

GIT PRADMENYS

Antanas Bastys

TURINYS

- Java Technologijų kurso apžvalga
- Versijų kontrolės istorija
- Git įvadas
- Resursai/Namų darbai

JAVA TECHNOLOGIJŲ KURSO APŽVALGA

- Gaminsime supaprastintą el. parduotuvės variantą
- Vartotojo sąsaja - Javascript + ReactJs + Bootstrap CSS
- Serveris - REST Api (Spring Boot + Java 8)
- Duomenų bazė - H2 reliacinė duomenų bazė + JPA (Java Persistence Api)

EL. PARDUOTUVĖS FUNKCIJOS

- Peržiūrėti prekių pasiūlą
- Peržiūrėti prekės detales
- Įsidėti prekę į krepšelį
- Peržiūrėti krepšelio turinį
- Pašalinti iš krepšelio
- Pirkti prekes krepšelyje
- Administruoti prekes: pridėti, išmesti, atnaujinti

VERSIJŲ KONTROLĖ

- Angliškai - version control, revision control, source control
- Skirta suvaldyti pakeitimus atliekamus kelių vartotojų grupei dokumentų
- Įsiviazduokite, jog visi turite įrašyti savo vardą į tą patį tekstinį failą

VERSIJŲ KONTROLĖS KARTOS

Karta	Tinklas	Pvz
Pirma	Jokio	RCS, SCCS
Antra	Centralizuotas	CVS, Subversion, Team Foundation Server
Trečia	Išskirstytas	Bazaar, Git, Mercurial

```
$ git help <command>
```

Komandos ir jos parametrų aprašymas

```
$ git init - paruošia dabartinį katalogą kaip git  
reositorija
```



```
$ git add <path>
```

Populiariu pridėti viską iš dabartinio katalogo: `$ git add`

▪

```
$ git status - dabartinē repositrijos būsena.
```

```
→ testproject git:(master) X git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md
        renamed:    oldname.txt -> newname.txt

Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md
        deleted:    test.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        newfile.txt

→ testproject git:(master) X
```

Sucommitinti bus žalių failų pakeitimai

```
$ git commit -m <message>
```

Komanda atliekama pabaigus darbą ar jo etapą su failais.
Susikuria changeset'as turintis informaciją apie
pasikeitusius failus.

\$ git log - commit'ų istorija dabartinėje šakoje

```
commit c0326d2386dd1227f35f46f1c75a8f87e2e93076
Author: Hudson @ build.clojure.org <build@clojure.com>
Date:   Fri Oct 28 14:24:47 2016 -0500

    [maven-release-plugin] prepare for next development iteration

commit e3c4d2e8c7538cfda40accd5c410a584495cb357
Author: Hudson @ build.clojure.org <build@clojure.com>
Date:   Fri Oct 28 14:24:47 2016 -0500

    [maven-release-plugin] prepare release clojure-1.9.0-alpha14

commit b80e1fe4b14654d943e2f8b060b0bc56e18b4757
Author: Nicola Mometto <brobronsa@gmail.com>
Date:   Fri Oct 7 21:23:39 2016 +0100

    CLJ-1242: equals doesn't throw on sorted collections

    Signed-off-by: Stuart Halloway <stu@cognitect.com>

commit 2ff700ede3866f97d7b1f53342e201df94384aee
Author: Nicola Mometto <brobronsa@gmail.com>
Date:   Sat Nov 7 02:58:40 2015 +0100

    CLJ-1790: emit a cast to the interface during procol callsite

    Signed-off-by: Stuart Halloway <stu@cognitect.com>

commit c6b76fadb4750c8f73d842cfdcf882b4a05683cae
Author: Alex Miller <alex.miller@cognitect.com>
Date:   Fri Oct 28 08:42:26 2016 -0500

    CLJ-2024 stest/check should resolve function spec

    Signed-off-by: Stuart Halloway <stu@cognitect.com>
```

UŽDUOTĖLĖ

- susikurkite katalogą: `studentai`
- inicijuokite jame git repositoriją
- sukurkite failą `studentai.dat`
- atverkite failą ir įrašykite į jį savo vardą ir pavardę
- sucommitinkite pakeitimus
- peržvelkite ar istorijoje atsirado commitas

