

من المتصفح إلى حين HTTP عندما يتلقى طلبًا Laravel عبارة عن عملية تتبعها إطار العمل Request cycle life :تجاوز الرد على هذا الطلب، تتكون تلك الدورة من عدة مراحل وهي

Apache (مثل) من المتصفح عن طريق خادم الويب HTTP بتلقي الطلب Laravel تلقي الطلب: تبدأ دورة حياة الطلب في الذي يعمل كنقطة index.php وملف PHP باستخدام نواة Laravel يتم تحويل الطلب إلى إطار العمل (Nginx أو). دخول للتطبيق.

التوجيه: بعد تلقي الطلب، يتم تحديد العملية أو الوظيفة التي يجب تشغيلها للرد على الطلب. يتم ذلك باستخدام المسار المحدد routes/web.php في Laravel المرسل من المتصفح. يتم تحديد الوظيفة باستخدام ملف التوجيه URL في عنوان

التحقق: بعد تحديد العملية أو الوظيفة، يتم التحقق من صحة الطلب. يتم التحقق من صحة المدخلات المرسل في الطلب، ومن حقوق الوصول للمستخدم الذي يقوم بالطلب، ومن أي شروط أخرى يجب توافرها لإتمام العملية بنجاح.

التنفيذ: بعد التحقق، يتم تنفيذ العملية أو الوظيفة المحددة. يمكن أن تشمل هذه العملية أي شيء من استرجاع البيانات من قاعدة البيانات، إلى تحديث الملفات أو إرسال رسائل البريد الإلكتروني، أو أي نوع من العمليات الأخرى.

الاستجابة: بعد تنفيذ العملية بنجاح، يتم إرسال الاستجابة إلى المتصفح. يتم إنشاء الاستجابة باستخدام النموذج الخاص بـ تتم عرضها على المتصفح. يتم إرسال الاستجابة HTML للعرض والإخراج. يتم عادةً إنشاء الاستجابة في شكل صفحة Laravel لتحديد ما إذا كان الطلب قد تم معالجته HTTP وتتضمن الاستجابة عادةً رمز حالة HTTP إلى المتصفح باستخدام بروتوكول بنجاح أم لا، ورسالة نصية توضح نتيجة الطلب.

وتتوقف العملية. يتم إغلاق جميع الاتصالات Laravel النهائية: بعد إرسال الاستجابة إلى المتصفح، تنتهي دورة حياة الطلب في النشطة مع قاعدة البيانات وملفات النظام وغيرها، ويتم إعادة تعيين المتغيرات والكائنات المحددة في ذاكرة الوصول العشوائي (RAM) إلى القيم الافتراضية.

يمكن تعديلها وتخصيصها باستخدام العديد من الأدوات والوظائف المتاحة في Laravel يجب ملاحظة أن دورة حياة الطلب في ويمكن أيضًا إنشاء مراحل إضافية وفقًا لاحتياجات التطبيق المحدد Laravel.

Laravel: مثال بسيط يوضح دورة حياة الطلب في

Laravel مثال بسيط عن دورة حياة الطلب في //

```
// استقبال الطلب
Route::get('/users', function () {
    // 2. الكود المعالج للطلب
    $users = DB::table('users')->get();

    // 3. إرسال الاستجابة
    return view('users', ['users' => $users]);
});
```

// 4. الإنشاء: تحديد المسار والعرض المستخدم للإدخال

// 5. المعالجة: تنفيذ العملية وإنشاء الاستجابة

// 6. النهاية: توقف العملية وإغلاق جميع الاتصالات النشطة

ومعالجته عن طريق الحصول على جميع المستخدمين من قاعدة البيانات users/ على GET في المثال، يتم استقبال طلب . ثم إنشاء الاستجابة وإرسالها إلى المتصفح، وتنتهي العملية ويتم إغلاق جميع الاتصالات النشطة users وإرسالهم إلى عرض