

HUGO

A statikus oldalgenerátor

Palócz István

Ez a mű az FSF.hu Alapítvány SzabadSzoftver Pályázat 2022 támogatásával készült.

Liszensz: CC-BY-SA

Verzió: 2023-01-13-2ac13b4

Tartalomjegyzék

Bevezetés	3
Hogyan induljunk el?	3
Tároló létrehozása	4
Ismerkedés a menüvel	5
Code	6
Actions	6
Settings	6
Segédeink a robotok	6
Az oldal publikálása	7
Adatvédelmi figyelmeztetés	10
Hol vannak az adatok és hogyan vannak tárolva?	10
Módosítás	11
Feladat	15
Markdown röviden	15
Formázás	16
Új oldal létrehozása	16

Ábrák jegyzéke

1. Hugo logo	3
2. Új tároló létrehozása. 1. lépés	4
3. Új tároló létrehozása. 2. lépés	5
4. Github menürendszere	6
5. Hogyan állnak a robotok a munkával	6
6. Settings oldal	7
7. Pages oldal	8
8. Pages oldal	8
9. Készül az oldal	9
10. Environments menü, ahonnan a weboldal elérhető	9
11. Environments menü, ahonnan a weboldal elérhető	10
12. Code menü és a mappaszerkezet	11
13. content mappa tartalma	12
14. content mappa tartalma	12
15. content mappa tartalma	13
16. content mappa tartalma	14
17. content mappa tartalma	15
18. Action oldal és a New Page, New Post mintafeladat	17
19. New Page mintafeladat	17
20. New Page mintafeladat	18
21. New Page mintafeladat	18

Bevezetés



1. ábra. Hugo logo

A Hugo egy olyan megoldás, ami szövegfájlokból generál HTML fájlokat. Egy Hugoval készült weboldal módosításához, szerkesztéséhez nincs szükségünk speciális programra, csak egy egyszerű szövegszerkesztőre. A forrásfájlok könyvtár szerkezete tükröződni fog az előállított weboldal útvonalaiban. Tehát, ha egy forrásfájl a **termekek** nevű könyvtárban szerepel, **sportcipo** néven, akkor az a weboldalon a `/termekek/sportcipo` útvonalon lesz elérhető. Amennyiben egy könyvtárban több szövegfájl található, akkor a Hugo generál hozzájuk egy lapozható listát. Például, ha a **termekek** könyvtárban még egy **lakkcipo** forrásfájl is van, akkor a `/termekek` útvonalon egy lista jelenik meg, melyben két elem lesz, a **sportcipo** és a **lakkcipo**. Nem kell tehát mást tennünk, mint egyszerű szövegfájlokat szerkeszteni, amikből egy sablon segítségével több összefüggő HTML oldalt, vagyis egy teljes weboldalt állít elő a Hugo.

Az így elkészített HTML oldalakat azután egy webszerver megfelelő könyvtárába kell elhelyeznünk, hogy azok elérhetőek legyenek az interneten keresztül is.

Mindent összevetve tehát a Hugoval készített weboldalak módosítása és publikálás két lépésből áll. Először a módosított szövegfájlokból le kell generálni a weboldalt. A második lépésként a legenerált HTML állományokat fel kell másolni a megfelelő helyükre.

Természetesen ez csak akkor ilyen egyszerű, ha minden elő van készítve. Van megfelelő domain név, hozzá tartozó tárhellyel és beállított webszerverrel, valamint a Hugo program telepítve van a gépünkre és egy könyvtár fel van töltve a szükséges konfigurációs állományokkal, hogy a Hugo tudja hol keresse a szövegfájlokat, milyen sablont használjon és hogyan állítsa elő belőle a weboldalt.

Azért, hogy a belépési küszöböt, vagyis az elinduláshoz szükséges lépések számát lecsökkentsem készítettem egy megoldást, ami segítségével pár egyszerű lépés után egy működő megoldásunk van. Ez a megoldás alkalmas arra, hogy az alapokkal megismerkedjünk, vagy a jövőben egy kisebb oldalt üzemeltessünk vele. Ez lehet egy személyes blog vagy egy kisebb cég bemutatkozó oldala.

Az itt található tananyag mélységét tekintve igazodik a Nemzeti Alaptantervben és a Kerettantervben megfogalmazott Tartalomkezelő témakör mélységéhez, annak szélesítéséhez próbál meg hozzájárulni.

Hogyan induljunk el?

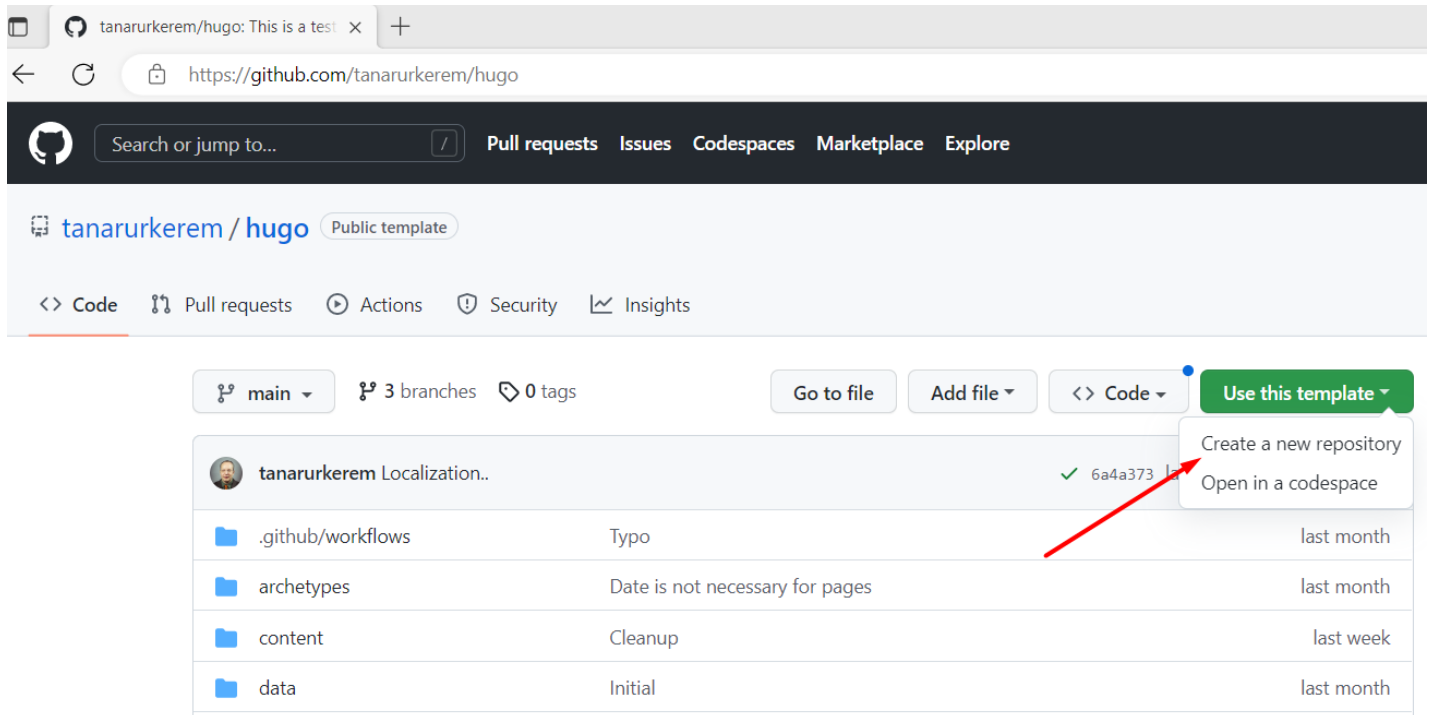
Első körben szükségünk lesz egy GitHub felhasználóra, melyet <https://github.com/> oldalon a Sign up gombra kattintva hozhatunk létre. Adjuk meg az email címünket és egy megfelelően erős jelszót. A bejelentkezési név (username) választásakor vegyük figyelembe, hogy ez a név majd szerepelni fog az elkészített weboldal címében. A legjobb, ha itt követed a tanárod ajánlását, én a teljes név ékezetellen verzióját és valamilyen az iskolára jellemző adatot javaslok. Fontos, hogy az elválasztásra

