Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: (Upisuje redakcija, ostaviti ovaj red)
DOI: (Upisuje redakcija, ostaviti ovaj red)

UPOTREBA GITHAB AKCIJA ZA AUTOMATIZACIJU IZRADE I OCENJIVANJA STUDENTSKIH ZADATAKA

USING GITHUB ACTIONS TO AUTOMATE THE CREATION AND EXAMINATION OF STUDENT ASSIGNMENTS

Jelena Dokić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – Računarstvo i automatika

Kratak sadržaj – *Poslednjih godina, na tržištu se pojavljuju mnogi alati za automatizaciju procesa razvoja softvera. Ovaj rad opisuje jedan od takvih alata - Github akcije, kao i prednosti i mane njegovog korišćenja. Opis je dat kroz primer razvoja dodatnih funkcionalnosti u aplikaciji koja služi za automatizaciju procesa kreiranja i ocenjivanja studentskih zadataka.*

Ključne reči: *Github akcije, automatizacija, izrada i ocenjivanje studentskih zadataka*

Abstract – *In the past few years, a lot od software automation tools have appeared on the market. This paper describes the usage of one of those tools – GitHub Actions, with the advantages and disadvantages of its usage. The descriptionis given through the development of features on the application that is used for the creation and examination of student assignments.*

Keywords: *GitHub actions, automation, creation and examination of student assignments*

1. UVOD

Od samog početka razvoja računarstva, ideja je bila da računari i softverski alati koji se na njima koriste olakšaju ljudske aktivnosti koje zahtevaju mnogo uloženog vremena ili resursa da bi se izvršile jednom ili više puta. Ove aktivnosti su vremenom postajale sve kompleksnije, a alati koji ih rešavaju sve moćniji. Jedan od takvih alata, koji je nastao pre svega nekoliko meseci, su GitHub akcije.

Zadatak ovog rada je da kroz realan primer razvoja softvera otvorenog koda testira primer upotrebe GitHub akcija i sve prednosti i mane koje ovaj alat donosi. Projekat otvorenog koda koji se koristi u ovom primeru je pjisp-assignment-template. U pitanju je softverski alat koji se koristi za automatizovano kreiranja zadataka za studente korišćenjem šablona na predmetu Programski jezici i strukture podataka na Fakultetu tehničkih nauka u Novom Sadu.

Proces kreiranja zadatka upotrebom pjisp-assignment-template alata sastoji se iz nekoliko koraka, te postoji mogućnost da se napravi greška. Pomoću GitHub akcija, u malopre pomenuti alat se integrišu drugi alati, koji služe za vršenje provera ispravnosti nekih od koraka i koji dodatno olakšavaju proces kreiranja zadatka. Ovi alati predstavljaju programe koji se mogu instalirati i nezavisno od pjisp-assignment-template alata, a i sami za pripremanje izvršne verzije koriste GitHub akcije.

Ovaj rad opisuje sve prethodno pomenute alate i način njihove integracije u celinu upotrebom GitHub akcija. Na kraju samog rada, izvedeni su zaključci o GitHub akcijama stečeni na osnovu njihovog korišćenja u procesu izrade ovog rada.

2. ALATI KOJI SE KORISTE U PROCESU AUTOMATICIJE IZRADE I OCENJIVANJA STUDENTSKIH ZADATAKA

Projekat o kojem ovaj rad govori sastoji se iz grupe alata koje je bilo potrebno napisati i povezati da bi se dodale nove funkcionalnosti u postojeći projekat - pjisp-assignment-template [1]. Pomenuti alat služi za automatizovano kreiranje studentskih zadataka, nihovo puštanje u učionicama u kojima studenti rade zadatke i, na kraju, pregledanje zadataka. Prvenstveno je namenjen da nastavnom osoblju olakša ceo proces od kreiranja do pregledanja kolokvijuma ili zadataka koje studenti rade.

