Technologie Java Enterprise
Persistence API #3

Łukasz Rybka · Gdańsk 2015

JPQL

JPQL

Java Persistence Query Language

JPQL

Java Persistence Query Language

Pozwala wykonywać zapytania bazodanowe w pseudo-SQL'owym języku

JPQL

Java Persistence Query Language

Pozwala wykonywać zapytania bazodanowe w pseudo-SQL'owym języku

Wsparcie dla natywnych query

JPQL

Java Persistence Query Language

Pozwala wykonywać zapytania bazodanowe w pseudo-SQL'owym języku

Wsparcie dla natywnych query

Dostępne operatory JOIN (ale z pewnymi ograniczeniami)

Critera API

Critera API

Deklaratywna forma JPQL

Critera API

Deklaratywna forma JPQL

Posiada te same możliwości co JPQL

Critera API

Deklaratywna forma JPQL

Posiada te same możliwości co JPQL

Przyjemniejsze dla niektórych developerów (kwestia gustu)

Paginacja

```
@Stateless
public class MessageStorageService {
    @PersistenceContext
    EntityManager manager;

    public List<Message> getMessages(int offset, int limit){
        return manager.createNamedQuery("message.all").
                setFirstResult(offset).
                setMaxResults(limit).
                getResultList();
    }

    public Long getCount() {
        return (Long) manager.createQuery("select count(m) from
Message m")
                    .getSingleResult();
    }
}
```

Podstawowe zagadnienia

Wyszukiwanie po wartości

Wyszukiwanie po wartości
Named queries

Wyszukiwanie po wartości

Named queries

Wyszukiwanie po dacie

Podstawowe zagadnienia

- Wyszukiwanie po wartości
- Named queries
- Wyszukiwanie po dacie
- Wyszukiwanie po połączonej tabeli

Pokazywanie zapytania

Pokazywanie zapytania

Inna konfiguracja dla każdego provider'a

Pokazywanie zapytania

Inna konfiguracja dla każdego provider'a

Domyślny provider GlassFish 4.X to EclipseLink

Pokazywanie zapytania

Inna konfiguracja dla każdego provider'a

Domyślny provider GlassFish 4.X to EclipseLink

Inna konfiguracja w różnych wersjach

Pokazywanie zapytania

```
<persistence>
  <persistence-unit name="MessagesManagement">
    <jta-data-source>jdbc/hsqlapps</jta-data-source>

    <properties>
      <property name="javax.persistence.schema-
generation.database.action"
                value="drop-and-create"/>
      <property name="javax.persistence.sql-load-script-source"
                value="META-INF/sql/load.sql" />

      <property name="eclipselink.logging.level" value="FINE"/>
      <property name="eclipselink.logging.level.sql" value="FINE"/>
      <property name="eclipselink.logging.parameters" value="true"/>
    </properties>
  </persistence-unit>
</persistence>
```

Funkcje JPQL

CONCAT(string1, string2)

Funkcje JPQL

CONCAT(string1, string2)

SUBSTRING(string, startIndex, length)

Funkcje JPQL

CONCAT(string1, string2)
SUBSTRING(string, startIndex, length)
LOWER(string)

Funkcje JPQL

CONCAT(string1, string2)
SUBSTRING(string, startIndex, length)
LOWER(string)
UPPER(string)

Funkcje JPQL

CONCAT(string1, string2)
SUBSTRING(string, startIndex, length)
LOWER(string)
UPPER(string)
LENGTH(string)

Funkcje JPQL

Funkcje JPQL

ABS(number)

Funkcje JPQL

ABS(number)

SQRT(number)

Funkcje JPQL

ABS(number)
SQRT(number)
MOD(number, divisor)

