Table des matières

Jenkins 3
Exemple simple de build.xml (compilation) 3
Le fichier build.xml 3
Un autre exemple de build.xml 5
Classe TypeDeBien 11
D'autres classes Java 11

Jenkins

Intégration continue

Automatiser :

- la compilation
- la génération de la documentation de l'API de l'application
- les tests :
  - unitaires
  - fonctionnels, intégration, acceotation
  - ihm
- la génération des exécutables
- le déploiement
- ...

Exemple simple de build.xml (compilation)

http://192.168.108.111:8080/scm/hg/build_xml/file/c336a7d188b0/build.xml

Le fichier build.xml

build_xml

```
<?xml version="1.0" ?>
<project name="utilisation de checkstyle" default="compile"
basedir=".">
  <!-- Definition des proprietes du projet -->
  <property name="dossier.sources.projet"       value="src/main/java"/>
  <property name="projet.bin.dir"           value="target/classes"/>
  <property name="dossier.outils.projet"
value="src/main/resources"/>
  <property name="dossier.outils.checkstyle"
value="${dossier.outils.projet}/checkstyle"/>
  <property name="projet.temp.dir"          value="target/temp"/>
  <property name="projet.outils.dir"
value="src/main/resources/outils"/>
  <property name="projet.outils.lib.dir"
value="${dossier.outils.projet}/lib"/>
  <property name="dossier.rapports"    value="target/rapports/"/>
  <property name="dossier.rapports.checkstyle"
value="${dossier.rapports}/checkstyle"/>
  <!-- Definition du classpath du projet -->
```

```
  <path id="projet.classpath">
    <fileset dir="${dossier.outils.projet}">
      <include name="*.jar"/>
    </fileset>
    <fileset dir="${projet.outils.lib.dir}">
      <include name="*.jar"/>
    </fileset>
    <fileset dir="${dossier.outils.checkstyle}">
       <include name="*.jar"/>
    </fileset>
    <pathelement location="${projet.bin.dir}" />
  </path>
  <echo message="${env.CHECKSTYLE_HOME}"/>
  <!-- Declaration de la tache Ant permettant l'execution de checkstyle
-->
  <taskdef resource="checkstyletask.properties"
    classpathref="projet.classpath" />
    <!--
ou
<taskdef resource="checkstyletask.properties"
classpath="${basedir}/lib/checkstyle-5.6-all.jar"  ???/> -->

  <!-- execution de checkstyle -->
  <target name="checkstyle" description="CheckStyle">
    <checkstyle config="${dossier.outils.checkstyle}/sun_checks.xml">
      <fileset dir="${dossier.sources.projet}" includes="**/*.java"/>
      <formatter type="xml"
tofile="${dossier.rapports.checkstyle}/checkstyle.xml"/>
    </checkstyle>

  </target>
   <target name="rapport.checkstyle" description="Generation d'un
rapport HTML  pour Checkstyle." depends="checkstyle">
    <xslt in="${dossier.rapports.checkstyle}/checkstyle.xml"
          out="${dossier.rapports.checkstyle}/html/index.html"
          style="${env.CHECKSTYLE_HOME}/contrib/checkstyle-
noframes.xsl">
      <param name="title" expression="Checkstyle Report" />
      <param name="module" expression="${system.name}" />
    </xslt>
  </target>
  <!-- Compilation des classes du projet -->
  <target name="compile" depends="checkstyle" description="Compilation
des classes">
    <javac srcdir="${dossier.sources.projet}"
      destdir="${projet.bin.dir}"
      debug="on"
      optimize="off"
      deprecation="on">
```

```
      <classpath refid="projet.classpath"/>
    </javac>
  </target>

</project>
```

