPF) tipih aplikacij, kasneje v Universal Windows Platform (UWP) in sedaj čisto na koncu tudi v WinUI 3, ki ga .NET MAUI uporablja za prikaz aplikacije znotraj Windows operacijskega sistema.

Mimogrede, .NET for Android skrbi na drugi strani za prikaz aplikacije znotraj Android okolja, .NET for iOS je seveda za iOS, .NET for Mac pa skrbi za Mac preslikavo XAML kontrol v ustrezne kontrole ciljne naprave.

Seveda se razvijajo še novi (za Linux, za Tizen itd).



.NET MAUI Blazor predstavlja razvoj UI s pomočjo spletnih tehnologij kot so HTML + CSS + JS + vsa ostala orodja, ki jih želite uporabljati v spletnem razvoju oz. katere ste vajeni (naj bo to node.js in poljuben NPM paket in framework, ki ga preferirate).

Kot vidite lahko izberete eno izmed zgornjih poti, ali preko uporabe spletnih tehnologij, ali pa preko uporabe Microsoftovega označevalnega jezika XAML za razvoj UI. Vsak ima svoje prednosti in slabosti.

.NET MAUI XAML

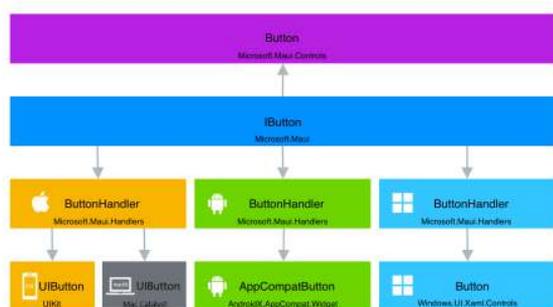
Prednosti:

- Uporabljate kontrole, ki se na koncu pretvorijo v kontrole ciljne naprave, katere ste vajeni – torej izgled, kot ste ga vajeni na napravi preko drugih aplikacij.
- Uporabljate lahko določene že spisane kontrole, ki bi jih morali po .NET MAUI Blazor poti razviti samostojno. Npr. obstajajo že spisane kontrole kot so Barcode scanner, SwipeView kot ste ga vajeni na telefonih da npr. z »slajdanjem« v desno izbrišete nek element v seznamu itd. Seveda to ne pomeni, da se to preko .NET MAUI Blazor platforme ne da narediti ali da mogoče taka komponenta še ne obstaja.

Slabosti:

- Pomanjkanje kontrol: platforma je še vedno zelo nova, kontrole se hitro razvijajo, ampak še vedno lahko pridete do primera, ko točno take, kot bi si jo želeli ne boste našli, ali pa vsaj, da obstoječe kontrole ne morete prilagoditi, da bo delovala po vaših željah.
- Take aplikacije ne moreš objaviti kot spletno stran, saj ne uporabljaš web technology stack (html + css + ...)
- Aplikacija lahko na vsaki platformi izgleda malenkost drugače, saj uporablja specifikke oz. kontrole ciljne platforme (npr. postavitev menujev, izgleda gumbov itd).

Torej, kot lahko iz vsega skupaj že razberete, ima MAUI platforme svoje kontrole, ki se nahajajo znotraj namespacea Microsoft.Maui.Controls in so le te na koncu, ob prenosu na ciljno napravo, pretvorjene v kontrole, ki so specifične za ciljno platformo, kar tudi pomeni, da se obnašajo kot so specificirane v ciljni platformi – torej, kot bi pisali native kodo.



.NET MAUI Blazor

Prednosti:

- Uporablja lahko enak web technology stack, ki ga verjetno že dobro poznate (HTML, CSS, JS + ostalo)
- Ker uporabljate web technology stack lahko tako mobilno ali desktop aplikacijo objavite tudi kot spletno stran.
- Če ne želite uporabljati JS lahko namesto njega uporabljate C# (Blazor)
- Ni potrebnega čakanja na izdelavo manjkajočih kontrol, saj imate tukaj popolnoma proste roke (uporabite lahko karkoli, saj je splet neomejen)
- Design aplikacije je lahko na vsaki napravi popolnoma enak + responsive design uredite preko spletnih tehnologij.

