

## ENSINO MÉDIO INTEGRADO - INFORMÁTICA

### Disciplina de Desenvolvimento Web

#### Aula 09: Eloquent ORM (*Object-Relational Mapper*)

---

*Gil Eduardo de Andrade*

#### Conceitos Preliminares

(<https://laravel.com/docs/8.x>)

### Eloquent ORM

O Laravel disponibiliza ao desenvolvedor um mapeador de objeto-relacional denominado Eloquent ORM ou Eloquent. Ele torna mais dinâmico e prático a interação com o banco de dados da aplicação. Ao utilizar o Eloquent cada tabela do banco de dados possui um “Modelo” correspondente, que permite interagir com a mesma. Sendo assim, torna-se possível, via modelo Eloquent, recuperar, inserir, atualizar e excluir registros da tabela.



### Codificação – Laravel / Eloquent ORM

(<https://laravel.com/docs/8.x/eloquent>)

---

*Para execução dos procedimentos a seguir, considera-se que o arquivo de configuração do ambiente (.env) já possui as informações necessárias para conexão com o banco de dados.*

### Relacionamentos Um-para-Um (One-to-One)

(No terminal - dentro da pasta do projeto)

#### Modelo e Migração - Eixos

```
php artisan make:model Eixo -m
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan
make:model Eixo -m
Model created successfully.
Created Migration: 2022_07_17_140426_create_eixos_table
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

### Arquivos Gerados

- /app/Models/Eixo.php
- /database/migrations/2022\_07\_17\_140426\_create\_pessoas\_table.php

### Modelo e Migração - Salas

`php artisan make:model Sala -m`

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan
make:model Sala -m
Model created successfully.
Created Migration: 2022_07_17_141436_create_salas_table
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

### Arquivos Gerados

- /app/Models/Sala.php
- /database/migrations/2022\_07\_17\_141436\_create\_salas\_table.php

### Configuração da Migração Eixo

(Arquivo: "/database/migrations/2022\_07\_17\_140426\_create\_eixos\_table.php")

```
...

Schema::create('eixos', function (Blueprint $table) {

    $table->id();

    $table->string('nome');

    $table->text('descricao');

    $table->timestamps();

});
```

```
});
```

```
...
```

Arquivo de Migração "Eixo" - Definindo a Estrutura da Tabela

---

(Arquivo-fonte: 08-ORM/database/migration/...\_create\_eixos\_table.php)

### Configuração da Migração Sala

(Arquivo: "/database/migrations/2022\_07\_17\_141436\_create\_salas\_table.php")

```
...
```

```
Schema::create('salas', function (Blueprint $table) {  
    $table->id();  
    $table->string('nome');  
    $table->text('descricao');  
    $table->unsignedBigInteger('eixo_id');  
    $table->foreign('eixo_id')->references('id')->on('eixos');  
    $table->timestamps();  
});
```

```
...
```

Arquivo de Migração "Sala" - Definindo a Estrutura da Tabela

---

(Arquivo-fonte: 08-ORM/database/migration/...\_create\_salas\_table.php)

### Efetuando a Migração

(No terminal - dentro da pasta do projeto)

```
php artisan migrate
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan migrate
Migrating: 2014_10_12_000000_create_users_table
Migrated:  2014_10_12_000000_create_users_table (69.63ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated:  2014_10_12_100000_create_password_resets_table (76.06ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated:  2019_08_19_000000_create_failed_jobs_table (79.11ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated:  2019_12_14_000001_create_personal_access_tokens_table (93.77ms)
Migrating: 2022_07_17_140426_create_eixos_table
Migrated:  2022_07_17_140426_create_eixos_table (34.61ms)
Migrating: 2022_07_17_141436_create_salas_table
Migrated:  2022_07_17_141436_create_salas_table (121.21ms)
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

### Eixo

Nome	Tipo
id 🔑	bigint
nome	varchar(255)
descricao	text
created_at	timestamp
updated_at	timestamp

### Sala

Nome	Tipo
id 🔑	bigint
nome	varchar(255)
descricao	text
eixo_id 🔑	bigint
created_at	timestamp
updated_at	timestamp

## Utilizando a cláusula “hasOne()”

A cláusula `hasOne('nome_modelo')` permite selecionar os dados contidos na tabela de eixos, juntamente com os dados contidos na tabela “salas”, que estão vinculados a cada um dos eixos. Numa tradução literal, *hasOne* significa *possui um*. Em outras palavras, temos um relacionamento *um-para-um*, onde cada eixo *possui uma* única sala.

(Arquivo: “/app/Model/Eixo.php”)

```
...
class Eixo extends Model {
```

```
use HasFactory;

public function sala() {

    return $this->hasOne('\App\Models\Sala');

}

}

...
```

Arquivo de Modelo “Eixo” - Definindo a Relação “Um-Para-Um” / hasOne()

---

(Arquivo-fonte: 08-ORM/app/Models/Eixo.php)

### Utilizando a cláusula “*belongsTo()*”

A cláusula *belongsTo(‘nome\_modelo’)* permite selecionar os dados contidos na tabela de salas, juntamente com os dados contidos na tabela “eixos”, que estão vinculados a cada uma das salas. Numa tradução literal, *belongsTo* significa *pertence a*. Em outras palavras, temos um relacionamento um-para-um, onde cada sala pertence a um único eixo.

