

التزامن والأقفال في قواعد البيانات باستخدام Laravel

لماذا التزامن مهم؟

عندما تصل عدة طلبات لنفس البيانات في نفس الوقت قد تحدث مشاكل مثل تكرار الدفع أو ضياع تحديثات. التحكم في التزامن يحافظ على سلامة البيانات.

مشاكل شائعة

- ضياع التحديثات: تحديثان على نفس السجل ويتفوق أحدهما على الآخر.
- قراءة بيانات غير معتمدة.
- نتائج غير ثابتة داخل نفس العملية.

لتزامن أدوات Laravel

1) المعاملات (Transactions)

تنفيذ العمليات كوحدة واحدة.

```
DB::transaction(function () {
    // read
    // write
});
```

2) الأقفال المتشائمة (For Update)

قفل الصفوف أثناء التحديث.

```
DB::transaction(function () {
    $wallet = DB::table('wallets')
        ->where('id', $id)
        ->lockForUpdate()
        ->first();

    // update safely
});
```

3) الأقفال المشتركة

قراءة مسموحة مع منع الكتابة.

```
DB::transaction(function () {
    $report = DB::table('reports')
        ->where('id', $id)
        ->sharedLock()
```

```
->first();  
});
```

(الأقفال المترافقية (يدويًا (4)

والتحقق منه قبل التحديث `version` إضافة عمود

```
$updated = DB::table('orders')  
->where('id', $id)  
->where('version', $version)  
->update([  
    'status' => 'paid',  
    'version' => $version + 1,  
]);  
  
if ($updated === 0) {  
    حدث تعديل آخر، أعد المحاولة //  
}
```

نصائح عملية

- أي تغييرات مالية يجب أن تكون داخل transaction.
- عندما تحتاج كاتباً واحداً فقط (`lockForUpdate()`) استخدم
- اجعل المعاملات قصيرة لتجنب الأقفال الطويلة.

الخلاصة

يوفر معاملات وأقفال على مستوى الصنف. استخدمها لضمان سلامة البيانات Laravel. التزامن ضروري في الإنتاج.