INSTITUTO DO EMPREGO
E FORMAÇÃO PROFISSIONAL

AEP

cesae digital

Centro para o Desenvolvimento
de Competências Digitais

Software Developer

PHP / Laravel

Sara Monteiro

API

cesae digital
Centro para o Desenvolvimento de Competências Digitais

- Para finalizar o estudo do Laravel, iremos usar o recurso de criar uma API e faremos a mesma usando a arquitectura RESTful.

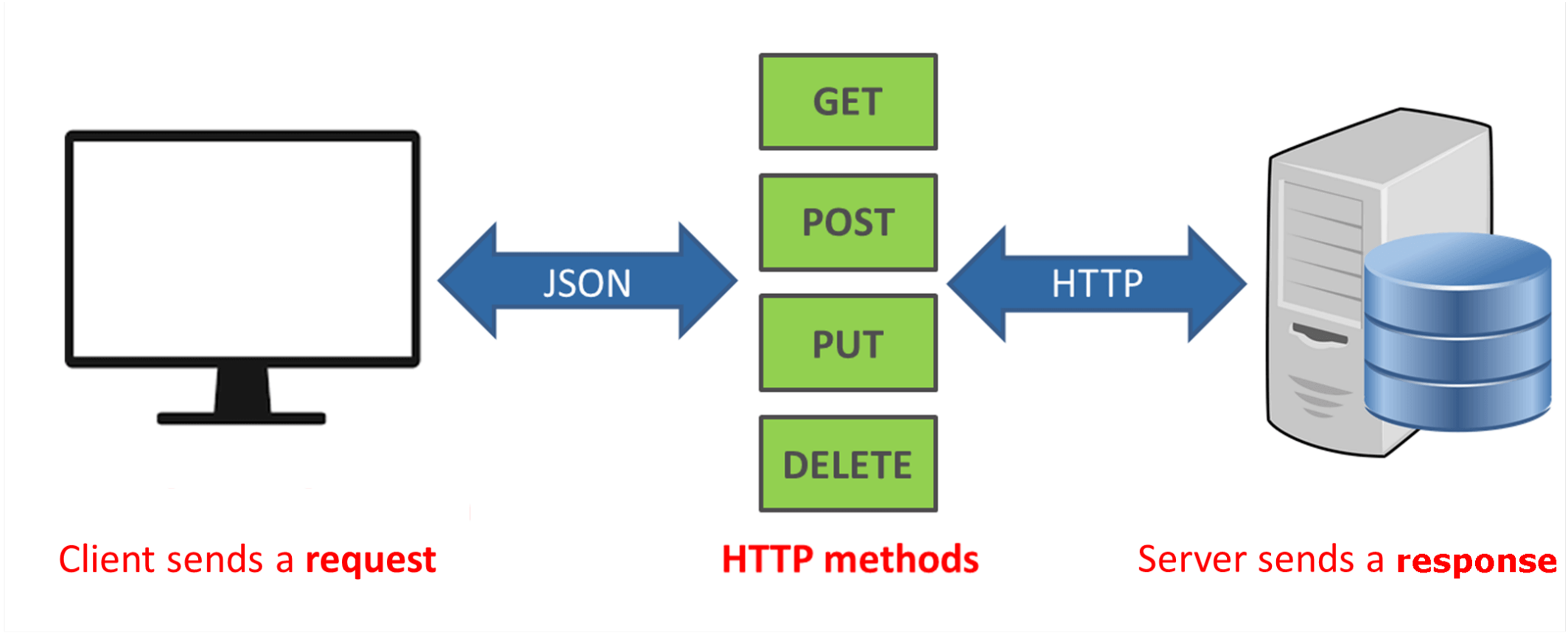
- Uma API RESTful (Representational State Transfer) é um estilo que usa solicitações HTTP para acesso e uso de dados. Esses dados podem ser usados para os tipos de dados GET, PUT, POST e DELETE e referem-se à leitura, atualização, criação e exclusão de operações relativas a recursos.

- Temos ainda o protocolo de acesso a objetos simples (SOAP), mantido pelo World Wide Web Consortium (W3C).

- A principal diferença é que SOAP é um protocolo enquanto REST é um conjunto de princípios de arquitectura.

