Instrukcja: Konfiguracja i Publikacja Pakietu Maven w GitHub Packages

Niniejsza instrukcja przeprowadzi Cię przez proces konfiguracji konta GitHub oraz projektu Maven site, a następnie publikacji Twojej biblioteki w rejestrze pakietów GitHub (GitHub ackages).

Wymagania wstępne:

- Konto GitHub.
- Projekt Maven o nazwie site, który chcesz opublikować.

Krok 1: Generowanie Personal Access Token (PAT) na GitHubie i newego repozytorium

Będziesz potrzebować tokenu dostępu osobistego (PAT) do uwierzytelnienia Mavena w GitHub Packages.

1. Zaloguj się do swojego konta GitHub w przeglądarce internetowej.
2. W prawym górnym rogu ekranu, kliknij swoje zdjęcie profilowe, a następnie wybierz "Settings" (Ustawienia).
3. W lewym menu bocznym, przewiń w dół i kliknij "Developer settings" (Ustawienia deweloperskie).
4. W menu "Developer settings", wybierz "Personal access tokens" (Osobiste tokeny dostępu), a następnie "Tokens (classic)" (Tokeny klasyczne).
    - Ważna uwaga: Obecnie tokeny z precyzyjnymi uprawnieniami (Fine-grained tokens) nie działają poprawnie z Mavenem dla GitHub Packages. Musisz użyć "Tokens (classic)".
5. Kliknij przycisk "Generate new token (classic)" (Generuj nowy token klasyczny).
6. Skonfiguruj nowy token:
    - Note: Wpisz opisową nazwę tokenu, np. Maven Package Publisher lub site-package-deploy.
    - Expiration: Wybierz termin wygaśnięcia tokenu (np. 90 dni, 1 rok lub No expiration – ale pamiętaj o ryzyku bezpieczeństwa przy braku wygaśnięcia).
    - Select scopes: Zaznacz następujące pola wyboru (scopes):
        - repo (Pełna kontrola nad repozytoriami prywatnymi, często wymagane do publikacji)
        - write:packages (Do publikowania pakietów)
        - read:packages (Do pobierania pakietów, jeśli będziesz używać pakietów z GitHub Packages)
7. Na dole strony kliknij przycisk "Generate token" (Generuj token).
8. SKOPIUJ WYGENEROWANY TOKEN NATYCHMIAST! GitHub wyświetli ten token tylko raz. Jeśli go nie skopiujesz, będziesz musiał wygenerować nowy. Przechowuj go w bezpiecznym miejscu.

9. Utwórz nowe publiczne repozytorium na github o nazwie `site` (New repository).

Krok 2: Konfiguracja pliku settings.xml **dla Mavena**

Musisz poinformować swojego lokalnego Mavena, jak uwierzytelnić się w GitHub Packages.

1. Zlokalizuj plik settings.xml:
    - Dla systemów Linux/macOS: ~/.m2/settings.xml
    - Dla systemu Windows: C:\Users\TwojaNazwaUzytkownika\.m2\settings.xml
    - Jeśli plik settings.xml nie istnieje, utwórz go w tym miejscu.
2. Otwórz plik settings.xml w edytorze tekstu.
3. Dodaj następującą zawartość. Jeśli masz już inne sekcje, upewnij się, że dodajesz te fragmenty w odpowiednich miejscach (np. <server> wewnątrz <servers>, <profile> wewnątrz <profiles>).

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.01
http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <servers>
    <server>
      <id>github</id>
      <username>TWOJA_NAZWA_UZYTKOWNIKA_GITHUB</username>
      <password>TWÓJ_PERSONAL_ACCESS_TOKEN</password>
    </server>
  </servers>

  <profiles>
    <profile>
      <id>github</id>
      <repositories>
        <repository>
          <id>central</id>
          <url>https://repo1.maven.org/maven2</url>
          <releases><enabled>true</enabled></releases>
          <snapshots><enabled>true</enabled></snapshots>
        </repository>
        <repository>
          <id>github</id>
          <name>GitHub OWNER Apache Maven Packages</name>
<url>https://maven.pkg.github.com/TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_G
ITHUB/*</url>
          <snapshots>
            <enabled>true</enabled>
          </snapshots>
        </repository>
      </repositories>
    </profile>
  </profiles>