Zadatak nastavnog osoblja je pre svega da iz repozitorijuma projekta koji je šablonski repozitorijum kreira svoj repozitorijum sa zadatkom. Važno je da svoj repozitorijum nazove u skladu sa predefinisanim načinom imenovanja. Nakon toga se u repozitorijumu kreiraju svi potrebni materijali koji su potrebni za generisanje studentskog zadatka. Zadatak nastavnog osoblja je da pokrene generisanje zadatka za jedan od predefinisanih testova na predmetu Programski jezici i strukture podataka. Jedan od alata o kojem ovaj rad govori - pjisp-template-name dodaje mogućnost automatizovanog kreiranja šablona za zadatak za određeni test na osnovu naziva repozitorijuma koji autor zadatka navede, uz proveru ispravnosti naziva.

Po završetku pisanja zadatka i njegovog testiranja, nastavnik ili saradnik je dužan da obavesti drugog nastavnika o tome, da bi on mogao da proveri ispravnost zadatka i odobri korišćenje zadatka u nastavi, naravno ako je dobar i pogodan za studente. Kao olakšicu, autor zadatka može da pokrene alat za testiranje zadatka i proveri ispravnost. Projekat o kojem ovaj rad govori

---

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Petar Marić, docent

uvodi isto to testiranje i nakon objavljivanja nove verzije zadatka na GitHub-u. Kao dodatnu proveru,jedan od alata iz ovog projekta - pjisp-diff proverava da li su izmenjene sve potrebne datoteke, da bi zadatak bio kompletan, odnosno da bi imao korektan tekst zadatka, primer rešenja i testove. Ako svi ovi alati vrate rezultat pozitivnog ishoda, u GitHub repozitorijumu se pojavljuje oznaka da su svi uslovi ispunjeni i da je zadatak spreman za pregledanje. U suprotnom, stoji oznaka da zadatak ne ispunjava sve potrebne uslove.

Naredna poglavlja opisuju alate koji su korišćeni da bi se postigla opisana unapređenja pjisp-assignment-template alata, a poslednje poglavlje opisuje njihovu integraciju.

2.1. pjisp-template-name

Alat pjisp-template-name [2] je aplikacija koja služi za proveru ispravnosti naziva repozitorijuma. Napisana je u obliku konzolne aplikacije otvorenog koda u programskom jeziku Python.

Za imenovanje repozitorijuma koristi se predefinisani šablon, koji izgleda ovako:

```
pjisp-{SCHOOL_YEAR}-{COURSE_ID}-{TEST_ID}-
{GROUP_ID}
```

Alat pre svega proverava da li je dužina naziva ispravna, odnosno da li naziv ima sve celine odvojene znakom “-”. Ako nema, ispisuje se poruka “Repository name length not valid” i program izlazi sa izlaznim kodom 1. U suprotnom, program nastavlja sa izvršavanjem. Nakon provere dužine, proverava se ispravnost ostalih tokena, prema malopre opisanom načinu davanja imena. Ako bilo gde dođe do greške, ispisuje se poruka“Repository name not valid. Error on <token>” i izlazi se sa izlaznim kodom 1. U ovom slučaju, <token> predstavlja jednu od celina odvojenih znakom“-”. Ako naziv u potpunosti ispunjava šablon, kao izlazna vrednost vraća se identifikator testa, da bi na osnovu toga mogle da se generišu datoteke potrebne za kreiranje tog testa.

2.2. pjisp-diff

Alat pjisp-diff [3] je konzolna aplikacija u programskom jeziku Python, koja služi kao pomoćni alat nastavnom osoblju koji proverava da li su izmenili sve potrebne datoteke pri kreiranju zadataka za studente.

Pored provere da li su promenjene sve potrebne datoteke, ovaj alat proverava i da li su ostale nepromenjene sve pomoćne datoteke koje nije dozvoljeno menjati. Datoteke koje je potrebno menjati zavise od ulaznog parametra, šablona na osnovu kojeg se kreira zadatak.

Datoteke koje je potrebno uvek menjati su assignment_solution.c i assignment.rst.

Ako je vrednost ulaznog parametra jednaka “T12”, potrebno je menjati fixtures/stdio-numbers.yaml datoteku.