API

cesae digital
Centro para o Desenvolvimento de Competências Digitais

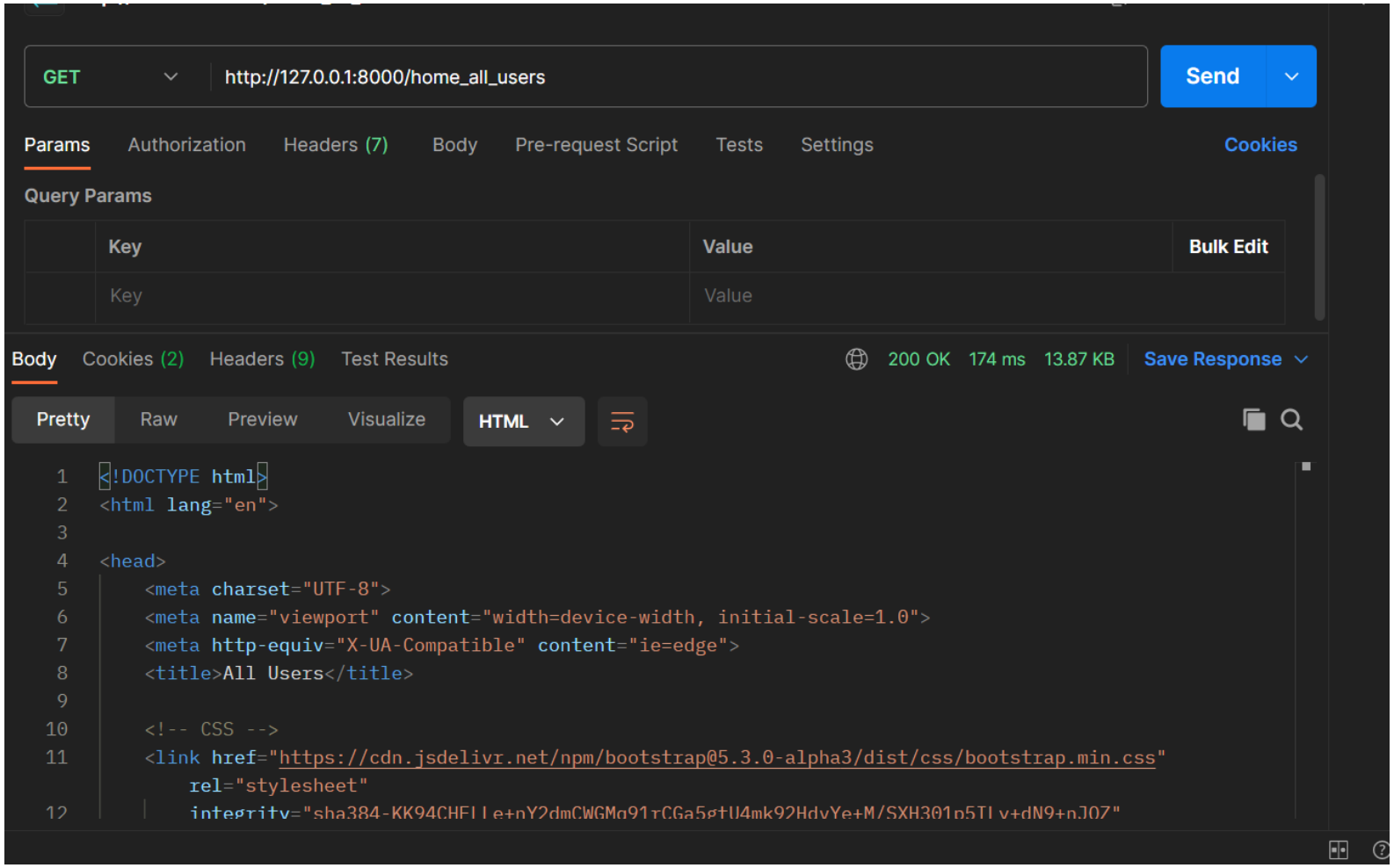


API – Postman

cesae digital
Centro para o Desenvolvimento de Competências Digitais

- O melhor recurso para testar API's é o Postman e pode ser instalado através do site oficial.

- Se testarmos uma das nossas rotas podemos verificar que nos retorna o Body tal como o browser o recebe.



Construir uma API – Show

cesae digital
Centro para o Desenvolvimento de Competências Digitais

```
api.php M
routes > api.php
12    | Here is where you can register API routes for your application. T
13    | routes are loaded by the RouteServiceProvider and all of them wil
14    | be assigned to the "api" middleware group. Make something great!
15    |
16    */
17
18    Route::get('/task/{task}', [TaskAPIController::class, 'show']);
19
```

Iremos então construir uma API com a tabela de Tasks que já temos. Para tal, criaremos um novo controller com recursos chamado TasksAPIController e o modelo da tabela, que se chamará Task.