  <activeProfiles>
    <activeProfile>github</activeProfile>
  </activeProfiles>
```

```
</settings>
```

4. Zastąp następujące wartości:
    - TWOJA_NAZWA_UZYTKOWNIKA_GITHUB: Twoja dokładna nazwa użytkownika na GitHubie.
    - TWÓJ_PERSONAL_ACCESS_TOKEN: Token PAT, który skopiowałeś w Kroku 1.
    - TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_GITHUB: Ponownie, Twoja nazwa użytkownika GitHub (lub nazwa organizacji, jeśli publikujesz dla organizacji). Upewnij się, że jest identyczna w <url> dla repozytorium.

Krok 3: Konfiguracja pliku pom.xml **projektu** site

Musisz określić w projekcie site, gdzie Maven ma wdrożyć artefakty.

1. Otwórz plik pom.xml w katalogu głównym projektu site.
2. Upewnij się, że groupId jest ustawione poprawnie. Zalecana konwencja dla GitHub Packages to io.github.TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_GITHUB.

Przykład:

```
<groupId>io.github.twojanazwauzytkownika</groupId>
<artifactId>site</artifactId>
<version>1.0.0-SNAPSHOT</version> <packaging>jar</packaging>
```

3. Dodaj sekcję <distributionManagement> wewnątrz głównego tagu <project> w pom.xml. Ta sekcja informuje Mavena, gdzie ma wdrażać artefakty.

```
<distributionManagement>
  <repository>
    <id>github</id>
    <name>GitHub Packages</name>

<url>https://maven.pkg.github.com/TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_G
ITHUB/site</url>
  </repository>
  <snapshotRepository>
    <id>github</id>
    <name>GitHub Packages Snapshots</name>

<url>https://maven.pkg.github.com/TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_G
ITHUB/site</url>
  </snapshotRepository>
</distributionManagement>
```

    - Zastąp: TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_GITHUB Twoją nazwą użytkownika GitHub (lub nazwą organizacji).
    - site w URL: Upewnij się, że site w URL-u (.../site) odpowiada dokładnie nazwie Twojego repozytorium GitHub, w którym znajduje się kod źródłowy projektu site.

4. Dodaj wtyczki do budowania JAR, źródeł i Javadoc (zalecane): Aby zapewnić, że Twój pakiet będzie kompletny z kodem źródłowym i dokumentacją Javadoc, dodaj poniższe wtyczki do sekcji <build> -> <plugins> w pom.xml.

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-source-plugin</artifactId>
            <version>3.3.0</version> <executions>
                <execution>
                    <id>attach-sources</id>
                    <goals>
                        <goal>jar-no-fork</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-javadoc-plugin</artifactId>
            <version>3.6.3</version> <executions>
                <execution>
                    <id>attach-javadocs</id>
                    <goals>
                        <goal>jar</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-deploy-plugin</artifactId>
            <version>3.1.2</version> </plugin>
    </plugins>
</build>
```

Krok 4: Opublikowanie biblioteki site **do GitHub Packages**

Po skonfigurowaniu plików XML, możesz wdrożyć swoją bibliotekę.

1. Przejdź do zakładki Maven po lewej stronie.
2. Wybierz site->Lifecycle->clean
    - clean: Czyści katalog target (gdzie Maven przechowuje skompilowane pliki).
3. Wybierz site->Lifecycle->deploy
    - deploy: Kompiluje projekt, tworzy pliki JAR (w tym źródła i Javadoc, jeśli skonfigurowano) i przesyła je do skonfigurowanego repozytorium GitHub Packages.
4. Monitoruj dane wyjściowe w terminalu. Jeśli proces zakończy się sukcesem, zobaczysz komunikaty takie jak [INFO] Uploading main artifact… i na koniec [INFO] BUILD SUCCESS. Jeśli wystąpią błędy, sprawdź dokładnie poprzednie kroki i komunikaty błędów. W razie problemów możesz uruchomić mvn clean deploy -X dla bardziej szczegółowych logów.