kötőjelet (-), használj és semmi esetre se pontot! Például: szabo-klara-suli-14b Lehetséges, hogy ki kell raknod majd egy kirakóst, hogy ellenőrizték nem vagy robot és az email címedet is validálnod kell majd.

Sikeres regisztráció után látogass el a <https://github.com/tanarurkerem/hugo> oldalra, hogy új tárolót tudj készíteni. Mutatom hogyan.

Tároló létrehozása

Első lépésként a **Use this template** gombra kattintva válaszd ki a **Create a new repository** menüpontot.



2. ábra. Új tároló létrehozása. 1. lépés

Második lépésként állítsd be a tároló nevét (**Repository name**). Első próbálkozásnál legyen nyugodtan az, hogy **hugo**. Fontos, hogy a tároló láthatósága maradjon nyilvános (Public), különben nem lesz elérhető a weboldal. A **Create repository from template** gombra kattintva létre is jön a tároló.

Create a new repository from hugo


The new repository will start with the same files and folders as [tanarurkerem/hugo](#).


Owner * Repository name *

/


Great repository names are short and memorable. Need inspiration? How about [curly-guacamole?](#)

Description (optional)

Public  Anyone on the internet can see this repository. You choose who can commit.

Private  You choose who can see and commit to this repository.

Include all branches
Copy all branches from tanarurkerem/hugo and not just main.

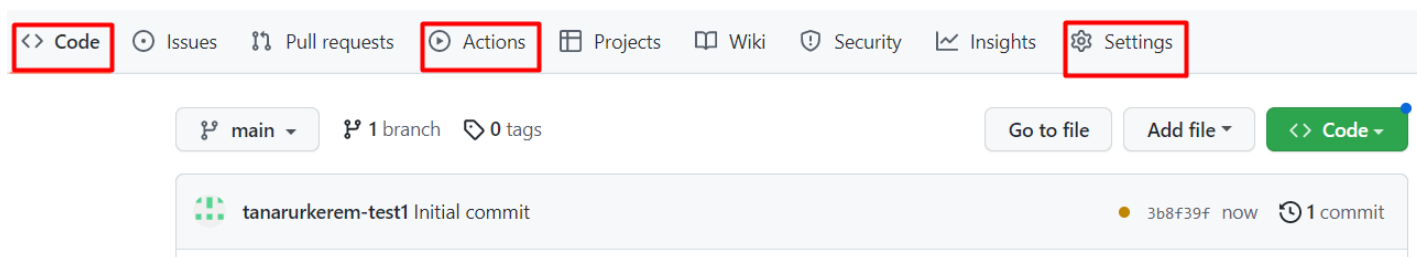
 You are creating a public repository in your personal account.

[Create repository from template](#)

3. ábra. Új tároló létrehozása. 2. lépés

Ismerkedés a menüvel

A GitHub rendszer jóval többet tud mint amire nekünk szükségünk van, ezért most ismerjük meg azt a három fő menüpontot amit használni fogunk.



4. ábra. Github menürendszere

Code

A **Code** (magyarul **Kód**) menüpont lesz ahol a szerkesztéseket fogjuk elvégezni. Ez az origó, mindig ide térünk vissza és innen indítjuk a módosításokat.

Actions

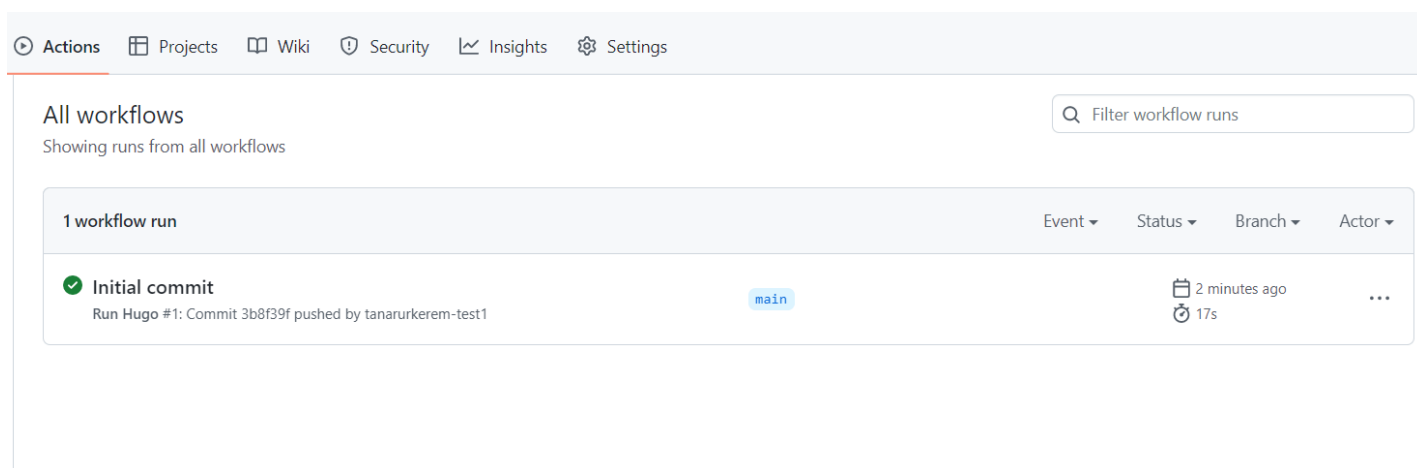
A munkánkat robotok fogják segíteni. Itt tudjuk nyomonkövetni a munkájukat és itt tudjuk majd utasítani őket, ha új tartalmat szeretnénk hozzáadni az oldalainkhoz.