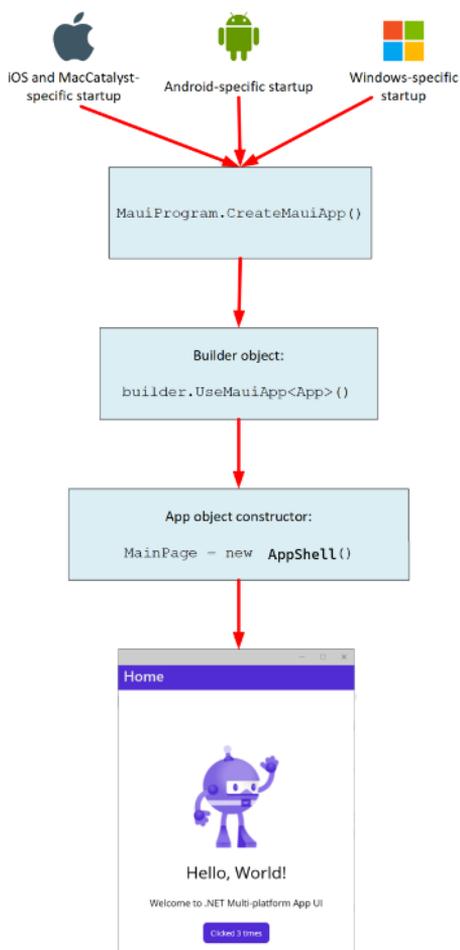
Slabosti:

- Dejansko ni slabosti. Blazor je definitivno nov način razvoja spletnih aplikacij, ki omogoča lažje in učinkovitejše pisanje client-side kode, s pomočjo C# jezika. Obstajajo seveda različne implementacije Blazorja (Blazor Server, Blazor WebAssembly, Blazor Hybrid), ampak več o tem mogoče kdaj drugič. Tukaj je za vas dovolj zgolj to, da se zavedate, da se v MAUIju uporablja Blazor Hybrid.

Hybrid apps with .NET & Blazor



Prerez MAUI aplikacije



MAUI aplikacija se prične v startup kodi, ki je specifična za vsako platformo posebej. Tam veliko večino začetnih nastavitev in konfiguracije pripravi sama MAUI platforma, seveda pa lahko v ta predel tudi sami dodamo marsikaj, ki je specifičnega samo za določeno platformo (npr. kodo prikaza push notifikacij, načina odpiranja dokumentov, ...) – torej uporabljamo neke npr. Android ali Apple specific knjižnice.

Vse skupaj se nadaljuje v skupni začetni točki imenovani MauiProgram, kjer se sestavijo skupne zadeve naše aplikacije kot so npr. registracija nekih fontov po želji, registracija mikroservisov, kateri v naši aplikaciji vsak zase ali povezano opravljajo določene naloge (mikroservis za shranjevanje globalnih nastavitev, mikroservis za delo z podatkovno bazo v telefonu za shranjevanje lokalnih podatkov oz. offline mode, mikroservis za dostop do WebAPIja, preko katerega mobilna aplikacija pridobiva podatke, mikroservis za prikazovanje push notifikacij, mikroservis za pošiljanje e-pošte itd). Vsem, ki ste že kadarkoli programirali kakšno .NET Core aplikacijo vam je verjetno že domače to, da je Program.cs (v tem primeru MauiProgram.cs) fajl tisti začetni, kjer se registrira vse take pomembne skupne servise, ki so potem injectani v vse vaše dele kode preko Dependency Injectiona (DI). Vsi .NET Core projekti se torej začnejo na istem mestu, na enak način. Med njimi so pri MAUI aplikaciji tudi nekateri spisani s strani

Microsofta, ki so nujno potrebni za delovanje – npr. mikroservis za delo z MAUI frameworkom. Za Blazor tip MAUI aplikacije je tu še par specifičnih, kot je mikroservis za uporabo Blazor Web View komponente, uporabo F12 developer konzole v DEBUG modu itd.

Vse skupaj se nadaljuje v Application razredu (znotraj App.xaml datoteke), kjer imamo možnost definirati skupne lastnosti kot so resourci, ki so uporabljeni vsepovsod (npr. app-wide colors, default styles itd). Tu lahko tudi izvedeš »handlanje« nekih skupnih metod kot so začetek naše aplikacije (OnStart), ko gre naša aplikacija v ozadje (OnSleep) ali ko pride nazaj na površje (OnResume). Vse skupaj se nadaljuje v Shell (znotraj AppShell.xaml datoteke), kjer imamo možnost definirati skupno navigacijo – ta je lahko tipa Flyout ali Hamburger menu, lahko je to Tab navigation, ali pa enostavno URI based navigacija.

Od tu pot sledi naprej do prve strani aplikacije (MainPage.xaml), ki pa je dejansko glavni peskovnik naše aplikacije. Tukaj lahko definiramo svoj UI, dodajamo kontrole (slike, labele, gumbe, sezname itd), ki jih zlagamo na več različnih načinov, ki jim pravimo Layouti (StackLayout, Grid, AbsoluteLayout, FlexLayout, ...).