Em seguida, é seguir o processo de retornar uma task registando a rota no ficheiro das API's.

```
/**
 * Display the specified resource.
 */
public function show(Task $task)
{
    return $task;
}
```

API em Laravel

cesae digital
Centro para o Desenvolvimento de Competências Digitais

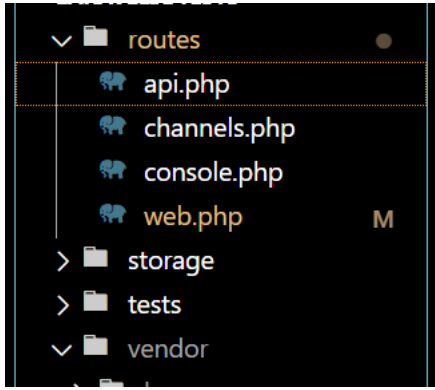
RouteServiceProvider.php

app > Providers > RouteServiceProvider.php > RouteServiceProvider > configureRateLimiti

```php
 */
public const HOME = '/home';

/**
 * Define your route model bindings, pattern filters, and other r
 */
public function boot(): void
{
    $this->configureRateLimiting();

    $this->routes(function () {
        Route::middleware('api')
            ->prefix('api')
            ->group(base_path('routes/api.php'));

        Route::middleware('web')
            ->group(base_path('routes/web.php'));
    });
}
```

No nosso RouteServiceProvider existe já um mapeamento para rotas de API construídas com um prefixo API.

Na nossa pasta de routes existe também um ficheiro para registar as rotas de API.

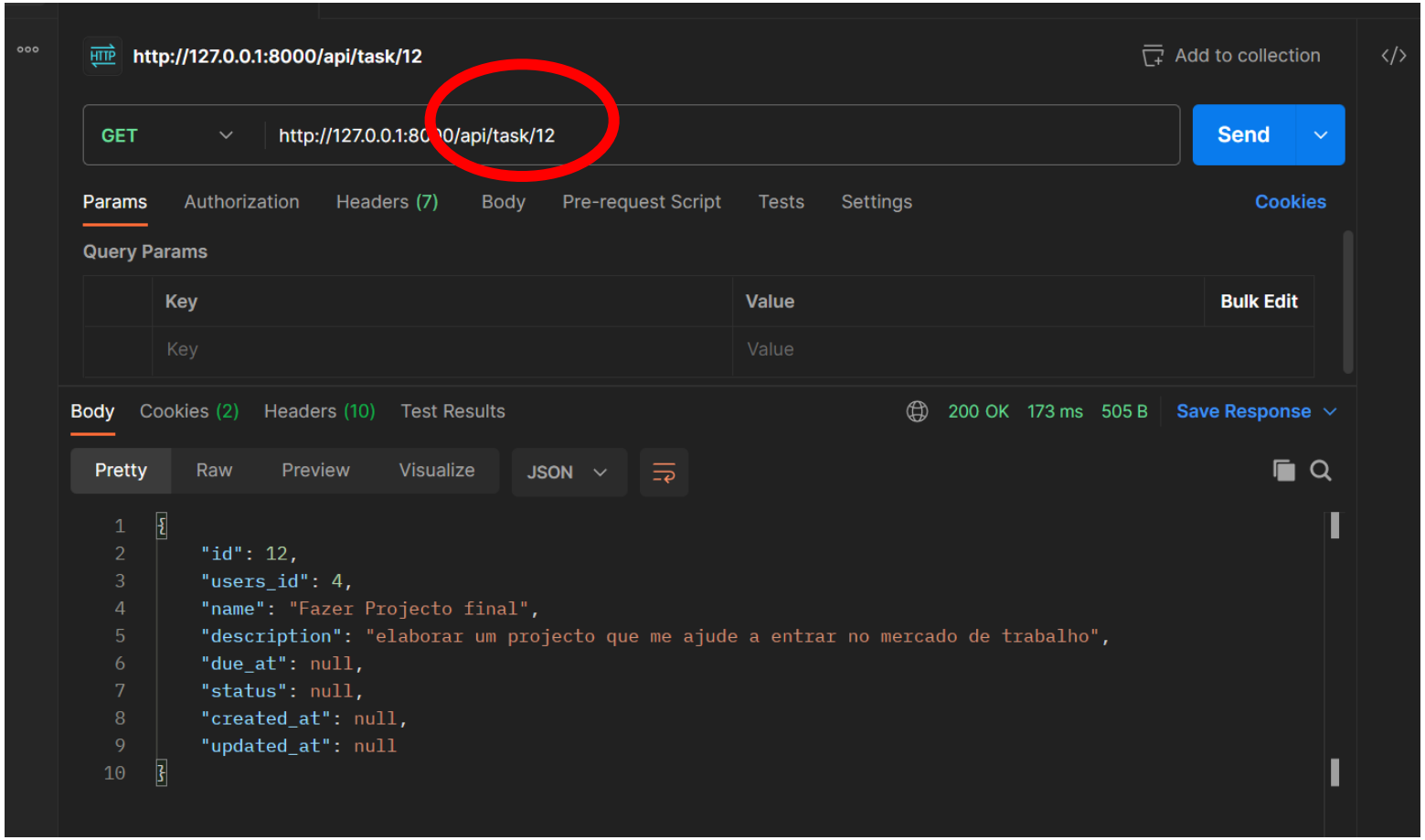



```php
<?php       You, 4 weeks ago • first comm

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|--------------------------------------
| API Routes
|--------------------------------------
|
```

