Timer Job

Plugins können neue Timer-Tasks für die im Framework eingebaute Zeitsteuerung mitbringen. Diese können dann beim Anlegen neuer Zeitsteuerungen ausgewählt und parametrisiert werden.

```
fun registerTimerJob(jobId: String, jobInfo: JobInfo)
```

`jobId` ist eine frei wählbare ID, die über alle Timer Jobs eines Plugins hinweg eindeutig sein muss. Es ist kein Problem, wenn mehrere Plugins einen Job mit derselben ID bereitstellen, da zum Ansprechen eines Jobs auch die `id` des Plugins herangezogen wird.

Der Titel des Jobs, sowie eine Instanz der implementierenden Klasse und die Standard-Parameter werden in `jobInfo` festgehalten. Die Standard-Parameter sind ein String-Array - jedes Element davon wird in der Oberfläche im Textfeld in eine eigene Zeile geschrieben (Kommentarzeilen beginnen mit `#`).

> **Thread-Safety**
> Die Instanz wird für alle Ausführungen nachgenutzt, sollte also thread-safe programmiert werden.

Die Klasse muss die `run(JobContext)` Methode aus `com.batix.plugins.Job` implementieren. `JobContext` enthält Informationen zum aktuellen Aufruf, wie z. B. die vom Benutzer eingestellten Parameter (`properties`) oder das Projekt, in dem die Zeitsteuerung definiert wurde (`web`).

Beispiel

Der Timer Job wird dem Framework in der Plugin-Hauptklasse bekannt gegeben.

```
override fun load() {
  registerTimerJob(
    "logString", // jobId
    JobInfo("Log String", LogStringJob(), arrayOf("level", "#text"))
  )
}
```

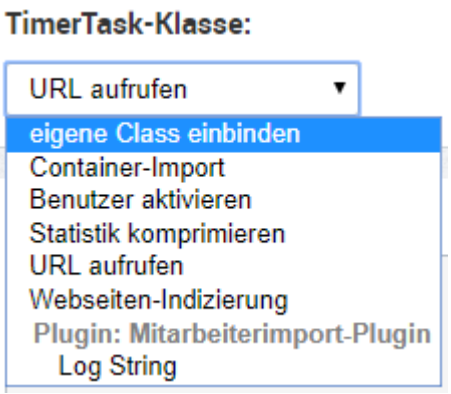
Diese Beispiel-Job-Klasse liest die eingestellten Parameter und loggt die festgelegte Nachricht mit dem entsprechenden Level.

```
import com.batix.Log
import com.batix.plugins.Job
import com.batix.plugins.JobContext

class LogStringJob : Job {
  override fun run(context: JobContext) {
    val level = context.properties.getProperty("level", "INFO")
    val text = context.properties.getProperty("text", "DEFAULT TEXT")

    when (level.lowercase()) {
      "error" -> Log.error(text)
      "warn" -> Log.warn(text)
      "notice" -> Log.notice(text)
      "debug" -> Log.debug(text)
      else -> Log.info(text)
    }
  }
}
```

Der Job ist dann in der TimerTask-Auswahl einer Zeitsteuerung sichtbar.



Wählt man diesen aus, werden die Steuerparameter vorausgefüllt.

```
level=
#text=
```