
	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	1
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015

Metodika č. 2015 – 3

Verziovane zdrojového kódu

	VYPRACOVAL
FUNKCIA	Manažér verzí
MENO, TITUL	Helmut Posch, Bc.
DÁTUM	6/12/2015
PODPIS	

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	2
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015


1. Účel a dedikácia metodiky

Účelom verziovania kódu je jednoduchšia kolaborácia pri jeho tvorbe. Každý programátor vytvára svoj vlastný kód (verziu) a všetky sa neskôr spoja do jedného funkčného celku. Taktiež je jednoduchý návrat k rôznym verziám čo zjednodušuje hľadanie chýb.

Slovník pojmov

Bitbucket - úložisko pre repozitáre spravované verziovacími systémami napr. Git


Git - verziovací systém zdrojového kódu

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	3
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015

2. Inicializácia lokálneho úložiska

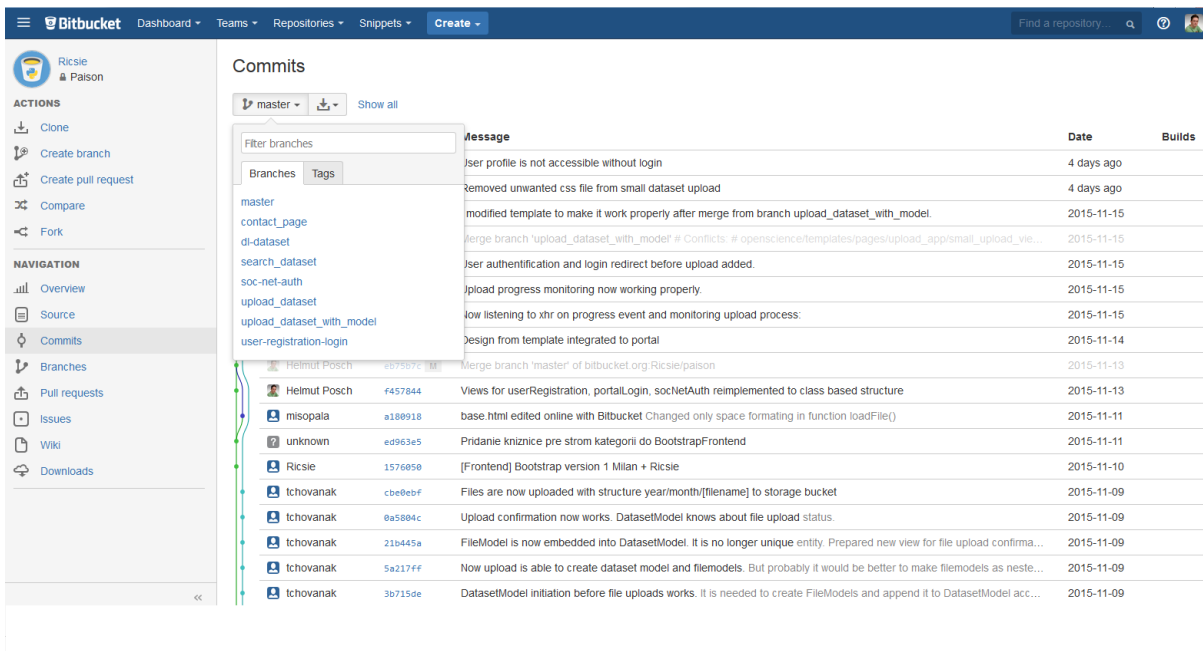
V projekte používame verziovane kódu pomocou úložiska Bitbucket (*bitbucket.org*) a verziovací systém Git. K nasledujúcim častiam sa predpokladá, že programátor má pridelený prístup k projektu v aplikácii Bitbucket-u a nainštalovaný systém Git.

Aby bolo možné pridávať nový kód do projektu, je nutné si vytvoriť lokálne úložisko s projektom. Programátor si vytvorí vlastný priečinok pre projekt kdekoľvek vo svojom počítači. Po nastavení sa do vytvoreného priečinku spustí príkaz `git init`. Po tomto kroku si naklonuje projekt k sebe príkazom `git clone git@bitbucket.org:Ricsie/paision.git`. V tomto stave je lokálne úložisko úspešne inicializované.