Ker bi lahko govorili o tem še dolgo v dan ali noč, je potrebno na tej točki končati, sicer bi bil ta članek lahko dolg čez celo to Piko. Zato nadaljujemo naslednjič, v novi Pika številki.



Domen Gričar
MCSD, MCSA, MCT
domen.gricar@kompas-xnet.si



JAVASCRIPT NOVOSTI V LETU 2022

JavaScript je programski jezik, namenjen dodajanju dinamičnosti na spletne strani. Standardiziran je po ECMAScript standardu. Standard se konstantno nadgrajuje in posodablja. Zadnja, 13. verzija, ECMAScript 2022, je bila objavljena julija 2022 in prinaša kar nekaj zanimivih novosti, kot so await na najvišjem nivoju, izven async metod, public in private instance polj in metod ter novo metodo at(), ki omogoča iskanje po zbirkah tudi po negativnih indeksih.

Kadar želimo pri asinhronem programiranju počakati rezultat klica, moramo uporabiti besedo await. Beseda await pa se je do sedaj lahko uporabljala le v funkcijah, ki so bile označene kot async. Če smo uporabili besedo await izven async funkcije je brskalnik vrnil napako `SyntaxError: await is only valid in async function.` Kadar je bilo treba uporabiti await na najvišjem nivoju je bilo zato treba narediti anonimno funkcijo, ki smo ji določili, da je async.

```
(async function() {
  const response = await fetch(url);
})();
```

Po novem načinu pa se lahko kliče await na najvišjem nivoju, izven funkcije in bo klic pravilno izveden.

```
const response = await fetch(url);
```

Kadar smo v JavaScriptu želeli dobiti element iz zbirke elementov je bilo vedno treba iskati po indeksu. Ker JavaScript ne pozna negativnih indeksov (pri npr. Pythonu je sintaksa `arr[-1]` za pridobitev zadnjega elementa pravilna), je bilo treba najprej preveriti dolžino zbirke in nato na podlagi dolžine dobiti zadnji element. Ko smo dolžino preverili z lastnostjo `length` smo od le te odšteli 1 za zadnji element, 2 za predzadnji itn.

```
let zadnji = arr[arr.length - 1];
let predzadnji = arr[arr.length - 2];
```

S pomočjo nove funkcije `at()`, pa lahko dobimo direktno s pomočjo

negativnega indeksa.

```
let zadnji = arr.at(-1);
let predzadnji = arr.at(-2);
```

Ker metoda `at()` sprejme tudi pozitivne vrednosti se jo lahko uporablja na splošno, za pridobivanje vseh elementov po indeksih.

```
let prvi = arr.at(0);
let drugi = arr.at(1);
```

Še ena nova funkcija je funkcija `hasOwn()`, ki preveri ali ima objekt lastno, ne podedovano, lastnost z določenim ključem. Prej se je to dalo preveriti preko lastnosti `hasOwnProperty` in se s klicem preverjalo resničnost.

```
const prototip = {
  prototipProp: 'lastnost',
};
const objekt = {
  __proto__: prototip,
  objektProp: 'lastnost',
}
let nepodedovanaLastnost =
Object.prototype.hasOwnProperty
if (nepodedovanaLastnost.call(objekt,
'prototipProp')) {
  //
}
```

Z novim načinom pa se lahko preverja direktno z metodo.

```
if (Object.hasOwn(objekt,
'prototipProp')) {
  //
}
```

JavaScript je precej sproščen jezik, zato je tudi nekaj novosti, ki dodajajo k njegovi strogosti. Z ES2022 je dodan način pisanja privatnih lastnosti in metod s pomočjo znaka `lojtra(#)`, ki se ga doda pred ime spremenljivke.

```
class Test {
  #zasebnaLastnost = 'Vrednost';
  #zasebnaFunkcija(besedilo) {
    this.besedilo = besedilo;
  }
}
const novalnstanca = new Test();
test.zasebnaLastnost;
test.zasebnaFunkcija('Vrednost');
```

V tem primeru `zasebnaLastnost` vrne `undefined`, `zasebnaFunkcija` pa `this` is not a function.