Construir uma API – Show

cesae digital
Centro para o Desenvolvimento de Competências Digitais

Uma vez que a nossa função show está associada ao Model, basta trocar o Id e ele irá retornar a tarefa em questão.



Construir uma API – Recurso Customizado

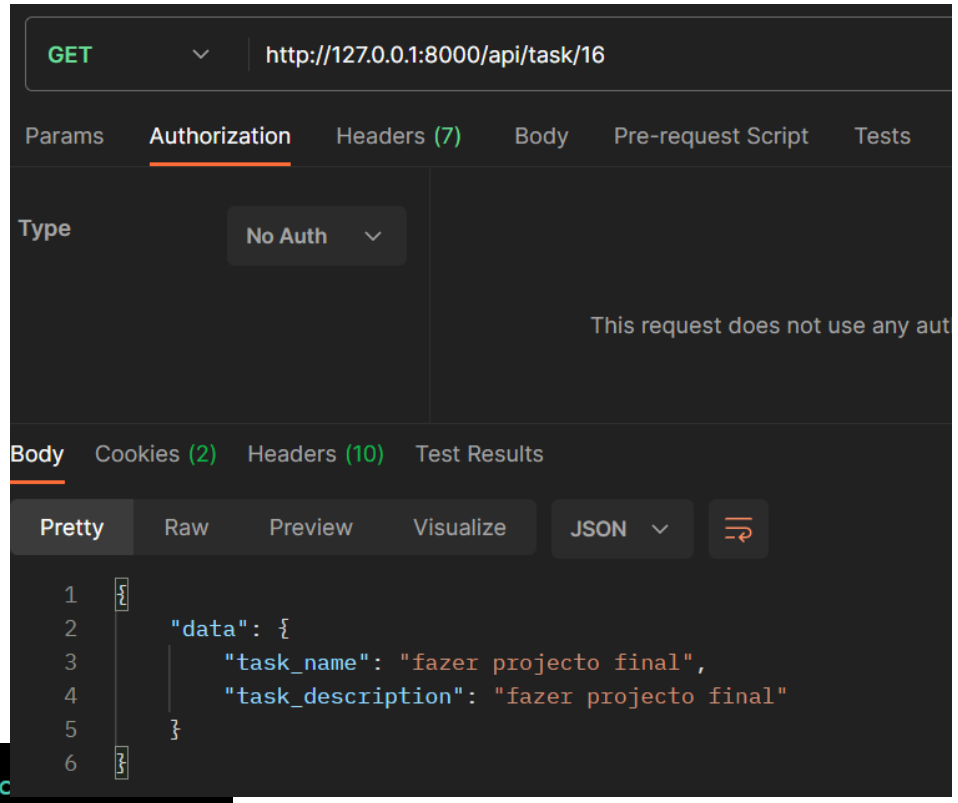
cesae digital
Centro para o Desenvolvimento de Competências Digitais

Podemos customizar a maneira como recebemos os dados da nossa BD e os enviamos na nossa Api. Para tal, iremos construir um recurso para as Tasks através de php artisan make:resource TaskResource.

Na função devemos adicionar o novo Recurso e ele passa a retornar o objecto JSON dentro de um array data apenas com os campos que queremos.

```php
 */
public function toArray(Request $request): array
{
    // return parent::toArray($request);

    return [
        'task_name' => $this->name,
        'task_description' => $this->description,
    ];
}
```

```php
public function show(Task $task): TaskResourc
{
    return new TaskResource($task);
}
```



Construir uma API – Recurso API em Laravel

cesae digital
Centro para o Desenvolvimento de Competências Digitais

O Laravel dispõe de um [recurso de API's chamado apiResource] que nos permite com um único registo nas rotas criar todas as rotas para os métodos da API.

```
Route::apiResource('/task', TaskAPIController::class);
```

Correndo a lista das rotas vemos que todas foram adicionadas à função correspondente.

