الموضوع الثالث

هيكل مشروع لارافل

بعد أن أنشأنا مشروع لارافل على desktop ، فلنقم الآن باستعراض مجلدات و ملفات لارافل .

اختر أي محرر للتعليمات البرمجية ، سأستخدم vs code ،

من **file** ← **open folder** ثم حدد مجلد المشروع

على يسار النافذة تظهر جميع المجلدات الرئيسية للمشروع ، سنتعرف عليها واحداً تلو الآخر ، لكن قبل ذلك دعنا نقوم بتشغيل المشروع ، و يتم ذلك بطريقة جداً سهلة ، قم بفتح موجه الأوامر ثم انتقل إلى مجلد المشروع عن طريق الأمر cd ، أو إذا كنت تعمل على ويندوز انقر بالزر الأيمن للماوس على مجلد المشروع مع الضغط باستمرار على **مفتاح shift** ← اختر **فتح نافذة الأوامر هنا** ، ستجد أنه انتقل مباشرة إلى دليل مشروع لارافل ، الآن سنستخدم ما يسمى بالـ artisan لتشغيل المشروع و هي واجهة سطر الأوامر مدرجة مع لارافل. توفر عدداً من الأوامر المفيدة التي تساعدك أثناء بناء التطبيق الخاص بك ، يمكنك استعراض جميع الأوامر عن طريق الأمر php artisan ، مثلاً ستجد الأمر artisan serve و مهمته تشغيل السيرفر الخاص بالمشروع ، فلنقم باستخدام هذا الأمر بالشكل التالي

Php artisan serve

سيقوم بتشغيله على البورت ٨٠٠٠ بشكل إفتراضي ، الآن سأكتب http://localhost:8000/ على المتصفح و ستعرض الصفحة الرئيسية مباشرة .

فلنلقي نظرة الآن على أهم الملفات و المجلدات في مشروع لارافل :

المجلد app: هو أحد المجلدات الرئيسية في لارافل ، يحوي التعليمات البرمجية الأساسية للتطبيق ، يشمل معظم كلاسات التطبيق كالـmodel و Controllers

يتفرع من هذا المجلد أربع مجلدات أهمها المجلد HTTP و الذي يتضمن أيضاً مجلدين فرعيين : Controllers سيضم جميع المتحكمات في التطبيق

و الـ Middleware المسؤول عن ترشيح طلبات HTTP الواردة للتطبيق

يضم الدليل app أيضاً مجلد console و الذي يتضمن أوامر الـ artisan ، حيث يمكننا هنا إنشاء artisan commands مخصصة .

ربما تتساءل الآن أين المجلد الخاص بالـmodel ، الـmodels يقع في جذر الـ App مباشرةً ، كما ترى هنا الكلاس User و هو model يمثل جدول الـusers ، و أي نموذج (model) ستقوم بإنشاءه سيظهر تلقائياً على نفس هذا المسار .

لاحظ artisan:make ، و بالتحديد عن طريق الأمر artisanيتم إنشاء العديد من الملفات في تطبيق لارافل عن طريق أوامر ال سنكتب هذا السطر على موجه الأوامر **controller** هذه قائمة بجميع الأوامر المتاحة ، فمثلاً إذا أردنا إنشاء ملف يمثل متحكم

php artisan make:controller Controllername

كما ترى أنشيء الملف مباشرة بداخل مجلد controller .

مجلد bootstrap : يحوي الملف app.php الذي يقوم بتهيئة الإطار ، كما يتفرع منه المجلد cache الذي يحتوي على ملفات يتم إنشاؤها لتحسين الأداء .

المجلد config : يحوي كافة ملفات تهيئة و إعداد التطبيق ، كالمصادقة(authentification)، التخزين المؤقت(cashe)،قواعد البيانات ، (sessions) .

المجلد database : يحتوي ملفات التهجير migration (و كما ترى فهو وصف لحقول جداول قاعدة البيانات ، و يتم إنشاء ملفات الmigration باستخدام أوامر artisan) ، أيضاً يتضمن ملفات البذر seed و عمله إدراج بيانات وهمية لقاعدة البيانات لغرض الإختبار .

المجلد Public : يتضمن هذا المجلد الملف index.php و الذي يعتبر نقطة الدخول لجميع الطلبات الداخلة للتطبيق ، كما يحتوي هذا المجلد على الملفات المتعلقة بالعرض مثل ملفات css و javascript و الصور حيث يستطيع المتصفح الوصول للملفات في هذ المجلد بشكل مباشر .

المجلد Resources : يتفرع منه ثلاث مجلدات **assests** و يحوي الملفات التي تتطلب تجميع مثل ملفات LESS,SASS ، **view** و يتضمن ملفات العرض ، و**lang** الخاص باللغة ، و كما ترى يوجد بداخله مجلد واحد يمثل النصوص باللغة الإنجليزية ، و إذا أردنا إضافة ترجمة أخرى و لتكن العربية سنضع مجلد آخر بالرمز ar .

المجلد routes : يحتوي جميع المسارات الخاصة بالتطبيق ، و يتضمن أربع ملفات web.php ، api.php ، console.php ، channels.php **(سنتحدث عنه بشكل أكثر تفصيلاً في المحاضرة القادمة)**

storage : يحتوي على الملفات المجمعة compiled كـ ملفات الجلسات sessions، ملفات التخزين المؤقت ، وغيرها من الملفات المنشأة ديناميكياً بواسطة الإطار ، و المجلد public المتفرع من app متاح لتخزين الملفات التي يرفعها المستخدم مثل صورة الملف الشخصي .

المجلد tests : خاص بإنشاء وحدات إختبار التطبيق (unit testing).

المجلد vendor : يحوي اعتمادات الـcomposer ذات العلاقة بالتطبيق .

هناك أيضاً بعض الملفات التي تقع ضمن الدليل الجذر للمشروع

كالملف env. و بمجرد النظر ستعرف أنه من خلاله يتم ضبط إعدادت التطبيق ، فمثلاً من هنا تحدد اسم قاعدة البيانات التي تتعامل معها مع اسم المستخدم و كلمة المرور .

من المهم أيضاً أن تتعرف على الملف composer.json

هذا الملف يتضمن قائمة بالمكتبات و الحزم التي يحتاجها مشروعك ، و إذا أردت استخدام مكتبة جديدة في مشروعك فما عليك سوى إضافة اسمها في هذا الملف ثم تنفيذ الأمر composer install ، و سيبدأ بتحميلها مباشرة إلى المجلد vendor ، عند الإنتهاء من تحميلها سيقوم بكتابة أسماء هذه الحزم (التي ثبتت فعلياً) مع الإصدارات الدقيقة منها إلى ملف composer.lock ، و يقفل الملف على الإصدارات التي تم تحميلها.

حسناً ماذا ظهرت تحديثات جيدة للحزم المذكورة في composer.json ، هل تنفيذ الأمر composer install سيعمل على تنصيب آخر التحديثات لها ؟

الإجابة هي **لا** ، لأن الأمر composer install سيستدعي فقط الإصدارات المحددة على composer.lock ، و هي نفس الإصدارات التي قمت بتثبيتها في أول مرة .

و في حالة ما أردت تحديث composer.lock بآخر الإصدارات ، قم بتنفيذ الأمر composer update ، سيعمل هذا على تحديث ملف القفل composer.lock باستخدام الإصدارات الجديدة .

المصادر:

https://getcomposer.org/doc/01-basic-usage.md

https://laravel.com/docs/5.5/structure#the-storage-directory