

PIKA

AKTUALNOSTI S PODROČJA IT STORITEV | september 2026, št. 4, letnik XXIX

**DESIGN TOKENS &
AVTOMATIZACIJA PREHODA IZ
FIGME V KODO**

stran 7

**SHAREPOINT ONLINE PREMIUM: CONTENT
UNDERSTANDING IN DOCUMENT
PROCESSING – SAMODEJNO PRIDOBIVANJE
METAPODATKOV Z AUTOFILL COLUMNS**

stran 15

**NASVETI ZA HITREJŠE DELOVANJE
SPLETNE STRANI**

stran 30

**ZAKAJ ORGANIZACIJE
POTREBUJEJO »VOZNIŠKI IZPIT«
ZA UPORABO UMETNE
INTELIGENCE?**

stran 37

Xnet
vaš zanesljiv IT partner



IT strokovnjaki

Infrastruktura in
oblačne storitve

Podatkovne baze
in analitika

Poslovne
aplikacije in
avtomatizacija

Upravljanje in
varnost
podatkov

Upravljanje
uporabnikov in
naprav

Razvoj in
integracije

Umetna
inteligenca in
avtomatizacija
procesov



Pokrivamo širok nabor izobraževanj za IT strokovnjake in končne uporabnike. Vsa izobraževanja lahko prilagodimo potrebam vaše organizacije.

📍 Kompas Xnet d.o.o., Stegne 7,
1000 Ljubljana

☎ 01 5136 990

✉ info@kompas-xnet.si

🌐 kompas-xnet.si

Osnove Microsoft
365 & Office

Excel & Power
BI

SharePoint &
OneDrive

Umetna
inteligenca

Digitalna higiena
in varnost

Power Automate
& Power Apps

Projektni
management



Končni uporabniki

KAZALO

KOLOFON

ISSN: 1408-7863
Kompas Xnet d.o.o.
Stegne 7
1000 Ljubljana

Telefon: 01 5136 990
Email: info@kompas-xnet.si
Web: <https://www.kompas-xnet.si>

Direktorica:
Branka Slinkar

Urednik in oblikovalec:
Aleš Frelih

Člani uredništva:

Branka Slinkar, Aleš Lipušček, Robi Vončina,
Klemen Vončina, Domen Gričar, Petra
Militarev, Aida Kalender Avdić, Lenart Svetek,
Dijana Babić, Žiga Brinc, Jan Žugelj

POVEŽIMO SE:

-  facebook.com/KompasXnet/
-  twitter.com/kompasxnet
-  linkedin.com/company/kompas-xnet-d-o-o-/
-  linkedin.com/showcase/thriveconf/
-  youtube.com/channel/UCfoMnj355AbRMeG66ePS8BQ

7

UX/UI

Design Tokens & avtomatizacija prehoda iz Figma v kodo

Aleš Frelih
UX/UI oblikovalec, projektni vodja

11

ADMINISTRACIJA

Powershell kotiček

Aleš Lipušček
MCP, MCTS, MCITP

15

SHAREPOINT

SharePoint Online Premium: Content Understanding in Document Processing – samodejno pridobivanje metapodatkov z Autofill Columns

Robi Vončina
MVP, MCT, MCITP, MCSA, MCTS

22

POWER AUTOMATE

Power Automate – HTTP GET na SharePoint seznamu z več kot 5000 elementi

Klemen Vončina
Microsoft Office Specialist Master, MCT

25

RAZVOJ

Azure Functions

Lenart Svetek
Frontend programer

28

RAZVOJ

Two-Factor Authentication (2FA)

Žiga Brinc
Frontend programer

30

RAZVOJ

Nasveti za hitrejšo delovanje spletne strani

Domen Gričar
Frontend programer, MCSA, MCT

KAZALO

33

PIKIN VITAMIN

Ali ste vedeli?

Dijana Babić
Računovodstvo

37

DRUGO

Zakaj organizacije potrebujejo »vozniški izpit« za uporabo umetne inteligence?

Petra Militarev
Vodja izobraževanja

40

DOGODKOVNIK

Tečaji, webinarji, seminarji in drugi pomembni prihajajoči dogodki

Kolofon

“

Vsak vikend eden
Da bo čimbolj uravnoteženo
Avtomatik je prav v redu
Poskrbi, da je pravočasno
Tokrat odkriva drug otok
Novi bo lepši in boljši
Še zadnji dopustnik
Čitanje te čisto omaga 🤗
Moramo jih napolnit
O vadbah ve vse
V online je čisto doma
Po malem še Italijan
Zaključuje magistrsko
Spet rešuje vdor
V vodo pa samo, če pade
3 tedne je čisto malo

Robi
Dijana
Aleš F.
Domen
Petra
Petra
Klemen
Žiga
Aida, Petra
Aleš
Klemen
Jože
Jan
Miha
Tomaž
Robi



Matic Vuković
predavatelj

TEČAJ M365-SK



Microsoft 365 v praksi: Sodelovanje in komunikacija

📅 8.9.2026 ⌚ 7 ur 🗣️ Osnovni

✉️ info@kompas-xnet.si ☎️ +386 1 5136 990
🌐 kompas-xnet.si/izobrazevanja/koledar-tecajev-t

Jesenska akademija

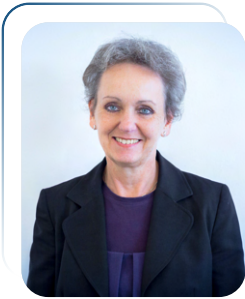
Microsoft Copilot

Manj rutine. Več časa za
pomembno delo.

📅 15.10., 22.10., 29.10., 5.11.2026
🕒 od 13:00 – 15:30

👉 Na daljavo

Oglejte si program →



UVODNIK

BRANKA SLINKAR

Direktorica

Srčno upam, da ste imeli lep dopust, predvsem pa, da ste si lahko resnično oddahnili, prebrali kakšno dobro knjigo, ..., naredili / obiskali nekaj, kar je bilo že dolgo na »TO DO« seznamu. Upam, da vam tudi peklenska vročina ni prišla preveč do živnega, da ste imeli primerno osvežitev vselej na dosegu roke.

Pri nas smo večinoma že nazaj z dopustov, polni energije in novih idej. Sedaj z velikim veseljem pričakujemo, da nas počastite z obiskom na katerem od tečajev ali nas povabite k sodelovanju na vaš IT projekt.



Kot veste, je naša izobraževalna ponudba pestra in pokrivamo praktično vse Microsoft (a ne izključno) tehnologije. V programu imamo tako šolanja za končne uporabnike, kot tudi za IT strokovnjake.

Veliko izvedb prilagodimo specifičnim potrebam naročnika, saj je tako dodana vrednost delavnice veliko večja. Prilagojene delavnice za uporabnike so še posebej dobro sprejete, saj se udeleženci veliko lažje naučijo novih funkcionalnosti, če vadijo na primerih iz svoje prakse.

Zato vas vabim, da našim svetovalkam zaupate svoje cilje glede nivoja znanja sodelavcev. Na podlagi t.i. uvrstitvenih testov in s pomočjo izkušenih predavateljev bomo pripravili zanimive in učinkovite delavnice, ki bodo ljudem dala potrebna znanja. Zagotavljamo vam, da bodo udeleženci z izvedbami delavnic zelo zadovoljni. V nasprotnem primeru vam šolanja ni potrebno plačati.

Tudi vaši IT sodelavci lahko pri nas nadgradijo svoje znanje in se ustrezno pripravijo za prihajajoče izzive. Vsi naši predavatelji so vrhunsko usposobljeni in prihajajo iz prakse. Zato so šolanja toliko boljša, z veliko dodano vrednostjo, saj predavanja obogatijo z realnimi primeri iz dolgoletne prakse in praktičnimi napotki za vsakdanje delo. Vse to, česar v učbenikih ne boste našli – neprecenljivo!

V primeru, da so vaše potrebe po znanju zelo napredne ali posebej specifične, vam lahko pomagamo do ustreznega šolanja, tudi v okviru mednarodnega združenja IC - [LLPA](#), katerega član smo in hkrati predstavnik za Slovenijo.

V sodelovanju s članicami LLPA so vam na voljo še programi ostalih vendorjev, kot so AWS, Cisco, Oracle, EC-Council, ISQI, Vmware, Linux Professional Institute (LPI)...

Kot rečeno, bo šolanje izjemnega pomena tudi v prihodnje. Morda celo bolj kot do sedaj, če se bomo želeli ogniti hujšim napakam in slediti novostim. Vabimo vas, da se posvetujete z našo ekipo, da vam predlaga učni program, ki bo najbolj primeren za vas in vaš kolektiv. Pomagali vam bomo, da bo vaša učna pot optimalna in čas kar najbolj učinkovito porabljen.

Vsekakor pa spremljajte naša obvestila in [Koledar tečajev >](#), ki ga redno posodabljam.

Konec septembra bo v Portorožu potekala NTK, na kateri bosta Robi in Klemen ponovno delila svoje znanje in izkušnje. Dodatno pa bodo na NTK odru tudi nekateri predavatelji, ki jih dobro poznate s [Thrive konference](#): Scott, Mustafa, Saša, Frane, Nenad, ...

Vabimo vas, da si že sedaj **rezervirate**

čas za naš dogodek, 16. Thrive konferenco, ki bo **25. – 26. 5. 2027**. Kot vedno lahko pričakujete ogromno zanimivih predavanj, odlično predavateljsko zasedbo in bogat obkonferenčni program. Prisrčno vabljeni!

Ko boste potrebovali partnerja za vaš Microsoft projekt, nas povabite k sodelovanju, saj imamo priznane kompetence za kar 5 področij



Naj bo **Xnet** **vaša prva izbira, ko gre za IT rešitve in storitve**. Microsoft tehnologije so naša strast že prek 30 let. Nismo številčna ekipa, imamo pa zato **»jagodni izbor« strokovnjakov**, ki bodo kos vsem izzivom vašega projekta.



Lep dan in vse dobro vam želimo
Branka Slinkar



UX/UI oblikovalec, projektni vodja
ales.frelih@kompas-xnet.si



morali razbirati fiksne vrednosti, kot so
 HEX zapis barv: #0078D4, zaobljenost
 robov: border-radius: 8px, senčenje
 elementov: box-shadow, in druge, ter
 jih ročno prepisovati v svojo izvirno
 kodo.

Kaj sploh so Design Tokens?

Design Tokens niso zgolj preimenovane barve ali CSS spremenljivke. So **najmanjši atomi** -

7

delci oblikovalskega sistema, ki nosijo določeno oblikovsko odločitev v platformno nevtralnem formatu (najpogosteje JSON ali YAML).

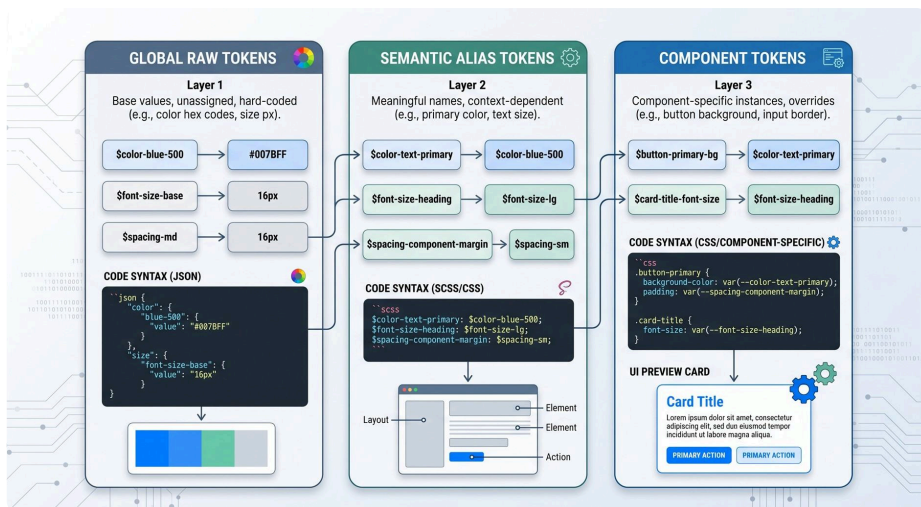
Troslojna arhitektura žetonov:

- 1. Globalni (Raw) žetoni:** Absolutne vrednosti brez konteksta (npr. color.blue.500: #0078D4, space.4:16px).