Ako je vrednost ulaznog parametra jednaka “T34” ili “SOV”, potrebno je menjati file-error-input-not-readable.yaml, file-error-output-not-readable.yaml i file-text.yaml datoteke u fixtures direktorijumu.

Datoteka koju nije dozvoljeno menjati je assignment_notes.rst.

Ako su po završetku programa izmenjene sve potrebne datoteke i nisu izmenjene one koje ne smeju da se menjaju, alat vraća 0 kao vrednost izlaznog koda. U suprotnom, alat vraća vrednost 1 uz jednu ili više poruka u obliku: “Please change the <filename> file.” ili “Please do not change the <filename> file.”. U ovom slučaju, <filename> predstavlja naziv datoteke koja nije izmenjena, a treba da bude ili je izmenjena, a ne treba da bude menjana.

2.3. poetry-publish

Alati koji su opisani u prethodna 2 poglavlja napravljeni su tako da mogu biti instalirani i korišćeni kao Python paketi. Oba paketa su dostupna na PyPI repozitorijumu Python paketa. Za postavljanje na repozitorijum, korišćen je alat Poetry. Pošto je potrebno objaviti novu verziju paketa na repozitorijumu nakon svake značajne izmene u samom paketu, korišćena je poetry-publish [4] GitHub akcija koja pri svakom novom objavljivanju verzije alata na GitHub-u, kreira i novu verziju Python paketa i postavlja je na PyPI repozitorijum.

Akcija poetry-publish je kreirana od strane autora ovog rada pre nego što se razvila ideja za projektom koji je u ovom radu opisan, ali je znatno olakšala kreiranje malopre pomenutih paketa, tako da je imala veoma važnu ulogu i u ovom projektu. Akcija je projekat otvorenog koda i dostupna je za preuzimanje u GitHub prodavnici [5]. Dostupna je pod BSD 3-Clause licencom.

Poetry-publish je GitHub akcija koja služi za kreiranje paketa i njihovo objavljivanje na Python repozitorijumu upotrebom alata Poetry. Kreirana je u obliku Docker akcije. U Docker kontejneru se instaliraju svi potrebni alati za korišćenje Python-a i zatim se pokrenu Poetry komande za kreiranje i objavljivanje paketa:

`poetry build` - da kreira Python paket

`poetry publish` - da objavi paket na repozitorijumu koji je prethodno podešen

Paramtri koje ova akcija očekuje su:

- python_version - verzija Python-a koja se instalira u kontejneru i koristi za kreiranje paketa. Ako se ne navede, podrazumeva se da se koristi najnovija verzija. Za bolje performanse, poželjno je koristiti predefinisanu - najnoviju verziju.

- poetry_version - verzija Poetry-a koja se instalira u kontejneru i koristi za kreiranje paketa. Ako se ne navede, podrazumeva se da se koristi najnovija verzija. Navodi se u PIP sintaksi za specifikaciju verzija

- pypi_token - jedini obavezan parametar. Služi kao API token za autentifikaciju pri objavljivanju paketa na PyPI.

- repository_name - naziv repozitorijuma na kojem se objavljuje paket. Ako se ne navede, podrazumeva se da se objavljuje na PyPI repozitorijumu.

- repository_url - adresa repozitorijuma na kojem se objavljuje paket. Ako se ne navede, podrazumeva se da se objavljuje na PyPI repozitorijumu.

Takođe, potrebno je specificirati pyproject.toml datoteku na osnovu koje će se kreirati Python paket. Ova datoteka treba da se nalazi u korenskom direktorijumu repozitorijuma koji koristi ovu akciju.

U pjisp-template-name i pjisp-diff, ova akcija se izvršava samo kada se napravi novi privezak u repozitorijumu u obliku v*.*.*.Tok poslova se sastoji od jednog posla koji se pokreće na poslednjoj verziji Ubuntu operativnog sistema i 3 koraka. Prvi korak preuzima sadržaj repozitorijuma. Drugi korak menja verziju u pyproject.toml datoteci na verziju priveska koji je prouzrokovao pokretanje ove akcije. Poslednji korak koristi poetry-publish akciju za objavljivanje nove verzije paketa. Svi ulazni parametri koriste predefinisanu verziju, samo je pypi_token parametar postavljen na vrednost PyPI tokena koji se preuzima iz GitHub-ovog konteksta okruženja - secrets.