Settings

A **Settings** (magyarul **Beállítások**) menüpontot mindjárt itt az elején fogjuk használni, valamint itt tudod majd törölni a tárolót, ha már nincs szükséged rá.

Segédeink a robotok

A GitHub robotjai nem személyes használatra adott robotok. Ezért akkor fogják elkészíteni a rájuk bízott feladatot, amikor idejük engedi. Ez általában a fél perctől az egy-két percig terjed, de nagy terhelésnél nem ritka az, hogy ritkábban jut ránk idejük. Türelem! A robotok munkáját a **Action** fülön tudod követni.



5. ábra. Hogyan állnak a robotok a munkával

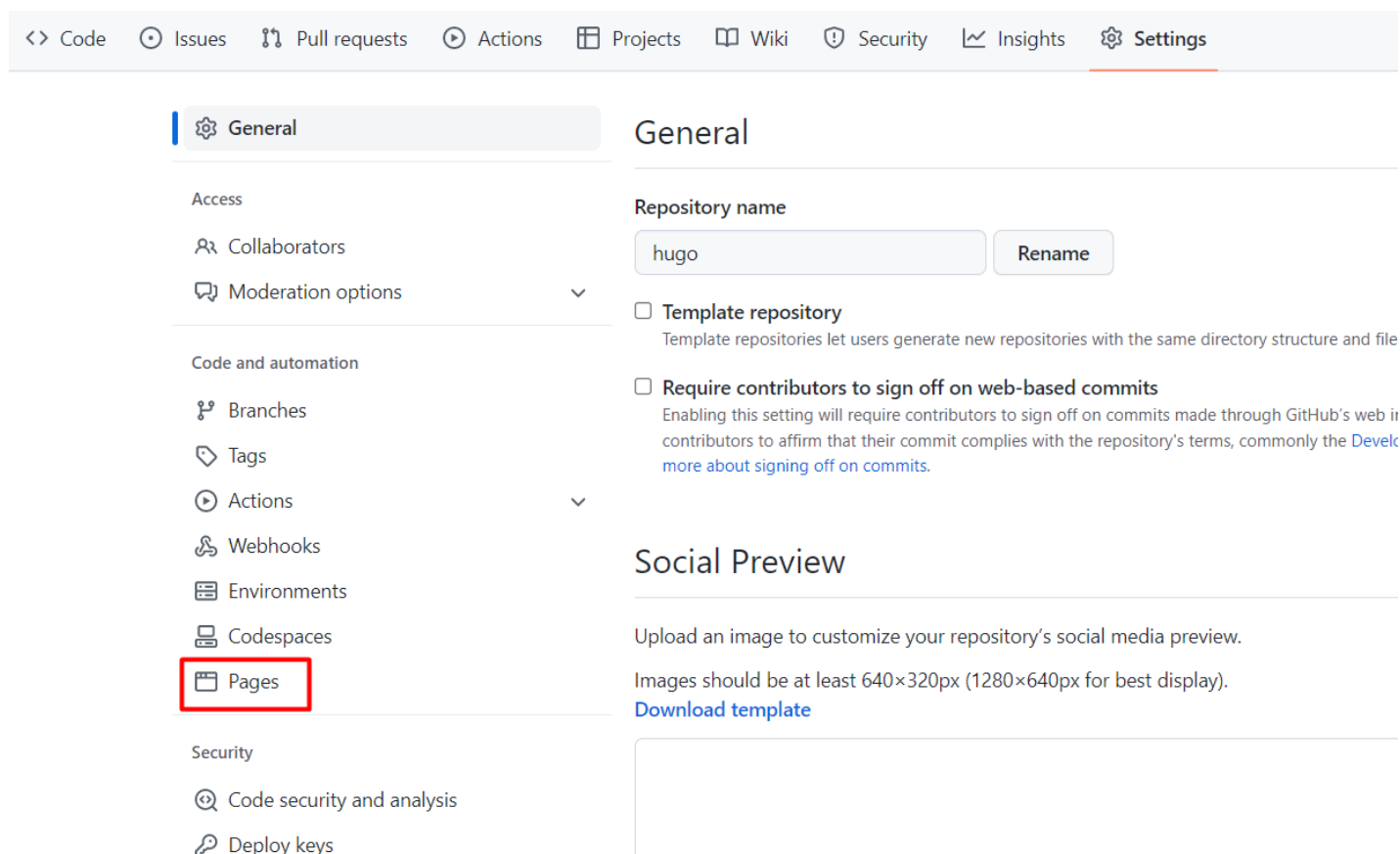
A képen azt láthatod, hogy egy feladattal végzett egy robot. Azért tudom, hogy végzett, mert zöld színű előtte a pipa. Amennyiben narancssárga, akkor vagy várakozik a feladat, vagy éppen végrehajtás alatt van. Amennyiben piros a feladat előtti jelölő, szólj az oktatónak, mert valami olyan probléma van, amit csak ő, vagy ő sem tud elhárítani.

A képen azt láthatod, hogy a legelső feladat befejeződött. Az a feladat amely során a szövegfájlokból HTML fájlok készültek a Hugo segítségével. Mint korábban említettem még egy lépés van, hogy élesben is elérhető legyen a weboldalad, amihez utasítani kell a GitHubot, hogy publikálja azt. Ezt addig nem tudtuk megtenni, amíg el nem készült a weboldalunk. Most, hogy kész van, vagyis az első feladat előtt lévő jelzés zöldre váltott utasíthatjuk a GitHubot, hogy ezt is tegye meg. Amennyiben nálad még nem zöld a feladat, úgy várj addig, amíg zöldre vált. Ne lépj tovább, amíg ez meg nem történik. Ugyan így minden módosítás után ellenőrizd, hogy zöld-e a jel, mert addig nem fogod látni a változást! Türelem!

Az oldal publikálása

Az oldal publikálásához csak egyetlen egyszer kell beállítanunk az alábbiakat. Amint ezzel megvagyunk, a rendszer minden egyes módosítás után automatikusan elvégzi a publikálást.

Irány a **Settings** menü és azon belül a **Pages** menüpont.



The screenshot shows the GitHub Settings interface. At the top, there's a navigation bar with options like Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The left sidebar has a 'General' section with sub-items: Access (Collaborators, Moderation options), Code and automation (Branches, Tags, Actions, Webhooks, Environments, Codespaces, Pages), and Security (Code security and analysis, Deploy keys). The 'Pages' item is highlighted with a red box. The main content area is titled 'General' and includes a 'Repository name' field with 'hugo' and a 'Rename' button. Below that are two unchecked checkboxes: 'Template repository' and 'Require contributors to sign off on web-based commits'. The 'Social Preview' section follows, with instructions to upload an image and a 'Download template' link.