Figma / Tokens Studio (JSON):

JSON

```
{
  "color": {
    "brand": {
      "action": { "$value":
        "{color.blue.500}" }
    }
  }
}
```



- 2. Semantični (Alias) žetoni:** Žetoni s pomenom in namenom (npr. color.background.primary: {color.neutral.50}, color.brand.action: {color.blue.500}). Omogočajo preprosto preklapljanje tem (Dark/Light Mode).

- 3. Komponentni žetoni:** Specifične vezave za posamezen gradnik vmesnika (npr. button.primary.bg: {color.brand.action}, card.padding: {space.4}).

Samodejno generiran CSS (Front-end):

```
:root {
  --color-brand-action: #0078d4;
}
.pika-button-primary {
  background-color: var(--color-brand-action);
  padding: var(--space-4);
}
```

Avtomatizirana veriga: Od Figue do repozitorija

Prava moč žetonov se pokaže z vzpostavitev avtomatiziranega cevovoda (CI/CD za dizajn):

Primerjave takšnega zapisa (JSON > CSS)

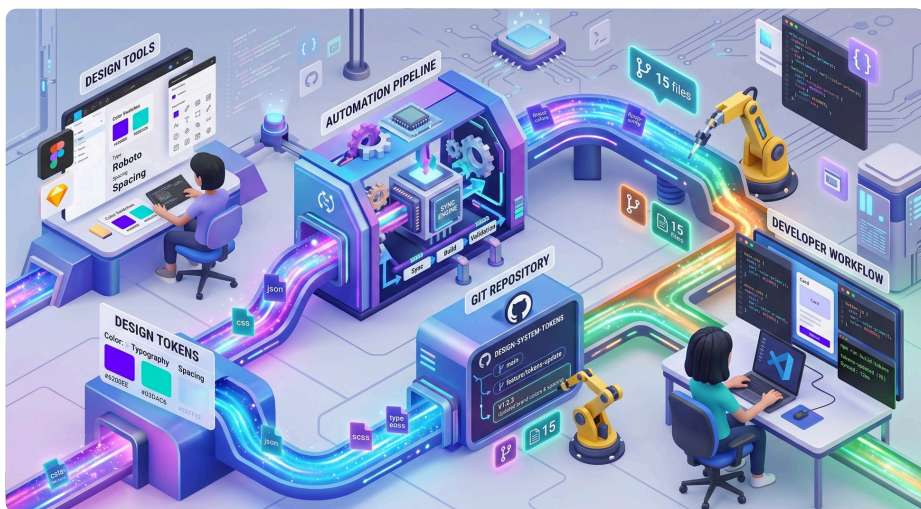
Faza 1 (Definicija): Oblikovalec v orodju Figma preko Figma Variables ali Tokens Studio določi barve, razmike in semantične tokene.

Faza 2 (Sinchronizacija): Vtičnik v Figma samodejno potisne spremembe v obliki JSON datoteke v GitHub/GitLab repozitorij (Pull Request).

- **Prilaganje za več naročnikov (Multi-branding):** Ista koda spletnih delov se z zamenjavo paketa žetonov takoj prilagodi barvni shemi različnih podjetij ali oddelkov.

Poslovni vpliv in AI

Uvedba avtomatiziranega sistema žetonov zahteva začetno investicijo v



Faza 3 (Transformacija): Orodje, kot na primer (Amazon Style Dictionary) pretvori nevtralni JSON v ciljne formate: CSS/SCSS, TypeScript ter tematske datoteke za uporabo na SharePoint SPFx.

Uporaba v praksi (SharePoint, M365 in Fluent UI)

V sodobnem Microsoftovem okolju (SharePoint, Teams, Fluent UI 2) žetoni omogočajo:

- **Brezhibno podporo za temni način (Dark Mode):** S preklpom semantičnih žetonov ostanejo vsi SPFx gradniki kontrastni in berljivi brez podvajanja kode.

strukturo in orodja, vendar se povrnitev investicije (ROI) pokaže zelo hitro:

- **Do 70 % manj časa za predajo dizajna:** Odpravljeni so neskončni sestanki za usklajevanje heksa kod in popravljanje vizualnih napak.
- **Takojšnja globalna posodobitev:** Sprememba celostne grafične podobe (rebranding), ki je včasih trajala mesece, se lahko izvede v nekaj dneh.
- **Manj stresa in boljše sodelovanje:** Oblikovalci dobijo

neposreden nadzor nad vizualno konsistenco, razvijalci pa se lahko nemoteno posvetijo logiki, stabilnosti in odzivnosti sistema, aplikacije.

Ker se zadnji čas vse vrti okoli umetne inteligence (AI) ter njene implementacije vsepovsod, se uporaba te pravitako dotika tudi Design Tokens področja.

Vloga AI pri Design Tokens se najbolj kaže v načinu kot je:

- **Avtomatsko luščenje in strukturiranje:** AI iz obstoječe kode ali zasnov samodejno prepozna barve, pisave in razmike ter jih uredi v semantični JSON format.
- **Nadzor nad dostopnostjo (WCAG):** V realnem času preverja barvne kontraste za svetli in temni način ter samodejno popravlja neskladne odtenke.
- **Takojšnje prevajanje v kodo:** En sam nabor žetonov zanesljivo pretvori v specifične oblike za

branje v spletu (CSS, Tailwind, React, Angular, SPFx) in mobilne platforme (Swift, Flutter, React Native).

• Čiščenje sistema (Design Linting):

Zaznava napačne fiksne vrednosti (hardcoded piksele) v vmesnikih ter predlaga ustrezne uradne "default" žetone.

Zaključek

Prehod iz statičnih grafičnih priročnikov v žive variabilne dizajne, z avtomatizacijo podprte Design Tokene je evolucijski korak digitalnega oblikovanja.

Za slovenska podjetja in razvojne ekipe, ki gradijo kompleksne in obsežne digitalne produkte, kot so aplikacije ali druge spletne rešitve, to pomeni konec improvizacije in kompliciranja ter začetek pravega inženirskega partnerstva med oblikovanjem in razvojem (kodo).

Kam vas bo pripeljalo novo znanje?

Vsaka nova veščina odpira nove priložnosti. Jesenski termini so odprti.

Oglejte si jih tukaj!



POWERSHELL KOTIČEK



Aleš Lipušček

MCP, MCTS, MCITP

ales.lipuscek@kompas-xnet.si

Pred kratkim sem spet naletel na obvestilo, da Microsoft pospravlja wmic.exe. Saj ni ravno novica WMIC je deprecated že vrsto let. V Windows 11 24H2 ni bil več privzeto nameščen, avgusta 2026 pa ga je Microsoft začel dokončno odstranjevati tudi iz Windows 11 24H2 in 25H2. Izginja samo wmic.exe; WMI kot tehnologija ostaja.

Če ste PowerShell oziroma Windows administracijo začeli dovolj pozno, obstaja celo možnost, da WMIC-a nikoli niste uporabljali.

Dolga leta pa je bil eden tistih ne preveč lepih, a koristnih pripomočkov, ki jih je imel administrator vedno nekje pri roki.

```
wmic os get
Caption,Version,BuildNumber
```

Nekaj podobnega danes precej raje naredimo tako:

```
Get-CimInstance
Win32_OperatingSystem |
    Select-Object Caption, Version,
BuildNumber
```

Ni namen tega članka primerjati

WMIC-a in CIM-a. Pravzaprav bomo WMIC čez nekaj odstavkov skoraj pozabili.

Ampak za začetek nam da zelo uporaben problem.

Nova koda me ne skrbi preveč. Bolj zanimiva je mapa s skriptami, ki se je skozi leta selila z računalnika na računalnik in v kateri so stvari, za katere sem že skoraj pozabil, zakaj obstajajo.

Recimo:

C:\Scripts

V njej je na desetine .ps1 datotek.

Vprašanje je preprosto:

Katere od teh skript še vedno uporabljajo WMIC?

Za demonstracijo bomo med njimi skrili majhno staro skripto.

```
C:\Scripts\Legacy\Get-PCInfo.ps1:
# Stara skripta za osnovni pregled
računalnika.
# TODO: nekoč zamenjaj WMIC.
$ComputerName =
$env:COMPUTERNAME
Write-Host "Computer:
$ComputerName"
```

wmic.exe os get

```
Caption,Version,BuildNumber
wmic.exe computersystem get
Manufacturer,Model
```

Nič posebnega. Pravzaprav ravno zato je dober primer. Take stvari ostanejo po diskih leta.

Prva rešitev je skoraj avtomatska. Če iščemo tekst po datotekah, ima PowerShell za to Select-String.

```
Get-Childitem C:\Scripts -Filter *.ps1 -
Recurse |
    Select-String -Pattern 'wmic'
```

In dobimo nekaj takega:

```
C:\Scripts\Legacy\Get-PCInfo.ps1:2:#
TODO: nekoč zamenjaj WMIC.
C:\Scripts\Legacy\Get-
PCInfo.ps1:8:wmic.exe os get
Caption,Version,BuildNumber
C:\Scripts\Legacy\Get-
PCInfo.ps1:9:wmic.exe
computersystem get
Manufacturer,Model
```

Našli smo ga.

No... pravzaprav smo našli besedo WMIC.

Kar še ni nujno isto.

Nekje drugje v direktoriju imamo recimo:

```
# WMIC smo iz te skripte odstranili
leta 2024.
```

```
$message = 'Stare skripte z WMIC je
treba pregledati.'
```

```
Get-CimInstance
```

Win32_OperatingSystem

Tudi ta datoteka se bo pojavila v rezultatih Select-String.

Če iščemo po petih skriptah, nam je vseeno. Pogledamo rezultat in gremo naprej.

Če iščemo po desetinah skript ali celo po več repozitorijih skript, pa je problem precej drugačen. Dobivamo komentarje, nize, primere v dokumentaciji in neke vmes dejanske ukaze.

In tukaj sem se prvič vprašal nekoliko drugače. Ne, kje se v datotekah pojavi beseda wmi ampak kje PowerShell vidi wmic kot ukaz?

To se na prvi pogled sliši kot ista stvar, pa ni.

In tukaj pridemo do nečesa, za kar moram priznati, da sem se s PowerShellom ukvarjal zelo dolgo, preden sem ga sploh zares opazil.

AST.

Abstract Syntax Tree.

Moram priznati, da sem ta del PowerShella spoznal šele nedavno. Danes si ogledamo samo površje, v naslednjih Pikah pa morebiti skočimo malo globlje.

Ko PowerShell dobi skripto, je ne more samo prebrati od leve proti desni in upati na najboljše. Najprej jo mora razčleniti.

Zelo poenostavljeno:

PowerShell koda



AST je drevesni opis strukture programa.

Na primer:

Get-Service -Name Spooler

ni samo niz znakov.

Parser ve, da ima pred sabo ukaz, parameter in vrednost. Nekje v tej strukturi zato obstaja objekt tipa:

System.Management.Automation.Language.CommandAst

In najbolj zanimivo je, da nam PowerShell dovoli pogledati prav to strukturo.