Funkcje JPQL

ABS(number)
SQRT(number)
MOD(number, divisor)
CURRENT_DATE

Funkcje JPQL

ABS(number)
SQRT(number)
MOD(number, divisor)
CURRENT_DATE
CURRENT_TIME

Funkcje JPQL

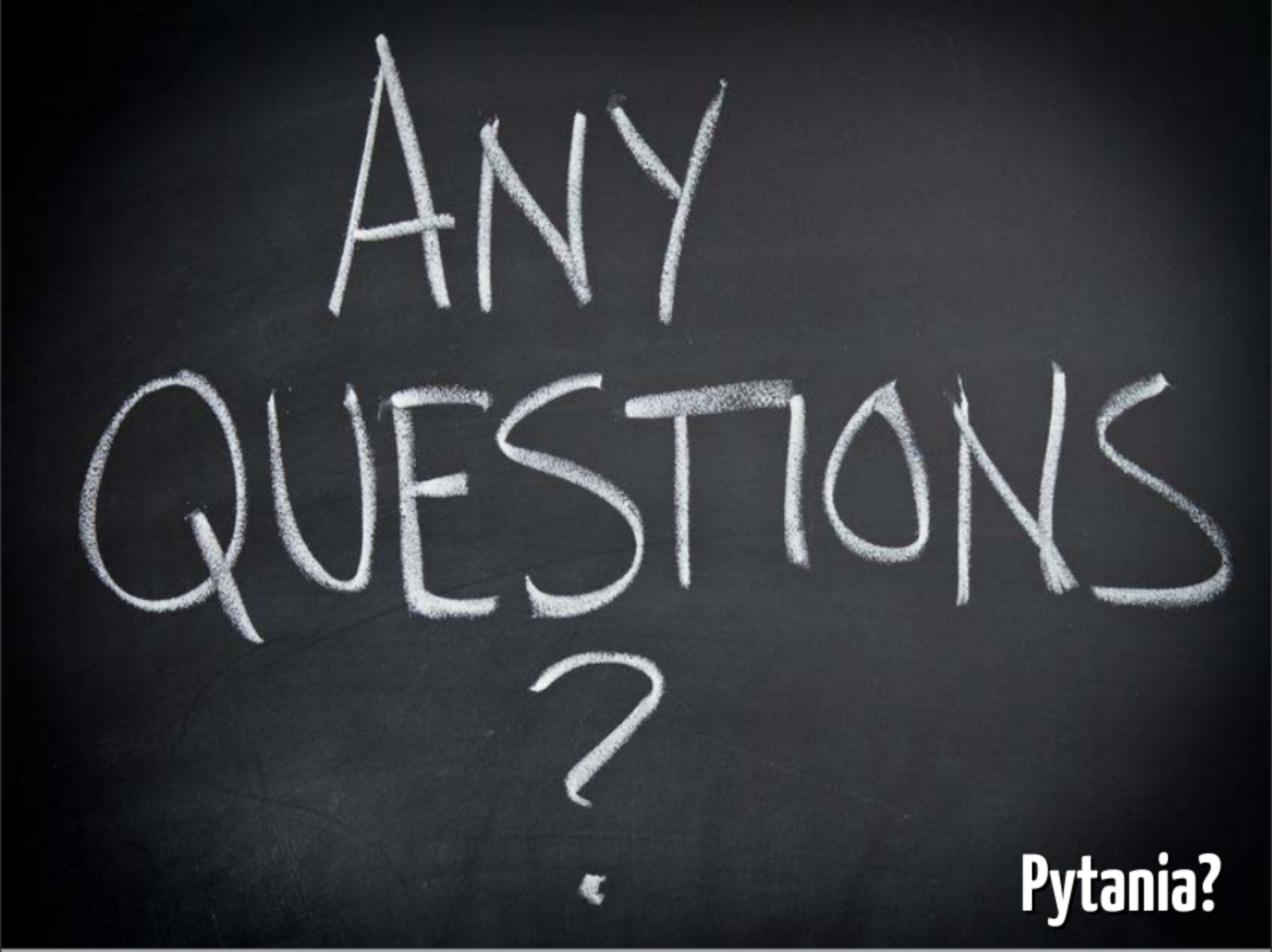
ABS(number)
SQRT(number)
MOD(number, divisor)
CURRENT_DATE
CURRENT_TIME
CURRENT_TIMESTAMP

Query execution

```
@Stateless
public class MessageStorageService {
    @PersistenceContext
    EntityManager manager;

    public void deleteAllMessagesByAuthor(String author) {
        em.createQuery ("DELETE FROM Message m WHERE m.author =
:author")
            .setParameter ("author", author)
            .executeUpdate();
    }

    public void updateMessagesAuthorName(String author, String
newName) {
        em.createQuery ("UPDATE Message m SET m.author = :newName
WHERE m.author = :author")
            .setParameter("author", author)
            .setParameter("newName", newName)
            .executeUpdate();
    }
}
```



Pytania?