(Arquivo: “/app/Models/Sala.php”)

```
...

class Sala extends Model {

    use HasFactory;

    public function eixo() {

        return $this->belongsTo('\App\Models\Eixo');

    }

}

...
```

Arquivo de Modelo “Sala” - Definindo a Relação “Um-Para-Um” / belongsTo()

---

(Arquivo-fonte: 08-ORM/app/Models/Sala.php)

### Utilizando os Recursos: “*hasOne()*” / “*belongsTo()*” / “*associate()*”

As implementações a seguir serão feitas dentro do arquivo de rotas *web.php* apenas para facilitar o entendimento com relação ao uso dos recursos apresentados. Contudo, para construção de uma aplicação web, e pensando em boas práticas de programação, tal codificação estaria dentro de uma classe de controle.

(Arquivo: “/app/Http/Middleware/VerifyCsrfToken.php”)

```
...  
  
protected $except = [  
    'eixos/add',  
];  
  
...  
  
Definindo uma Exceção para o uso do Token CSRF / Rota 'eixo/add'
```

---

(Arquivo-fonte: 08-ORM/app/Http/Middleware/VerifyCsrfToken.php)

### Inserindo um Novo Eixo / Sala

(Arquivo: “/route/web.php”)

```
use \Illuminate\Http\Request;  
  
use App\Models\Eixo;  
use App\Models\Sala;  
  
...  
  
Route::post('eixos/add', function(Request $request) {  
  
    // Eixo  
  
    $obj_eixo = new Eixo();  
  
    $obj_eixo->nome = mb_strtoupper($request->nome_eixo, 'UTF-8');  
  
    $obj_eixo->descricao = mb_strtoupper($request->desc_eixo, 'UTF-8');  
  
    $obj_eixo->save();  
  
    // Sala
```

```

$obj_sala = new Sala();

$obj_sala->nome = mb_strtoupper($request->nome_sala, 'UTF-8');

$obj_sala->descricao = mb_strtoupper($request->desc_sala, 'UTF-8');

// Associa o Eixo a Sala

$obj_sala->eixo()->associate($obj_eixo);

$obj_sala->save();

return "<h1>Eixo Cadastrado com Sucesso!</h1>";

});

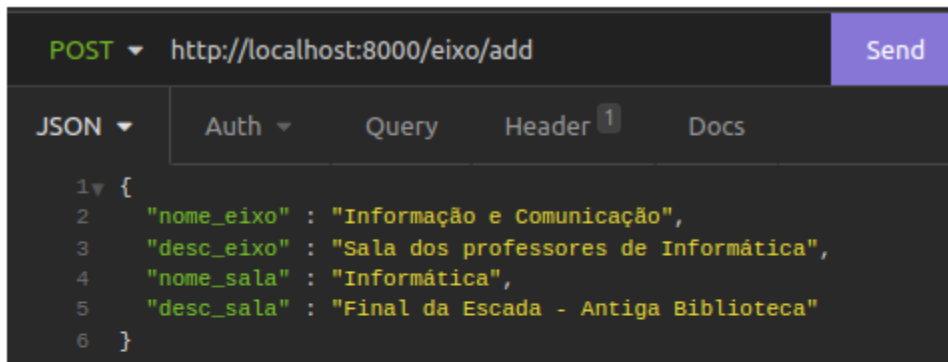
...

```

Arquivo de Rotas - Definindo uma Nova Rota "eixo/add"

(Arquivo-fonte: 08-ORM/routes/web.php)

## Insomnia



POST http://localhost:8000/eixo/add Send

JSON Auth Query Header 1 Docs

```

1 {
2   "nome_eixo" : "Informação e Comunicação",
3   "desc_eixo" : "Sala dos professores de Informática",
4   "nome_sala" : "Informática",
5   "desc_sala" : "Final da Escada - Antiga Biblioteca"
6 }

```

## Tabela de Eixos

id	nome	descricao
1	INFORMAÇÃO E COMUNICAÇÃO	SALA DOS PROFESSORES DE INFORMÁTICA

## Tabela de Salas

id	nome	descricao	eixo_id
1	INFORMÁTICA	FINAL DA ESCADA - ANTIGA BIBLIOTECA	1

(Execução da codificação anterior)

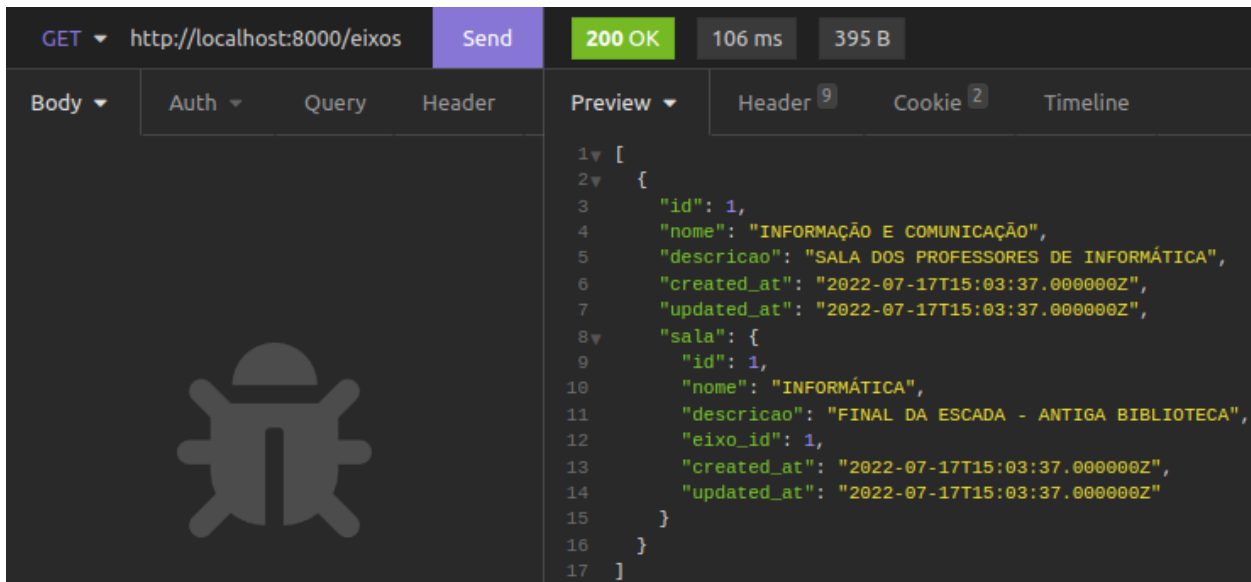
## Selecionando os Dados de Eixos e Salas