Poglejmo primer:

```
$tokens = $null
```

```
$errors = $null
```

```
$ast = [System.Management.Automation.Language.Parser]::ParseFile(
    'C:\Scripts\Legacy\Get-PCInfo.ps1',
    [ref]$tokens,
    [ref]$errors
)
```

```
$commands = $ast.FindAll(
    { param($node)
```

```
$node -is [
```

```
System.Management.Automation.Language.CommandAst
    ]
```

```
},
    $true )
```

```
$commands |
    ForEach-Object {
        $_.GetCommandName() }

```

Rezultat naše male skripte bi bil približno:

```
Write-Host
wmic.exe
wmic.exe
```

Komentar:

TODO: nekoč zamenjaj WMIC.

je izginil.String z besedo WMIC bi prav tako izginil. PowerShellov parser namreč ve, da to nista ukaza.

Eno datoteko smo pregledali. Zdaj naredimo isto čez celo drevo. Logika se pravzaprav skoraj ne spremeni.

Za vsako datoteko naredimo AST, v AST-ju poiščemo CommandAst, nato pa vprašamo, kako se ukaz imenuje.

```
$results = foreach (
    $file in Get-ChildItem C:\Scripts -
    Filter *.ps1 -Recurse
){
    $tokens = $null
    $errors = $null
    $ast = [System.Management.Automation.Language.Parser]::ParseFile(
        $file.FullName,
        [ref]$tokens,
```

```
[ref]$errors
$wmicCommands = $ast.FindAll(
{
    param($node)
    $node -is [
System.Management.Automation.Lan
guage.CommandAst
] -and
    $node.GetCommandName() -in
@(
    'wmic'
    'wmic.exe'
)
    },
    $true
)
foreach ($command in
$wmicCommands) {
    [pscustomobject]@{
        File = $file.FullName
        Line
$command.Extent.StartLineNumber
        Code = $command.Extent.Text
    }
}
$results |
    Format-Table -AutoSize
```

Zdaj dobimo

File	Line Code
----	-----
C:\Scripts\Legacy\Get-PCInfo.ps1	8 wmic.exe os get Caption,Version,BuildNumber
C:\Scripts\Legacy\Get-PCInfo.ps1	9 wmic.exe computersystem get Manufacturer,Model

In naša druga skripta, tista s komentarjem in stringom? Ni je. Torej ne iščemo več samo besede WMIC. Iščemo ukaz. In to je že nekaj drugega. To je precej velika razlika.

V prejšnjem primeru se je pojavil še en zanimiv del:

```
$command.Extent.StartLineNumber
$command.Extent.Text
```

AST vozlišče ne ve samo, kaj to predstavlja. Zapomni si tudi, od kod v kodi je ta kos prišel. Zato lahko vprašamo:

```
$command.Extent.StartLineNumber
$command.Extent.EndLineNumber
$command.Extent.StartColumnNumb
er
$command.Extent.Text
```

Kar pomeni, da lahko iz relativno majhnega kosa kode naredimo že čisto uporaben migracijski report.

```
$results |
    Sort-Object File, Line |
    Select-Object File, Line, Code |
    Export-Csv .\WMIC-Audit.csv -
NoTypeInfoInformation
```

Zdaj imamo seznam skript, vrstic in dejanskih WMIC ukazov, ki jih je treba pregledati. Ampak že precej bolj uporabno od petsto rezultatov tekstovnega iskanja.

In tu se začne zanimivejši del. Ker v naši skripti ni nič posebnega, kar bi bilo vezano na WMIC. Namesto:

```
$node.GetCommandName() -in @(
```

wmic 'wmic.exe')

Lahko napišemo:

```
$node.GetCommandName() -eq 'Get-WmiObject'
```

ali:

```
$node.GetCommandName() -eq 'Invoke-Expression'
```

ali iščemo Add-Type, Start-Process, stare ukaze, ki jih želimo zamenjati, ukaze, ki jih pri code reviewu želimo posebej preveriti...

In na tej točki WMIC skoraj ni več pomemben.

Pravzaprav nas je samo pripeljal do precej bolj zanimivega vprašanja: Kaj vse lahko izvemo o PowerShell kodi, ne da bi jo sploh zagnali?

AST zna poiskati funkcije. Parametre. Spremenljivke. Pipeline. Assignment statemente. if, foreach, metode, properties...

Lahko začnemo izdelovati lastni pregledovalnik kode, dokumentacijo modulov, migration scanner ali zelo primitiven static analyzer.



SHAREPOINT ONLINE PREMIUM: CONTENT UNDERSTANDING IN DOCUMENT PROCESSING – SAMODEJNO PRIDOBIVANJE METAPODATKOV Z AUTOFILL COLUMNS



Robi Vončina

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

Eden največjih izzivov pri upravljanju dokumentov v SharePoint Online ni več samo vprašanje, **kam dokument shraniti**, ampak predvsem, **kako ga pravilno opremiti z metapodatki**.

Dobro načrtovana SharePointova knjižnica dokumentov običajno uporablja stolpce, kot so *Vrsta dokumenta*, *Stranka*, *Projekt*, *Datum pogodbe*, *Datum poteka*, *Odgovorna oseba* ali *Status*. Metapodatki omogočajo učinkovito iskanje, filtriranje, izdelavo pogledov, avtomatizacijo procesov in upravljanje

življenjskega cikla dokumentov. Težava nastane, ko morajo uporabniki te podatke vnašati ročno. Pri večjih količinah dokumentov to pomeni veliko administrativnega dela, hkrati pa prihaja do napak, nedoslednega vnosa in pogosto tudi do situacije, ko uporabniki metapodatkov sploh ne izpolnijo.

SharePoint Online Premium oziroma njegove storitve za obdelavo dokumentov ta problem rešujejo z uporabo umetne inteligence.

Ena najzanimivejših funkcionalnosti so **Autofill columns**, s katerimi lahko SharePoint samodejno prebere vsebino dokumenta, iz nje pridobi zahtevane informacije in jih zapiše v stolpec knjižnice.

Microsoft Autofill columns opisuje kot funkcionalnost, ki uporablja velike jezikovne modele (LLM) za ekstrakcijo, klasifikacijo, povzemanje ali generiranje informacij iz vsebine datotek. Rezultat se nato shrani v ustrezen SharePointov stolpec.

Kaj so Autofill Columns?

Dokument	Vrsta dokumenta	Stranka	Številka ponudbe	Datum	Vrednost
Pogodba_001.pdf	Pogodba	ACME d.o.o.	P-2026-001	1. 1. 2026	25.000 €
Pogodba_002.pdf	Pogodba	Contoso d.o.o.	P-2026-002	15. 2. 2026	42.500 €

Autofill column je običajen SharePointov stolpec, ki mu dodamo **AI navodilo oziroma prompt**.

Namesto, da uporabniku rečemo: »Ko naložiš pogodbo, vnesi ime stranke, številko pogodbe in datum poteka.« lahko SharePointu določimo: »Iz dokumenta pridobi naziv stranke, ki je navedena kot pogodbeni partner.«

SharePoint nato analizira dokument in rezultat zapiše v ustrezen stolpec. Na ta način lahko iz enega dokumenta

samodejno pridobimo več različnih informacij.

Primer knjižnice **Ponudbe**:

Uporabniku ni treba ročno izpolnjevati teh vrednosti. Ko je dokument dodan v knjižnico, Autofill obdela njegovo vsebino in zapolni ustrezne stolpce. To je pomembna razlika v primerjavi s klasičnimi SharePointovimi metapodatki: metapodatkov ne določamo več samo na podlagi vnosa uporabnika, ampak jih lahko pridobimo neposredno iz vsebine dokumenta.

Kako pripraviti knjižnico za Autofill Columns?

Preden začnemo ustvarjati AI-stolpce, je zelo pomembno, da knjižnico pravilno načrtujemo.

AI namreč ne more rešiti slabo zasnovanega informacijskega modela.

Najprej določimo:

1. katere dokumente bomo shranjevali,
2. katere informacije želimo pridobiti,
3. katere informacije so za uporabnike pomembne,

4. katere podatke želimo uporabljati pri filtriranju in iskanju,
5. katere podatke bomo uporabljali v avtomatizacijah.

Primer:

Predpostavimo, da imamo knjižnico **Ponudbe**.

Najprej ustvarimo klasične SharePointove stolpce:

- **Stranka** – Single line of text
- **Številka ponudbe** – Single line of text
- **Datum ponudbe** – Date and time
- **Število** – Number
- **Znesek** - Currency

Nato posameznim stolpcem določimo, da bodo njihove vrednosti pridobljene z AI. Pri načrtovanju je pomembno, da je vsak stolpec odgovoren za **eno jasno informacijo**.

Ni priporočljivo ustvariti stolpca:

»Podatki o ponudbi« in pričakovati, da bo AI vanj zapisal vse podatke.

Veliko bolje je imeti:

- Številka ponudbe
- Stranka
- Datum začetka
- Vrednost
- Število

Vsak stolpec ima tako jasno definiran namen in lastno AI navodilo.

Kako ustvarimo Autofill Columnn?

Create a column ✕

[Learn more about column creation.](#)

Name *

Stranka

Description

Type

Single line of text

Autofill ⓘ ☒ On

Prompt

Extract the Stranka entity information from the document. Provide a clear...

Pridobi informacije o nazivu stranke ali customer, če je dokument angleški.

Naša navodila

Test the prompt

Select a test file

Purchase Order 12345... Test

Result **Test na dokumentu**

Mercator

Rezultat testa

Autofill stolpec lahko ustvarimo neposredno v SharePointovi knjižnici dokumentov.

Pri ustvarjanju novega stolpca izberemo možnost **Use autofill**, nato določimo navodilo, ki opisuje, kaj naj SharePoint poišče oziroma ustvari iz vsebine dokumenta. Autofill columns omogočajo uporabo naravnega jezika za določitev informacij, ki jih želimo pridobiti iz datotek.

Na primer za stolpec **Številka ponudbe** lahko uporabimo navodilo: »Iz dokumenta pridobi številko ponudbe. Če številka ponudbe ni navedena, vrni prazno vrednost.« Za stolpec **Datum**: »Iz dokumenta pridobi datum, ko pogodba preneha veljati. Datum vrni v obliki DD.MM.LLLL.«

Kako pripraviti dobro navodilo (prompt)? Pri pripravi Autofill stolpca je priporočljivo jasno določiti **kaj iščemo, kje naj AI išče in kakšen rezultat pričakujemo**.

Dobro navodilo običajno vsebuje:

1. Kaj naj AI poišče

»Poišči datum začetka veljavnosti pogodbe.«

2. Kako naj ga interpretira

»Uporabi datum, naveden kot začetek veljavnosti pogodbe, in ne datuma podpisa.«

3. Kaj naj naredi, če podatka ni

»Če datum ni naveden, vrni prazno vrednost.«

4. Kakšen naj bo rezultat

»Rezultat vrni kot datum.«

Primer dobrega prompta:

»Iz dokumenta pridobi datum začetka veljavnosti pogodbe. Uporabi datum, ki določa začetek pogodbenega obdobja, in ne datuma podpisa. Če datum začetka ni naveden, vrni prazno vrednost.«

Takšno navodilo je bistveno bolj uporabno kot preprost prompt:

»Poišči datum.«

Copilot lahko Autofill Columns tudi predlaga

Ena najbolj zanimivih možnosti je, da administratorju oziroma uporabniku **ni treba vnaprej poznati vseh metapodatkov, ki bi jih bilo smiselno ustvariti**. Copilot v SharePointu lahko analizira dokumente v knjižnici in **predlaga stolpce ter ustrezna navodila za pridobivanje metapodatkov**.