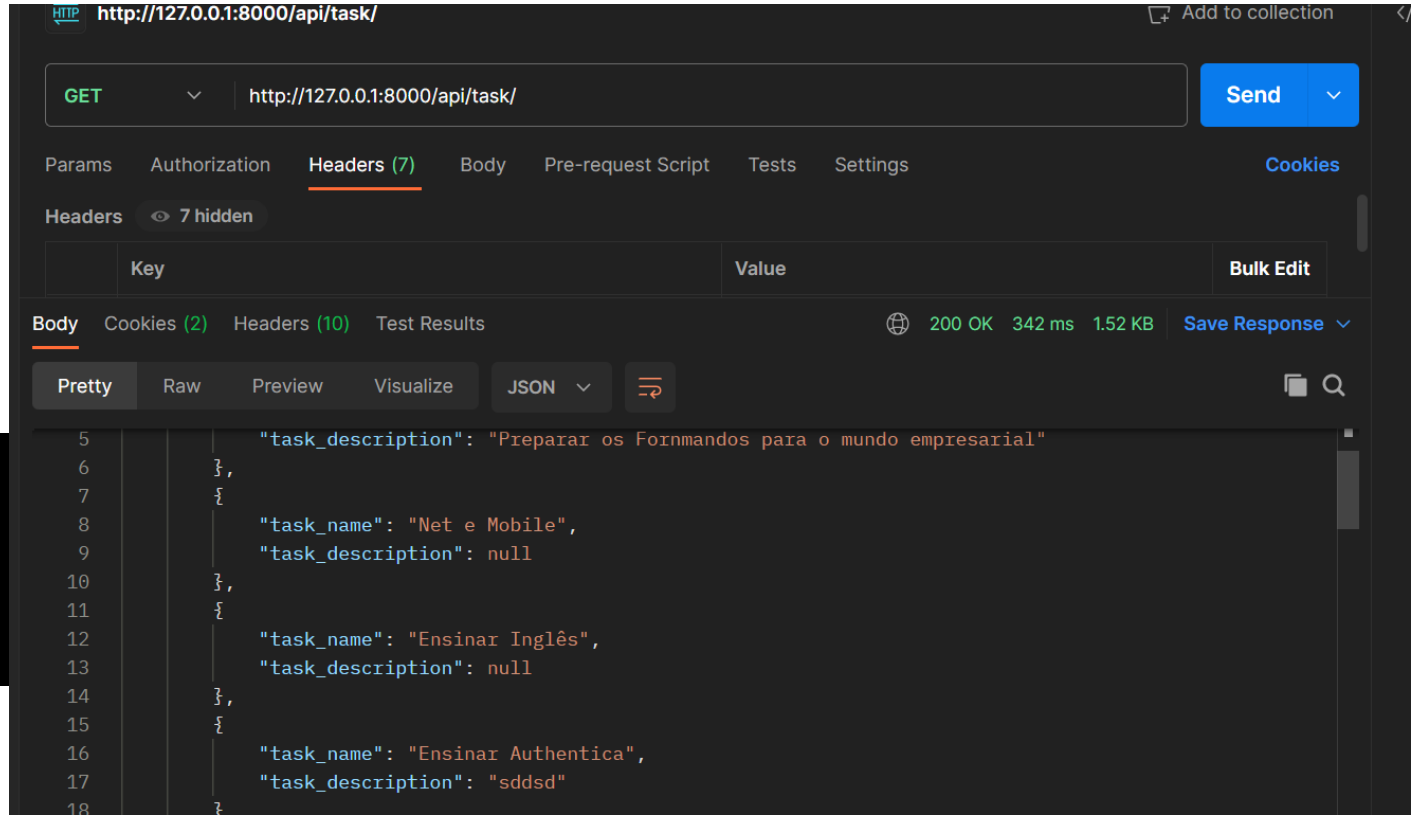
```
POST      _ignition/update-config .. Ignition.updateConfig › Spatie\LaravelIgnition › UpdateConfigController
GET|HEAD  api/task/{task} ..... {task}.index › TaskAPIController@index
POST      api/task/{task} ..... {task}.store › TaskAPIController@store
GET|HEAD  api/task/{task}/{{task}} ..... {task}.show › TaskAPIController@show
PUT|PATCH api/task/{task}/{{task}} ..... {task}.update › TaskAPIController@update
DELETE    api/task/{task}/{{task}} ..... {task}.destroy › TaskAPIController@destroy
POST      create_task ..... create_task › UserController@storeTask
```

