**Joda-Time 2.4 - New Methods, Improved Concurrency and Performance**

* Joda-time 2.4 thư viện về ngày và thời gian của Java đã được phát hành. Nó được phát hành đầu tiên vào năm 2014, nó bao gồm việc cải tiến, sửa lỗi và cập nhập múi giờ. Nó được phát hành theo giấy phép của Apache Ver 2 và các yêu cầu của JDK 5+
* Những cải tiến của Joda-Tiem 2.4:
  + Các phương pháp Duration.multipliedBy(long multiplicand), Duration.dividedBy(long divisor), và Duration.negated()  đã được thêm vào giao diện chung.
  + Hash Trong LocalDate.HashCode() Các sửa đổi biến động không cần thiết lưu trữ trên hashcode(int) đã được gỡ bỏ.
  + Nội dung của Date TimeparserBucket đã được thay đổi mà không ảnh hưởng đến CharSequence của các API chung.
  + Sự hỗ trợ cho java.lang.appendable đã được thêm vào trong DateTimeFormat, DateTimeFormatter và DateTimeprinter đã được dùng thay vì phải dùng Stringbuffer.
  + Cải thiện hiệu suất thời gian của trình tự tra cứu trong các lớp của gói joda.time.chrono. Bây giờ nó sử dụng ConcurrentHashMaps thay vì Maps đồng bộ. Tra cứu bộ nhớ lưu trữ của Gjchronologies cũng được đơn giản hóa.
  + Cải thiện hiệu suất của DateTimeFormat tra cứu mô hình bộ nhớ cache bằng cách sử dụng AtomicReferenceArray.
  + Đơn giản hóa và tăng hiệu quả hoạt động tra cứu ký tự của GJLocaleSymbols..
  + Dữ liệu DateTimeZone data đã được cập nhập lên phiên bản 2014e .

## Những lỗi đã được fix trong Joda-Time 2.4:

* + Thế kỷ và thời đại đã được xử lý không chính xác trong DateTimeField.getDurationField() and DateTimeField.getRangeDurationField().
  + DateTimeZone.getoffsetFormLocal() đã được sửa chữa lại để xử lý các lịch sử cuối cùng đã bị bỏ qua trong DST
  + Trong quá trình tạo giai đoạn, Nếu một đầu trong DST chồng lên 1 đầu bên ngoài thì thời kỳ là không chính xác. Việc tính toán lại đã được sửa chữa và không còn thêm giờ được nữa
  + Phương pháp DateTimeUtils.isContiguous(ReadablePartial) có thể ném một NullpointerException khi đánh giá các phần lạ
  + UnsupportedOperationException đã được sửa lại trong Period.normalizedStandard(PeriodType) trong khi PeriodType đã bị hỏng vài năm hoặc vài tháng
  + Thêm xác nhận hỏng trong Partial.with(DateTimeFieldType, int) để ngăn chặn các phần bị lỗi được tạo ra
  + Thêm số không vào MutableDateTime để bù đắp những thay đổi trong việc chồng chéo lên nhau của DST
  + Thêm xác nhận trong DateTime/MutableDateTime có cấu trúc mili giây. Datetime(Long.Max\_Value) không còn giá trị và bị từ chối
  + Nhấn mạng year > weekyear khi tạo các phần để so sánh
  + Xử lý các TimeZone mới với ID NULL. Gọi DateTimeZone.forTimeZone() cho vào NullPointerException khi thông qua một ID null.
  + Concurrency tốt hơn trong DateTimeFormatterBuilder sử dụng phân tích cú pháp concurrentHashMap.
  + DateTimeZone được gọi không chính xác ThreadGroup.uncaughtException() và được thay đổi để ném 1 thay thế RuntimeException.