2.4. smoke-test

Još jedan alat otvorenog koda koji je korišćen i izmenjen u ovom radu je smoke_test [6]. To je konzolna aplikacija i API koji služi za testiranje studentskih zadataka na dim.

Ovaj alat se koristi i za testiranje zadataka koje nastavno osoblje piše kao primer tačno urađenog zadatka i samim tim se koristi u okviru pjisp-assignment-template alata. Zadatak koji ovaj alat očekuje treba da bude napisan u programskom jeziku C ili nekom jeziku od kojeg je GCC progrmaski prevodilac sposoban da napravi izvršnu datoteku. Zadatak se prvobitno kompajlira pomoću GCC-a. Nakon toga zadatak se testira na slučajeve korišćenja opisane u .yaml datotekama koje nastavno osoblje priprema prilikom kreiranja zadatka. Rezultat je u ljudski čitljivom formatu, ali je izlazni kod u svakom slučaju, bilo pri pozitivnoj ili negativnoj evaluaciji studentskog rešenja uvek bio 0. Da bi smoke_test alat mogao da se upotrebi u okviru GitHub akcije koja proverava ispravnost rešenja koje nastavno osoblje kreira, i da bi se izvršavanje akcije prekinulo u slučaju negativne evaluacije studentskog rešenja, izmenjen je izlaznikod na vrednost 1 ako ponuđeno rešenje ne ispunjava sve uslove navedene u .yaml datotekama i ako je prosleđen argument -e kao argument komandne linije. Kada GitHub akcija u bilo kojem koraku dobije rezultat različit od 0, smatra se da taj korak nije uspešno izvršen i ne prelazi se na naredne korake. U slučaju pjisp-assignmen-template alata, to znači da autor zadatka nije dobro napisao primer ispravnog rešenja i da mora da ga ispravi.

2.5. pjisp-assignment-template

Alat pjisp-assignment-template nudi veliki broj funkcionalnosti za kreiranje i proveru ispravnosti kreiranog zadatka za studente. Ove funkcionalnosti su u velikoj meri učestvovale u dodavanju GitHub tokova poslova čije funkcionalnosti su opisane na samom početku poglavlja. Ipak, neke od funkcionalnosti je trebalo malo izmeniti i bilo je potrebno kreirati nekoliko novih.

Za spisak postojećih funkcionalnosti potrebno je pročitati dokumentaciju.

Nove funkcionalnosti koje uvodi projekat o kojem ovaj rad govori su:

- assignment-diff - za proveru da li su izmenjene sve datoteke koje je potrebno menjati i da lisu ostale iste sve datoteke koje se ne smeju menjati

- assignment-check - za proveru izmena datoteka i proveru ispravnosti primera rešenja

- get-template - za pronalaženje naziva testa na osnovu naziva repozitorijuma i proveru ispravnosti naziva repozitorijuma

Takođe, uvedene su sitne izmene u neke od postojećih funkcionalnosti:

- init - dodatno, nakon generisanja datoteka na osnovu izabranog identifikatora testa, kreira se .template datoteka u koju se smešta identifikator za kasniju upotrebu

- test-solution - dodato je da pre pokretanja funkcionalnosti zahteva da postoji assignment_solution.c datoteka

- assignment-build - dodato je da pre pokretanja funkcionalnosti zahteva da postoji assignment_solution.c datoteka

- assignment-pack - dodato je da se pre pakovanja, pored provere ispravnosti primera rešenja, proveri da li su izmenjene sve datoteke koje je potrebno menjati i da li su ostale iste sve datoteke koje se ne smeju menjati

Kao zavisnosti u projektu dodati su pjisp-template-name i pjisp-diff korišćenjem Pipenv alata. Get-template funkcionalnost poziva pjisp-template-name sa nazivom repozitorijuma kao ulaznim parametrom i identifikatorom testa kao povratnom vrednošću. Assignment-diff poziva pjisp-diff sa identifikatorom testa kao ulaznim parametrom.