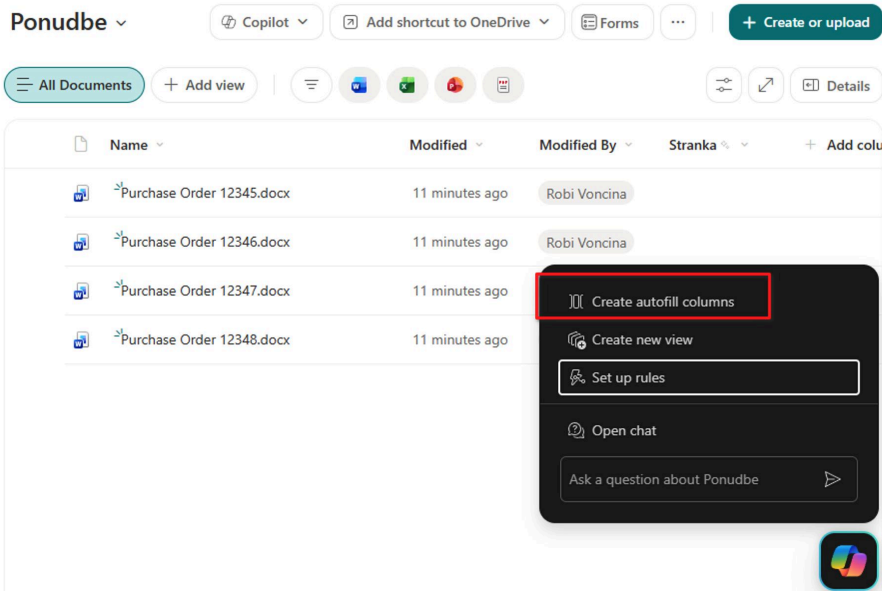
V knjižnici lahko odpremo Copilot in izberemo **Create autofill columns**. Copilot analizira reprezentativen vzorec dokumentov in na podlagi njihove vsebine predlaga, katere informacije bi bilo smiselno shranjevati kot metapodatke. Pri začetni konfiguraciji lahko analizira do 20 datotek.

Če izberemo konkretne datoteke, analizira samo izbrane dokumente. Če je to prvič, ko uporabljamo to funkcionalnost v knjižnici, lahko Copilot samodejno predlaga do tri stolpce.

Primer:

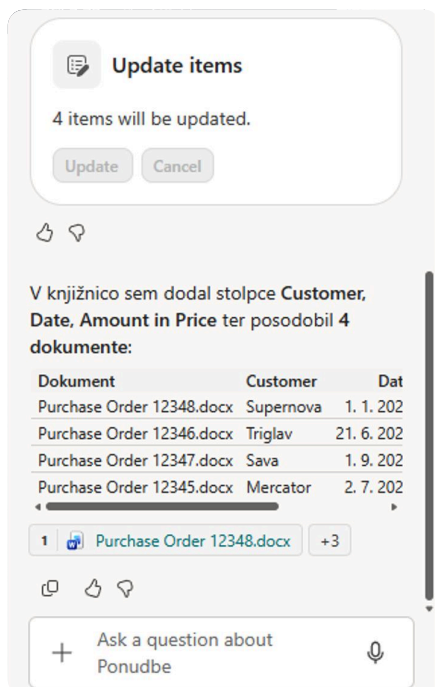
V knjižnici imamo 50 ponudb, vendar smo ustvarili samo stolpca **Vrsta dokumenta** in **Stranka**.

Copilot lahko pri analizi dokumentov



ugotovi, da se v vseh pogodbah pogosto pojavljajo tudi:

- številka ponudbe,
- datum,
- količina,
- cena.



Predlaga lahko torej nove Autofill stolpce, skupaj z navodili, kako naj se vrednosti pridobijo. Administrator oziroma uporabnik predlogov ne sme obravnavati kot dokončne konfiguracije. Predlagane stolpce je treba pregledati, po potrebi preimenovati, spremeniti navodilo in preveriti rezultate.

Microsoft omogoča tudi testiranje spremenjenega navodila na največ desetih datotekah, preden spremembo shranimo. To je zelo pomembno, saj lahko s testiranjem ugotovimo, ali AI pravilno razume naše dokumente.

Kaj se zgodi z novim dokumentom?

Ko je Autofill stolpec nastavljen in shranjen, postane del knjižnice. Ko uporabnik naloži nov dokument:

Nalaganje dokumenta → SharePoint → AI analiza dokumenta → ekstrakcija podatkov → zapis v stolpce

Na primer uporabnik naloži:

Purchase order 12345.docx

SharePoint lahko samodejno izpolni:

- Stranka = Sava
- Številka ponudbe = P-2026-015
- Datum = 01.09.2026
- Število = 150
- Vrednost = 150 EUR

Posebnost je, da uporabniku, ki dokument naloži, ni treba imeti Copilot licence za samodejno obdelavo novih

dokumentov v že konfigurirani knjižnici.

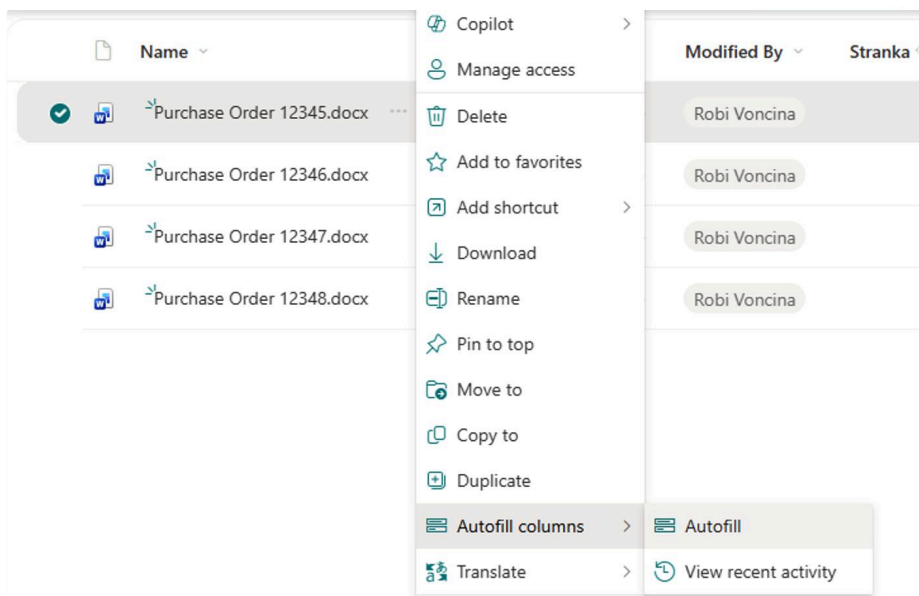
Kaj pa obstoječi dokumenti?

Pri obstoječih dokumentih lahko izberemo datoteke, ki jih želimo obdelati, in zaženemo **Autofill**.

To je pomembno pri uvajanju funkcionalnosti v obstoječe knjižnice, kjer je lahko že več tisoč dokumentov. Pri večjih knjižnicah je zato smiselno najprej izvesti pilotni projekt na omejenem številu dokumentov, preveriti kakovost rezultatov in šele nato obdelavo razširiti. SharePoint omogoča tudi spremljanje napredka obdelave prek aktivnosti Autofill.

Omejitve, ki jih je treba poznati

Autofill Columns niso univerzalna rešitev za vse tipe SharePointovih stolpcev.



Trenutno so podprti stolpci, kot so:

- Text,
- Multiple lines of text,
- Number,
- Yes/No,
- Date and time,
- Choice,
- Hyperlink,
- Currency,
- Managed metadata.

Nekateri tipi, kot so **Person or Group**, **Location**, **Image in Lookup**, trenutno niso podprti za Autofill. Microsoft za optimalno delovanje priporoča tudi največ približno 10 Autofill stolpcev na knjižnico in dokumente do 65 strani.

Pomembna je tudi vsebina dokumentov. Trenutna funkcionalnost lahko obdeluje več podprtih formatov, med drugim Office dokumente in PDF.

Zato je treba pred produkcijsko uvedbo preveriti dejanske dokumente organizacije in kakovost rezultatov.

Licenciranje in Pay-As-You-Go

Autofill Columns sodijo med Document processing for Microsoft 365 storitve. Te storitve uporabljajo model Pay-As-You-Go, pri katerem je uporaba obračunana glede na vrsto in število transakcij. Za uporabo je zato treba Azure Subscription povezati z okoljem Microsoft 365, kot je bilo navedeno v 1. delu serije člankov.

Po povezavi na naročnino Azure, se

Autofill Columns lahko omogoči za SharePointova mesta. Privzeto je lahko funkcionalnost omogočena za vsa SharePointova mesta, administrator pa lahko uporabo omeji na izbrana mesta.

Konfiguracijo najdemo v:

Microsoft 365 admin center → Settings → Org settings → Pay-as-you-go services → Document & image services → Autofill columns

Zaključek

Autofill Columns so ena najbolj praktičnih funkcionalnosti SharePoint Online Premium oziroma storitev Document Processing. Njihova največja prednost je preprost koncept: **uporabnik naloži dokument, umetna inteligenca pa iz njegove vsebine pridobi informacije, ki jih organizacija potrebuje kot metapodatke.**

Pri uvedbi pa je pomembneje od samega vklopa funkcionalnosti dobro načrtovanje. Treba je pravilno določiti informacijski model knjižnice, izbrati smiselne stolpce in predvsem pripraviti kakovostna AI navodila.

Dodatno lahko Copilot pomaga tudi pri začetni analizi knjižnice in predlaga, katere Autofill Columns bi bile glede na dejansko vsebino dokumentov smiselne. Tako se lahko začetna konfiguracija, ki bi jo administrator sicer pripravljaval ročno, bistveno pospeši.



POWER AUTOMATE – HTTP GET NA SHAREPOINT SEZNAMU Z VEČ KOT 5000 ELEMENTI



Klemen Vončina

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

Send an HTTP request to SharePoint akcija v Power Automate je zelo zmogljivo orodje za raznovrstne operacije na SharePoint Online. V tem članku si bomo ogledali, kako lahko uporabimo to akcijo, da pridobimo naziv vseh elementov nekega SharePoint seznama. Za primer bom uporabil seznam, ki ima 7000 elementov.

Prvotno pridobivanje elementov

V delovnem toku bomo najprej pridobili prvih 5000 elementov. To akcijo lahko izvedemo tudi, če seznam še nima 5000 elementov, ampak pričakujemo, da jih bo nekoč imel.

Z namenom pridobivanja nazivov vrnjenih elementov, sem v delovni tok dodal Select akcijo, ki iz nabora

 **Get items from list** ⋮ <<

Parameters Settings Code view Testing About

Site Address *

Site A - <https://1nxpk.sharepoint.com/sites/SiteA> ▼

Method *

GET ▼

Uri *

_api/web/lists/GetByTitle('MoreThan5K')/items?\$top=5000

Advanced parameters

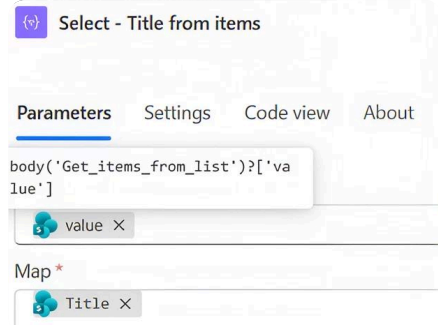
Showing 1 of 2 ▼ Show all Clear all

Headers

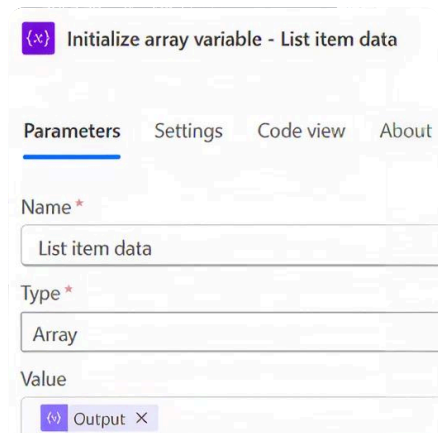
accept application/json;odata=nometadata

 ✕

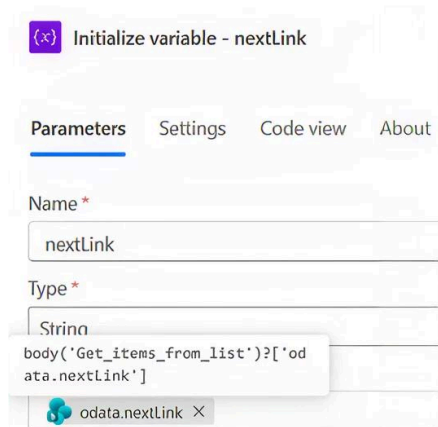
vrnjenih elementov prejšnje akcije izlušči njihov naziv.