(Arquivo: “/route/web.php”)

```
...  
  
Route::get('/eixos', function () {  
  
    $eixos = Eixo::with(['sala'])->get();  
  
    return $eixos->toJson();  
  
});  
  
Route::get('/salas', function () {  
  
    $salas = Sala::with(['eixo'])->get();  
  
    return $salas->toJson();  
  
});  
  
...
```

Arquivo de Rotas - Definindo Novas Rotas: “/eixos” e “/salas”

(Arquivo-fonte: 08-ORM/routes/web.php)



GET http://localhost:8000/eixos 200 OK 106 ms 395 B

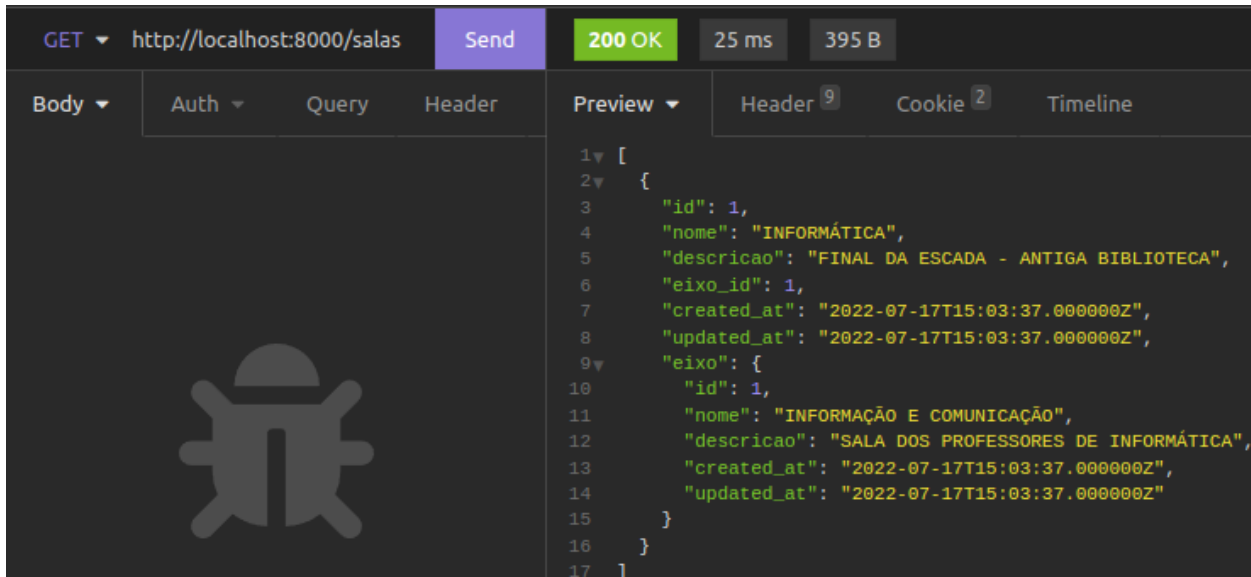
Body Auth Query Header Preview Header 9 Cookie 2 Timeline

```
1 [  
2 {  
3   "id": 1,  
4   "nome": "INFORMAÇÃO E COMUNICAÇÃO",  
5   "descricao": "SALA DOS PROFESSORES DE INFORMÁTICA",  
6   "created_at": "2022-07-17T15:03:37.000000Z",  
7   "updated_at": "2022-07-17T15:03:37.000000Z",  
8   "sala": {  
9     "id": 1,  
10    "nome": "INFORMÁTICA",  
11    "descricao": "FINAL DA ESCADA - ANTIGA BIBLIOTECA",  
12    "eixo_id": 1,  
13    "created_at": "2022-07-17T15:03:37.000000Z",  
14    "updated_at": "2022-07-17T15:03:37.000000Z"  
15  }  
16 }  
17 ]
```

(Execução da codificação anterior)



**OBS.:** A cláusula **“with()”** permite indicar quais dados da relação **“hasOne()”** devem ser trazidos juntamente com os dados dos eixos. Isso é possível pela codificação efetuada na classe de modelo Eixo.



```
GET http://localhost:8000/salas Send 200 OK 25 ms 395 B
Body Auth Query Header Preview Header 9 Cookie 2 Timeline
1 [
2 {
3   "id": 1,
4   "nome": "INFORMÁTICA",
5   "descricao": "FINAL DA ESCADA - ANTIGA BIBLIOTECA",
6   "eixo_id": 1,
7   "created_at": "2022-07-17T15:03:37.000000Z",
8   "updated_at": "2022-07-17T15:03:37.000000Z",
9   "eixo": {
10    "id": 1,
11    "nome": "INFORMAÇÃO E COMUNICAÇÃO",
12    "descricao": "SALA DOS PROFESSORES DE INFORMÁTICA",
13    "created_at": "2022-07-17T15:03:37.000000Z",
14    "updated_at": "2022-07-17T15:03:37.000000Z"
15  }
16 }
17 ]
```

*(Execução da codificação anterior)*

**OBS.:** A cláusula **“with()”** permite indicar quais dados da relação **“belongsToMany()”** devem ser trazidos juntamente com os dados das salas. Isso é possível pela codificação efetuada na classe de modelo Sala.