Nakon dodavanja malopre pomenutih izmena u kod pjisp-assignment-template alata, uvedene su sve funkcionalnosti potrebne za kreiranje GitHub tokova poslova čije funkcionalnosti su opisane na početku trenutnog poglavlja. Kreirane su dve nove datoteke na mestu koje je predviđeno za pisanje GitHub tokova poslova - github/workflows. Tokovi poslova nazvani su “Project create” i “PJISP assignment” i nalaze se u init-repo.yml i test-solution.yml datotekama respektivno.

2.5.1 Project create

“Project create” je tok poslova koji se pokreće, kako mu i samo ime kaže, izazvan događajem kreiranja projekta, odnosno nakon pravljenja repozitorijuma na osnovu šablon repozitorijuma pjisp-assignment-template. Uloga ovog toka posla je da postavi inicijalno stanje projekta. Na osnovu naziva repozitorijuma, zaključuje se na koji test se repozitorijum odnosi i onda se na osnovu identifikatora testa kreiraju datoteke namenjene tom testu. Identifikator testa može biti T12, T34 ili SOV. U README.rst datoteku, koja sadrži sve bitne informacije o projektu i datotekama u njemu, postavlja bedž koji daje informacije o tome da li je zadatak kreiran kako treba, a menja se na osnovu rezultata drugog toka poslova - “PJISP assignment”.

Bedž može biti u jednom od tri stanja: “no status” kada nema status, “failing” kada se “PJISP assignment” neuspešno izvrši i “passing” kada se uspešno izvrši.

“Project create” tok posla u sebi ima jedan posao koji se sastoji od šest koraka. Posao se pokreće na Ubuntu

operativnom sistemu na najnovijoj verziji. Prvi korak preuzima sadržaj repozitorijuma. Drugi korak instalira Python u verziji 2.7 u okruženju pokretača. Treći korak je zadužen za instaliranje Pipenv-a, alata za upravljanje Python paketima koji pjisp-assignment-template koristi. Četvrti korak je zadužen za inicijalizaciju datoteka na osnovu naziva repozitorijuma. Naziv repozitorijuma dobija se iz promenjive github.repository koja se preuzima iz konteksta okruženja. Nakon toga se instaliraju zavisnosti korišćenjem Pipenv alata i pokreće se get-template funkcionalnost. Ova funkcionalnost proverava ispravnost naziva repozitorijuma i na osnovu njega pronalazi identifikator testa. Ako je naziv repozitorijuma ispravan, poziva se init funkcionalnost sa identifikatorom testa kao ulaznim parametrom.Peti korak služi za kreiranje bedža i njegovo dodavanje u README.rst datoteku. Poslednji, šesti korak je zadužen da postavi sve novonastale izmene na GitHub repozitorijum. Nakon postavljanja ovih izmena, sve nove datoteke nalaze se na GitHub repozitorijumu i bedž je vidljiv na vrhu README.rst datoteke. Bedž u početku ima failing vrednost, pošto zadatak nije spreman za studentei očekuje se od nastavnog osoblja da ga kreira.

2.5.2 PJISP assignment

"PJISP assignment" tok posla se pokreće na događaj push ili pull-request na glavnoj - master grani. Zadužen je da svaki put nakon izmene koda na repozitorijumu proveri da li je repozitorijum spreman za pregledanje od strane nastavnika i davanje studentima.

3. ZAKLJUČAK

Iz prethodnog poglavlja vidimo da se mnoge nove funkcionalnosti oslanjaju na funkcionalnosti koje GitHub Actions servis nudi. Ovaj servis je dosta mlad, nastao je u novembru 2019. godine i to sa sobom donosi neke probleme. Još uvek nije u potpunosti pouzdan, pošto su neke od njegovih funkcionalnosti povremeno nedostupne.