Nazive pa potem shranil v spremenljivko tipa array.



Kot je vidno v prvem HTTP klicu, smo dobili le prvih 5000 elementov, kar je tudi limit. A v primeru, da seznam

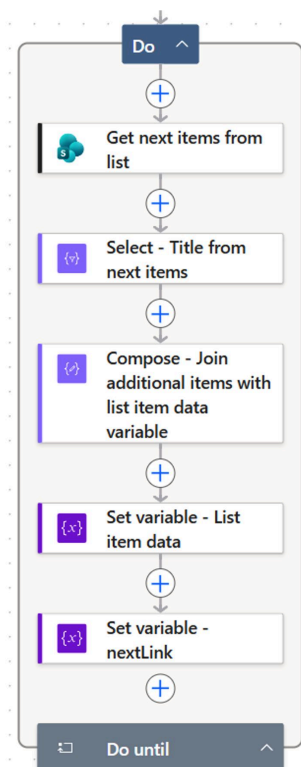


vsebuje več elementov, bo ta klic vrnil lastnost nextLink, ki sem jo prav tako shranil v spremenljivko tipa string.

Zankanje po dodatnih sklopih elementov

Tem akcijam sledi Do until zanka, v kateri bomo ponavljali klic na seznam, vse dokler ne bo lastnost nextLink prazna.

Implementacija tega izgleda tako, da v zanki skorajda ponovimo prvotni klic, le da v URI namesto prejšnjega API endpointa uporabimo nextLink, ki smo jo shranili v spremenljivko, in ji odstranimo pod do mesta, na katerem se seznam nahaja.



Do until

Parameters Settings Code view About

length(variables('nextLink'))
 length(...) is equal to 0

Get next items from list

Parameters Settings Code view Testing About

Site Address*
 Site A - https://1nxjpk.sharepoint.com/sites/SiteA

Method*
 replace(variables('nextLink'),'https://1nxjpk.sharepoint.com/sites/SiteA','')
 replace(...)

Da v vsaki iteraciji shranimo dodatne elemente, ponovno izvedemo Select akcijo na novih elementih.

Select - Title from next items

Parameters Settings Code view About

body('Get_next_items_from_list')['value']
 value

Map*
 Title

Nato novo pridobljene elemente združimo z elementi iz List item data spremenljivke.

Compose - Join additional items with list item data variable

Parameters Settings Code view About

union(variables('List item data'),body('Select_-_Title_from_next_items'))
 union(...)

In rezultat te akcije nato določimo kot novo vrednost List item data

spremenljivke.

Set variable - List item data

Parameters Settings Code view About

Name*
 outputs('Compose_-_Join_additional_items_with_list_item_data_variable')
 Outputs

Da ne naredimo neskončne zanke, moramo seveda popraviti tudi vrednost spremenljivke nextLink.

Set variable - nextLink

Parameters Settings Code view About

Name*
 nextLink

Value*
 odata.nextLink

Celoten delovni tok, v katerem smo pridobili 7000 elementov in iz njih izluščili podatek o nazivu, je trajal pičle 4 sekunde. Za primerjavo – če bi uporabili standardno Get items akcijo, ki je na voljo v Power Automate, bi samo ta akcija v mojem primeru trajala 3 sekunde. Seveda pa smo pri vgrajeni Get items akciji omejeni na 5000 elementov, do preostanka elementov niti ne moremo dostopati.

↓ 3s ✓

Get items



AZURE FUNCTIONS



Lenart Svetek

Frontend programmer

lenart.svetek@kompas-xnet.si

Azure Functions je brezstrežniška (serverless) rešitev, ki omogoča razvoj zmogljivih aplikacij z manj kode in enostavnejšo infrastrukturo. Namesto da bi morali ob povečanju ali zmanjšanju obremenitve ročno prilagajati zmogljivost strežnikov, se Azure Functions samodejno prilagajajo obremenitvi in skalirajo glede na potrebe aplikacije – vse do vnaprej določene omejitve.

Splošne lastnosti

Azure Functions so privzeto stateless, kar pomeni, da si med posameznimi zagoni funkcije ne zapomnijo stanja oziroma sprememb, ki so se zgodile med izvajanjem. Če potrebujemo stateful funkcije, lahko uporabimo možnost Durable Functions, ki omogoča ohranjanje stanja in izvajanje bolj kompleksnih procesov.

Ker so Azure Functions serverless, je eden od možnih stranskih učinkov ta, da se funkcije ob daljšem obdobju neaktivnosti lahko ustavijo. Ko je funkcija ponovno poklicana, mora Azure ponovno vzpostaviti potrebno izvajalno okolje, kar lahko povzroči tako imenovani Cold Start.

To pomeni, da lahko na primer API, ki

temelji na Azure Functions, ob prvem klicu po daljšem obdobju neaktivnosti potrebuje nekoliko več časa za odgovor kot običajno.

To lahko rešimo tako, da v Azure Portalu pod »Scale and concurrency« nastavimo možnost »Always-ready instances«. S tem lahko določimo število instanc, ki bodo za določeno funkcijo oziroma vrsto funkcije vedno pripravljene na izvajanje.

Na ta način zmanjšamo verjetnost Cold Startov, saj Azureju ni treba vsakič znova vzpostavljati izvajalnega okolja za neaktivno funkcijo.

Seveda pa ima takšna nastavitve tudi svojo ceno. Always-ready instance namreč pomenijo, da določeno število instanc ostaja ves čas pripravljeno na izvajanje, kar posledično prinese nekoliko višje stroške.

Sprožilci (Triggers)

Triggers oziroma sprožilci določajo, kdaj in na kakšen način se Azure Function izvede. Funkcija se lahko na primer sproži ob prejemu HTTP zahtevka, dodanem sporočilu v vrsto (queue), spremembi podatkov v bazi ali ob določenem časovnem intervalu.

Vsaka Azure Function ima lahko en sprožilec, ki določa dogodek, na katerega se funkcija odzove.

HTTP sprožilec

HTTP sprožilec je eden najosnovnejših sprožilcev, saj Azure Function izpostavi kot HTTP endpoint, ki ga lahko pokličemo z običajno HTTP zahtevo. Uporabimo ga lahko na primer za izdelavo API-jev, prejemanje podatkov iz drugih aplikacij ali obdelavo zahtevkov uporabnikov.

```
[Function("HttpFunction")]
public IActionResult
Run([HttpTrigger(AuthorizationLevel.Anonymous, "get")] HttpRequest req)
{
    return new OkObjectResult($"Welcome to Azure Functions!");
}
```

Pri HttpTriggerju lahko z nastavitvijo AuthorizationLevel določimo, kdo lahko dostopa do funkcije oziroma jo lahko pokliče. S tem lahko nadzorujemo, ali je funkcija javno dostopna ali pa je za njen klic potrebno ustrezno preverjanje oziroma ključ.

Service Bus sprožilec

Service Bus sprožilec omogoča, da se Azure Function samodejno izvede, ko je v Azure Service Bus vrsti (queue) ali temi (topic) na voljo novo sporočilo.

To je uporabno predvsem pri asinhroni obdelavi sporočil, saj funkciji ni treba neprestano preverjati, ali je prispelo novo sporočilo. Ko se sporočilo pojavi, ga sprožilec zazna in zažene funkcijo, ki

ga lahko nato obdela.

```
[Function(nameof(ServiceBusReceivedMessageFunction))]
[ServiceBusOutput("outputQueue",
    Connection = "ServiceBusConnection")]
public string
ServiceBusReceivedMessageFunction(
    [ServiceBusTrigger("queue",
        Connection = "ServiceBusConnection")]
    ServiceBusReceivedMessage message)
{
    _logger.LogInformation("Message ID: {id}", message.MessageId);
    _logger.LogInformation("Message Body: {body}", message.Body);
    _logger.LogInformation("Message Content-Type: {contentType}",
        message.ContentType);
    var outputMessage = $"Output message created at {DateTime.Now}";
    return outputMessage;
}
```

Časovni sprožilec

Časovni sprožilec omogoča, da se Azure Function samodejno izvede ob določenem času oziroma v določenih časovnih intervalih. Urnik izvajanja določimo s CRON izrazom, zato lahko funkcijo nastavimo na primer tako, da se izvede vsakih 5 minut, vsak dan ob določeni uri ali enkrat tedensko.

Uporaben je predvsem za periodična opravila, kot so čiščenje podatkov, generiranje poročil, preverjanje stanja sistemov ali izvajanje različnih ozadnih procesov.

```
[Function(nameof(TimerFunction))]
[FixedDelayRetry(5, "00:00:10")]
public static void Run([TimerTrigger("0
*/5 * * * *")] TimerInfo timerInfo,
```

```
FunctionContext context)
{
    var logger =
context.GetLogger(nameof(TimerFunction));
    logger.LogInformation($"Function
Ran. Next timer schedule =
{timerInfo.ScheduleStatus?.Next}");
}
```

Ostali sprožilci

Na voljo je veliko različnih sprožilcev, vendar so zgoraj naštetih med bolj uporabnimi za začetek. Ostale sprožilce in podrobnejšo dokumentacijo si lahko ogledate na <https://learn.microsoft.com/en-us/azure/azure-functions>.

Scenariji

V nadaljevanju bom predstavil nekaj primerov oziroma scenarijev, kjer lahko uporabimo Azure Functions.

Scenarij 1: REST API

Predstavljajmo si spletno trgovino, kjer uporabnik odda naročilo. Ob potrditvi naročila aplikacija pošlje HTTP zahtevek na Azure Function. Funkcija preveri podatke naročila, ga shrani v bazo in po potrebi sproži nadaljnjo obdelavo, na primer pošlje potrditveno e-pošto ali ustvari račun.

Prednost takšnega pristopa je, da nam ni treba vzdrževati strežnika, ki bi ves čas čakal na nova naročila. Azure Functions se samodejno prilagajajo številu zahtevkov, zato lahko brez večjih sprememb obravnavamo tako nekaj naročil na uro kot tudi velik porast naročil.

Scenarij 2: Service Bus

Predstavljajmo si, da obdelava naročila v API-ju traja predolgo, zato ne želimo, da uporabnik čaka na zaključek celotnega procesa. V tem primeru lahko uporabimo HTTP sprožilec, ki sprejme naročilo in ga zapiše v Service Bus Queue.

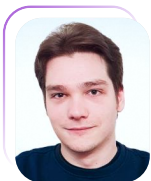
Nato Service Bus sprožilec zazna novo sporočilo v vrsti in zažene Azure Function, ki naročilo obdelava v ozadju. Na ta način lahko API uporabniku hitro odgovori, medtem ko se obdelava naročila izvede asinhrono in neodvisno od začetnega zahtevka.