## Relacionamentos Um-para-Muitos (One-to-Many)

(No terminal - dentro da pasta do projeto)

```
php artisan make:model Curso -m
```

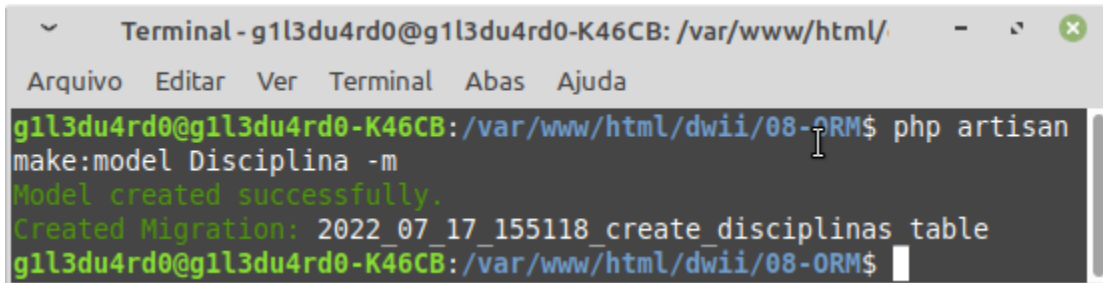


```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan
make:model Curso -m
Model created successfully.
Created Migration: 2022_07_17_155007_create_cursos_table
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

## Arquivos Gerados

- /app/Models/Curso.php
- /database/migrations/2022\_07\_17\_155007\_create\_cursos\_table.php

```
php artisan make:model Disciplina -m
```



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan
make:model Disciplina -m
Model created successfully.
Created Migration: 2022_07_17_155118_create_disciplinas_table
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

### Arquivos Gerados

- /app/Models/Disciplina.php
- /database/migrations/2022\_07\_17\_155118\_create\_disciplinas\_table.php

### Configuração da Migração Curso

(Arquivo: "/database/migrations/2022\_07\_17\_155007\_create\_cursos\_table.php")

```
...
Schema::create('cursos', function (Blueprint $table) {
    $table->id();
    $table->string('nome');
    $table->text('descricao');
    $table->timestamps();
});
...

Arquivo de Migração "Curso" - Definindo a Estrutura da Tabela

(Arquivo-fonte: 08-ORM/database/migration/..._create_cursos_table.php)
```

### Configuração da Migração Disciplina

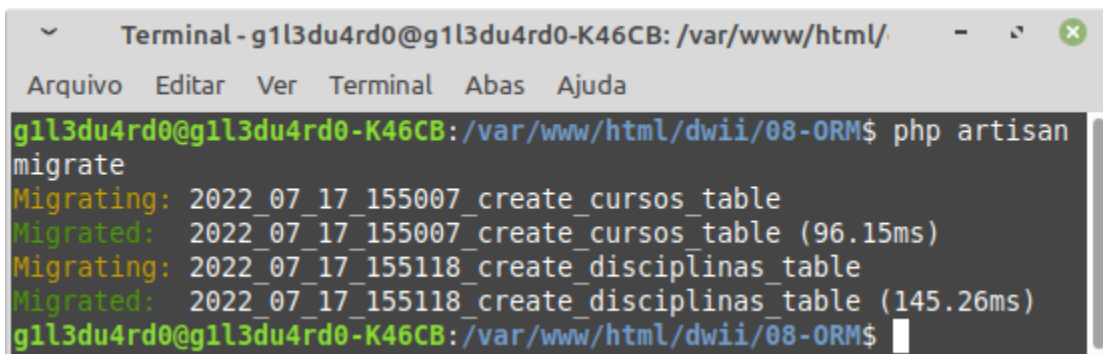
(Arquivo: "/database/migrations/2022\_07\_17\_155118\_create\_disciplinas\_table.php")

```
...  
  
Schema::create('disciplinas', function (Blueprint $table) {  
  
    $table->id();  
  
    $table->string('nome');  
  
    $table->unsignedBigInteger('curso_id');  
  
    $table->foreign('curso_id')->references('id')->on('cursos');  
  
    $table->timestamps();  
  
});  
  
...  
  
Arquivo de Migração "Disciplina" - Definindo a Estrutura da Tabela  
  
-----  
(Arquivo-fonte: 08-ORM/database/migration/..._create_disciplinas_table.php)
```

## Efetutando a Migração


(No terminal - dentro da pasta do projeto)

```
php artisan migrate
```





```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/  
Arquivo Editar Ver Terminal Abas Ajuda  
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan  
migrate  
Migrating: 2022_07_17_155007_create_cursos_table  
Migrated: 2022_07_17_155007_create_cursos_table (96.15ms)  
Migrating: 2022_07_17_155118_create_disciplinas_table  
Migrated: 2022_07_17_155118_create_disciplinas_table (145.26ms)  
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

### Cursos

Nome	Tipo
id 	bigint
nome	varchar(255)
descricao	text
created_at	timestamp
updated_at	timestamp

### Disciplinas

Nome	Tipo
id 	bigint
nome	varchar(255)
curso_id 	bigint
created_at	timestamp
updated_at	timestamp

### Utilizando a cláusula “*hasMany()*”

A cláusula *hasMany('nome\_modelo')* permite selecionar os dados contidos na tabela de cursos, juntamente com os dados contidos na tabela de “disciplinas”, que estão vinculados a cada um dos cursos. Numa tradução literal, *hasMany* significa *possui vários*. Em outras palavras, temos um relacionamento *um-para-muitos*, onde cada curso *possui muitas* disciplinas.