6. ábra. Settings oldal

A **Pages** oldalon a **Branch** beállításnál a **None** értéket kell átállítani **gh-pages** értékre. A mentés gomb megnyomása után el is indul a weboldal publikálása.

The screenshot shows the GitHub Pages configuration interface. On the left, a sidebar lists various settings categories: General, Access (Collaborators, Moderation options), Code and automation (Branches, Tags, Actions, Webhooks, Environments, Codespaces, Pages), and Security (Code security and analysis, Deploy keys). The 'Pages' section is currently selected. The main area is titled 'GitHub Pages' and contains the following settings:

- Build and deployment:** A section with a 'Save' button highlighted in red.
- Source:** A dropdown menu set to 'Deploy from a branch'.
- Branch:** A dropdown menu set to 'gh-pages', with a 'Save' button highlighted in red. Below it, a 'Select branch' dialog is open, showing a list of branches: 'main', 'gh-pages' (checked and highlighted in red), and 'None'.

7. ábra. Pages oldal

A folyamatot az Action menüben tudjuk nyomonkövetni. A képen azt láthatod, ahogy a feladat arra vár (Queued), hogy egy robot ráérjen és elvégezze.

The screenshot shows the GitHub Actions interface. The left sidebar has 'Actions' selected, with options for 'New workflow', 'All workflows', 'New Page', 'New Post', 'pages-build-deployment', 'Run Hugo', 'Management', and 'Caches'. The main area is titled 'All workflows' and shows a table of workflow runs:

Event	Status	Branch	Actor
pages build and deployment	Queued		
Initial commit	Completed	main	

8. ábra. Pages oldal

Amint ez a feladat itt zöld lesz navigáljunk el a **Code** menübe.

The screenshot shows a GitHub repository interface. At the top, there are navigation tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation, there's a header with 'main' branch selected, '2 branches', and '0 tags'. A warning box states 'Your main branch isn't protected' with a 'Protect this branch' button. The main content area displays a file list for the 'tanarurkerem-test1' repository, showing various folders and files, all marked as 'Initial commit' and '6 minutes ago'. On the right side, there's an 'About' section with 'No description, website, or topics provided', and sections for 'Releases' (no releases published) and 'Packages' (no packages published). The 'Environments' section shows one environment named 'github-pages' which is 'Active'.

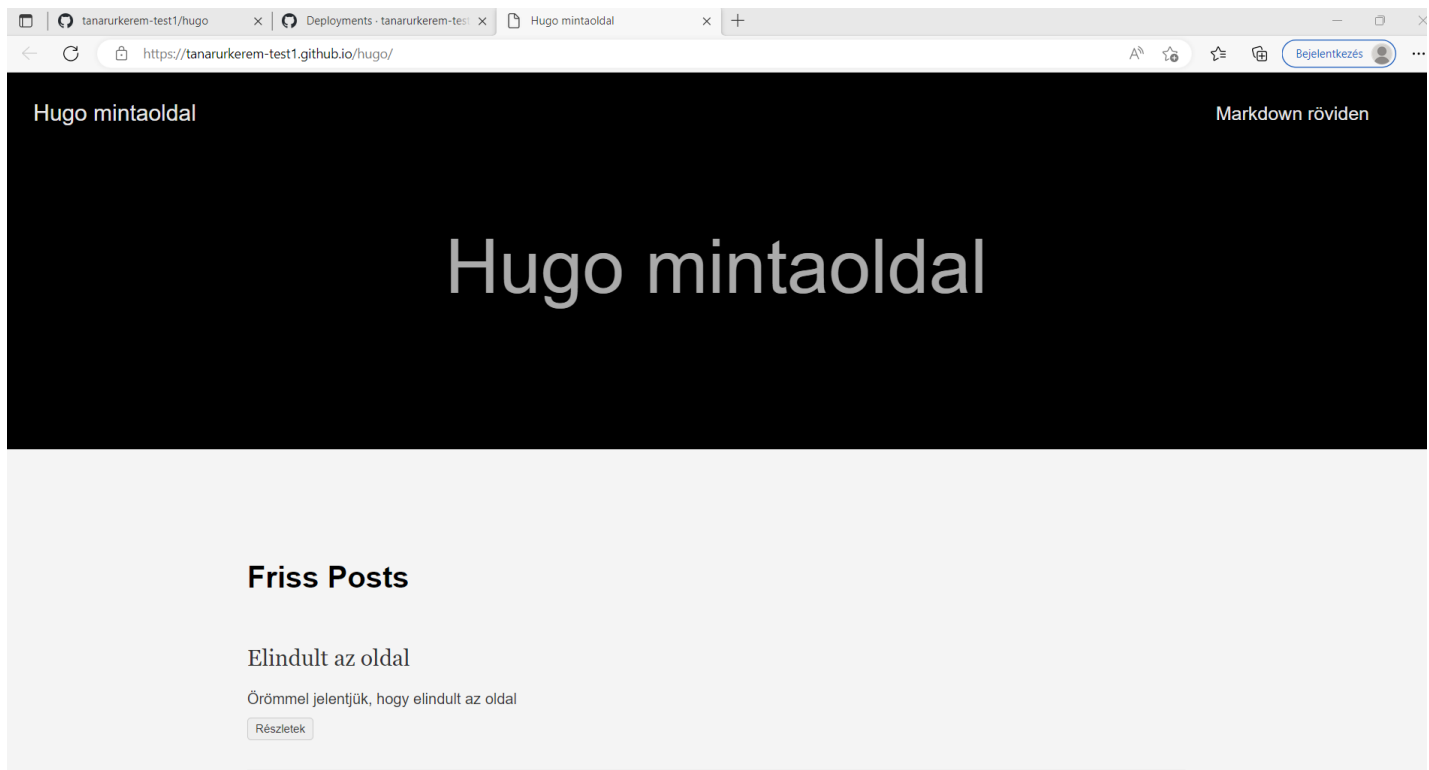
9. ábra. Készül az oldal

Itt jobb alul látsz majd egy Environments blokkot, amiben a **github-pages** linkre kell kattintanod.

The screenshot shows the 'Deployments / History' page in a GitHub repository. At the top, there are navigation tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation, there's a header with 'Deployments / History' and a 'Show: github-pages' dropdown. The main content area displays a deployment entry for 'github-pages' at '1d4bcd1', deployed by GitHub Pages on behalf of 'tanarurkerem-test1' 2 minutes ago, with an 'Active' status. A 'View deployment' button is highlighted with a red box.

10. ábra. Environments menü, ahonnan a weboldal elérhető

Végül a **View deployment** gombra kell kattintani, hogy láthatóvá váljon az oldal. Valami ilyesmit kell kapnod:



11. ábra. Environments menü, ahonnan a weboldal elérhető

Ha megfigyeled az oldal címét, akkor láthatod, hogy a bejelentkezési neveddel kezdődik és a tároló nevével végződik közepén a github.io domain névvel. Ez lesz az a cím, amin keresztül elérhető lesz az oldalad a világ számára.

