

هما اثنين من الأدوات المستخدمة للتفاعل مع قاعدة Query Builder و Eloquent ORM ، Laravel في البيانات. ومع ذلك، هناك فرق بينهما

Eloquent ORM:

، والتي تتيح لك تمثيل الجداول والعلاقات في شكل كائنات Laravel هو طريقة تفاعلية للتعامل مع قاعدة البيانات في إنشاء علاقات بين الجداول والتعامل معها بسهولة، وتنفيذ العديد من العمليات Eloquent ORM برمجية. يتيح لك مثل الإدخال والتحديث والحذف والاستعلام بطريقة سلسلة وسهلة الاستخدام.

Query Builder:

برمجيًا. يمكن استخدامها لبناء استعلامات معقدة SQL التي تسمح للمطورين بإنشاء استعلامات Laravel هي أداة بطريقة سهلة ومرنة، وتنفيذ العديد من العمليات مثل الإدخال والتحديث والحذف والاستعلام. يمكن استخدامها للتعامل مع الجداول والبيانات بشكل عام بطريقة برمجية.

بطريقة مرنة وبسيطة، حيث Laravel للتعامل مع البيانات في Eloquent ORM بشكل عام، يمكن استخدام يمكنك تمثيل الجداول والعلاقات بينها في شكل كائنات برمجية، والتعامل معها باستخدام العديد من الدوال والميثودز بشكل برمجي ومرنة، وتنفيذ العديد SQL لبناء استعلامات Query Builder المتاحة. بالمقابل، يمكن استخدام من العمليات على البيانات بطريقة برمجية.

يسهل عليك تمثيل العلاقات بين الجداول وتعامل Eloquent ORM وبالنسبة للاستفادة من كل منهما، فإن استخدام معقدة بطريقة SQL لبناء استعلامات Query Builder معها بطريقة أكثر مرونة وسهولة، بينما يمكن استخدام مرنة وسهلة الاستخدام. في النهاية، يعتمد الخيار الأنسب على نوع المشروع واحتياجاته الخاصة.

قد تكون أسرع في بعض الحالات DB على الرغم من أن واجهة Laravel Eloquent هناك عدة أسباب لاستخدام

1. السهولة في الاستخدام:

، حيث يمكن للمطورين العمل DB يوفر واجهة برمجية أكثر سهولة في الاستخدام بالمقارنة مع واجهة Eloquent SQL لأنه يستخدم تمثيلًا كائنيًا للجداول والعلاقات بدلاً من استخدام كود Eloquent ORM بشكل أسهل مع الصلب. وهذا يجعل العملية أكثر مرونة وسهولة في الصيانة والتطوير.

2. القابلية للتعديل:

يسمح للمطورين بإجراء تعديلات أكثر سهولة على نموذج البيانات، حيث يمكن تغيير الخصائص Eloquent صلب SQL والعلاقات بسهولة بدون الحاجة إلى تغيير أي كود.

3. التعامل مع العلاقات بشكل أفضل:

(Models) يتيح للمطورين التعامل مع العلاقات بشكل أكثر سهولة ومرونة من خلال استخدام النماذج Eloquent. والتعامل مع العلاقات بينها وبين الجداول بشكل أسهل.

4. المزايا الإضافية:

يوفر العديد من المزايا الإضافية مثل التحكم في العلاقات، والتحقق من صحة البيانات، والتحكم في Eloquent الحقول الذي يجعلها خيارًا أفضل للتطبيقات الكبيرة والمعقدة.

قد تكون أسرع في DB على الرغم من أن واجهة Eloquent Laravel وبشكل عام، يمكن أن يستخدم المطورون بعض الحالات، لأنه يوفر واجهة برمجية أكثر سهولة في الاستخدام والتعديل، ويتيح التعامل مع العلاقات بشكل أفضل، ويوفر المزايا الإضافية التي تجعلها خيارًا أفضل للتطبيقات الكبيرة والمعقدة.