(Arquivo: “/app/Models/Curso.php”)

```
...
class Curso extends Model {
    use HasFactory;

    public function disciplina() {
        return $this->hasMany('\App\Models\Disciplina');
    }
}
...
```

Arquivo de Modelo “Curso” - Definindo a Relação “Um-Para-Muitos” / *hasMany()*

(Arquivo-fonte: 08-ORM/app/Models/Curso.php)

### Utilizando a cláusula “*belongsTo()*”

A cláusula *belongsTo('nome\_modelo')* permite selecionar os dados contidos na tabela de disciplinas, juntamente com os dados contidos na tabela de “cursos”, que estão vinculados a cada uma das disciplinas. Numa tradução literal, *belongsTo* significa *pertence a*. Em outras

palavras, temos um relacionamento um-para-muitos, onde muitas disciplinas pertencem a um único curso.

(Arquivo: “/app/Models/Disciplina.php”)

```
...  
  
class Disciplina extends Model {  
  
    use HasFactory;  
  
    public function curso() {  
  
        return $this->belongsTo('App\Models\Curso');  
  
    }  
  
}  
  
...
```

Arquivo de Modelo “Disciplina” - Definindo a Relação “Um-Para-Muitos” / belongsTo()

---

(Arquivo-fonte: 08-ORM/app/Models/Disciplina.php)

### Utilizando os Recursos: “*hasMany()*” / “*belongsTo()*” / “*associate()*”

As implementações a seguir serão feitas dentro do arquivo de rotas *web.php* apenas para facilitar o entendimento com relação ao uso dos recursos apresentados. Contudo, para construção de uma aplicação web, e pensando em boas práticas de programação, tal codificação estaria dentro de uma classe de controle.

(Arquivo: “/app/Http/Middleware/VerifyCsrfToken.php”)

```
...  
  
protected $except = [  
  
    "eixos/add",  
  
    "cursos/add",  
  
    "disciplinas/add",  
  
];
```

...

Definindo uma Exceção para o uso Token CSRF / Rotas 'cursos/add' e 'disciplinas/add'

---

(Arquivo-fonte: aula09/Middleware/VerifyCsrfToken.php)

## Inserindo Novos Cursos e Disciplinas

(Arquivo: "/route/web.php")

...

```
use App\Models\Curso;
use App\Models\Disciplina;
...
Route::post('cursos/add', function (Request $request) {
    $obj_curso = new Curso();
    $obj_curso->nome = mb_strtoupper($request->nome, 'UTF-8');
    $obj_curso->descricao = mb_strtoupper($request->desc, 'UTF-8');
    $obj_curso->save();
    return "<h1>Curso Cadastrado com Sucesso!</h1>";
});

Route::post('disciplinas/add', function (Request $request) {
    $obj_curso = Curso::find($request->curso_id);
    if(isset($obj_curso)) {
        $obj_disciplina = new Disciplina();
        $obj_disciplina->nome = mb_strtoupper($request->nome, 'UTF-8');
        $obj_disciplina->curso()->associate($obj_curso);
        $obj_disciplina->save();
        return "<h1>Disciplina Cadastrada com Sucesso!</h1>";
    }
});
```

```

}

return "<h1>[ERRO] Curso NÃO ENCONTRADO!</h1>";

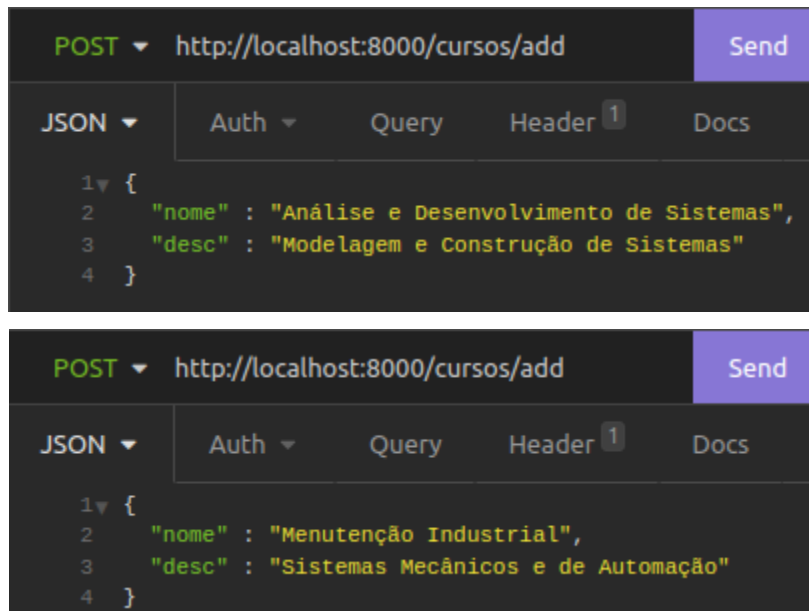
});

...

