Jenkins

- description : 인텔리제이 엔티티 자동생성 코드
- author : 와프
- email : iyyeo@repia.com
- lastupdate : 2021-09-15

```
import com.intellij.database.model.DasTable
import com.intellij.database.model.ObjectKind
import com.intellij.database.util.Case
import com.intellij.database.util.DasUtil
import com.intellij.psi.codeStyle.NameUtil

import javax.swing.*

/**
 * @author shirohoo
 * @link https://github.com/shirohoo/create-automation-jpa-entity
 * @param pakageName, primaryKey
 *
 * <pre>
 *
 *     this script's
```

```
default primary key strategy is @GeneratedValue(strategy = GenerationType.IDENTITY)
 *     and specialized in Microsoft SQL Server
 *     and finally implemented Serializable so recommend that create serial version UID
 *
 *
first. enter your project package name. for example:
 *     > com.intelliJ.psi
 *
 *
second. enter primary key column name of target database table.
 *     this script is convert input to camel case. for example 1:
 *     >
```

```
table
primary key
column name
= MEMBER_ID
 *     >
enter
primary key
= memberId
 *
 *
example 2:
 *     >
table
primary key
column name
= ID
 *     >
enter
primary key
= id
 *
 * </pre>
 */

columnType
= [
(~/(?i)bigi
nt/)
: "Long",
(~/(?i)int/
)
:
"Integer",
(~/(?i)bit/
)
:
"Boolean",
(~/(?i)deci
mal/)
:
"BigDecimal
",
(~/(?i)floa
t|double|re
al/) :
"Double",
(~/(?i)date
time|timest
amp/):
"LocalDateT
ime",
```

```
(~/(?i)time
/)
:
"LocalTime"
,
(~/(?i)date
/)
:
"LocalDate"
,
(~/(?i)nvar
char/)
:
"nvarchar",
(~/(?i)varc
har/)
:
"varchar",
(~/(?i)char
/)
: "String",
(~/(?i)text
/)
: "String"
]
def input =
{
    JFrame
jframe =
new
JFrame()
    String
answer =
JOptionPane
.showInputD
ialog(jfram
e, it)
jframe.disp
ose()
    answer
}
packageName
=
input("Ente
r your
package
name")
primaryKey
=
input("Ente
r column
```

```
name of primary key")
FILES.chooseDirectoryAndSave("Choose directory", "Choose where to store generated files") {
dir ->
SELECTION.filter {
        it instanceof DasTable && it.getKind() == ObjectKind.TABLE
    }.each {
generate(it, dir)
    }
}

def generate(table, dir) {
    def tableName = table.getName()
    def className = convertFieldName(tableName + "Entity", true)
    def fields = categorizeFields(table)
    new File(dir,
```

```
className +
".java").wi
thPrintWrit
er {
        out
->
generate(ou
t,
tableName,
className,
fields)
    }
}

def
generate(ou
t,
tableName,
className,
fields) {
out.println
"package
$packageNam
e;"
out.println
""
out.println
"import
javax.persi
stence.*;"
out.println
"import
java.time.L
ocalDateTim
e;"
out.println
"import
lombok.Data
;"
out.println
""
out.println
"@Entity"
out.println
"@Data"
out.println
"@Table(nam
e =
\"$tableNam
e\")"
out.println
```

```
"public class $className"
out.println "{"
fields.each() {
        if (it.annos != "") {
out.println "${it.annos}"
        }
        if (it.name == primaryKey) {
out.println "   @Id"
out.println "@GeneratedValue(strategy = GenerationType.IDENTITY)"
        }
        if (it.type == 'nvarchar') {
out.println "@Nationalized"
out.println "@Column(name = \"${it.colName}\")"
out.println "   private String ${it.name};"
        }
else if
```

```
(it.type == 'varchar') {
out.println "    @Column(name = \"${it.colName}\")"
out.println "    private String ${it.name};"
        }
else {
out.println "    @Column(name = \"${it.colName}\")"
out.println "    private ${it.type} ${it.name};"
        }
out.println ""
    }
out.println "}"
}

def categorizeFields(table) {
DasUtil.getColumns(table).reduce([]) {
fields, col ->
        def spec = Case.LOWER.apply(col.getDataType().getSpecif
```

```
ication())
        def
typeStr =
columnType.
find {
p, t ->
p.matcher(s
pec).find()
}.value
fields +=
[[
colName:
col.getName
(),
name   :
convertFiel
dName(col.g
etName(),
false),
type   :
typeStr,
annos  :
""]]
    }
}

def
convertFiel
dName(str,
capitalize)
{
    def s =
NameUtil.sp
litNameInto
Words(str)
.collect {
Case.LOWER.
apply(it).c
apitalize()
}
.join("")
.replaceAll
(/[^\p{java
JavaIdentif
ierPart}[_]
]/, "_")
capitalize
||
s.length()
== 1 ? s :
Case.LOWER.
```

```
                                                    apply(s[0])
                                                    + s[1..-1]
                                                    }
```

Tip

Troubleshooting

Ref

와프, Entity, IntelliJ, 인텔리제이, 엔티티, 자동생성

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:user:iyyeo:entity&rev=1631664974

Last update: **2022/03/10 19:52**