Takođe, zbog brzog razvoja softvera, dokumentacija nije u svakom momentu usklađena sa alatom, pa to nekada može predstavljati problem za programere koji žele da koriste nove funkcionalnosti alata čim se one pojave. I pored pomenutih problema, za potrebe ovakvog projekta, GitHub akcije donese mnoge beneficije i znatno olakšavaju razvoj i korišćenje svih prethodno pomenutih alata. Integracija alata je bila veoma jednostavna, pošto se šablon repozitorijum već nalazi na GitHub-u i onda nije bilo potrebno uvoditi dodatne alate u proces integracije.

Funkcionalnosti koje ove akcije uvode mogu znatno da smanje učestalost grešaka koje nastavno osoblje pravi prilikom kreiranja zadataka za studente. Olakšavaju i procenu spremnosti zadataka za davanje studentima, pošto bedž na početku README.rst datoteke u repozitorijumu zadatka već naznačava da je zadatak prošao sve testove koje alat trenutno pokreće nad njim i da je spreman za pregledanje od strane drugog nastavnika.

Ovaj tok posla u sebi ima jedan posao koji se sastoji od četiri koraka. Posao se pokreće na Ubuntu operativnom sistemu na najnovijoj verziji. Prvi korak preuzima sadržaj repozitorijuma. Drugi korak instalira Python u verziji 2.7 u okruženju pokretača. Treći korak je zadužen za instaliranje Pipenv-a. Svi koraci osim prvog imaju uslov pod kojim se izvršavaju, a to je da u svom nazivu ne sadrže "pjisp-assignment-template". Ovaj uslov osigurava se na pjisp-assignment-template repozitorijumu neće pokretati ovi koraci, pošto je on samo šablonski repozitorijum i nikada neće sadržati konkretan zadatak za studente, pa samim tim nema smisla testirati njegovu ispravnost. Poslednji, četvrti korak je zadužen za pokretanje testiranja zadatka. Prvo se instaliraju zavisnosti projekta korišćenjem Pipenv alata, a onda se pokreće assignment-check funkcionalnost. Ova funkcionalnost prvo pokreće assignment-diff. koji proverava da li su izmenjene sve datoteke koje je potrebno menjati i da li su ostale nepromenjene sve datotekekoje ne bi trebalo menjati. Ako se uspešno izvrši, prelazi se na test-solution funkcionalnost. Ako se ne izvrši uspešno, onda se izvršavanje koraka, a zatim i celog toka posla prekida i kreiranje zadatka se smatra neuspešnim. Test-solution funkcionalnost ispituje ispravnost primera rešenja na osnovu testova opisanih u .yaml datotekama. Potrebno je da se svi testovi uspešno izvrše da bi se zadatak smatrao ispravnim i onda se bedž postavlja na "passing" vrednost.

4. LITERATURA

[1] Petar Marić, "pjisp-assignment-template", https://github.com/petarmaric/pjisp-assignment-template (pristupljeno u avgustu 2020.)

[2] Jelena Dokić, "pjisp-template-name", https://github.com/JRubics/pjisp-template-name (pristupljeno u avgustu 2020.)

[3] Jelena Dokić, "pjisp-diff", https://github.com/JRubics/pjisp-diff (pristupljeno u avgustu 2020.)

[4] Jelena Dokić, "poetry-publish", https://github.com/JRubics/poetry-publish (pristupljeno u avgustu 2020.)

[5] Jelena Dokić, "publish-python-poetry-package", https://github.com/marketplace/actions/publish-python-poetry-package (pristupljeno u avgustu 2020.)

[6] Petar Marić, "smoke-test", https://github.com/petarmaric/smoke_test (pristupljeno u avgustu 2020.)

Kratka biografija:

Jelena Dokić rođena je u Novom Sadu 1996. god. Gimnaziju je završila 2015. godine, i iste godine upisala OAS na Fakultetu tehničkih nauka, smer Računarstvo i automatika. Diplomirala je 2019. god i upisala MAS na istom smeru. Master rad na Fakultetu tehničkih nauka iz oblasti Računarstva i automatike odbranila je2020.god. kontakt: jelena.dokic@uns.ac.rs