Scenarij 3: Preverjanje Dead-letter Queue

Predstavljajmo si, da želimo ob začetku delovnega dne preveriti, ali so se v Service Bus Dead-letter Queue čez noč znašla sporočila, ki jih ni bilo mogoče uspešno obdelati.

Za to lahko uporabimo časovni sprožilec, ki se vsak delovni dan ob določeni uri zažene. Funkcija preveri Dead-letter Queue in, če najde kakšno sporočilo, o tem obvesti odgovorno ekipo, na primer prek e-pošte ali Teamsa.

Na ta način lahko avtomatiziramo jutranje preverjanje in hitro zaznamo sporočila, ki zahtevajo dodatno obravnavo.



TWO-FACTOR AUTHENTICATION (2FA)



Žiga Brinc

Frontend programer

ziga.brinc@kompas-xnet.si

Dvostopenjsko preverjanje pristnosti 2FA (angl. Two-Factor Authentication) je varnostna metoda, pri kateri mora uporabnik pri dostopu do računa ali sistema predložiti dva različna faktorja preverjanja pristnosti. Je oblika večfaktorskega preverjanja pristnosti MFA (angl. Multi-Factor Authentication). Namen 2FA je dodati dodatno plast zaščite, tako da ukradeno geslo samo po sebi ni dovolj za dostop do računa.

Faktorji preverjanja pristnosti se na splošno delijo v tri kategorije:

- Nekaj, kar veš (informacija, ki jo uporabnik pozna, na primer geslo ali PIN)
- Nekaj, kar imaš (nekaj, kar uporabnik ima in nadzoruje, na primer pametni telefon ali strojni varnostni ključ)
- Nekaj, kar si (biometrična značilnost uporabnika, na primer prstni odtis ali prepoznavo obraza)

Pri 2FA naj bi faktorja prihajala iz različnih kategorij. Na primer, kombinacija gesla in začasne kode OTP uporablja nekaj, kar veš, in nekaj, kar imaš.

Zakaj uporabljati 2FA?

Gesla so ena izmed pogostih šibkih točk pri varnosti računov. Lahko jih je uganiti, ukrasti z lažnim predstavljanjem (angl. phishing), razkriti pri vdoru v podatkovne zbirke ali pa jih uporabniki ponovno uporabijo pri več različnih storitvah.

Če je geslo edina zaščita, lahko njegovo pridobitev napadalec izkoristi za prevzem računa.

2FA zagotavlja dodatno varnostno oviro. Tudi če napadalec pridobi geslo, za dokončanje preverjanja pristnosti še vedno potrebuje drugi faktor.

Različne metode 2FA

Vse metode 2FA ne zagotavljajo enake ravni zaščite. Kode, poslane prek SMS-sporočil, so na primer široko podprte, vendar veljajo za šibkejšo obliko preverjanja pristnosti, saj lahko napadalec s tehnikami, kot je zamenjava kartice SIM (angl. SIM swapping), pridobi nadzor nad telefonsko številko in tako prejme tudi avtentikacijske kode.

Avtentikacijske aplikacije, ki ustvarjajo časovno omejena enkratna gesla TOTP (angl. Time-Based One-Time Password), so na splošno boljše

možnost. Vendar so lahko tudi te še vedno ranljive za napade z lažnim predstavljanjem.

Napadalec lahko uporabi napad AiTM (angl. Adversary-in-the-Middle), pri katerem zlonamerno spletno mesto deluje kot posrednik med žrtvijo in legitimno storitvijo. Žrtev lahko nevede vnese svoje geslo in kodo 2FA na napadalčevo spletno mesto.

Napadalec lahko nato podatke posreduje legitimni storitvi in potencialno pridobi že preverjeno uporabniško sejo.

Za močnejšo zaščito je priporočljiva avtentikacija, odporna proti lažnemu predstavljanju (angl. phishing-resistant authentication), predvsem varnostni ključi na osnovi FIDO2/WebAuthn in passkeys.

Te metode uporabljajo kriptografsko preverjanje pristnosti, ki je vezano na legitimno spletno mesto, zaradi česar so napadi, pri katerih napadalec posreduje poverilnice med žrtvijo in storitvijo, kot je AiTM, veliko težji.

Omejitve 2FA

Čeprav 2FA zagotavlja močno zaščito med postopkom preverjanja pristnosti, ne ščiti pred vsemi vrstami napadov. Pomemben primer je ugrabitev uporabniške seje (angl. session hijacking).

2FA zato ne bi smeli obravnavati kot popolno zaščito računa. Namesto tega ga je treba kombinirati z varnim

upravljanjem uporabniških sej, ustreznim časom trajanja sej, zaznavanjem sumljive dejavnosti in ponovnim preverjanjem pristnosti pri posebej občutljivih dejanjih.

Ena izmed težav pri varnosti je, da si morajo uporabniki že zdaj zapomniti gesla za številne različne storitve.

Zahteva po dodatnem preverjanju pristnosti pri vsakem dejanju lahko postane nepraktična in uporabnike spodbudi k iskanju načinov za obhod varnostnih mehanizmov.

Sodobni sistemi lahko to nevšečnost zmanjšajo s tehnologijami, kot je enotna prijava SSO (angl. Single Sign-On).

SSO uporabniku omogoča, da se preveri pri osrednjem ponudniku identitete in nato dostopa do več pooblaščenih storitev brez ponavljajočega se vnosa podatkov za prijavo.





NASVETI ZA HITREJŠE DELOVANJE SPLETNE STRANI



Domen Gričar

MCSO, MCSA, MCT

domen.gricar@kompas-xnet.si

Hitrost spletnega mesta neposredno vpliva na uporabniško izkušnjo, uporabniki pričakujejo, da se bodo strani hitro nalagale, zlasti na mobilnih napravah in pri počasnejših povezavah. Že majhna zamuda lahko poveča zapuščanje strani in zmanjša zanimanje. Izboljšanje spletne zmogljivosti običajno ne zahteva celotne prenove. V tem članku bom opisal nekaj praktičnih načinov za pospešitev delovanja spletnega mesta.

Slike so pogosto največja sredstva na spletni strani. Nalaganje nepotrebno velikih slik lahko zelo poveča čas nalaganja. Najbolje je spremeniti velikost slik na dejansko potrebne dimenzije in uporabiti sodobne formate, kot sta WebP ali AVIF. Uporabi se lahko tudi prilagodljive slike z ukazom srcset, da mobilne naprave ne bodo prenašale datotek velikosti namiznega računalnika.

```
<picture>
  <source srcset="slika.avif"
  type="image/avif">
  <source srcset="slika.webp"
  type="image/webp">
  
</picture>
```

Za slike prikazane nižje na strani se

lahko uporabi tudi zakasnjeno nalaganje (lazy loading) s pomočjo atributa loading, ki naloži datoteke šele, ko se jim uporabnik približa.

```

```

Enak princip lahko uporabimo tudi za nalaganje ostalih multimedijskih vsebin, kot so video posnetki, avdio datoteke in druge vdelane datoteke (embed, iframe).

Na spletni strani je običajno naloženih tudi kar nekaj datotek za urejanje stilov in dinamičnosti vsebine. Presledki, komentarji in nepotrebni znaki povečajo velikost datoteke, ne da bi to vplivalo na funkcionalnost, zato je datoteke dobro stisniti (minify). CSS kodo kot je

```
.gumb {
  background-color: #094879;
  padding: 10px 20px;
}
```

lahko brez izgube funkcionalnosti stisnemo na

```
.gumb{background-
color:#094879;padding:10px 20px;}
```

S pomočjo orodij, kot je webpack pa lahko postopek tudi avtomatiziramo. JavaScript lahko znatno vpliva na

zmogljivost, saj ga mora brskalnik prenesti, razčleniti, prevesti in izvesti. Bolj kot zmanjševanje same datoteke, je pomembno zmanjševanje količine kode, ki je nepotrebna za posamezno stran. Za skripte, ki jih ni treba takoj izvesti, se lahko uporabi defer

```
<script src="aplikacija.js" defer></script>
```

Za neodvisne skripte tretjih oseb pa se lahko uporabi asinhrono nalaganje

```
<script src="analitika.js" async></script>
```

Namesto da prenesemo vsa sredstva iz glavnega strežnika lahko uporabimo CDN (content delivery network), ki shranjuje datoteke na drugih strežnikih. Namesto da bi se vsak uporabnik povezal z glavnim strežnikom, lahko prejme vire z lokacije, ki mu je geografsko bližje. CDN je še posebej uporaben za spletna mesta z mednarodnimi uporabniki ali veliko količino statične vsebine.

Pri nalaganju lahko tudi zmanjšamo količino datotek, ki se naloži ob vsaki osvežitvi strani. Predpomnjenje omogoča brskalnikom, da lokalno shranjujejo pogosto uporabljene vire. Ko se obiskovalci vrnejo na spletno mesto, lahko ponovno uporabijo predpomnjene datoteke CSS, JavaScript in slike, namesto da bi jih ponovno prenesli. Na strežniku lahko določimo header, ki sporoči brskalniku kako naj shranjuje sredstva. Določimo mu lahko dostopnost (public/private) in čas trajanja. Za datoteke, ki se ne

bodo spreminjale oziroma imajo že v imenu dodano verzijo lahko uporabimo tudi immutable, kar sporoči brskalniku, da je ni treba preverjati.

```
Cache-Control: public, max-age=31536000, immutable
```

Hiter uporabniški vmesnik ne more nadomestiti počasnega strežnika. Če strežnik predolgo ustvarja začetni odgovor HTML, se celotna stran zažene počasi. Poizvedbe v bazo, neučinkovita koda in prekomerni klici API-ja lahko prispevajo k dolgim odzivnim časom strežnika. Predpomnjenje pogosto zahtevanih strani ali rezultatov baze lahko pomaga izboljšati zmogljivost in precej pospeši nalaganje.

```
if (cache.ContainsKey(page))
{
    return cache[page];
}
var generatedPage = GeneratePage();
cache[page] = generatedPage;
return generatedPage;
```

Za besedilne datoteke kot so HTML, CSS, JavaScript, SVG in JSON se lahko tudi omogoči stiskanje s pomočjo Gzip ali Brotli. Stiskanje zmanjša velikost poslanih podatkov in s tem pospeši nalaganje. Primer implementacije na Node.js strežniku

```
import express from 'express';
import compression from 'compression';
const app = express();
app.use(compression());
app.get('/api/data', (req, res) => {
```

```
res.json({
  message: 'Pozdravljen svet!',
  data: '...'
});
});
app.listen(3000);
import express from 'express';
import compression from
'compression';
const app = express();
app.use(compression());
app.get('/api/data', (req, res) => {
  res.json({
    message: 'Pozdravljen svet!',
    data: '...'
  });
});
});
app.listen(3000);
```

Pred in po optimizaciji strani lahko uporabimo orodja za merjenje kot so Chrome DevTools, Lighthouse in PageSpeed Insights, ki nam omogočajo hitro prepoznavanje ozkih grl. Pri analizi je, poleg hitrosti nalaganja datotek, dobro spremljati tudi največje vsebinsko slikanje (LCP - Largest Contentful Paint), ki pove kako hitro postane glavna vsebina vidna, interakcija z naslednjim slikanjem (INP - Interaction to Next Paint), ki pove kako odzivna je stran na interakcije uporabnikov in kumulativni premik postavitve (CLS - Cumulative Layout Shift), ki pove koliko se postavitev nepričakovano premakne. Učinkovitost delovanja je treba redno spremljati, saj lahko nove slike, vtičniki, skripte in spremembe oblikovanja postopoma upočasnijo delovanje spletnega mesta.

odločitev in ne ene same optimizacije. Stiskanje slik, zmanjšanje JavaScripta, omogočanje predpomnjenja, izboljšanje odzivnih časov strežnika in uporaba CDN-ja lahko bistveno vplivajo na hitrost nalaganja strani.