Krok 5: Weryfikacja w GitHubie

Po pomyślnym wdrożeniu, możesz sprawdzić, czy pakiet został dodany do GitHub Packages.

1. Przejdź do swojego repozytorium GitHub, w którym znajduje się kod źródłowy projektu site.
2. Kliknij zakładkę "Packages" (Pakiety) po prawej stronie lub poszukaj sekcji "Packages" na głównej stronie repozytorium.
3. Powinieneś zobaczyć opublikowaną bibliotekę site z jej wersją. Kliknięcie na nią pokaże szczegóły pakietu, w tym instrukcje, jak dodać ją jako zależność w innych projektach.
4. Zmień widoczność pakietu na "Public"

Krok 6: Użycie biblioteki site **jako zależności w innym projekcie Maven (Opcjonalnie)**

Aby użyć opublikowanej biblioteki w innym projekcie Maven, musisz dodać ją jako zależność i skonfigurować repozytorium GitHub Packages.

- W pliku pom.xml nowego projektu, dodaj sekcję <repositories> (jeśli jej nie ma) i zdefiniuj repozytorium GitHub Packages:

```
<repositories>
    <repository>
        <id>github</id>
        <name>GitHub Packages</name>

<url>https://maven.pkg.github.com/TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_G
ITHUB/site</url>
        <snapshots>
            <enabled>true</enabled>
        </snapshots>
    </repository>
</repositories>
```

  - Zastąp: TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_GITHUB Twoją nazwą użytkownika GitHub (lub nazwą organizacji).
  - site w URL: Upewnij się, że site w URL-u odpowiada nazwie repozytorium, w którym znajduje się Twój pakiet.

- Dodaj zależność do biblioteki site:

```
<dependencies>
    <dependency>
        <groupId>io.github.twojanazwauzytkownika</groupId>
<artifactId>site</artifactId>
        <version>1.0.0-SNAPSHOT</version> </dependency>
</dependencies>
```

- Upewnij się, że twój pakiet jest publiczny. Jeśli pakiet jest prywatny, **każda osoba, która chce go pobrać, musi użyć własnego PAT**, aby uwierzytelnić się na GitHubie i skonfigurować plik settings.xml.

Po wykonaniu tych kroków Twoja biblioteka site będzie dostępna w GitHub Packages i gotowa do użycia w innych projektach Maven.

Użycie biblioteki site w innym projekcie

Dodaj do pliku pom.xml projektu repository i dependency do biblioteki `site` i innych wymaganych w `site`.

```
<repositories>
    <repository>
        <id>github</id>
        <name>GitHub Packages</name>

<url>https://maven.pkg.github.com/TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_G
ITHUB/site</url>
        <releases>
            <enabled>true</enabled>
        </releases>
        <snapshots>
            <enabled>true</enabled> </snapshots>
    </repository>
</repositories>

<dependencies>
    <dependency>
        <groupId>io.github.mdpiekarz</groupId>
        <artifactId>site</artifactId>
        <version>1.0-SNAPSHOT</version>
    </dependency>


    <!-- https://mvnrepository.com/artifact/org.xerial/sqlite-jdbc -->
    <dependency>
        <groupId>org.xerial</groupId>
        <artifactId>sqlite-jdbc</artifactId>
        <version>3.41.2.1</version>
    </dependency>
    <!-- https://mvnrepository.com/artifact/at.favre.lib/bcrypt -->
    <dependency>
        <groupId>at.favre.lib</groupId>
        <artifactId>bcrypt</artifactId>
        <version>0.10.2</version>
    </dependency>
</dependencies>
```

- Zastąp: TWOJA_NAZWA_UZYTKOWNIKA_LUB_ORGANIZACJA_GITHUB Twoją nazwą użytkownika GitHub (lub nazwą organizacji).