

# Git - zadania

## Spis treści

Wstęp .....	1
Zadanie 1 .....	2
Zadanie 2 .....	3
Zadanie 3 .....	3
Zadanie 4 .....	3

## Wstęp

Stworzyliśmy mini program, który pomoże Ci przygotować przypadki opisane w kolejnych zagadnieniach. Program ten możesz pobrać pod adresem [zajavka-git-gradle-boot](#).

Repozytorium zdalne z przygotowanymi wstępnymi przypadkami do zabawy znajdziesz pod adresem [zajavka-git-workshop-playground](#).

Sklonuj oba repozytoria. Repozytorium **zajavka-git-gradle-boot** zawiera kod źródłowy programu, który ma przygotowywać przypadki wyjściowe do zadań. Zanim uruchomisz ten program, musisz dostosować parę rzeczy w tym repozytorium. Repozytorium **zajavka-git-workshop-playground** jest natomiast piaskownicą, którą możesz psuć do woli. Schemat zabawy z poniższymi zadaniami będzie wyglądał w ten sposób, że wykonując odpowiednią metodę z repozytorium **zajavka-git-gradle-boot**, przygotujesz sobie historię zmian w repozytorium **zajavka-git-workshop-playground**, na podstawie której będziemy dalej realizować zagadnienia.

Będziesz potrzebować jeszcze jednego repozytorium zdalnego. Stwórz na swoim koncie **GitHub** repozytorium zdalne, które będzie wykorzystywane w ćwiczeniach. Lokalnie będziemy operować na repozytorium **zajavka-git-workshop-playground**, ale zepniemy je z Twoim repozytorium zdalnym. Czyli sklonujesz **zajavka-git-workshop-playground** z repozytorium zdalnego **zajavka**, ale wypychać zmiany będziesz już do swojego własnego repozytorium.

### Przejdźmy do konfiguracji:

Otwórz klasę **GitConfig** w projekcie **zajavka-git-gradle-boot** i dostosuj tam odpowiednie ustawienia:

```
public interface GitConfig {  
  
    String YOUR_LOCAL_USER_NAME = "Testowy Koleszka"; ①  
    String YOUR_LOCAL_USER_EMAIL = "testowykoleszka@zajavka.pl"; ②  
    String YOUR_LOCAL_PLAYGROUND_REPOSITORY_LOCATION = "./zajavka-git-workshop-playground/.git"; ③  
  
    String YOUR_GITHUB_USERNAME = "your_user_name"; ④  
    String YOUR_GITHUB_TOKEN = "your_token"; ⑤  
    String YOUR_REMOTE_PLAYGROUND_REPOSITORY_NAME = "repo_name"; ⑥  
  
    String GITHUB = "https://github.com";  
    String ORIGINAL_GITHUB_USERNAME = "zajavka";  
}
```

```
String ORIGINAL_GITHUB_PLAYGROUND_REPOSITORY_NAME = "zajavka-git-workshop-playground";
String ORIGINAL_GITHUB_PLAYGROUND_REPOSITORY_LOCATION = String.format("%s/%s/%s",
    GITHUB, ORIGINAL_GITHUB_USERNAME, ORIGINAL_GITHUB_PLAYGROUND_REPOSITORY_NAME);
String YOUR_GITHUB_PLAYGROUND_REPOSITORY_LOCATION = String.format("%s/%s/%s",
    GITHUB, YOUR_GITHUB_USERNAME, YOUR_REMOTE_PLAYGROUND_REPOSITORY_NAME);

}
```

- ① Wprowadź tutaj nazwę swojego użytkownika, wartość ta będzie ustawiona w lokalnym pliku *config*.
- ② Wprowadź tutaj email swojego użytkownika, wartość ta będzie ustawiona w lokalnym pliku *config*.
- ③ Wprowadź tutaj ścieżkę absolutną Twojego lokalnego repozytorium **zajavka-git-workshop-playground**, zgodnie z podanym schematem, czyli podaj lokalizację katalogu **.git**. Lokalnie będziemy pracować z tym repozytorium.
- ④ Tutaj wpisz Twoją nazwę użytkownika **GitHub**.
- ⑤ Tutaj podaj swój wygenerowany token do **GitHub**. W notatkach był podany link z opisem jak wygenerować taki token.
- ⑥ Podaj tutaj nazwę Twojego zdalnego repozytorium. Czyli tego repozytorium, które musisz stworzyć na swoim koncie, żeby móc dalej przejść przez te ćwiczenia. Program **zajavka-git-gradle-boot** skonfiguruje automatycznie Twoje **YOUR\_LOCAL\_PLAYGROUND\_REPOSITORY\_LOCATION**, by wskazywało ono na repozytorium zdalne **YOUR\_GITHUB\_PLAYGROUND\_REPOSITORY\_LOCATION**.

Pozostałych wartości nie zmieniaj.

