Annotation Guide

RELEVANCE: A SECTION is relevant to an API ELEMENT if: It would help a reader unfamiliar with the corresponding API to decide when or how to use the ELEMENT to complete a programming task;

NON RELEVANCE:

1. When the element is mentioned for **completeness in a list of elements**, without being otherwise referenced. Example:

   Introduction

   . . . In particular, we cover the usage of the key DateTime, Interval, Duration and Period classes. . .

2. When the element is mentioned just to **give more information/supplement for relevant** API type by comparison or by other ways. Example:

   Accessing Fields

   . . . For a complete reference, see the documentation for the base class AbstractReadableInstantFieldProperty. . . .

3. when the element is mentioned as an example which is not showing the use of the element. Example:

   Fields

. . . So, for instance, the 'day-of-year' calendar field would be retrieved by calling the getDayOfYear() method...

4. The information is **structural** which can be derived from Javadoc or source code. Example:

   Querying DateTimes

   int iDoW = dt.getDayOfWeek(); where iDoW can take the values (from class DateTime-Constants). *public static final int MONDAY = 1; public static final int TUESDAY = 2; ...*

5. when the information shows a **static relationship**. Example:

   Date fields

   The DateTime implementation provides a complete list of standard calendar fields: dt.getEra(), dt.getYear(), dt.getWeekyear() ...

6. when too little information is given which is can be considered as **non-information**. Example:

   Querying DateTimes

   ...For instance, the direct way to get the day of week for a particular DateTime, involves calling the method int iDoW = dt.getDayOfWeek(),

7. if **not sure** whether it is relevant, then it cannot be relevant. Example:

   Instants

   ..Within Joda-Time an instant is represented by the ReadableInstant interface...Other implementations of ReadableInstant include Instant and DateMidnight