Adatvédelmi figyelmeztetés

Itt most meg kell állnom és meg kell jegyeznem valami fontosat. Ez az oldal inentől kezdve elérhető az interneten. Ezért **el kell kerülni, hogy bármilyen személyes adat vagy érzékeny információ felkerüljön**, mert az egyből láthatóvá válik mindenki számára. Ráadásul mivel a GitHub egy verziókezelő rendszer ezért, az összes verziója az oldalnak elérhető lesz a későbbiek folyamán. Tehát, ha felraksz az oldalra valamilyen érzékeny adatot, majd törlöd, attól az még elérhető marad. Szóval csak óvatosan ezzel!

Hol vannak az adatok és hogyan vannak tárolva?

Kiindulás mindig a **Code** menü legyen. Legalábbis kezdetben. Később amikor már kiismertük eléggé a GitHubot, akkor már nem lesz szükségünk erre a mankóra, de most az elején jól jön majd. Bárhol vagyunk is az oldalon, ha megnyomjuk a **Code** fület, akkor visszajutunk ide. Itt látjuk a Hugo projektünk mappaszerkezetét.

tanarurkerem-test1 Initial commit		✓ 3b8f39f 3 days ago	🕒 1 commit
📁 .github/workflows	Initial commit		3 days ago
📁 archetypes	Initial commit		3 days ago
📁 content	Initial commit		3 days ago
📁 data	Initial commit		3 days ago
📁 layouts	Initial commit		3 days ago
📁 public	Initial commit		3 days ago
📁 static	Initial commit		3 days ago
📁 themes	Initial commit		3 days ago
📄 .gitignore	Initial commit		3 days ago
📄 .gitmodules	Initial commit		3 days ago
📄 README.md	Initial commit		3 days ago
📄 config.toml	Initial commit		3 days ago

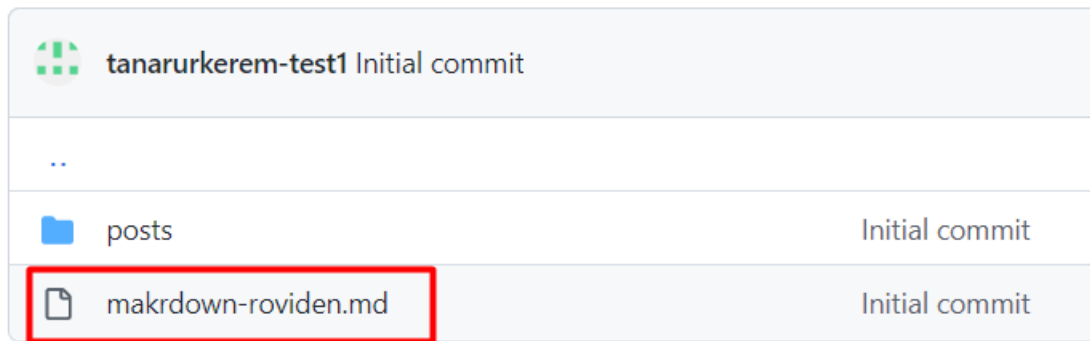
12. ábra. Code menü és a mappaszerkezet

Ebben a rövid anyagban csak a **content** és a **public** nevű mappákról és a **config.toml** fájlról lesz szó. A Hugo jóval több lehetőséget rejt, mint amit ebben a rövid bevezető anyagban tárgyalunk, ezért nem lesz szó a többiről. Amennyiben érdekel részletesebben is a téma, úgy az angol nyelvű Hugo Dokumentációból (<https://gohugo.io/documentation/>) tájékozódhatsz.

A **content** mappában találhatóak a szöveges tartalmak, a **public** mappában pedig az egyéb fájlok, mint például képek, dokumentumok. A **config.toml** fájlban pedig az oldal beállításai találhatóak.

Módosítás

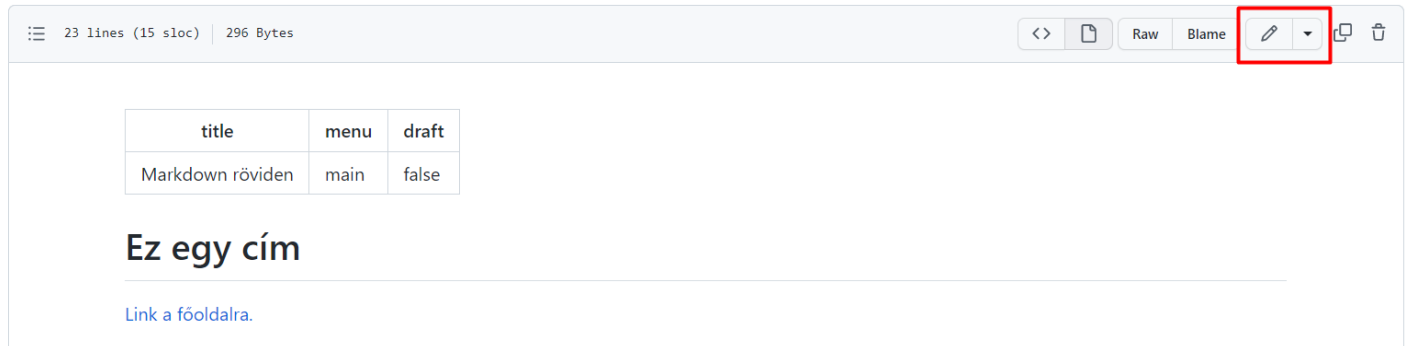
Először nézzük meg a content mappát.



13. ábra. content mappa tartalma

Találunk benne egy **posts** mappát és egy **markdown-roviden.md** nevű fájlt. Ebben a példában kétfajta tartalom típus van. Page(Oldal) és Post(Bejegyzés, Írás, Blogpost). A **Page** egy önmagában álló oldal, ami megjelenik a menüben. Ilyen a weboldalon található **Markdown röviden** című oldal is. A **Post** az egy olyan tartalom, ami egyrészt megjelenik a főoldalon listázva, másrészt önmagában is megjeleníthető, tehát ennek a tartalom típusnak két nézete - más szóval - megjelenése van.