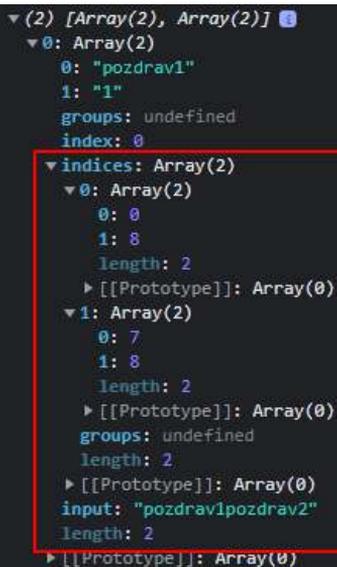
Poleg načina za pisanje zasebnih spremenljivk, je dodana tudi beseda `static` za pisanje statičnih spremenljivk in metod.

```
class Test {
  static ime = 'Ime';
  static pozdrav(){
    console.log('Pozdrav');
  }
}
Test.firstName;
Test.pozdrav();
```

Še ena novost je možnost /d pri regularni izrazih. S pomočjo te lastnosti dobimo objekt, ki zabeleži začetno pozicijo in indeks vsake zajete skupine.

```
const besedilo = 'pozdrav1pozdrav2';
const regexStr = /pozdrav(\d)/g;
const regexStr2022 = /pozdrav(\d)/dg;
console.log([...besedilo.matchAll(regexStr)]);
console.log([...besedilo.matchAll(regexStr2022)]);
```

V tem primeru nam vsak vrnjen objekt dodatno vrne tudi lastnost `indices` z zbirkami o različnih ujemanjih izraza.



```
▼ (2) [Array(2), Array(2)]
  ▼ 0: Array(2)
    0: "pozdrav1"
    1: "1"
    groups: undefined
    index: 0
    ▼ indices: Array(2)
      ▼ 0: Array(2)
        0: 0
        1: 8
        length: 2
        ► [[Prototype]]: Array(0)
      ▼ 1: Array(2)
        0: 7
        1: 8
        length: 2
        ► [[Prototype]]: Array(0)
    groups: undefined
    length: 2
    ► [[Prototype]]: Array(0)
  ► [[Prototype]]: Array(0)
```

Pri razvoju je pomembno, da si v konzolo sproti izpisujemo napake. Še ena novost je `cause` lastnost pri zapisovanju napak. S pomočjo `cause` lastnosti si lahko bolj instinktivno zapišemo napake. Do razloga dostopamo preko `stack trace-a` in ga dobimo preko `error.cause`.

```
const funkcija = () => {
  try {
    //
  } catch (err) {
    throw new Error('Prišlo je do napake',
      {cause: err});
  }
}

try {
  const rezultat = funkcija();
} catch (err) {
  console.error(err, err.cause);
}
```

Nove funkcionalnosti v JavaScriptu so uporabne in lahko olajšajo delo, ker pa se jezik pogosto spreminja, včasih niso podprte v vseh brskalnikih. Naštete novosti delujejo v novejših posodobitvah modernih brskalnikov, če pa mora biti naša aplikacija kompatibilna tudi s starejšimi, pa je bolje, da najnovejših funkcionalnosti ne uporabljamo.



Alen Dedić
alen.dedici@kompas-xnet.si

METODA »LEAN DEVELOPMENTS« V PRAKSI

Eden izmed bolj poznanih pristopov razvijanja programske opreme v industriji je »lean development« oziroma »vitki razvoj«, katerega se poslužujejo predvsem manjša podjetja in ekipe, ki hočejo relativno hitro dostaviti končan in delujoč projekt naročnikom. Ker so razpoložljiva sredstva v takšnih okoliščinah dokaj omejena (bodisi število razvijalcev, ki so na razpolago in/ali skromen proračun, s katerim podjetje operira), je primaren fokus v metodi vitkega razvoja hitrost, zaradi katere pa kvaliteta dostavljene programske opreme ne sme trpeti. Za doseg tega cilja pa je potrebno postaviti jasne cilje in točno opredeliti katere funkcionalnosti aplikacije so ključnega pomena, in katere se lahko postavi na stranski tir ter se jih implementira kasneje.

Da bi metoda vitkega razvoja dosegla svoj namen, je najprej potrebno določiti katere tehnologije se bodo uporabile za razvoj aplikacije (frontend, backend in podatkovna baza), nato pa smotrno usmeriti razvijalce v ekipo in jim določiti

ustrezne naloge in odgovornosti na podlagi njihovih sposobnosti ter znanj. Ko je to vse usklajeno, se lahko ekipa loti postavljanja okolja za razvoj, in se loti prvih nalog potrebnih za predajo MVPja (minimal viable product) stranki. Zadnji faktor, ki je izjemno pomemben za uspešen vitki razvoj, pa je komunikacija v ekipi. Že zelo zgodaj v procesu razvoja pride do situacij, kjer je en razvijalec v ekipi odvisen od drugega. Npr. backend razvijalec, zadolžen za implementacijo funkcionalnosti, ki poskrbi da pri vsakem vpisu v aplikacijo, sistem preveri do katerih strani ima dotična vloga pravico dostopa, ostale pa uporabniku skrije, mora imeti v projektu že spisane vse strani, ki so prikazane v meniju aplikacije, kar je pa dolžnost front end razvijalca.