Construir uma API – Index

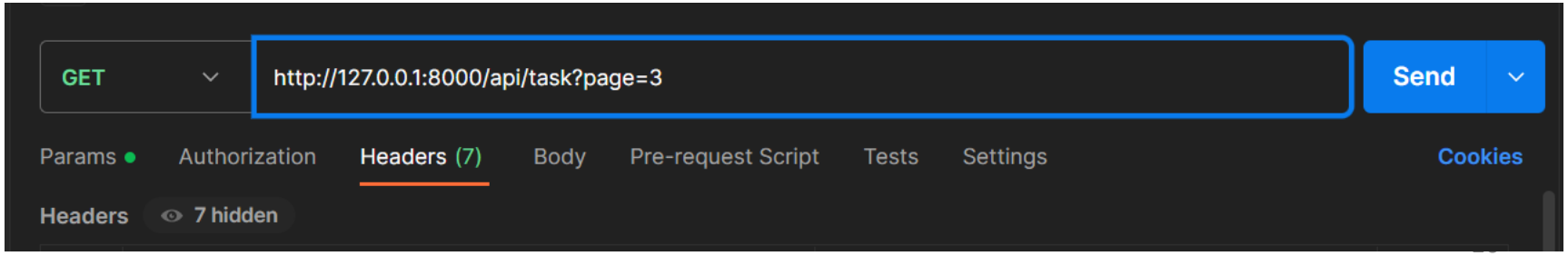
cesae digital
Centro para o Desenvolvimento de Competências Digitais

Para retornar todas as tasks iremos criar uma colecção através de php artisan make:resource TaskResourceCollection –collection.

```php
 */
public function index(): TaskResourceCollection
{
    return new TaskResourceCollection(resource: Task::paginate());
}

/**
 * Show the form for creating a new resource
```



O método das colecções paginate() recebe uma linha da BD por página.



Construir uma API – Store

cesae digital
Centro para o Desenvolvimento de Competências Digitais

```php
use Illuminate\Database\Eloquent\Model;

class Task extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'description',
        'users_id',
    ];
}
```

A primeira coisa a fazer é no Model Task adicionar os campos que têm que ser preenchidos.

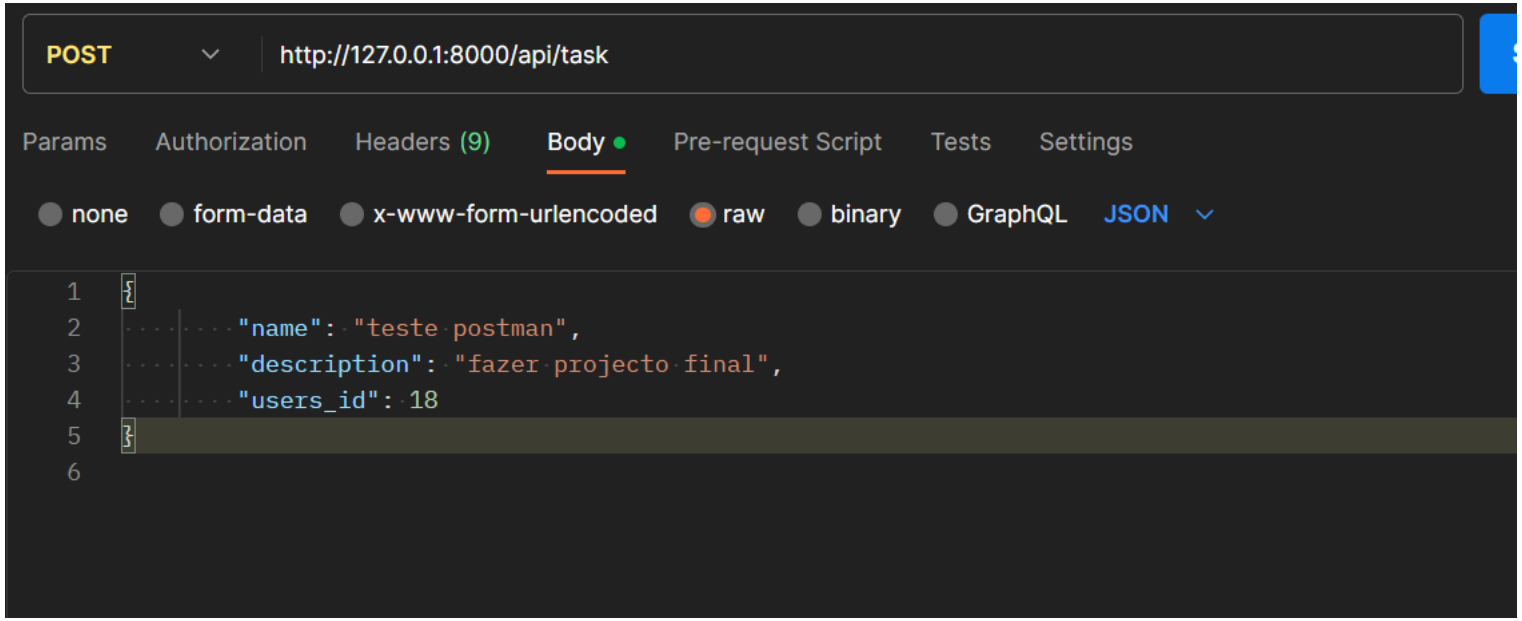
Em seguida editamos o store para criar uma nova task com a validação e criação de um novo recurso Task.

```php
 */
public function store(Request $request)
{

    $request->validate([
        'name' => 'required',
        'description' => 'required',
        'users_id' => 'required',
    ]);

    $task = Task::create($request->all());

    return new TaskResource($task);
}

/**
```

