

Zweck

Der Job **recordEventualities** sammelt täglich alle aktuell erkannten Eventualitäten (z. B. Konflikte oder Konsistenzverletzungen in HR-Daten) und hält sie in der Datenbank auf dem neuesten Stand. Er sorgt dafür, dass neue Ereignisse erfasst und erledigte Fälle automatisch historisiert werden.

Ablauf

1. Beim Start ruft der Job alle registrierten Eventualitäten-Detektoren auf. Jeder Detektor liefert eine Liste der aktuell bestehenden Eventualitäten seines Typs.
2. Parallel dazu werden alle bereits gespeicherten Eventualitäten gleichen Typs mit dem heutigen Datum ermittelt.
3. Neue Ereignisse, die noch nicht in der Datenbank stehen, werden gespeichert und gelten ab sofort als aktiv.
4. Eventualitäten, die zuletzt gespeichert wurden, jetzt aber nicht mehr auftreten, werden historisiert und gelten damit als erledigt.
5. Fehler einzelner Detektoren beeinflussen die übrigen Detektoren nicht; sie werden protokolliert und der Job arbeitet mit den verbleibenden Detektoren weiter.

Bekannte Eventualitäten-Detektoren

- **HrPersonJobUnassignedTaskEventualitiesDetector**: Meldet Personen, die zwar einem Job zugeordnet sind, aber einzelne Aufgaben dieses Jobs nicht besitzen.
- **HrPersonTaskUnassignedJobEventualitiesDetector**: Meldet Personen, die eine Aufgabe erhalten haben, ohne dem zugehörigen Job zugeordnet zu sein.
- **HrPersonJobUnassignedLevelEventualitiesDetector**: Meldet Personen, die einem Job-Level zugeordnet sind, das in der Jobdefinition nicht freigegeben ist.

Zusätzliche Detektoren können projektabhängig vorhanden sein. Sie werden automatisch berücksichtigt, sobald sie als Spring-Beans registriert sind.

Konfiguration

Der Job besitzt keine eigene Konfigurationsoberfläche. Er verarbeitet automatisch alle im System registrierten Detektoren. Anpassungen erfolgen über die jeweiligen Detektor-Implementierungen (z. B. Aktivierung/Deaktivierung per Spring-Konfiguration oder Projekterweiterungen).

Bedienhinweise

- Planen Sie den Job täglich oder häufiger ein, damit Eventualitäten zeitnah aktualisiert werden.
- Überwachen Sie die Logs: Dort sehen Sie, wie viele Fälle pro Detektor erkannt, gespeichert oder historisiert wurden.
- Bei neu entwickelten Detektoren empfiehlt sich ein Testlauf in einer Staging-Umgebung, um sicherzustellen, dass Metadaten konsistent sind.
- Falls ein Detektor fehlerhaft arbeitet, deaktivieren Sie ihn über die Spring-Konfiguration, damit der Gesamtjob weiterlaufen kann.
- Historisierte Einträge bleiben erhalten und können für Analysen oder Audits genutzt werden.