V teoriji se vitki razvoj sliši idealno – cilji jasno zastavljeni, ekipa zbrana, tehnologija določena, MVP bo stranki dostavljen v treh mesecih. A kot vemo, se stvari v praksi ne odvijajo vedno tako kot narekuje teorija. Zato je ključno se izogniti nekaterim pastem. Seveda je izjemno pomembno, da

idejni vodja ne uničuje načrta v katerem so določene ključne funkcionalnosti, da lahko razvijalci nemoteno in hitro razvijajo projekt, a kaj kmalu lahko pride do problema, če eden izmed razvijalcev v ekipi iz kakršnegakoli razloga preneha s sodelovanjem na projektu. Potrebno je hitro reagirati in v najkrajšem možnem času poiskati ustrezno zamenjavo. Nato se mora novi razvijalec dodobra spoznati s projektom, kar je vedno izziv za vsakega programerja, ki začne s sodelovanjem na projektu, pri katerem je že dosti stvari narejenih. Na podlagi mojih izkušenj pa cel ta proces traja tudi 2 meseca, odvisno iz izkušenj novega člana ekipa.

Do zapletov pa lahko pride tudi, ko je potrebno implementirati funkcionalost, ki temelji na specifični komponenti, o kateri pa sploh ni veliko dokumentacije in jo obvladajo je redki. Hkrati pa gre pa za ključno funkcionalost, okoli katere temelji uporabnost celotne aplikacije. Seveda bi bil optimalen zaključek, da se razvijalska ekipa dodobra spozna s to komponento, in implementira vse željene funkcionalnosti, a to seveda ne pripomore pri hitrosti razvoja, kar je cilj vitkega razvoja. Če imaš srečo, najdeš nekoga, ki je že večkrat delal s takšno komponento, in s svojimi izkušnjami izjemno pripomore pri

razvoju tega dela aplikacije, razvojna ekipa pa se lahko osredotoči na ostale zastavljene cilje.

Potrebno je biti tudi pazljiv, da se redno komunicira s stranko, ter se ji redno pokaže napredek aplikacije, da lahko stranka vidi kako je stvar zastavljena. Če stranka pove, da si je določene zadeve drugače zamislila, je veliko bolje biti o tem obveščen čimprej, da se lahko naredijo hitri popravki, dokler je ekipa še v fazi razvoja, kjer bi lahko to hitro opravila. Nobena stran si namreč ne želi, da stranka dobi v roke že skoraj dokončan MVP, nakar pa razvijalci izvejo, da bi stranka rada imela nekatere strani čisto drugače zastavljene, za kar bi se potrebovalo nekaj tednov dela, da se vsi ti popravki uredijo po želji naročnika.

V kolikor se ekipa drži zastavljenih ciljev za MVP, poskrbi da so vsi razvijalci osredotočeni na projekt in zadovoljni z delovnim procesom in komunikacija v ekipi in s stranko teče redno in proaktivno, se lahko metoda »lean developmenta« izkaže za zelo uspešno. V kolikor se pri eni izmed zgoraj naštetih zadev kaj zalomi, se pa lahko cela zadeva zelo hitro obrne v dokaj kaotično situacijo za vse vpletene.



Premzl Urška
urska.premzl@kompas-xnet.si

ZA VEČJO STOPNJO KONVERZIJE JE POTREBNO IZBOLJŠATI UPORABNIŠKO IZKUŠNJO (UX) VAŠEGA SPLETNEGA MESTA

UX dizajn je hrbtenica vsakega spletnega mesta.

Uporabniška izkušnja (UX design) je proces povečevanja zadovoljstva uporabnikov z izboljšanjem uporabnosti, učinkovitosti in dostopnosti spletne strani. Prav tako je stopnja konverzije neposredno sorazmerna z UX zasnovo spletnega mesta.

Vendar dober dizajn UX vključuje veliko več kot le intuitiven uporabniški tok in osupljivo postavitev. Odlična uporabniška izkušnja bo takoj izboljšala izkušnjo brskanja uporabnikov. Ta pa bo povzročila več obiska strani in povečanje konverzij.

4 načini, kako izboljšati uporabniško izkušnjo vašega spletnega mesta:

1. Oblikujte gumbe, ki bodo 'pozivali k akciji' (Call to Action)

Call to Action Button so gumbi, ki se uporabljajo na spletnem mestu za usmerjanje uporabnika k povečanju konverzije.

Najpogostejši so: Pričnite preiskusno obdobje (30-dni), prijava na posodobitve, prenos aplikacije, rezervacija posveta in številni drugi.