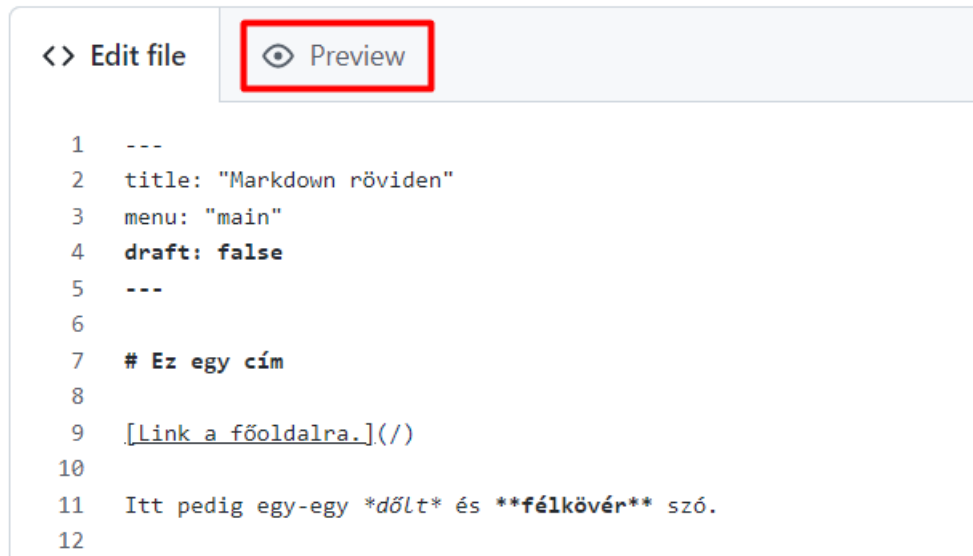
Kattintsunk a **markdown-roviden.md** fájlra, majd a jobb felső sarokban található ceruza ikonra, hogy szerkeszteni tudjuk a fájlt. Itt a legördülőben több opció is választható, többek között egy online VSCode editor is, de mi most az egyszerűség kedvéért az alapértelmezett beállítást használjuk.



14. ábra. content mappa tartalma

A megjelenő szerkesztő eszközzel fogjuk tudni módosítani a fájlt, valamint található egy **Preview** gomb, ami segítségével megnézhetjük, hogy hogyan fog kinézni a megszerkesztett oldal.

hugo / content / makrdown-roviden.md in main



```
1 ---
2 title: "Markdown röviden"
3 menu: "main"
4 draft: false
5 ---
6
7 # Ez egy cím
8
9 [Link a főoldalra.](/)
10
11 Itt pedig egy-egy *dőlt* és **félkövér** szó.
12
```

15. ábra. content mappa tartalma

Most ne foglalkozzunk semmivel, hanem csak a 7. sorban található # Ez egy cím szöveget írjuk át # Ez egy főcím szövegre. Amint ezt a módosítást megtettük, akkor az oldal alján található **Commit changes** feliratú gomb zöldre vált és megnyomva azt érvényre is jut a módosításunk.

Attach files by dragging & dropping, selecting or pasting them.



Commit changes

Update markdown-roviden.md

Add an optional extended description...

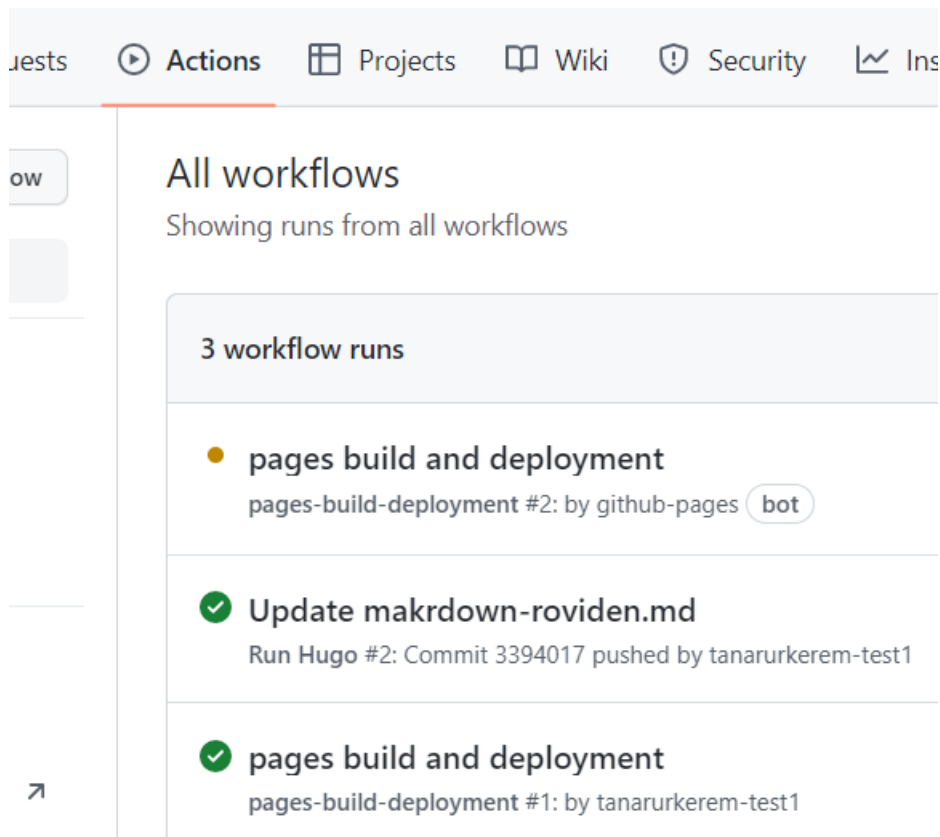
- Commit directly to the `main` branch.
- Create a **new branch** for this commit and start a pull request. [Learn more about pull requests](#)

Commit changes

Cancel

16. ábra. content mappa tartalma

Ne feledd: ezek a változtatások nem jutnak egyből érvényre, meg kell várnod, amíg a robotok befejezik a munkájukat. Az első robot a szövegfájlokból HTML fájlokat gyárt. A második robot pedig az így elkészített HTML állományokat másolja a megfelelő helyre. Az alábbi képen azt láthatod, hogy a szöveg/HTML konverzió már elkészült (*Update markdown-roviden.md* sor a képen), hisz előtte zöld gombócot látsz, a publikálás pedig még folyamatban van (*pages build and deployment* sor a képen), hisz előtte még narancssárga a gombóc.



17. ábra. content mappa tartalma

Amint mindkettő zöldre vált frissíthetsz az oldaladon és a **Markdown röviden** menüpontban meg is nézheted a módosításod eredményét.

Feladat

Most keressük meg a `content/posts` mappában az `elindul-az-oldal.md` bejegyzést és módosítsuk. A mondat végéről lemaradt egy pont. Javítsuk a hibát!

Markdown röviden

Eljött az idő, hogy megismerkedjünk a Markdown formátummal, amit korábban nagyvonalúan csak szövegfájlnak neveztem. A formátumot 2004-ben John Gruber dolgozta ki Aaron Swartz közreműködésével. Az első markdown/HTML fordítót is ő írta Perl nyelven. Aztán a formátum elterjedt és számos kisebb-nagyobb különbséget tartalmazó verziója készült. Ezért felmerült az igény arra, hogy szabványt dolgozzanak ki rá, ez lett a CommonMark.