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	4
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015


3. Prehliadky verzií a zmien kódu projektu

Verzie vo vzdialenom repozitári je možné si prezerať pomocou aplikácie *bitbucket.org*. Je možné vybrať si jednotlivé vetvy. Verzie je možné pozerať ako zoznam zmien alebo ako celý zdrojový kód. Vetvy sú tiež zobrazené v samostatnej časti, podľa toho či už sú spojené s hlavnou vetvou alebo nie.



Message	Date	Builds
User profile is not accessible without login	4 days ago	
Removed unwanted css file from small dataset upload	4 days ago	
modified template to make it work properly after merge from branch upload_dataset_with_model.	2015-11-15	
merge branch 'upload_dataset_with_model' # Conflicts: # openscience/templates/pages/upload_apps/small_upload_vie...	2015-11-15	
User authentication and login redirect before upload added.	2015-11-15	
upload progress monitoring now working properly.	2015-11-15	
now listening to xhr on progress event and monitoring upload process:	2015-11-15	
design from template integrated to portal	2015-11-14	
Merge branch 'master' of bitbucket.org:Ricsie/paision	2015-11-13	
Views for userRegistration, portalLogin, socNetAuth reimplemented to class based structure	2015-11-13	
base.html edited online with Bitbucket Changed only space formatting in function loadFile()	2015-11-11	
Pridanie kniznice pre strom kategorii do BootstrapFrontend	2015-11-11	
[Frontend] Bootstrap version 1 Milan + Ricsie	2015-11-10	
Files are now uploaded with structure year/month/[filename] to storage bucket	2015-11-09	
Upload confirmation now works. DatasetModel knows about file upload status.	2015-11-09	
FileModel is now embedded into DatasetModel. It is no longer unique entity. Prepared new view for file upload confirma...	2015-11-09	
Now upload is able to create dataset model and filemodels. But probably it would be better to make filemodels as neste...	2015-11-09	
DatasetModel initiation before file uploads works. It is needed to create FileModels and append it to DatasetModel acc...	2015-11-09	

Zmeny v lokálnom úložisku je možné sledovať pomocou príkazu *gitk --all*, čo predstavuje grafické rozhranie s mapou verzií projektu. Pre zobrazenie lokálnych zmien v textovej forme sa používa príkaz *git status*.

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	5
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015

File Edit View Help

- Added notification about not activated account during login by email
- User profile is not accessible without login
- Removed unwanted css file from small dataset upload
- I modified template to make it work properly after merge from branch upload_dataset_with_model.
- Merge branch 'upload_dataset_with_model'
- User authentication and login redirect before upload added.
- Upload progress monitoring now working properly.
- Now listening to xhr on progress event and monitoring upload process:
- Files are now uploaded with structure year/month/[filename] to storage bucket
- Upload confirmation now works. DatasetModel knows about file upload
- FileModel is now embedded into DatasetModel. It is no longer unique
- Now upload is able to create dataset model and filemodels.

SHA1 ID: 80f734cbae537c5e5fd1a17eb89e8918a695bda6

Find commit containing:

Search

Diff Old version New version Lines of context: 3 Ignore space change Line diff


Author: HelmutP <helmutposch4@gmail.com> 2015-11-18 14:13:01
 Committer: HelmutP <helmutposch4@gmail.com> 2015-11-18 14:13:01
 Parent: 605ed436f6b0e3f93ba4afbaa06ab198fd5f399 (User profile is not accessible without login)
 Branch: master
 Follows:
 Precedes:

```

Added notification about not activated account during login by email


----- openscience/portalLogin/views.py -----
index f7aa94d..4db623f 100644
@@ -52,6 +52,9 @@ class LoginView(TemplateView):
        paionuser.reset_pw_attempt_counter()
        paionuser.save()
        return redirect('/account/profile')
+
+         else:
+             info_message = "Your account is not activate
+             return render(request, self.template_name, {
+
+         else:
+             paionuser = PaionUser.objects.get(user_pk=user.id
+             if paionuser.pw_attempt_counter == 3:
  
```

Comments
 openscience/portalLogin/views.py

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	6
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015

4. Stiahnutie aktuálnej verzie projektu

Prvým krokom k možnosti zmeniť kód projektu je stiahnutie si jeho aktuálnej verzie. Každý programátor je povinný spraviť tento krok. Vykona tak príkazom *git pull*, čím stiahne všetky vetvy a verzie projektu zo vzdialeného repozitára do lokálneho úložiska vo svojom počítači.