UŽDUOTĖLĖ - GALIMAS SPRENDIMAS

```
$ mkdir studentai  
$ cd studentai  
$ git init  
$ echo "Jonas Petraitis" >> studentai.dat  
$ git add studentai.dat  
$ git commit -m "Added my name to studentai.dat"  
$ git log
```

ŠAKOS (BRANCH)

- branch'as kuris sukuriamas pagal nutylėjimą - 'master'
- nauji branch'ai kuriami vartotojų
- branch'ai dažniausiai naudojami gaminant didesnį funkcionalumą norint perkelti jį į master branch'ą pilnai išbaigus


```
$ git checkout -b test  
Switched to a new branch 'test'
```

```
$ git branch  
  master  
* test
```

```
$ git checkout master  
Switched to branch 'master'
```

```
$ git branch  
* master  
  test
```

```
$ git merge <branch-name> - Atlikti merge commitą  
perkeliant nurodyto brancho pakeitimus į dabartinį
```

```
$ git branch
* master
  test
$ echo "master branch tekstas" >> file
$ git add .
$ git commit -m "Master commitas"
[master 2a5c8f1] Master commitas
1 file changed, 1 insertion(+)
```

```
$ git checkout -b naujas
Switched to a new branch 'naujas'
$ echo "naujas branch tekstas" >> file
$ git add .
$ git commit -m "Naujas commitas"
[naujas 52f1927] Naujas commitas
 1 file changed, 1 insertion(+)
$ git checkout master
Switched to branch 'master'
```

```
$ cat file
master branch tekstas
$ git merge naujas
Updating 2a5c8f1..52f1927
Fast-forward
  file | 1 +
  1 file changed, 1 insertion(+)
$ cat file
master branch tekstas
naujas branch tekstas
```

GIT ISTORIJOJOS NARSYMAS

Dažnai labai patogiu tai daryti grafiniais įrankiais

```
$ sudo apt-get install gitg  
$ gitg
```

customerdataservices (~code/omni/online)

master

All commits

Branches

active-fees-bugfix

all-contracts

batch-poc

boolean-mapper-fix

cnget-cleanup

different-ds

discount-ad...onal-index

FEATURE-2651

FEATURE-p...nd-services

fix

fix2

itest

✓ master

OOP-4214

OOP-4291

OOP-4313

OOP-4315

OOP-4315-test-refactor

OOP-4316

OOP-4558

OOP-7809

OOP-8365-discounts

OOP-8365-discounts-api

OOP-8812

OOP-8826

OOP-8861

OOP-8936

pom-fix

purge-queues

qa-master

release-1.12.0

release-1.3.0

Remotes

origin

active-fees-bugfix

all-contracts

BATCH-LOG

batch-poc

boolean-mapper-fix

Removed crap

Updated Promotion model again

Merge branch 'OOP-87' into OOP-2981

Merge branch 'OOP-87' of online-tooling.omnitel.lan:cds-team/customerdataservices into OOP-87

Tag id fix

Added liquibase test that checks some good practices and rollback/update

Merge branch 'OOP-2957' into 'OOP-87'

OOP-2930 Enrichers refactored

OOP-2930 Static mapper, accept enriched data

Merge branch 'OOP-87' of online-tooling.omnitel.lan:cds-team/customerdataservices into OOP-87

OOP-2981: made tests pass

Dropped invoice address and contact phone, email from API response schema

Merge commit '245add60' into OOP-87

Added activation dates, removed unnecessary lazy paramters in hbm

Updated Promotion model

Merge branch 'OOP-87' of online-tooling.omnitel.lan:cds-team/customerdataservices into OOP-87

Merge branch 'OOP-2956' into 'OOP-87'

Merge branch 'master' into OOP-2930

OOP-2930 Added loyalty info for account subscription

Added Account settings atomic endpoint with tests

Account settings api schema

Merge branch 'jaxb-plugin-conf-fix' into 'master'

Removed lotusNodesPromotionCode from Promotion model

Merge branch 'OOP-2954-address' into 'OOP-87'

Merge branch 'OOP-2954' into 'OOP-87'

Populator impl from account setting class

Removed contact phone and contact address from db model. For now it is still in API which will be refactored by other tasks