Markdownnal formázhatod a szöveget többek között a Facebook Messenger, Slack, Microsoft Teams üzenetküldőkben, vagy az olyan platformokon mint a StackOverflow, vagy a Reddit. Inkább az a meglepő, ha egy platform nem ismeri a formátumot.

A formátum tárháza nagyon széles az egyszerű link beillesztéstől kezdve a bonyolultabb táblázatok, kódsorok beillesztéséig. Ráadásul különböző kiegészítőkkal ábrákat, képleteket is előállíthatunk.

Az egyszerű pár soros szöveges üzenetektől kezdve a több ezer oldalas, több szerző közreműködésével készülő kiadványokig bármi elkészíthető vele.

A kimenet ma már nem csak HTML, hanem szinte bármilyen formátum lehet. Például PDF, Epub, Mobi.

Formázás

Nyisd meg szerkesztésre a **markdown-roviden.md** fájlt és teszteld módosításaidat a **Preview** gomb segítségével. Ne feledd, ha mented, akkor 1-2 percet várnod kell, amíg megnézheted az eredményt. A **Preview** gomb segítségével viszont azonnal láthatod azt.

Mint ahogyan a HTML-ben is van egy metainformációt tartalmazó rész (HEAD), úgy a Markdown-ban is. Ezt láthatod a dokumentum elején. Itt csak a **title:** mezőnél található értéket változtasd.

```
---
title: "Markdown röviden"
menu: "main"
draft: false
---
```

Első sorban egy címet formázunk. Mint látható egy # jel van a sor elején, aztán egy szóköz, majd a cím. A jelek száma megegyezik a HTML-ben használt számozással. Egy jel H1 tag lesz, kettő H2 és így tovább.

```
# Ez egy cím
## Ez egy alcím
```

A link beszáúrása egyszerű. Ha egy szöveget linkelni szeretnénk, akkor az szögletes zárójelek ([,]) közé kell tenni, majd a szögletes zárójelek után közvetlenül kerek zárójelek ((),) közé be kell tenni az URL-t.

```
[Link a főoldalra.]()
```

Dólt vagy félkövér szöveget a * karakterrel tudunk formázni. Egy csillag dólt, két csillag félkövér szöveget eredményez. Kivéve, ha a csillag karaktert a sor elejére tesszük és egy szóköz követi, mert akkor felsorolást jelöl.

Itt pedig egy-egy *dólt* és **félkövér** szó.

```
* első elem
* második elem
* haramadik elem
```

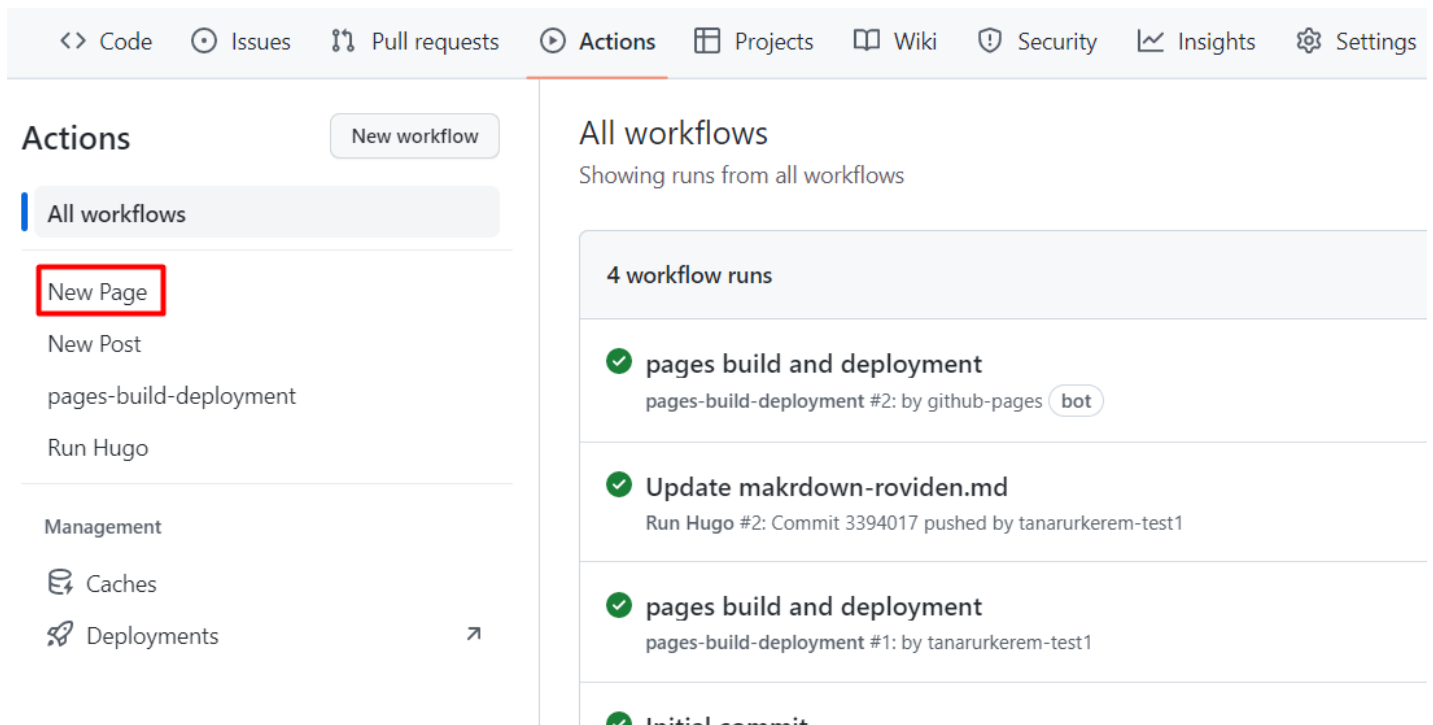
Képet beszúrni pedig úgy tudunk, hogy belinkeljük a képet és egy felkiáltójelet teszünk az egész elejére. A szöveg fog bekerülni a kép alt tulajdonságába. A képet a **static** könyvtárba kell elhelyeznünk.

```
![Ez a Kisbalaton](/kisbalaton.jpg)
```