Pomembno je tudi merjenje učinkovitosti delovanja pred in po implementaciji sprememb, da ugotovimo, katere optimizacije dejansko pomagajo. Hitro spletno mesto zagotavlja boljšo izkušnjo za uporabnike, porabi manj virov in lahko prispeva k boljšim poslovnim rezultatom.



Učinkovitost delovanja spletnega mesta je rezultat številnih majhnih



PIKIN VITAMIN - DOBRO JE VEDETI



Dijana Babić
Računovodstvo



Pomembne novosti in aktualne informacije

📌 Višje vrednosti nepremičnin za zdaj ne bodo vplivale na pravice iz javnih sredstev

🏠 Kaj je novega? Državni zbor je julija 2026 sprejel novelo ZMVN-1D. Pri ugotavljanju materialnega položaja za pravice iz javnih sredstev se bodo do 31. 1. 2028 še naprej uporabljale vrednosti nepremičnin po podatkih, evidentiranih na 26. 3. 2020.

👥 Zakaj je pomembno? Nove splošene vrednosti nepremičnin so zaradi rasti cen pri številnih nepremičninah višje. Če bi jih takoj začeli uporabljati pri presoji

materialnega položaja, bi lahko nekateri posamezniki in družine prejeli nižje socialne pravice ali jih izgubili, čeprav se njihov dejanski finančni položaj ni spremenil.

🇸🇮 Koliko ljudi bi lahko sprememba prizadela? Po oceni ministrstva bi brez podaljšanja prehodnega obdobja približno 3.000 gospodinjstev izgubilo denarno socialno pomoč ali varstveni dodatek, več kot 30.000 družin pa bi prejelo nižje letne pravice.

📅 Do kdaj? Prehodno obdobje velja do 31. 1. 2028; od 1. februarja 2028 naj bi bil sistem prilagojen uporabi aktualnejših podatkov.

🔍 Vir: Ministrstvo za demografijo, družino in socialne zadeve ter Geodetska uprava RS, julij 2026.

📌 Priložnost za delodajalce: do 8.640 € za novo zaposlitev

🏠 Kaj? ZRSZ ima odprto javno povabilo **Zaposli.me+ 2026**.

🇸🇮 Koliko? Delodajalec lahko za zaposlitev brezposelne osebe iz ciljne skupine prejme **od 450 do 720 € subvencije mesečno, največ 12**

mesecev.

 **Za koga?** Višina subvencije je odvisna od tega, koliko kriterijev ciljne skupine brezposelna oseba izpolnjuje.

 **Rok:** do porabe sredstev oziroma najpozneje do **30. 11. 2026**.

 **Vir:** Zavod RS za zaposlovanje – Zaposli.me+ 2026.

Sodelavci – več kot le ljudje, s katerimi delimo pisarno

Poletje in čas dopustov pogosto pokažeta nekaj, kar med letom hitro spregledamo – kako pomembni so sodelavci. Še posebej to opazimo v manjših kolektivih, kjer se v juliju in avgustu pisarne hitro izpraznijo in takrat postane delovni dan precej drugačen. Manj je pogovorov, manj spontanih vprašanj čez mizo, manj smeha in tudi manj tistega občutka, da je nekdo vedno nekje blizu.

In potem se sodelavec vrne z dopusta. Naenkrat je pisarna spet nekoliko bolj živa, kava bolj družabna, kakšna naloga lažja, predvsem pa se zavemo, da sodelavci niso le ljudje, s katerimi si delimo delovno mesto. So pomemben del našega vsakdana – in pogosto tudi naše poklicne poti.

Dobri odnosi niso samo prijeten dodatek

Na delovnem mestu preživimo velik del dneva, zato odnosi, ki jih imamo s sodelavci, pomembno vplivajo na naše počutje in način dela. To smo verjetno

že velikokrat slišali, pa se tega v resnici tudi zavedamo? Ni potrebno, da smo z vsemi prijatelji. Veliko pomembnejši so spoštovanje, korektnost in občutek, da se lahko drug na drugega zanesemo.

V okolju, kjer ljudje komunicirajo odkrito in spoštljivo, je običajno manj nesporazumov. Težave se rešujejo hitreje, informacije lažje krožijo, sodelavci pa si prej priskočijo na pomoč.

Včasih je že stavek »Povej, če ti lahko pomagam« v pravem trenutku vreden precej več, kot se zdi.

Pri sodelavcih zelo hitro opazimo, na koga se lahko zanesemo. To so običajno ljudje, ki se držijo dogovorov, pravočasno opozorijo, če naloge ne bodo mogli dokončati, priznajo napako in informacij ne zadržujejo zase. Prav slednje je lahko v nekaterih delovnih okoljih občutljiva tema. Zadrževanje informacij zaradi želje po lastnem uspehu namreč hitro načne zaupanje in dobre odnose v ekipi.

Zaradi takšnega vedenja se zaupanje ne ustvari čez noč. Nastaja skozi številne majhne vsakodnevne situacije.

Pomembno je tudi, da znamo sami prositi za pomoč. Včasih imamo občutek, da moramo vse rešiti sami, vendar je prav sodelovanje pogosto znak profesionalnosti, ne šibkosti.

Vsak človek v ekipo prinese nekaj

svojega. Nekdo je odličen organizator, drugi hitro rešuje zapletene težave, tretji dobro komunicira s strankami, četrti zna v najbolj stresnem trenutku umiriti situacijo, peti pa zna iz najbolj tihega in zadržanega sodelavca izvabiti besedo ali dve, včasih pa tudi celo zgodbo.

Če smo odprti za drugačne načine dela, lahko od sodelavcev prevzamemo marsikatero dobro navado ali znanje. Morda se sodelovanja začnemo učiti že veliko prej, kot si mislimo. Otroci v vrtcu najprej raziskujejo svet predvsem vsak zase, kmalu pa odkrijejo, da je v družbi bolj zabavno, marsikaj lažje in da lahko skupaj dosežejo nekaj, česar sami ne bi. Tudi na delovnem mestu ni prav nič drugače.

Tudi nestrinjanje je lahko koristno. Če dve osebi na problem gledata drugače, lahko skupaj prideta do rešitve, na katero posameznik sam morda ne bi pomislil. Težava nastane šele takrat, ko različno mnenje razumemo kot osebni napad.

Nekateri sodelavci nas vlečejo naprej ...

Verjetno je skoraj vsak že imel sodelavca, ob katerem je postal boljši pri svojem delu.

Takšni ljudje delijo znanje, spodbujajo ideje, opozorijo na napako brez poniževanja in se veselijo tudi uspeha drugih. Ob njih se lažje lotimo novih nalog in si upamo prevzeti več odgovornosti.

Včasih prav sodelavec prvi opazi, da smo v nečem dobri, in nas spodbudi, da to sposobnost razvijemo naprej. Takšni odnosi imajo lahko dolgoročen vpliv na našo kariero.

... drugi pa nam lahko jemljejo energijo

Vsi odnosi seveda niso takšni.

Obstajajo tudi odnosi, ob katerih imamo občutek, da moramo nenehno paziti, kaj bomo rekli. Nekateri ljudje v delovno okolje prinašajo veliko negativnosti, drugi neradi delijo informacije, tretji pa uspeh sodelavca težko doživijo tudi kot uspeh ekipe.

Takrat je pomembno postaviti meje.

Profesionalen odnos namreč ne pomeni, da moramo biti z vsemi prijatelji. Pomeni predvsem, da znamo sodelovati korektno tudi z ljudmi, s katerimi se osebno ne ujamemo.

Včasih je prav sposobnost sodelovanja z zelo različnimi ljudmi ena najbolj dragocenih veščin, ki jih pridobimo skozi delovno življenje.

Današnji sodelavec je lahko jutri poslovni partner

Še ena stvar, ki jo med vsakodnevnim delom hitro pozabimo, je dolgoročnost odnosov.

Ljudje menjajo podjetja, napredujejo, začnejo svoje projekte ali odidejo na popolnoma druga področja. Nekdanji sodelavec lahko čez nekaj let postane stranka, poslovni partner, priporočitelj

ali oseba, ki nam pove za zanimivo zaposlitveno priložnost.

Zato se profesionalne mreže pogosto gradijo precej bolj spontano, kot si predstavljamo.

Ni potrebno, da poznamo na stotine ljudi. Nekaj kakovostnih odnosov, zgrajenih na zaupanju in korektnosti, je lahko veliko dragocenejših.

In ko je pisarna spet polna

Morda se zato ob koncu poletja nekoliko bolj razveselimo znanega glasu iz sosednje pisarne, vprašanja »Gre kdo na kavo?« ali sodelavca, ki se po dveh tednih dopusta pojavi na vratih.

Dopusti nam za nekaj časa pokažejo, kako bi bilo, če teh ljudi ne bi bilo.

In nas mimogrede spomnijo, da dobro delovno okolje ne nastane samo

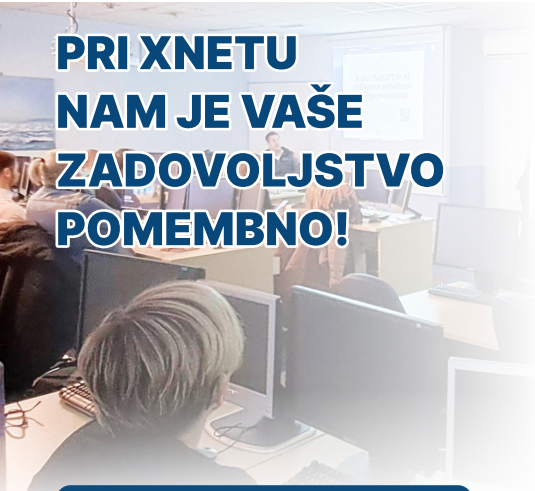
zaradi zanimivega dela, dobre plače ali urejene pisarne.

V veliki meri ga ustvarjajo ljudje. Tisti, s katerimi vsak dan delimo naloge, težave, uspehe, kakšno slabo voljo in – če imamo srečo – tudi precej smeha.

Ste vedeli?

Gallupove raziskave kažejo, da so zaposleni, ki imajo na delovnem mestu osebo, ki jo štejejo za svojega najboljšega prijatelja, kar **sedemkrat pogosteje zavzeti pri delu**. Njihova obsežna analiza več kot 180.000 ekip pa je pokazala tudi, da bi lahko več tesnih odnosov v ekipah prispevalo k **višji kakovosti dela, večji dobičkonosnosti in manj varnostnim incidentom**.

Vir: Gallup, Why Work Friends Matter: Unlocking Powerful Partnerships with CliftonStrengths, 2025.



**PRI XNETU
NAM JE VAŠE
ZADOVOLJSTVO
POMEMBNO!**

Pridruži se zadovoljnim tečajnikom

✉ info@kompas-xnet.si ☎ +386 1 5136 990

☞ Matej, NUK

Na predavanju mi je bilo všeč, da je predavatelj artikulirano in strokovno predstavil uporabo ChatGPT. Njegova razlaga je bila jasna, strukturirana in razumljiva, kar je omogočilo, da sem temo dobro razumel in se poglobil v njene možnosti in omejitve.

☞ Jean, HSE d.o.o.

Jože je izjemen strokovnjak na področju SCCM orodja. Zahtevna poglavja je predstavil na način, da smo razumeli tudi uporabniki, ki z orodjem še nimamo veliko izkušenj. Delil je tudi veliko koristnih informacij in dodatnih orodij, ki olajšajo delo s SCCM-jem. Vsekakor izjemno zadovoljen s predavateljem.

☞ Aljaž, DEM d.o.o.