Edmondas Girkantas

Edmondas Girkantas

Zygimantas Gatelis

Zygimantas Gatelis

Antanas Bastys

Antanas Bastys

Antanas Bastys

Tadas Kvietkauskas

Tadas Kvietkauskas

Zygimantas Gatelis

Zygimantas Gatelis

Antanas Bastys

Antanas Bastys

Antanas Bastys

Edmondas Girkantas

Zygimantas Gatelis

Zygimantas Gatelis

Tadas Kvietkauskas

Tadas Kvietkauskas

Antanas Bastys

Antanas Bastys

Antanas Bastys

Edmondas Girkantas

Antanas Bastys

Antanas Bastys

Antanas Bastys

Antanas Bastys

Antanas Bastys

Apr 15 2015, 16:52

Apr 15 2015, 16:49

Apr 15 2015, 16:45

Apr 15 2015, 16:45

Apr 15 2015, 16:27

Apr 15 2015, 16:23

Apr 15 2015, 16:19

Apr 15 2015, 16:12

Apr 15 2015, 15:34

Apr 15 2015, 14:35

Apr 15 2015, 13:43

Apr 15 2015, 13:36

Apr 15 2015, 13:30

Apr 15 2015, 13:28

Apr 15 2015, 12:14

Apr 15 2015, 12:04

Apr 15 2015, 11:48

Apr 15 2015, 08:59

Apr 15 2015, 08:58

Apr 14 2015, 18:58

Apr 14 2015, 16:43

Apr 14 2015, 16:09

Apr 14 2015, 15:16

Apr 14 2015, 13:54

Apr 14 2015, 13:53

Apr 14 2015, 11:51

Apr 14 2015, 10:46

Antanas Bastys <antanas.bastys@tieto.com>

04/15/2015 01:28:33 PM +0300

Added activation dates, removed unnecessary lazy paramters in hbm

Expand all

025cf7291e547c24b3d38c191bd615caf52cd2bc

2

cds-commons/src/main/java/lt/omnitel/ws/cds/common/utis/AsyncUtils.java

9

cds-datamodel/src/main/java/lt/omnitel/ws/cds/model/account/AccountModel.java

44

cds-datamodel/src/main/java/lt/omnitel/ws/cds/model/account/AccountSettingModel.java

1

cds-repository/src/main/java/lt/omnitel/ws/cds/repository/impl/AccountRepository.java

3

cds-repository/src/main/java/lt/omnitel/ws/cds/repository/impl/AccountSettingRepository.java

1

cds-repository/src/main/java/lt/omnitel/ws/cds/repository/impl/SubscriptionRepository.java

8

cds-repository/src/main/resources/database/changelogs/1.1.0.xml

8

cds-repository/src/main/resources/mappings/mappings.hbm.xml

54

cds-repository/src/test/java/lt/omnitel/ws/cds/repository/account/AccountRepositoryTest.java

20

cds-repository/src/test/java/lt/omnitel/ws/cds/repository/account/AccountSettingRepositoryTest.java

GIT RODYKLĒS (POINTERS)

- HEAD - dabartinās būsēnos rodyklē
- Gudresni atvejai
<http://www.paulboxley.com/blog/2011/06/git-caret-and-tilde>

COMMITO ID

- 52f192703d9951f5bbaa55fc41c0c78bba811a49
- praktikoje unikaliai nusakyti pakanka 7 pirmų simbolių
52f1927
- žinant commit'o id galima atlikti visokių veiksmų. Tarkime nukreiptį HEAD'ą į šį commit'ą

```
$ git checkout 52f1927
```

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you do so (now or later) by using -b with the checkout command again. Example:

```
git checkout -b <new-branch-name>
```

HEAD is now at 52f1927... Naujas commitas

MERGE KONFLIKTAI

- Įvyksta kai ta pati failo vieta buvo modifikuota kelių vartotojų
- Galimi merge ir pull komandų metu
- Ištaisius konfliktus failuose juos reikia add'inti ir sucommitin'ti nekeičiant commit'o pranešimo

KONFLIKTO PAVYZDYS

```
Jonas Petraitis
<<<<<<< HEAD
1985-01-09
=====
Vilnius
>>>>>>> gimimo-vieta
```