Új oldal létrehozása

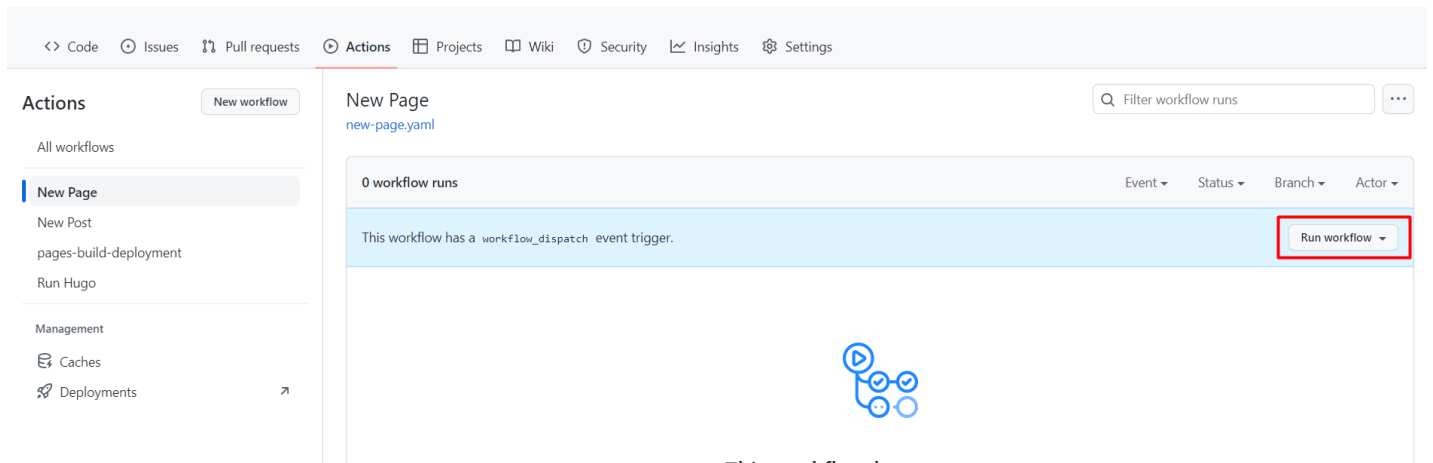
Az új oldal létrehozásához segítségül kell hívnunk a robotokat. Van két mintafeladat. Az egyikkel új oldalt tudsz létrehozni, a másikkal új bejegyzést. Mindkettőnél meg kell majd adni a fájl nevét, aminél arra kel figyelni, hogy a fájl neve szerepelni fog az útvonalban és a végére kell az .md kiterjesztés. A Page(Oldal) típusú tartalmakat a **content** mappában, míg a Post(Bejegyzés) típusú tartalmakat a **content/post** mappába hozza létre a robot. Nézzük akkor, hogy hogyan tudod ezt megtenni.

Első körben navigálj el az Action oldalra.



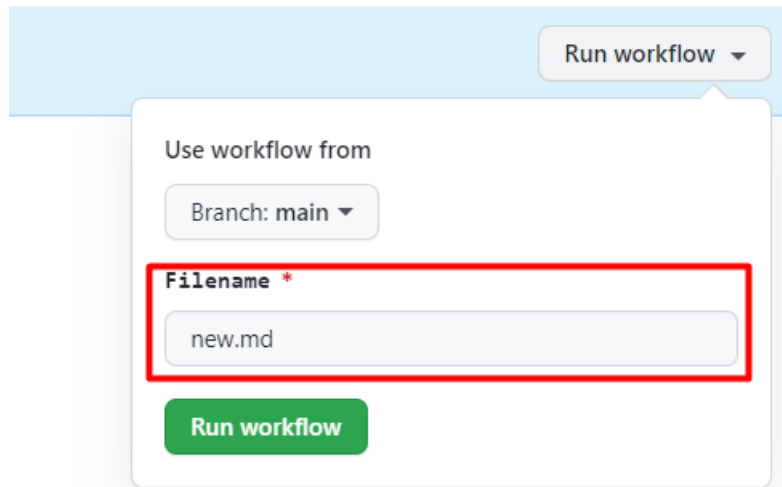
18. ábra. Action oldal és a New Page, New Post mintafeladat

Itt válaszd ki a **New Page** menüpontot.



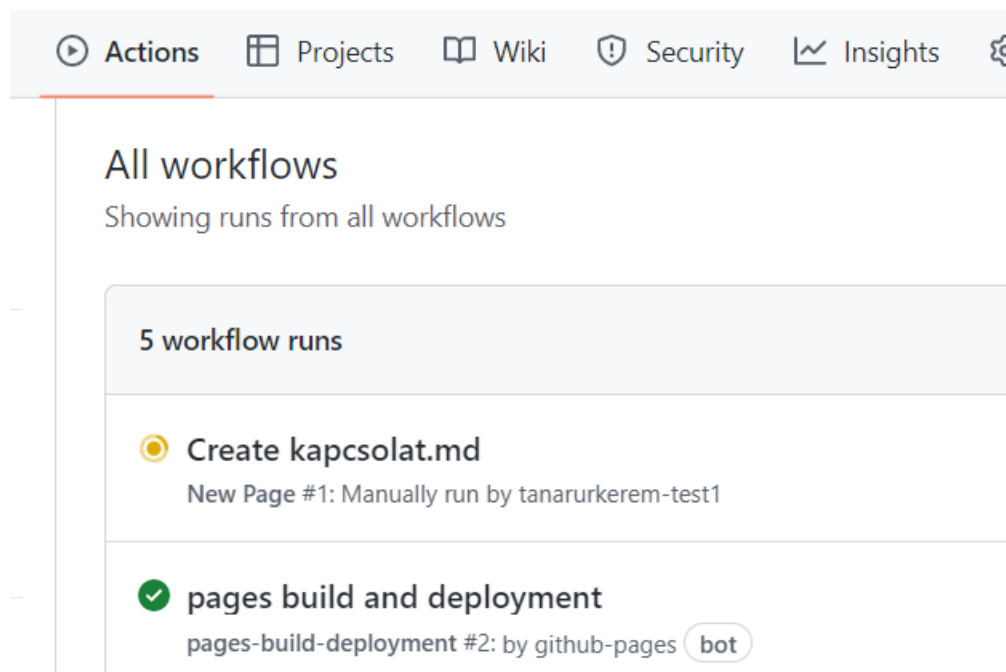
19. ábra. New Page mintafeladat

Ez az az oldal, ahol tudod majd futtatni ezt a mintafeladatot, valamint láthatod majd a korábban végrehajtott hasonló feladatok futásának eredményét. Kattints a **Run workflow** legördülőre és a **Filename** mezőben add meg a fájl nevét, amit létre kívánsz hozni. Ez most legyen **kapcsolat.md**. Majd kattints a Run workflow gombra.



20. ábra. New Page mintafeladat

Ha minden jól ment, akkor elindul a fájl készítése, melynek folyamatát szokás szerint az **Action** fülön követheted nyomon. Amint az ábrán látszik éppen fut a **Create kapcsolat.md** feladat.



21. ábra. New Page mintafeladat

Amint ez zöldre vált navigálhatunk is a **content** mappához és szerkeszthetjük a fájlt. Ne aggódj, amíg nincs tartalom a fájlban, addig nem fog megjelenni az oldalon. Azonban figyelj arra, hogy amint elmentetted egyből online lesz látható, amit beírtál. Ezért javaslom, hogy itt mindig valamilyen kitalált adatot adj meg. Email címnek használd mindig a @example.com domain-t, telefonszámnak meg +36 55 kezdetű bármilyen számot.