```

Arquivo de Rotas - Definindo Novas Rotas: "cursos/add" e "disciplinas/add"

(Arquivo-fonte: aula09/routes/web.php)



The image shows two screenshots of a REST client interface. Both show a POST request to the URL `http://localhost:8000/cursos/add`. The first screenshot shows a JSON payload with the following structure:

```

1 {
2   "nome" : "Análise e Desenvolvimento de Sistemas",
3   "desc" : "Modelagem e Construção de Sistemas"
4 }

```

The second screenshot shows a JSON payload with the following structure:

```

1 {
2   "nome" : "Manutenção Industrial",
3   "desc" : "Sistemas Mecânicos e de Automação"
4 }

```

## Cursos

id	nome	descricao
1	ANÁLISE E DESENVOLVIMENTO DE SISTEMAS	MODELAGEM E CONSTRUÇÃO DE SISTEMAS
2	MENUTENÇÃO INDUSTRIAL	SISTEMAS MECÂNICOS E DE AUTOMAÇÃO

*(Execução da codificação anterior)*

```
POST http://localhost:8000/disciplinas/add
JSON
1 {
2   "nome" : "Desenvolvimento Web II",
3   "curso_id" : "1"
4 }
```

```
POST http://localhost:8000/disciplinas/add
JSON
1 {
2   "nome" : "Automação I",
3   "curso_id" : "2"
4 }
```

### Disciplinas

id	nome	curso_id
1	DESENVOLVIMENTO WEB II	1
2	AUTOMAÇÃO I	2

*(Execução da codificação anterior)*

### Selecionando os Dados de Cursos e Disciplinas

(Arquivo: "/route/web.php")

```
...
Route::get('/cursos', function () {
    $obj_curso = Curso::with(['disciplina'])->get();
    return $obj_curso->toJson();
});

Route::get('/disciplinas', function () {
    $obj_disciplina = Disciplina::with(['curso'])->get();
    return $obj_disciplina->toJson();
});
```

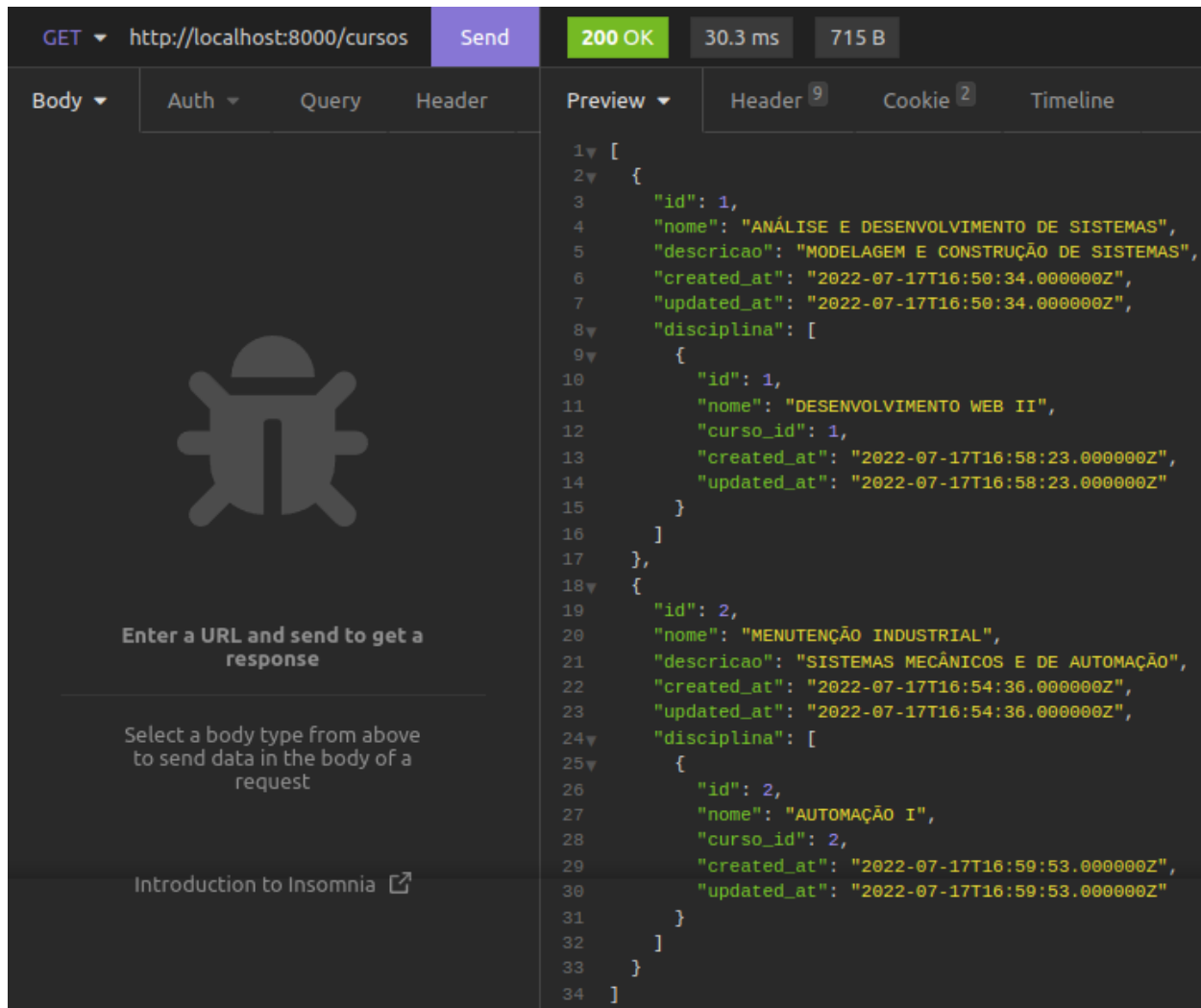


```
});
```

```
...
```

Arquivo de Rotas - Definindo Novas Rotas “/cursos” e “/disciplinas”

(Arquivo-fonte: 08-ORM/routes/web.php)



