1. Retrieving Boolean Input Values

When dealing with HTML elements like checkboxes, your application may receive "truthy" values that are actually strings. For example, "true" or "on". For convenience, you may use the boolean method to retrieve these values as booleans. The boolean method returns true for 1, "1", true, "true", "on", and "yes". All other values will return false:

$archived = $request->boolean('archived');

\_2Retrieving Boolean Input Values only /except

If you need to retrieve a subset of the input data, you may use the only and except methods. Both of these methods accept a single array or a dynamic list of arguments:

$input = $request->only([‘username’, ‘password’]);

$input = $request->only(‘username’, ‘password’);

$input = $request->except([‘credit\_card’]);

$input = $request->except(‘credit\_card’);

The only method returns all of the key / value pairs that you request; however, it will not return key / value pairs that are not present on the request.

3\_3Determining If Input Is Present

You may use the has method to determine if a value is present on the request. The has method returns true if the value is present on the request:

if ($request->has('name')) {

// ...

}

When given an array, the has method will determine if all of the specified values are present:

if ($request->has(['name', 'email'])) {

// ...

}

The hasAny method returns true if any of the specified values are present:

if ($request->hasAny(['name', 'email'])) {

// ...

}

The whenHas method will execute the given closure if a value is present on the request:

$request->whenHas('name', function (string $input) {

// ...

});

A second closure may be passed to the whenHas method that will be executed if the specified value is not present on the request:

$request->whenHas('name', function (string $input) {

// The "name" value is present...

}, function () {

// The "name" value is not present...

});

If you would like to determine if a value is present on the request and is not an empty string, you may use the filled method:

if ($request->filled('name')) {

// ...

}

The anyFilled method returns true if any of the specified values is not an empty string:

if ($request->anyFilled(['name', 'email'])) {

// ...

}

The whenFilled method will execute the given closure if a value is present on the request and is not an empty string:

$request->whenFilled('name', function (string $input) {

// ...

});

A second closure may be passed to the whenFilled method that will be executed if the specified value is not "filled":

$request->whenFilled('name', function (string $input) {

// The "name" value is filled...

}, function () {

// The "name" value is not filled...

});

To determine if a given key is absent from the request, you may use the missing and whenMissing methods:

if ($request->missing('name')) {

// ...

}

$request->whenMissing('name', function (array $input) {

// The "name" value is missing...

}, function () {

// The "name" value is present...

});

What is request life cycle in laravel ???

دورة حياة الطلب (Request Life Cycle) تشير إلى سلسلة الخطوات والعمليات التي يمر بها طلب من البداية إلى النهاية. يتم استخدام هذا المصطلح بشكل عام في عالم تطوير البرمجيات وتكنولوجيا المعلومات.

تبدأ دورة حياة الطلب عندما يتم إرسال طلب من جهاز العميل (مثل المتصفح) إلى الخادم. يمر الطلب بمراحل مختلفة أثناء معالجته واستجابته من الخادم. هنا هي المراحل الرئيسية في دورة حياة الطلب:

1. إرسال الطلب (Request Dispatch): يتم إرسال الطلب من العميل إلى الخادم. يحتوي الطلب على معلومات مثل نوع الطلب (GET أو POST) وعنوان الصفحة المستهدفة وبيانات الطلب الإضافية إن وجدت.

2. استقبال الطلب (Request Reception): يتم استقبال الطلب في الخادم وفحصه للتحقق من صحته وانتقاله للمرحلة التالية.

3. معالجة الطلب (Request Processing): يتم معالجة الطلب في الخادم. يتضمن هذا المرحلة قراءة بيانات الطلب والتعامل معها وتنفيذ العمليات المطلوبة، مثل الاستعلام عن قاعدة البيانات أو تنفيذ الأكواد المخصصة.

4. إنشاء الاستجابة (Response Generation): بعد معالجة الطلب، يتم إنشاء استجابة من الخادم للعميل. تتضمن الاستجابة معلومات مثل رمز الحالة (مثل 200 للنجاح أو 404 للصفحة غير موجودة) والبيانات المطلوبة لعرضها في العميل.

5. إرسال الاستجابة (Response Dispatch): ترسل الاستجابة من الخادم إلى العميل. يتم توجيه الاستجابة إلى العميل الصحيح الذي أرسل الطلب الأصلي.

6. عرض الاستجابة (Response Rendering): يقوم العميل بعرض الاستجابة التي تلقاها من الخادم. يتم عرض المحتوى المطلوب، مثل الصفحة الويب أو البيانات المستردة، للمستخدم النهائي.