- <<<<<<< ir >>>>>>> žymi pradžią ir pabaigą konfliktuojančios dalies
- <<<<<<< HEAD sako, jog konfliktuojantis blokas yra iš branch'o į kurį merginame
- ===== atskiria dalis tarp branch'ų
- >>>>>>> gimimo-vieta sako, jog dalis tarp ===== ir >>>>>>> yra iš branch'o gimimo-vieta

UŽDUOTĖ

- Įsitikinkite, kad esate master branch'e
- Sukurkite branch'ą pavadinimu gimimo-data
- Tame branch'e į failą studentai.dat naujoje eilutėje įrašykite savo gimimo datą
- Sucommitinkite
- Grįžkite į master branch'ą

UŽDUOTĖLĖ

- Sukurkite branch'ą gimimo - vieta
- Tame branch'e į failą studentai.dat naujoje eilutėje įrašykite savo gimimo vietą
- Sucommitinkite
- Grįžkite į master branch'ą

UŽDUOTĖ

- Įmerginkite gimimo datos pakeitimus į master branch'ą
- Įmerginkite gimimo vietos pakeitimus į master branch'ą

UŽDUOTĖLĖ - GALIMAS SPRENDIMAS

```
$ git checkout -b gimimo-data
Switched to a new branch 'gimimo-data'
$ echo "1985-01-03" >> studentai.dat
$ git add .
$ git commit -m "Gimimo data"
[gimimo-data 1da96ec] Gimimo data
1 file changed, 1 insertion(+)
$ git checkout master
Switched to branch 'master'
$ git checkout -b gimimo-vieta
Switched to a new branch 'gimimo-vieta'
$ echo "Vilnius" >> studentai.dat
$ git add .
$ git commit -m "Gimimo vieta"
[gimimo-vieta a309462] Gimimo vieta
1 file changed, 1 insertion(+)
```


DECENTRALIZACIJA

- Kiekviena gito repositorija yra kopija kažkurio laiko momento.
- Ji gali nepriklausomai vystytis
- Praktikoje dažniausiai yra Git Serveris, kuris yra centrinė repositorija (origin)
- Populiariausi serveriai: <https://github.com>, <https://gitlab.com>, <https://bitbucket.org>

REMOTE REPOSITORIJOS

- Praktikoje būna viena centrinė remote repozitorija vadinama 'origin'
- Kuriant naują git projektą ją reikia prisidėti pačiam (`git remote add`)
- Klonuojantis (`git clone`) egzistuojantį projektą ji pridedama automatiškai

\$ git remote add <name> <repository>

```
$ git remote add origin git@github.com:Antanukas/it-akademija-java-tech
$ git remote -v
origin  git@github.com:Antanukas/it-akademija-java-tech.git (fetch)
origin  git@github.com:Antanukas/it-akademija-java-tech.git (push)
```

\$ git clone <repository>

```
$ git clone git@github.com:Antanukas/it-akademija-java-tech.git
```

remote ar clone repositorio radimas

No description or website provided. — Edit

10 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Antanukas Implemented cart

mvn/wrapper	Initial spring project
src/main	Implemented cart
.eslintrc.json	Some ui stuff
.gitattributes	Update .gitattributes

2 months ago

Clone with SSH Use HTTPS

Use an SSH key and passphrase from account.

git@github.com:Antanukas/it-akademija-

Download ZIP

LOKALIOS REPO SINCHRONIZACIJA

```
$ git pull arba $ git fetch
```

PULL VS FETCH

- fetch - atsisiunčia pakeitimus iš repositrijos, bet nepritaiko jų lokaliai kopijai
- pull - padaro ta pati ką fetch, bet pritaiko gautus pakeitimus, ko pasekoje gali reikėti mergo
 - techniskai git pull atveju vyksta tokia seka jei esame master'yje:

```
$ git fetch  
$ git merge remotes/origin/master
```

- praktikoje visada darome `git pull` ir nesipariname

LOKALIŲ PAKEITIMŲ IŠSTUMIMAS Į REMOTE'Ą

```
$ git push origin <branch-name>
```

- jei remote'as pabėgęs į priekį bus atspausdinta klaida
 - tokiu atveju reikia atlikti `git pull`, išspręsti merge konfliktus ir bandyti vėl

UŽDUOTIS

- Susikurkite Githubo accounta
- Susikurkite Githubo repositoriją
- Prisidėkite remotą
- Papushinkite į remotą

Pabaiga