W jaki sposób lokalne repozytorium zostanie spięte ze zdalnym repozytorium? Program **zajavka-git-gradle-boot** po uruchomieniu podmieni wpis w **/git/config** na poniższy:

```
[remote "origin"]
  url = https://github.com/your_user_name/repo_name ①
  fetch = +refs/heads/*:refs/remotes/origin/*
```

- ① W tym miejscu program **zajavka-git-gradle-boot** podmieni wpis w taki sposób, żeby Twoje lokalne repozytorium **zajavka-git-workshop-playground** wskazywało na Twoje repozytorium utworzone specjalnie na potrzeby tych ćwiczeń. Cała ta kombinacja wynika z tego, że **GitHub** pozwoli Ci pobrać zmiany z repozytorium **\_zajavka/zajavka-git-workshop-playground\***, ale nie będziesz mieć możliwości, żeby wypchać zmiany do tego repo. Poza tym korzystać z niego będziesz nie tylko Ty, ale również inni uczestnicy. Dlatego właśnie każdy uczestnik musi stworzyć swoje repozytorium zdalne i spiąć repozytorium lokalnie **zajavka-git-workshop-playground** ze swoim repozytorium zdalnym.



Zaznaczę tutaj, że jeżeli program będzie zgłaszał błędy, bo nie będzie mógł wykonać swoich zadań ze względu na to, że stan Twojego repozytorium na to nie pozwala - to dobrze. Masz wtedy okazję poćwiczyć i wyprostować historię zmian w Twoim repozytorium.

## Zadanie 1

Otwórz klasę **GitRunner** w **zajavka-git-gradle-boot**. Jeżeli wszystko zostało poprawnie skonfigurowane, powinno Ci się udać poprawnie uruchomić przypadek **Case1**. Uruchom **GitRunner** z przypadkiem **Case1**.

W tym momencie powinna Ci się utworzyć historia w Twoim repozytorium lokalnym **zajavka-git-workshop-playground** oraz w Twoim repozytorium zdalnym. Zapoznaj się z utworzoną historią.

Przypadek wygląda w ten sposób. Pracujesz na branchu **feature**, udało Ci się już wypchać zmiany do **origin/feature**. Stwórz **PR** i zmerguj swoje zmiany do **origin/master**. Gdy już to zrobisz, zaktualizuj stan lokalnego **master** do **origin/master**. Teraz okazuje się, że w Twoich zmianach był błąd, musisz wycofać zmianę wprowadzoną w drugim z trzech dodanych commitów. Wycofaj zatem tę zmianę, wykorzystując znane Ci techniki. Pamiętaj, że nie możesz pushować bezpośrednio do **mastera**!

## Zadanie 2

Otwórz klasę **GitRunner** w **zajavka-git-gradle-boot**. Jeżeli wszystko zostało poprawnie skonfigurowane, powinno Ci się udać poprawnie uruchomić przypadek **Case1**. Uruchom **GitRunner** z przypadkiem **Case1**. W tym momencie powinna Ci się utworzyć historia w Twoim repozytorium lokalnym **zajavka-git-workshop-playground** oraz w Twoim repozytorium zdalnym. Zapoznaj się z utworzoną historią.

Przypadek wygląda w ten sposób. Pracujesz na branchu **feature**, udało Ci się już wypchać zmiany do **origin/feature**. Dochodzisz jednak do wniosku, że zamiast trzech commitów chcesz mieć jeden. Zmodyfikuj zawartość historii w taki sposób, żeby zamiast 3 commitów powstał jeden z tymi samymi zmianami. Korzystając z **PR**, zmerguj swoje zmiany do **origin/master**. Następnie wyrównaj stan lokalnego **mastera** do **origin/master**. Pamiętaj, że nie możesz pushować bezpośrednio do **mastera**!

## Zadanie 3

Otwórz klasę **GitRunner** w **zajavka-git-gradle-boot**. Jeżeli wszystko zostało poprawnie skonfigurowane, powinno Ci się udać poprawnie uruchomić przypadek **Case1**. Uruchom **GitRunner** z przypadkiem **Case1**. W tym momencie powinna Ci się utworzyć historia w Twoim repozytorium lokalnym **zajavka-git-workshop-playground** oraz w Twoim repozytorium zdalnym. Zapoznaj się z utworzoną historią.

Przypadek wygląda w ten sposób. Pracujesz na branchu **feature**, udało Ci się już wypchać zmiany do **origin/feature**. Dochodzisz jednak do wniosku, że musisz zmodyfikować zawartość przedostatniego commita. Zmodyfikuj zawartość commita nr 2, ale nadal pozostaw trzy commity. Korzystając z **PR**, zmerguj swoje zmiany do **origin/master**. Następnie wyrównaj stan lokalnego **mastera** do **origin/master**. Pamiętaj, że nie możesz pushować bezpośrednio do **mastera**!

## Zadanie 4

Otwórz klasę **GitRunner** w **zajavka-git-gradle-boot**. Jeżeli wszystko zostało poprawnie skonfigurowane, powinno Ci się udać poprawnie uruchomić przypadek **Case2**. Uruchom **GitRunner** z przypadkiem **Case2**. W tym momencie powinna Ci się utworzyć historia w Twoim repozytorium lokalnym **zajavka-git-workshop-playground** oraz w Twoim repozytorium zdalnym. Zapoznaj się z utworzoną historią.

Przypadek wygląda w ten sposób. Pracujesz na branchu **feature**, udało Ci się już wypchać zmiany do **origin/feature** i pojawiły się nowe zmiany na branchu lokal **master**. Doprowadź teraz do sytuacji, gdzie zmiany z Twojego brancha wciągną zmiany z **master** i zmiany te znajdą się następnie na branchu **origin/master** przy wykorzystaniu **PR**. Następnie wyrównaj stan Twojego lokalnego **master** z **origin/master**. Pamiętaj, że nie możesz pushować bezpośrednio do **mastera**!