Super izobraževanje, vse pohvale predavatelju Mihi. Soočil nas je s težavami, ki jih lahko imamo ter vso teorijo zelo dobro prenesel v prakso.

☞ Danijel, Arriva d.o.o.

Dejan je super predavatelj, z ogromno znanja ter razgledanostjo in smisla za humor.



ZAKAJ ORGANIZACIJE POTREBUJEJO »VOZNIŠKI IZPIT« ZA UPORABO UMETNE INTELIGENCE?



Petra Militarev

vodja izobraževanj

petra.militarev@kompas-xnet.si



Umetna inteligenca postaja del vsakodnevnega dela. Zaposleni jo uporabljajo za pripravo dokumentov, analizo podatkov, povzemanje sestankov in avtomatizacijo opravil. Vse več organizacij zato vlaga v orodja, kot je Microsoft 365 Copilot, saj želijo povečati produktivnost in hitrost dela.

Toda uvedba umetne inteligence odpira pomembno vprašanje: ali je dostop do tehnologije dovolj?

Primerjava z vozniškim izpitom ponuja zanimiv odgovor. Za uporabo avtomobila ne zadostuje, da imamo

ključ. Poznati moramo pravila, razumeti tveganja in prevzeti odgovornost za svoje ravnanje. Podobno velja za umetno inteligenco.

AI ne zmanjšuje potrebe po znanju

Pogosto slišimo, da bo umetna inteligenca poenostavila delo. To drži. Manj pogosto pa govorimo o tem, da uspešna uporaba AI zahteva nova znanja.

Orodja, kot so Microsoft 365 Copilot, ChatGPT in drugi digitalni asistenti, lahko ustvarijo osnutke dokumentov, pripravijo analize ali predlagajo rešitve.

Kakovost rezultatov pa je še vedno odvisna od uporabnika.

Ključna kompetenca prihodnosti ne bo hitrost tipkanja, temveč sposobnost postavljanja pravih vprašanj, razumevanja podatkov, preverjanja rezultatov in kritične presoje informacij. AI pismenost, podatkovna pismenost, digitalna varnost in razumevanje skladnosti postajajo enako pomembni kot poznavanje poslovnih procesov.

Kaj bi moral znati vsak uporabnik AI?

Predstavljajmo si novega zaposlenega, ki prejme službeni računalnik, dostop do Microsoft 365 in licenco za Copilot.

Preden začne orodje uporabljati pri vsakodnevnem delu, bi moral razumeti:

- katere informacije so zaupne in občutljive,
- kako preveriti pravilnost odgovorov AI,
- katere so omejitve umetne inteligence,
- kako ravnati v skladu s pravili informacijske varnosti,
- kako odgovorno uporabljati podatke organizacije.

Če teh znanj nima, tveganje ni več tehnično, temveč poslovno.

Neustrezen AI-povzetek lahko napačno predstavi ugotovitve sestanka. Slabo zastavljeno vprašanje lahko razkrije občutljive podatke. Avtomatiziran proces brez ustreznega nadzora pa lahko napake širi bistveno

hitreje kot človek.

Težava ni tehnologija, temveč način uporabe

V praksi so tveganja pogosto manj očitna, kot si predstavljamo. Zaposleni lahko uporabi Copilot za pripravo povzetka sestanka in ga brez preverjanja posreduje vodstvu. Če umetna inteligenca napačno interpretira ključne ugotovitve, lahko organizacija sprejema odločitve na podlagi netočnih informacij.

Podobno lahko AI pomaga pri analizi strank, vendar neustrezno zastavljena vprašanja hitro pripeljejo do napačnih zaključkov ali neprimerne uporabe podatkov.

V obeh primerih težava ni tehnologija, temveč pomanjkanje znanja in kritične presoje uporabnika.

AI pripravljenost postaja nova organizacijska kompetenca

Uspešna uvedba umetne inteligence ni odvisna le od tehnologije. Organizacije morajo vzpostaviti jasna pravila uporabe, ustrezno upravljanje podatkov, podporo vodstva ter sistematičen razvoj kompetenc zaposlenih.

Nakup licence je danes razmeroma preprost. Veliko zahtevnejše pa je zagotoviti, da zaposleni umetno inteligenco uporabljajo učinkovito, varno in v skladu s poslovnimi cilji.

Zato vse več organizacij uvaja programe AI pismenosti, interna usposabljanja in certifikacije.

Ne zaradi nadzora, temveč zato, da zmanjšajo tveganja in povečajo poslovne učinke uporabe umetne inteligence.

Znanje postaja nova infrastruktura

Vsaka večja tehnološka sprememba je organizacijam prinesla isto spoznanje: največje vrednosti ne ustvarja tehnologija sama, temveč ljudje, ki jo znajo uporabljati.

Tudi pri umetni inteligenci ne bo drugače. Najuspešnejše organizacije ne bodo nujno tiste z največ licencami ali najnaprednejšimi orodji. Prednost bodo imele tiste, ki bodo sistematično razvijale znanje zaposlenih, spodbujale odgovorno uporabo in merile dejanske poslovne učinke.

Prihodnost ne bo pripadala organizacijam, ki imajo umetno

inteligenco. Pripadala bo tistim, ki jo znajo uporabljati.

Pravo vprašanje danes zato ni več, ali uvesti Microsoft 365 Copilot ali drugo AI-orodje. Pomembnejše je vprašanje, ali imajo zaposleni znanje za njegovo učinkovito, varno in odgovorno uporabo.

Če smo se pri avtomobilih že zdavnaj strinjali, da za vožnjo potrebujemo usposabljanje in preverjeno znanje, bomo morali podobno razmišljati tudi pri umetni inteligenci.

Kajti v dobi umetne inteligence konkurenčne prednosti ne bo predstavljal dostop do tehnologije, temveč znanje, kako jo uporabljati odgovorno, varno in učinkovito.



Poslovni zajtrk

Xnet
vaš zanesljiv IT partner

Projektno vodenje in AI: Kaj mora vedeti sodobni projektni vodja?

Pridružite se brezplačnemu poslovnemu zajtrku in odkrijte, kako umetna inteligenca spreminja projektno vodenje. Spoznajte dobre prakse, konkretne primere uporabe AI ter možnosti nadaljnjega usposabljanja za projektne vodje.



11. 9. 2026



9.00-10.00



Online



Brezplačna prijava. Omejeno število mest.

Jesenski tečaji 2026

Več odličnih tečajev vas čaka na [tej povezavi](#)



Microsoft Copilot in umetna inteligenca

17.9.2026	PowerPoint Copilot: Ustvarjanje privlačnih predstavitev s pomočjo Copilota	Prijava >
18.9.2026	Word Copilot: Učinkovito oblikovanje in urejanje dokumentov s pomočjo Copilota	Prijava >
7.10.2026	MS-4018 Draft, analyze, and present with Microsoft 365 Copilot	Prijava >
8.10.2026	MS-4023 Transform Ideas into Action with Copilot Chat	Prijava >
9.10.2026	PL-7008 Create agents in Microsoft Copilot Studio	Prijava >
14.10.2026	Outlook Copilot: Učinkovito upravljanje e-pošte in koledarja s pomočjo Copilota	Prijava >



Microsoft Azure, infrastruktura in administracija

7.9. - 11.9.2026	AZ104 Microsoft Azure Administrator	Prijava >
14.9. - 18.9.2026	MD102 Microsoft 365 Endpoint Administrator	Prijava >
28.9. - 30.9.2026	AZ700 Microsoft 365 Endpoint Administrator	Prijava >
5.10. - 9.10.2026	Identity with Windows Server	Prijava >
5.10. - 8.10.2026	Supporting and Troubleshooting Windows 11	Prijava >
12.10. - 15.10.2026	AZ140 Configuring and Operating Microsoft Azure Virtual Desktop	Prijava >



Kibernetska varnost

7.9. - 11.9.2026	Hacking and Securing Windows Infrastructure, LIVE VIRTUAL	Prijava >
10.9.2026	Cloud Post-Exploitation Mitigation: Azure & Microsoft 365	Prijava >
14.9. - 17.9.2026	SC-300 Microsoft Identity and Access Administrator	Prijava >
14.9. - 17.9.2026	Incident Response in the Cloud	Prijava >
16.9. - 18.9.2026	Threat Hunting with AI Support	Prijava >
21.9. - 24.9.2026	SC-200 Defend against cyberthreats with Microsoft's security operations platform	Prijava >



Power BI, SQL Server in podatkovna analitika

7.9. - 11.9.2026	Administering a SQL Database Infrastructure	Prijava >
14.9. - 16.9.2026	Administracija in razvoj baz podatkov na strežniku SQL Server	Prijava >
16.9. - 17.9.2026	Uvod v Excel BI	Prijava >
21.9. - 23.9.2026	PL-300 Design and manage analytics solutions using Power BI	Prijava >
28.9. - 30.9.2026	Power BI	Prijava >
28.9.2026	T-SQL Language Beyond Fundamentals	Prijava >



SharePoint, Power Platform in avtomatizacija

15.9. - 17.9.2026	SharePoint Online for Administrators	Prijava >
28.9. - 1.10.2026	SharePoint Online Power User	Prijava >
5.10.2026	55302AC Microsoft Lists Super User	Prijava >
5.10.2026	PL-7001 Create and Manage Canvas Apps with Power Apps	Prijava >
6.10.2026	PL-7002 Create and Manage Automated Processes by using Power Automate	Prijava >
6.10. - 9.10.2026	55311AC SharePoint Server SE Super User	Prijava >
8.10.2026	PL-900 Introduction to Microsoft Power Platform	Prijava >
12.10. - 16.10.2026	PL400 Microsoft Power Platform Developer	Prijava >



Office in sodobno digitalno delo

2.9. - 4.9.2026	Microsoft Excel nadaljevalni	Prijava >
7.9.2026	Microsoft PowerPoint	Prijava >
8.9. - 10.9.2026	Microsoft Word	Prijava >
8.9.2026	Microsoft 365 v praksi: Sodelovanje in komunikacija	Prijava >
10.9.2026	Vrtilne tabele	Prijava >
11.9.2026	Excel mojster 2	Prijava >
14.9.2026	Excel mojster 4	Prijava >
16.9. - 17.9.2026	Sodobna pisarna in uporaba Microsoft 365 orodij 2.del	Prijava >



Projektno vodenje in ITSM

11.9.2026	Prince2 7th Edition Foundation	Prijava >
14.9. - 16.9.2026	PRINCE2 Foundation	Prijava >
16.9. - 18.9.2026	ITIL® V5: Foundation	Prijava >



Razvoj spletnih in poslovnih rešitev

21.9. - 25.9.2026	Programming with HTML, CSS, and JavaScript	Prijava >
20.10. - 21.10.2026	PowerShell for Digital Investigation & Triage	Prijava >

Osvoji nova znanja in postani specialist!



Copilot

Tips & Tricks

Xnet
vaš zanesljiv IT partner



Copilot Excel: Predlagaj stolpec s formulo

Copilot v Excelu na podlagi obstoječih podatkov predlaga nov stolpec s primerno formulo ter ga samodejno izračuna za celotno tabelo.



Poglej



Copilot Word: Priprava govornih točk

Copilot na podlagi vsebine Word dokumenta samodejno pripravi jasne in strukturirane govorne točke, ki olajšajo predstavitev in poudarijo ključne informacije.



Poglej



Copilot PowerPoint: Prilagajanje vsebine

Uporabi Microsoft Copilot za preureditev vsebine glede na podana navodila, pri tem pa samodejno prilagodi strukturo, poudarke in tok besedila.



Poglej



**ŠE VEČ COPILOT TRIKOV IN NASVETOV
VAS ČAKA NA NAŠEM KANALU TUKAJ**

kompas-xnet.si