Un autre exemple de build.xml

build.xml

```
<project name="AntDefault" default="run" basedir=".">
    <description>
        Ant build sample for running
        - findbugs,
        - pmd,
        - checkstyle,
        - cobertrua and
        - junit.
    </description>

     <!-- Change the paths to your individual installation directories
-->
    <property name="checkstyle.home.dir"
location="C:/EU/checkstyle/checkstyle-5.0" />
    <property name="cobertura.home.dir"
value="C:/EU/cobertrua/cobertura-1.9.2" />
    <property name="findbugs.home.dir"
value="C:/EU//findbugs/findbugs-1.3.9-rc2" />
    <property name="pmd.home.dir" value="C:/EU//pmd/pmd-4.2.5" />
    <property name="lib.dir" location="${basedir}/lib" />
    <property name="src.dir" location="${basedir}/src" />
    <property name="bin.dir" location="${basedir}/bin" />
    <property name="htm.dir" location="${basedir}/html" />
    <property name="report.dir" location="${basedir}/report" />

    <property name="report.checkstyle.dir"
location="${report.dir}/checkstyle" />
    <property name="report.junit.dir" location="${report.dir}/junit" />
    <property name="report.findbugs.dir"
location="${report.dir}/findbugs" />
    <property name="report.cobertura.dir"
location="${report.dir}/cobertura" />
    <property name="report.pmd.dir" location="${report.dir}/pmd" />

    <property name="instrumented.dir"
location="${basedir}/instrumented" />
```

```
    <property name="report.temp.dir" location="${report.dir}/temp" />
    <property name="cobertura.ser.file"
location="${report.temp.dir}/cobertura.ser" />

    <path id="run.classpath">
        <pathelement path="${bin.dir}" />
        <pathelement path="${basedir}/lib/junit-4.5.jar" />
    </path>

    <path id="cobertura.classpath">
        <path refid="run.classpath" />
        <fileset dir="${cobertura.home.dir}">
            <include name="cobertura.jar" />
            <include name="lib/**/*.jar" />
        </fileset>
    </path>
    <taskdef classpathref="cobertura.classpath"
resource="tasks.properties" />

    <!-- -->
    <target name="clean" description="Delete all result to start with a
clean build.">
        <delete dir="${report.junit.dir}" />
        <delete dir="${report.findbugs.dir}" />
        <delete dir="${report.cobertura.dir}" />
        <delete dir="${report.checkstyles.dir}" />
        <delete dir="${report.temp.dir}" />
        <delete dir="${bin.dir}" />
    </target>

    <target name="prepare.report.dir" description="Prepares the reports
folder">
        <copy todir="${report.dir}">
            <fileset dir="${htm.dir}">
                <include name="*.htm" />
            </fileset>
        </copy>
    </target>

    <!-- -->
    <target name="findbugs"
            depends="compile"
            description="Run code analysis over code to check for
problems."
    >
        <!-- Fail this target if FindBugs is not installed. -->
        <available file="${findbugs.home.dir}/lib/findbugs.jar"
property="findbugs.available" />
        <fail unless="findbugs.available"
```

```
            message="Error: FINDBUGS_HOME not set or findbugs.jar not
found."
        />
        <taskdef name="findbugs"
                 classname="edu.umd.cs.findbugs.anttask.FindBugsTask"
                 classpath="${findbugs.home.dir}/lib/findbugs-ant.jar"
        />

        <!-- Run FindBugs. -->
        <mkdir dir="${report.findbugs.dir}" />
        <findbugs home="${findbugs.home.dir}"
                  workHard="true"
                  output="xml:withMessages"
                  outputFile="${report.findbugs.dir}/findbugs.xml"
        >
            <class location="${bin.dir}" />
            <auxClasspath>
                <fileset file="${basedir}/lib/junit-4.5.jar" />
            </auxClasspath>
        </findbugs>
    </target>

    <!-- -->
    <path id="pmd2.classpath">
        <pathelement location="${build}" />
        <fileset dir="${pmd.home.dir}/lib/">
            <include name="*.jar" />
        </fileset>
    </path>
    <taskdef name="pmd" classname="net.sourceforge.pmd.ant.PMDTask"
        classpathref="pmd2.classpath" />
    <target name="report.pmd">
        <mkdir dir="${report.pmd.dir}" />
        <pmd rulesetfiles="rulesets/favorites.xml">
            <formatter type="xml"
toFile="${report.pmd.dir}/pmd_report.xml" />
            <fileset dir="${src.dir}">
                <include name="**/*.java" />
            </fileset>
        </pmd>
        <xslt in="${report.pmd.dir}/pmd_report.xml"
              style="${pmd.home.dir}/etc/xslt/pmd-report-per-
class.xslt"
              out="${report.pmd.dir}/pmd_report.html"
        />
    </target>

    <!-- -->
    <target name="compile" depends="clean" description="Compile the
entire project.">
        <mkdir dir="${bin.dir}" />
```

```
        <javac debug="true"
               debuglevel="lines, source"
               srcdir="${src.dir}"
               destdir="${bin.dir}"
               classpathref="run.classpath"
        />
    </target>

    <!-- -->
    <target name="cobertura.instrument"
            depends="clean, compile"
            description="Instrument the project for code coverage."
    >
        <mkdir dir="${report.temp.dir}" />
        <cobertura-instrument todir="${instrumented.dir}"
datafile="${cobertura.ser.file}">