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	7
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015

5. Pridávanie novej verzie

Postup pridávania nového kódu sa delí na dva spôsoby.

Prvým prípadom je, ak po pridaní nového kódu do projektu ostane projekt alebo vyvíjaná časť nefunkčná. Nefunkčnosťou chápeme zobrazovanie chybových hlášok alebo funkčnosť, ktorú nie je možné považovať za prezentácie schopný prototyp zákazníkovi / používateľovi. V tomto prípade je programátor povinný si vytvoriť novú vetvu (branch)(pozri nižšie.)

Druhým prípadom je ak programátor vkladá do projektu funkcionality, ktorá nebude mať zásadný vplyv na funkčnosť projektu. Ide predovšetkým o zmeny textového obsahu, jednoduchá zmena designu, jednoduché opravy chýb. V tomto prípade je možné vkladať kód (pozri nižšie) do hlavnej vetvy (master).

5.1. Pridanie novej verzie (commit-u)


Pred samotným pridaním prebieha výber súborov, ktoré do novej verzie patria. Po spustení príkazu *git status* sa červenou farbou zobrazia súbory, ktorých zmeny nie sú zahrnuté do novej verzie a zelenou farbou súbory, ktoré sú zahrnuté do novej verzie. Súbor do verzie pridáme príkazom *git add <nazov subora>* alebo *git rm <nazov subora>* podľa toho, či bol v novej verzii súbor pridávaný resp. modifikovaný alebo zmazaný. Potvrdenie novej verzie vykonáme príkazom *git commit -m <sprava>*. Správa sa píše v anglickom jazyku a má nasledovnú štruktúru:

[<typ zmeny>] <popis zmien>

kde typ zmeny je jedna z možností v nasledujúcej tabuľke.


Typ zmeny	Kedy použiť
FIX	Jednoduchá oprava chyby (zmena malého počtu riadkov, bez ovplyvnenia iných častí projektu)
FEATURE	Vytvorenie novej funkcionality, ktorá rozširuje tú v predchádzajúcej verzii
ISSUE	Zásadnejšia oprava alebo reimplementácia existujúcej funkcionality
MERGE	Zmeny k spojeniu viacerých verzí

Každá verzia by mala byť atomická a nie je žiaduce viac funkcionalít vkladať do jednej novej verzie. Predpokladá sa, že verzia sa pridáva práve do tej vetvy, v ktorej je práve programátor nastavený.

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	8
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015


5.2. Vytvorenie novej vetvy (branch-u)

Rozhodnutie, či vytváraný kód potrebuje novú vetvu by malo prebehnúť ešte pred písaním kódu. Ak chceme vytvoriť novú vetvu, nastavíme sa do verzie projektu, z ktorej chceme vetvu vytvoriť. Do vybranej verzie projektu sa nastavíme príkazom `git checkout <hash verzie>`. Novú verziu následovne vytvoríme príkazom `git checkout -b <nazov novej vetvy>`. Názov vetvy je v anglickom jazyku, slová sú oddelené pomlčkami a obsahuje najviac 5 slov. Tieto slová majú popisovať zmysel vetvy.

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	9
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015

6. Spojenie novej vetvy s hlavnou vetvou

Túto činnosť vykonáva programátor novej vetvy spolu s ostatnými programátormi, s ktorých veziami nastáva konflikt. Programátor novej verzie sa nastaví do hlavnej vetvy *git checkout master*. A vykoná príkaz *git merge <nazov novej vetvy>*. Pokiaľ nenastane konflikt, Git vykoná spojenie sám a taktiež programátor môže vykonať celý proces sám. V prípade konfliktu je potrebné kontaktovať autora kódu, v ktorom je konflikt a dohodnúť sa na spoločnom riešení. Do hlavnej vetvy sa spájajú iba vetvy, ktoré sú plne funkčné.

	METODIKA	Verzia :	2
	Verziovane zdrojového kódu	Strana :	10
		Číslo :	M2015-3
		Dátum vydania:	06/12/2015

7. Odovzdanie novej verzie do vzdialeného úložiska

Po vykonaní zmien a potvrdení novej verzie je možné odoslať lokálny stav repozitára do vzdialeného úložiska. Jednotlivé vetvy je možné odosielať príkazom `git push origin <nazov vetvy>`, prípadne odoslanie všetkých vetiev a zmien príkazom `git push`.