GET http://localhost:8000/cursos Send 200 OK 30.3 ms 715 B

Body Auth Query Header Preview Header 9 Cookie 2 Timeline

Enter a URL and send to get a response

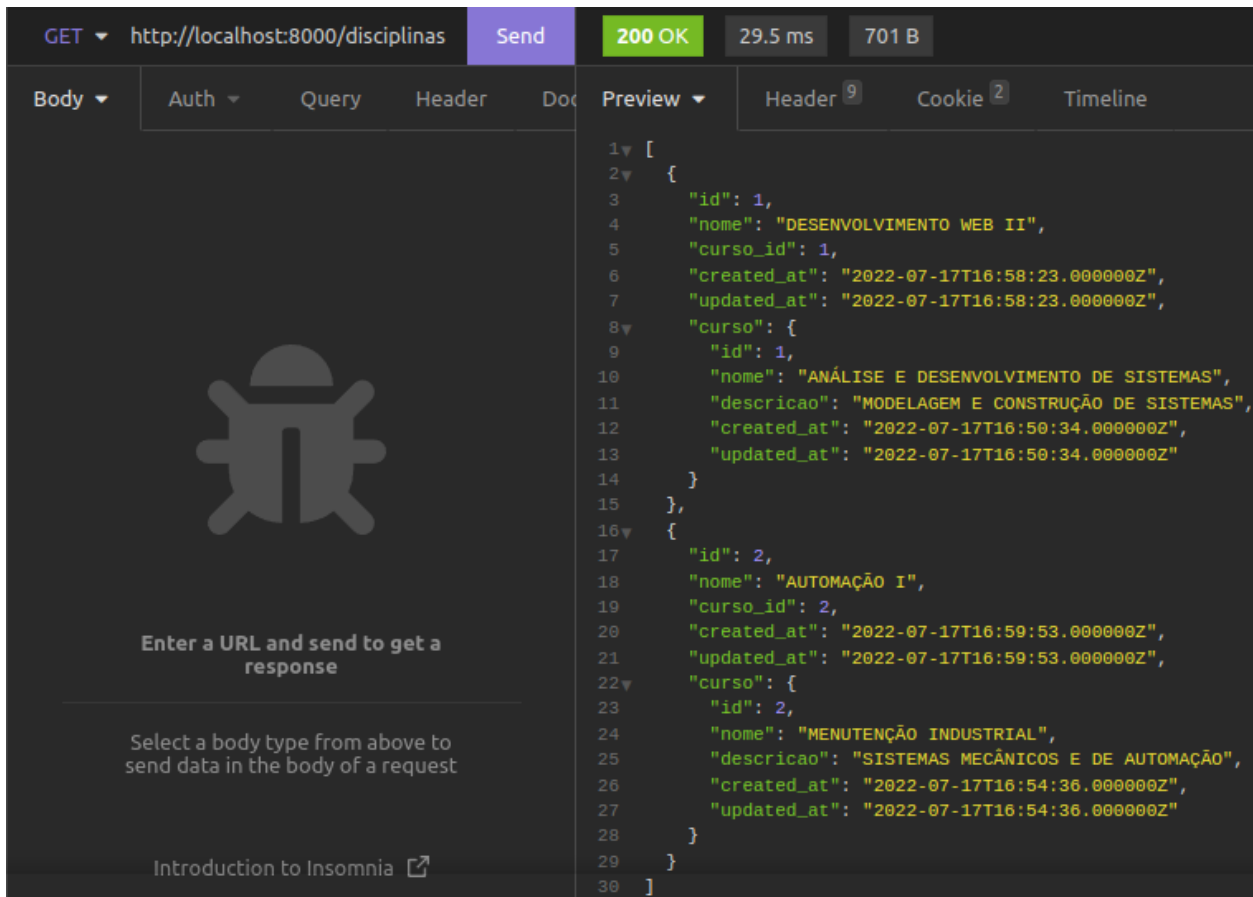
Select a body type from above to send data in the body of a request

Introduction to Insomnia

```
1 [
2   {
3     "id": 1,
4     "nome": "ANÁLISE E DESENVOLVIMENTO DE SISTEMAS",
5     "descricao": "MODELAGEM E CONSTRUÇÃO DE SISTEMAS",
6     "created_at": "2022-07-17T16:50:34.000000Z",
7     "updated_at": "2022-07-17T16:50:34.000000Z",
8     "disciplina": [
9       {
10        "id": 1,
11        "nome": "DESENVOLVIMENTO WEB II",
12        "curso_id": 1,
13        "created_at": "2022-07-17T16:58:23.000000Z",
14        "updated_at": "2022-07-17T16:58:23.000000Z"
15      }
16    ]
17  },
18  {
19    "id": 2,
20    "nome": "MENUTENÇÃO INDUSTRIAL",
21    "descricao": "SISTEMAS MECÂNICOS E DE AUTOMAÇÃO",
22    "created_at": "2022-07-17T16:54:36.000000Z",
23    "updated_at": "2022-07-17T16:54:36.000000Z",
24    "disciplina": [
25      {
26        "id": 2,
27        "nome": "AUTOMAÇÃO I",
28        "curso_id": 2,
29        "created_at": "2022-07-17T16:59:53.000000Z",
30        "updated_at": "2022-07-17T16:59:53.000000Z"
31      }
32    ]
33  }
34 ]
```

(Execução da codificação anterior)