            <ignore regex="org.apache.log4j.*" />
            <fileset dir="${bin.dir}">
                <include name="**/*.class" />
                <exclude name="**/*Test.class" />
            </fileset>
        </cobertura-instrument>
    </target>

    <!-- -->
    <target name="junit"
            depends="clean, compile, cobertura.instrument"
            description="Run all junit test cases."
    >
        <mkdir dir="${report.cobertura.dir}" />
        <mkdir dir="${report.temp.dir}" />

        <junit printsummary="yes" fork="yes" haltonfailure="no">

            <!-- Specify the name of the coverage data file to use.
                 The value specified below is the default.    -->
            <sysproperty key="net.sourceforge.cobertura.datafile"
                file="${cobertura.ser.file}" />

            <!-- Note the classpath order: instrumented classes are
before
                 the original (uninstrumented) classes.  This is
important. -->
            <classpath location="${instrumented.dir}" />
            <classpath location="${bin.dir}" />
            <classpath location="${basedir}/lib/junit-4.5.jar}" />

            <!-- The instrumented classes reference classes used by the
                 Cobertura runtime, so Cobertura and its dependencies
```

```
                must be on your classpath. -->
            <classpath refid="cobertura.classpath" />

            <formatter type="plain" />
            <formatter type="xml" />

            <batchtest todir="${report.temp.dir}">
                <fileset dir="${src.dir}">
                    <include name="**/*Test.java" />
                </fileset>
            </batchtest>
        </junit>
    </target>

    <!-- -->
    <target name="report.junit" depends="junit"
        description="Create a report for the test result.">
        <delete dir="${report.junit.dir}" />
        <mkdir dir="${report.junit.dir}" />
        <junitreport todir="${report.junit.dir}">
            <fileset dir="${report.temp.dir}">
                <include name="*.xml" />
            </fileset>
            <report format="frames" todir="${report.junit.dir}" />
        </junitreport>
    </target>

    <!-- -->
    <target name="report.findbugs" description="Generate a report on
error analysis.">
        <xslt in="${report.findbugs.dir}/findbugs.xml"
              style="${findbugs.home.dir}/src/xsl/fancy.xsl"
              out="${report.findbugs.dir}/findbugs-default.html"
        />
    </target>

    <!-- -->
    <target name="report.cobertura"
            depends="junit"
            description="Generate an HTML report on Cobertura."
    >
        <cobertura-report format="html"
                          datafile="${cobertura.ser.file}"
                          destdir="${report.cobertura.dir}"
        >
            <fileset dir="${src.dir}">
                <include name="**/*.java" />
            </fileset>
        </cobertura-report>
    </target>
```

```
    <!-- -->
    <target name="report.checkstyle"
        description="Generate a report of code convention violations.">
        <taskdef resource="checkstyletask.properties"
                 classpath="${checkstyle.home.dir}/checkstyle-
all-5.0.jar"
        />

        <!-- run verification of installation-->
        <available file="${checkstyle.home.dir}/checkstyle-all-5.0.jar"
                   property="checkstyle.available"
        />
        <fail unless="checkstyle.available"
              message="Error: CHECKSTYLE_HOME not set or checkstyle-
all-5.0.jar not found."
        />
        <mkdir dir="${report.checkstyle.dir}" />

        <!-- run analysis-->
        <checkstyle config="${checkstyle.home.dir}/sun_checks.xml"
                    failureProperty="checkstyle.failure"
                    failOnViolation="false"
        >
            <formatter type="xml"
tofile="${report.checkstyle.dir}/checkstyle_report.xml" />
            <fileset dir="${src.dir}" includes="**/*.java" />
        </checkstyle>

        <style in="${report.checkstyle.dir}/checkstyle_report.xml"
               out="${report.checkstyle.dir}/checkstyle_report.html"
               style="${checkstyle.home.dir}/contrib/checkstyle-
noframes.xsl"
        />
    </target>

    <!-- -->
    <target name="clean.temp" description="Delete all temporary files
and folders.">
        <delete dir="${instrumented.dir}" />
        <delete dir="${report.temp.dir}" />
    </target>

    <!-- -->
    <target name="run"
            description="Run the build"
            depends="clean,
                prepare.report.dir,
                report.checkstyle,
```

```
                report.pmd,
                compile,
                cobertura.instrument,
                junit,
                report.junit,
                findbugs,
                report.findbugs,
                report.cobertura,
                clean.temp"
    >
    </target>

</project>
```

Classe TypeDeBien

TypeDeBien.java

```
package domaine;

/**
 *
 * @author Br
 */
public class TypeDeBien {


}
```

D'autres classes Java

L'archive :domaine.zip

From:
https://wikisio.lyceejeanbart.fr/ - **wikiSio**

Permanent link:
https://wikisio.lyceejeanbart.fr/doku.php?id=ouvert_a_tous:dafop_2014:jenkins:accueil

Last update: **2022/12/03 07:45**