Construir uma API – Store

cesae digital
Centro para o Desenvolvimento de Competências Digitais

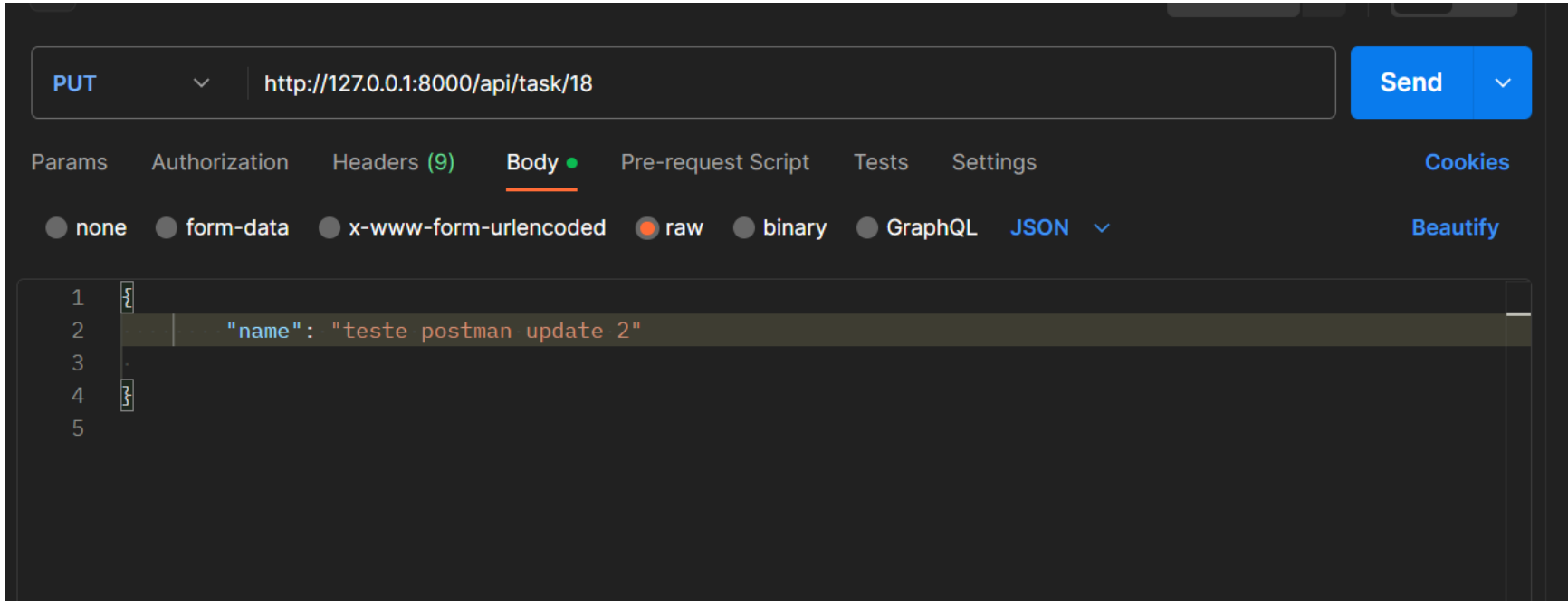
Por fim, no postman enviamos em JSON o recurso que queremos criar.
Não esquecer que o método é o POST.