Uporaba jasnega in privlačnega gumba Call to Action izboljša uporabniško izkušnjo katere koli spletne strani. Atribut Call to Action naj bo nameščen na vsaki strani spletnega mesta. Ugotovljeno je bilo, da imajo spletna mesta z uporabo tega gumba višjo stopnjo konverzije.

Pri načrtovanju gumba upoštevajte: Gumb mora biti dobro vidljiv zato je barva pomembna. Z uporabo barv lahko gumb izstopa in jim daje večjo pomembnost. Zato uporabite kontrastne barve v primerjavi z barvno shemo celotne spletne strani. Besedilo mora biti usmerjeno k dejanjem. Izogibajte se uporabi pasivnih glagolov v besedilu. Besedilo mora biti subtilno, a dovolj aktivno, da uporabnika pozove k zahtevanemu dejanju.

Pri pisanju besedila Call to Action naj število besed ne presega pet besed.

2. Odpravite napako 404

Med iskanjem uporabniki običajno pričakujejo, da bodo pristali na natančno določeni strani, ki so jo iskali. Če pride do napake, v večini primerov napake 404, se bodo pomaknili na drugo spletno mesto v iskanju hitrejše storitve.

Napake 404 lahko uporabnike odvrnejo od vaše spletne strani. Razumem pa, da napake 404 niso povsem neizogibne. Kako se torej lotiti problema?

Prvi korak je, da ugotovite, za katera iskanja so prikazane napake 404, in jih nato čim prej popravite.

Namesto da dovolite, da se vaše spletno mesto pomakne na standardno stran »Napaka 404: strani ni mogoče prikazati«, prilagodite sporočila o napakah, tako da bodo uporabnikom prijazna in privlačna. Uporabite ustrezne, zabavne in prijetne slike na strani z napako.

Dajte uporabnikom jasno vedeti, da boste zagotovili udobno in nemoteno izkušnjo brskanja. Če želite to doseči, prilagodite besedilo napake in mu dodajte osebno noto.

Čeprav sporočil o napakah ni mogoče popolnoma odpraviti, boste z odpravo

teh napak, ki jih ni bilo najdeno, pripeljali vašo spletno stran korak bližje oblikovanju dobre uporabniške izkušnje.

3. Uporabite izvirne fotografije

Slike takoj povzdignejo vsako spletno stran in naredijo vsebino vizualno privlačno. Toda vrsta slike, ki jo izberete, lahko povzroči, da je celoten dizajn strani videti dober ali slab.

Pravilo — vedno se izogibajte brezplačnim fotografijam iz drugih spletnih strani.

Izvirne fotografije pritegnejo več obiskovalcev, saj imajo do njih realističen pristop in se lahko uporabnik z njimi poveže. Medtem ko so brezplačne fotografije iz spletnih strani, ki si jih prenese več ljudi, ne pritegne uporabnikov.

Še ena stvar, ki jo morate imeti v mislih, je, da bo uporaba brezplačnih fotografij sporočala, da niste vložili veliko truda v oblikovanje spletne strani. Prav tako ne bo videti edinstveno, saj so ga morda že videli kje drugje.

Zato vedno uporabljajte pristne slike, ne glede na to, kako osnovne in preproste so videti.

4. Hitrejši čas nalaganja strani

Če se vaša spletna stran nalaga



počasi, bo to razočaralo uporabnike, ki bodo sčasoma zapustili vaše spletno mesto in prešli na drugo spletno mesto, ki ima hitrejši čas nalaganja.

Opaženo je bilo, da če se vaša stran nalaga več kot 2 sekundi, uporabnik zapusti spletno stran. Prav tako se uporabniki glede na čas nalaganja vašega spletnega mesta odločijo, ali bodo nadaljevali z obiskom vašega spletnega mesta ali prešli na drugo spletno mesto.

Ne pozabite, da samo izboljšanje hitrosti nalaganja strani spletnega mesta za namizne računalnike ni dovolj. Pomembno je optimizirati tudi za mobilne uporabnike. Glede na to, da je na Googlu na prvem mestu uporaba mobilnih naprav, je treba optimizirati in oblikovati svojo spletno stran za mobilne uporabnike brez napak.

Obstaja veliko spletnih orodij, s katerimi lahko preverite, ali vaše

spletno mesto zagotavlja hitro uporabniško izkušnjo na vseh platformah. Eden izmed njih je Googlov PageSpeed Insights. Če s tem orodjem vnesete URL spletnega mesta, ki ga želite preveriti, bo Google poudaril področja, kjer je hitrost vaše strani šibka, in vam dal predloge, kako lahko izboljšate.

