

“TypedQuery“ Interface und “Query“ Interface – Methodenüberblick:

1.) Query Interface bzw. TypedQuery Interface Referenz erhalten:

```
// Typsicherer Zugriff auf eine Query eines bestimmten Entity Type:
TypedQuery<Message> query = em.createQuery(queryString, Message.class);

// Zugriff auf eine Query eines bestimmten Entity Type:
Query query = em.createQuery( queryString );
```

2.) Operationen auf dem Query Interface bzw. TypedQuery Interface:

```
// Ergibt ein einzelnes unique Objekt oder eine NonUniqueResultException:
Object getSingleResult();           // Query Interface
<T>    getSingleResult();           // TypedQuery Interface

// Ergibt eine Liste von Objekten bzw. eine typsichere Ergebnis-Liste:
List   getResultList();             // Query Interface
List<T> getResultList();             // TypedQuery Interface

// Konfiguriert den Startindex und die Anzahl der Objekte in der ResultList:
// (Diese Methoden sollten nicht für Queries verwendet werden, die über Collections
// Relationen (OneToMany oder ManyToMany) JOINen da solche Queries doppelte Ergebnis-
// Elemente zurückgeben können. Doppelte Elemente im ResultSet verhindern dadurch die
// Verwendung einer logischen Result Position.):

// Pagination Support:
Query setFirstResult(int startPosition); // Query u. TypedQuery Interface
Query setMaxResults(int maxResults);    // Query u. TypedQuery Interface

int getFirstResult(); // Query u. TypedQuery Interface
int getMaxResults();  // Query u. TypedQuery Interface

// Ausführen einer saveOrUpdate (auch Bulk) Operation, gibt die Row Anzahl zurück:
int executeUpdate(); // Query u. TypedQuery Interface

// Provider-abhängige Konfigurationen setzen bzw. Persistenz Manager konfigurieren:
Query setHint(String hintName, Object value); // Query u. TypedQuery Interface
// Beispiele hierzu:
query.setHint("javax.persistence.query.timeout", 5000);
query.setHint("eclipselink.cache-usage", "DoNotCheckCache");

// Query Parameter setzen, wenn eine Query ersetzbare Parameter anbietet:
Query setParameter(String name, Object value); // Query u. TypedQuery Interface
Query setParameter(String name, Calendar value); // Query u. TypedQuery Interface
Query setParameter(String name, Date value); // Query u. TypedQuery Interface
// Bei Verwendung der Date.class sind bekannte Probleme (Thread-safety) evtl. zu beachten.

// Setzen des Flush Modes einer Query (Query u. TypedQuery Interface):
Query setFlushMode(FlushModeType flushMode); //AUTO(sieht Entityänderungen),COMMIT(nicht)

// Setzen des Lock Modes einer Query (Query u. TypedQuery Interface), mögliche Typen:
// PESSIMISTIC_WRITE, READ, OPTIMISTIC, OPTIMISTIC_FORCE_INCREMENT
Query setLockMode(LockModeType lockModeType);

// Beispiel zur Verwendung des LockModeType.OPTIMISTIC (2.0) bzw. LockModeType.READ (1.0):
// ➔ nur der erste bekommt Zugriffs-Lock: entweder lesende Transaktion od. Entity-Änderer:
TypedQuery<Employee> query = em.createQuery(
    "SELECT c FROM Customer c WHERE (c.name = '" + "Average_Joe' )", Customer.class);
query.setLockMode(LockModeType.READ); // same as LockModeType.OPTIMISTIC
query.setHint("javax.persistence.lock.timeout",5000);
```