play! + Bitbucket + Jenkins

Paweł Skowroński

Kiedy już mamy zrobione testy naszej aplikacji, warto zautomatyzować uruchamianie testów dla naszego repozytorium. Oznacza to mniej więcej tyle, że za każdym razem, jak zostanie przeprowadzony commit do repozytorium z naszą aplikacją testy zostaną wykonane i zostanie sprawdzona poprawność dokonanych zmian. Takie zachowanie ma nazwę integracji ciągłej (CI - Continuous Integration). Najpopularniejszymi narzędziami CI są Jenkins i Hudson. Oba są bardzo do siebie podobne (kiedyś to był jeden projekt), ale ja opiszę jak skonfigurować działanie Jenkinsa dla projektu pisanego przy użyciu frameworka play! i trzymanego w repozytorium na bitbuckecie.

Przygotowanie Jenkinsa dla play! i Mercuriala

Pierwszym krokiem jaki powinniśmy podjąć, jest przygotowanie Jenkinsa dla projektów pisanych w play! (a tak naprawdę dla jakichkolwiek projektów pisanych w scali).

1. Na serwerze, na którym uruchomiony jest Jenkins potrzebujemy sbt-launchera (simple-build-tool dla scali). Możemy go znaleźć tutaj: **http://scala-sbt.org/download.html**. Jeśli natomiast nasz Jenkins jest na http://www.cloudbees.com/, to nie musimy się przejmować tą paczką - jest już zainstalowana.
2. Teraz musimy w naszym Jenkinsie włączyć sbt plugin. W tym celu wchodzimy w Jenkins -> Manage Jenkins -> Manage Plugins -> Available, znajdujemy “Jenkins sbt plugin” i instalujemy go.
3. W tym samym miejscu znajdujemy plugin “Jenkins Mercurial plugin” i także go instalujemy

Przygotowanie Jenkinsa do współpracy z bitbucketem

Aby Jenkins mógł współpracować z bitbucketem, musimy mieć ustawioną autoryzację użytkowników, a nie jest to domyślna opcja Jenkinsa. Aby skonfigurować zabezpieczenia Jenkinsa należy:

1. Wejść w Jenkins -> Manage Jenkins -> Configure Global Security

2 Zaznaczmy opcje “Enable secutrity”, “Jenkins own user database”, “Allow users to sign up” i “Logged-in users can do anything”

Uwaga: Kiedy po tych krokach stworzymy konto dla siebie, możemy wyłączyć opcję “Allow users to sign up”, co zabezpieczy system przed obcymi ludźmi.

Przygotowanie projektu Jenkinsa

Kolejnym krokiem jest już przygotowanie projektu Jenkinsa.

1 Wchodzimy w Jenkins -> New Job
2 Podajemy nazwę i wybieramy “Build a free-style software project”
3 W sekcji Source Code Management wybieramy Mercurial. W pole “Repository URL” wpisujemy adres naszego repozytorium z projektem, zazwyczaj w postaci “https://**username**@bitbucket.org/**username/projectname**”, w “Repository browser” wybieramy bitbucket i wpisujemy URL naszrgo projektu (zazwyczaj “https://bitbucket.org/**username/projectname/**”)
4 W sekcji Build Triggers wybieramy “Trigger builds remotely” i w pole “Authentication Token” wpisujemy jakiś długi losowy łańcuch znaków.
5 W sekcji Build wybieramy opcję “Build using sbt” i wybieramy sbt launcher wcześniej skonfigurowany
6 W pole Actions wpisujemy “clean test”
7 Musimy jeszcze znaleźć klucz użytkownika: Jenkins -> People -> Nasz profil -> Configure i po kliknięciu Show API Token kopiujemy ten klucz do schowka

Konfiguracja bitbucketa

Teraz pozostaje nam tylko skonfigurować Bitbucketa tak, aby akceptował połączenia naszego Jenkinsa. W tym celu:

1 Wchodzimy w nasze repozytorium na bitbuckecie
2 Wchodzimy w zakładkę admina
3 Wybieramy Services
4 Z listy wybieramy Jenkins i klikamy Add service
5 W pole token wpisujemy ten klucz z punktu 4 poprzedniego etapu konfiguracji
6 W pole Project name wpisujemy nazwę naszego projektu
7 W pole endpoint wpisujemy adres dostępowy do naszego Jenkinsa. Jest on w następującym formacie: http://**username**:**key@jenkinsurl.com**, gdzie
    - **username** - nazwa użytkownika Jenkinsa
    - **key** - jego klucz do API (pobrany w ostatnim punkcie poprzedniego etapu)
    - **jenkinsurl** - adres dostępowy do Jenkinsa (może być adres url, może być adres IP z odpowiednim portem)

Uruchamianie testów

Jeśli wykonaliśmy wszystkie punkty z listy i nasz projekt i scala na naszym serwerze sa dobrze skonfigurowane (odpowiednie wersje), to przy każdym pushu do naszego repozytorium, Jenkins będzie pobierał zmiany i załączał testy aplikacji. Dodatkowo możemy do konfiugracji projektu dołożyć wiele dostępnych w Jenkinsie pluginów, jak chociażby JDepend. Build projektu oczywiście możemy także wywoływać ręcznie, klkając na “Build now” w widoku projektu Jenkinsa.