Odlična uporabniška izkušnja ni samo zagotavljanje koristnih informacij. Namesto tega vključuje zagotavljanje koristnih informacij na privlačen in prijeten način. Ne glede na to, kako dobra je vaša storitev ali izdelek; če ne pritegne pozornosti uporabnikov, ni dovolj dober.

Zato vložite čas in trud v ustvarjanje privlačnega dizajna spletnega mesta. Bodite prepričani, vaša osredotočenost in prizadevanja za izboljšanje uporabniške izkušnje vašega spletnega mesta bodo zagotovo ustvarili več prometa in izboljšali stopnje konverzije.



Aleš Lipušček
MCP, MCTS, MCITP
ales.lipuscek@kompas-xnet.si



POWERSHELL KOTIČEK

Grafikoni so priročno orodje za hiter in strnjen pregled velike količine informacij, primerjavi podatkovnih naborov in prikaz trendov v grafičnem načinu.

Tega dela se seveda lahko lotimo tudi v Powershellu.

Najprej moramo referencirati knjižnice, s pomočjo katerih bomo graf zgradili:

```
Add-Type -AssemblyName
System.Windows.Forms
Add-Type -AssemblyName
System.Windows.Forms.DataVisualization
```

Nato moramo pripraviti instance razredov Chart, ChartArea, Series and Legend

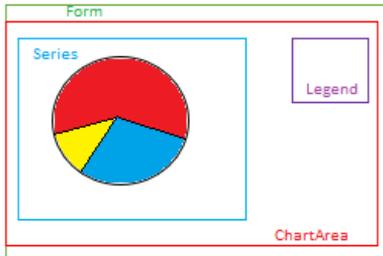
```
$chart = New-Object
System.Windows.Forms.DataVisualization
ion.Charting.Chart
$chart_area = New-Object
System.Windows.Forms.DataVisualization
ion.Charting.ChartArea
$series = New-Object
System.Windows.Forms.DataVisualization
ion.Charting.Series
```

```
$legend = New-Object
System.Windows.Forms.DataVisualization
ion.Charting.Legend
```

Chart objekt je tisti, na katerega se povezujejo vsi ostali. ChartArea uporabimo za nastavitve mreže, osi... S Series pa določimo podatke za prikaz pa tudi obliko grafa. Posamezen Chart objekt lahko vsebuje več ChartArea objektov, vsak od teh pa več podatkovnih Serij (seveda nekateri tipi grafov podpirajo samo enega)

Sedaj, ko imamo objekte pripravljene, jih opremimo z lastnostmi:

```
$chart.Series.Add($series)
$chart.ChartAreas.Add($chart_area)
$chart.Legends.Add($legend)
$chart.Dock =
[System.Windows.Forms.DockStyle]::Fi
ll
Seveda ne sme manjkati velikost,
širina, barve in naslovi.
$Chart.Width = 700
$Chart.Height = 400
$Chart.Left = 10
```



```
$Chart.Top = 10
$Chart.BackColor =
[System.Drawing.Color]::White
$Chart.BorderColor = 'Black'
$Chart.BorderDashStyle = 'Solid'
```

```
$ChartTitle = New-Object
System.Windows.Forms.DataVisualization.Charting.Title
$ChartTitle.Text = 'Test'
$Font = New-Object
System.Drawing.Font @('verdana','12',
[System.Drawing.FontStyle]::Bold)
$ChartTitle.Font = $Font
$Chart.Titles.Add($ChartTitle)
```

```
Nato nastavimo tip grafa in zapolnimo
z dvema nizoma: imeni in vrednostmi:
$series.ChartType =
[System.Windows.Forms.DataVisualization
```

```
tion.Charting.SeriesChartType]::Pie
$series.Points.DataBindXY(@('a', 'b',
'c'), @(2, 6, 4))
```

Na voljo nam je več kot 30 vrst grafikonov. Delo z nekaterimi je lažje (npr. tortni in kolobarni imata precej skupnih točk pristopa, zato je preklap med njima enostaven), dočim drugi zahtevajo drugačen pristop do podajanja podatkov (series).

Na koncu še pripravimo formo, nanjo »prilepimo« graf, in vse skupaj prikažemo

```
$window = New-Object
System.Windows.Forms.Form
$window.Text = "Graf"
$window.Controls.Add($chart);
$window.ShowDialog();
```

Chart objekt ima tudi to lepo lastnost, da pozna metodo `Savelmage`, ki nam izrisan graf shrani z določenim imenom in tipom slike.

Šolanje za vaše napredovanje

AKTUALNI TEČAJI

AZ700

Designing and Implementing Microsoft Azure Networking Solutions

Kdaj: 4. – 6. 1. 2023
Predava: Jože Markič



MS203

Microsoft 365 Messaging

Kdaj: 5. – 9. 12. 2022
Predava: Miha Pihler





Petra Militarev
Vodja izobraževanj
petra.militarev@kompas-xnet.si

ZAUPANJE, PREPOZNAVANOST, PREDANOST, SPOSOBNOST,..

