Guide to 4.2 Features

Version 4.2 (4.2.M1)

Table of Contents

1. Java Version 2
2. New Features 3
    2.1. Subqueries 3
    2.2. New Property API 3

Copyright © 2011-2020 Apache Software Foundation and individual authors

License

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at https://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This guide highlights the new features and changes introduced in Apache Cayenne 4.2. For a full list of changes consult RELEASE-NOTES.txt included in Cayenne download. For release-specific upgrade instructions check UPGRADE.txt.

Chapter 1. Java Version

Minimum required JDK version is 1.8 or newer. Cayenne 4.2 is fully tested with Java 1.8 and 11.

Chapter 2. New Features

2.1. Subqueries

Expressions are now support subqueries.

```
ColumnSelect<Long> subQuery = ObjectSelect
        .columnQuery(Artist.class, Artist.ARTIST_ID_PK_PROPERTY)
        .where(...);
List<Artist> artists = ObjectSelect.query(Artist.class)
        .where(Artist.ARTIST_ID_PK_PROPERTY.in(subQuery))
        .select(context);
```

2.2. New Property API

Property API are greatly revised. This API allows to use type aware expression factories aka Properties. These properties are normally generated as static constants in model classes, but they can also be created manually by `PropertyFactory` if needed.

Typical usage in select queries:

```
Painting painting = //...
Artist artist = ObjectSelect.query(Artist.class)
        .where(Artist.PAINTING_ARRAY.contains(painting))
        .and(Artist.DATE_OF_BIRTH.year().gt(1950))
        .and(Artist.ARTIST_NAME.lower().like("pablo%"))
        .selectOne(context);
```