Construir uma API – Update

cesae digital
Centro para o Desenvolvimento de Competências Digitais

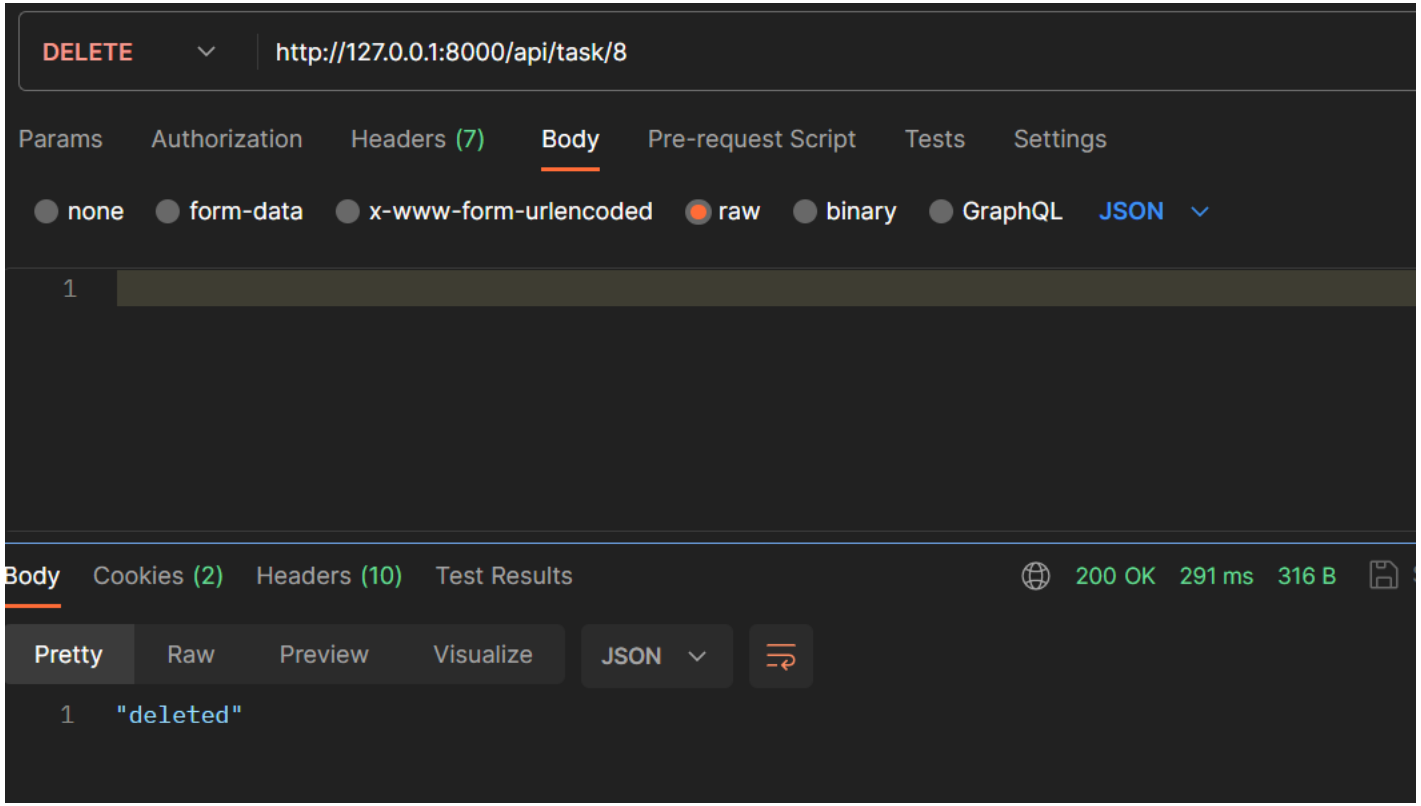
```php
    * Update the specified resource in storage.
    */
    public function update(Request $request, Task $task): TaskResource
    {

        $task = $task->update($request->all());

        return new TaskResource($task);
    }

    /**
     * Remove the specified resource from storage
```



Construir uma API – Delete

cesae digital
Centro para o Desenvolvimento de Competências Digitais

```php
    */
    public function destroy(Task $task)
    {
        $task = $task->delete();

        return response()->json('deleted');
    }
```



Recursos

cesae digital
Centro para o Desenvolvimento de Competências Digitais

- Documentação Laravel
- Laracasts