Vsi se zavedamo, da je znanje, ki ga imajo zaposleni ključno za uspešno, učinkovito in produktivno poslovanje vsake organizacije. Z digitalnim okoljem, ki se bliskovito spreminja so IT certifikati, znanje in pripravljenost bolj dragoceni kot kdaj koli prej, tako za posameznega zaposlenega kot za IT organizacijo. Oblak, hibridno delo, napredek pri upravljanju podatkov, varnost, vse to zahteva veliko bolj spretno in usposobljene IT ekipe.

Z Microsoftovimi certifikati posamezniki in organizacije izkazujejo, da ste v koraku z današnjimi tehničnimi vlogami in zahtevami. Za številna IT delovna mesta obstajajo različne poti in nivoji Microsoft certificiranja.

Poglejmo najprej še nekaj razlogov, zakaj bi se certificirali.

Prednosti certificiranja za posameznika:

- Povečana prepoznavnost – Microsoftove certifikate izda svetovno ugledna organizacija z velikim tržnim deležem v niši

poslovne programske opreme in poslovnega razvoja.

- Poudari vašo predanost osebnemu in poklicnemu razvoju
- Poudari vašo sposobnost reševanja tehničnih težav.
- Omogoča vam dostop do ekskluzivnih Microsoftovih virov (kot npr. vpogledi v nadgradnje infrastrukture in njihovo uporabo).

Earning certifications can lead to a variety of personal benefits:

91%

Feel increased confidence

84%

Get a boost in determination to succeed professionally

76%

Receive greater respect from their peers

76%

Achieve greater satisfaction in their jobs

74%

Are entrusted with more work autonomy and independence

Vir: [Pearson VUE, "2021 Value of IT Certification," 2021](#)

Prednosti certificiranja za podjetja:

- Poveča zaupanje strank – s certificiranim osebjem potencialnim strankam pokažete, da je vaše podjetje na tekočem z dogajanjem v IT panogi.
- Poudarja profesionalne standarde podjetja.
- Povečuje raven produktivnosti – študije dokazujejo, da so zaposleni, ki pridobijo poklicne certifikate na splošno bolj produktivni.
- Poveča vaše zaupanje v zaposlene – uspešna podjetja odpravijo potrebo po mikrovodenju vseh poslovnih operacij, ko verjamejo v svoje zaposlene.

Kandidati in organizacije pristopajo k izpitom in certifikaciji na različne načine. Skoraj vedno pa se začnejo priprave z ustreznim usposabljanjem. Večino tečajev, ki pokrivajo vsebine posameznega izpita najdete razpisane tudi na [našem urniku](#), kjer vam certificirani predavatelji v živo nudijo tudi vso podporo. Do znanja lahko pridete tudi v svojem tempu, s samoizobraževanjem preko naše Skills Akademije, kjer je tudi možnost Q&A konzultacij.

Tokrat izpostavljamo 6 Microsoft certifikacij, ki so vezane na IT delovne vloge in jih lahko osvojite z našo pomočjo:



Administrator
Administrators implement, monitor,



Developer
Developers design, build, test, and maintain cloud solutions.



Business User
Use Microsoft applications to move your business forward.



Data Analyst
Data analysts make data easy to understand through designing an...



DevOps Engineer
DevOps engineers combine people, process, and...



Security Engineer
Security engineers implement security controls and threat...

Kot vaš učni in izpitni partner naše storitve vključujejo tudi:

- da z vami ustvarimo prilagojene učne programe,
- da uskladimo izvedbo usposabljanja z vašimi zahtevami,
- vsi naši udeleženci na uradnih Microsoft tečajih prejmejo vavčer za brezplačno opravljanje izpita,
- nudimo vam akcijske pakete za zakup večjega števila Microsoft izpitov (akcija velja do marca 2023).

Vstopite na pot Microsoft certifikacije že danes! Vabljeni da se nam pridružite na enem izmed prihajajočih tečajev:

Administrator: [MS-700 Managing Microsoft Teams](#)

Developer: [MS-600 Building Applications and Solutions with Microsoft 365 Core Services](#)

Business User: [MS-900 Microsoft 365 Fundamentals](#)

Data Analyst: [PL-300 Microsoft Power BI Data Analyst](#)

DevOps Engineer: [AZ-400 Designing and Implementing Microsoft DevOps Solution](#)

Security Engineer: [AZ-500 Microsoft Azure Security Technologies](#)