**OBS.:** A cláusula “with()” permite indicar quais dados da relação “hasMany()” devem ser trazidos juntamente com os dados dos cursos. Isso é possível pela codificação efetuada na classe de modelo Curso.



```
GET http://localhost:8000/disciplinas 200 OK 29.5 ms 701 B
Body Auth Query Header Doc Preview Header 9 Cookie 2 Timeline
1 [
2 {
3   "id": 1,
4   "nome": "DESENVOLVIMENTO WEB II",
5   "curso_id": 1,
6   "created_at": "2022-07-17T16:58:23.000000Z",
7   "updated_at": "2022-07-17T16:58:23.000000Z",
8   "curso": {
9     "id": 1,
10    "nome": "ANÁLISE E DESENVOLVIMENTO DE SISTEMAS",
11    "descricao": "MODELAGEM E CONSTRUÇÃO DE SISTEMAS",
12    "created_at": "2022-07-17T16:50:34.000000Z",
13    "updated_at": "2022-07-17T16:50:34.000000Z"
14  }
15 },
16 {
17   "id": 2,
18   "nome": "AUTOMAÇÃO I",
19   "curso_id": 2,
20   "created_at": "2022-07-17T16:59:53.000000Z",
21   "updated_at": "2022-07-17T16:59:53.000000Z",
22   "curso": {
23     "id": 2,
24     "nome": "MANUTENÇÃO INDUSTRIAL",
25     "descricao": "SISTEMAS MECÂNICOS E DE AUTOMAÇÃO",
26     "created_at": "2022-07-17T16:54:36.000000Z",
27     "updated_at": "2022-07-17T16:54:36.000000Z"
28   }
29 }
30 ]
```

*(Execução da codificação anterior)*

**OBS.:** A cláusula **“with()”** permite indicar quais dados da relação **“belongsTo()”** devem ser trazidos juntamente com os dados das disciplinas. Isso é possível pela codificação efetuada na classe de modelo Disciplina.

## Relacionamentos Muitos-para-Muitos (Many-to-Many)

(No terminal - dentro da pasta do projeto)

```
php artisan make:model Aluno -m
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan
make:model Aluno -m
Model created successfully.
Created Migration: 2022_07_17_170738_create_alunos_table
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

### Arquivos Gerados

- /app/Models/Aluno.php
- /database/migrations/2022\_07\_17\_170738\_create\_alunos\_table.php

`php artisan make:model Matricula -m`

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan
make:model Matricula -m
Model created successfully.
Created Migration: 2022_07_17_170827_create_matriculas_table
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

### Arquivos Gerados

- /app/Models/Matricula.php
- /database/migrations/2022\_07\_17\_170827\_create\_matriculas\_table.php

(Arquivo: “/database/migrations/2022\_07\_17\_170738\_create\_alunos\_table.php”)

```
...
Schema::create('alunos', function (Blueprint $table) {
    $table->id();
    $table->string('nome');
    $table->timestamps();
});
...
```

Arquivo de Migração "Aluno" - Definindo a Estrutura da Tabela

---

(Arquivo-fonte: 08-ORM/database/migration/...\_create\_alunos\_table.php)

(Arquivo: "/database/migrations/2022\_07\_17\_170827\_create\_matriculas\_table.php")

```
...  
  
Schema::create('matriculas', function (Blueprint $table) {  
  
    $table->unsignedBigInteger('aluno_id');  
  
    $table->foreign('aluno_id')->references('id')->on('alunos');  
  
    $table->unsignedBigInteger('disciplina_id');  
  
    $table->foreign('disciplina_id')->references('id')->on('disciplinas');  
  
    $table->string('descricao');  
  
    $table->primary(['aluno_id', 'disciplina_id']);  
  
    $table->timestamps();  
  
});  
  
...
```

Arquivo de Migração "Matrícula" - Definindo a Estrutura da Tabela

---



(Arquivo-fonte: 08-ORM/database/migration/...\_create\_matriculas\_table.php)

## Efetuando a Migração

(No terminal - dentro da pasta do projeto)

```
php artisan migrate
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$ php artisan
migrate
Migrating: 2022_07_17_170738_create_alunos_table
Migrated: 2022_07_17_170738_create_alunos_table (85.66ms)
Migrating: 2022_07_17_170827_create_matriculas_table
Migrated: 2022_07_17_170827_create_matriculas_table (303.89ms)
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/08-ORM$
```

Alunos		Matriculas	
Nome	Tipo	Nome	Tipo
id 	bigint	aluno_id 	bigint
nome	varchar(255)	disciplina_id  	bigint
created_at	timestamp	descricao	varchar(255)
updated_at	timestamp	created_at	timestamp
		updated_at	timestamp

### Utilizando a cláusula “*belongsToMany()*”

A cláusula *belongsToMany('nome\_modelo', 'nome\_tabela')* permite selecionar os dados contidos na tabela alunos, juntamente com os dados contidos na tabela “disciplinas”, que estão vinculados com cada um dos alunos. Numa tradução literal, *belongsToMany* significa *pertence a muitos*. Em outras palavras, temos um relacionamento *mutios-para-muitos*, onde cada aluno *pertence a muitas* disciplinas.

(Arquivo: “/app/Models/Aluno.php”)

```
...
class Aluno extends Model {
    use HasFactory;

    public function disciplina() {
        return $this->belongsToMany('\App\Models\Disciplina', 'matriculas');
    }
}
...
```

Arquivo de Modelo “Aluno” - Definindo a Relação “Muitos-Para-Muitos” / belongsToMany()

---

(Arquivo-fonte: 08-ORM/app/Models/Aluno.php)

(Arquivo: “/app/Models/Disciplina.php”)

```
...  
  
class Disciplina extends Model {  
  
    use HasFactory;  
  
    public function curso() {  
  
        return $this->belongsTo('App\Models\Curso');  
  
    }  
  
    public function aluno() {  
  
        return $this->belongsToMany('\App\Models\Aluno', 'matriculas');  
  
    }  
  
}  
  
...
```

Arquivo de Modelo “Disciplina” - Definindo a Relação “Muitos-Para-Muitos” /  
belongsToMany()

---

(Arquivo-fonte: 08-ORM/app/Models/Disciplina.php)

(Arquivo: “/app/Model/Matricula.php”)

```
...  
  
class Matricula extends Model {  
  
    use HasFactory;  
  
    public function aluno() {  
  
        return $this->belongsTo('App\Models\Aluno');  
  
    }  
  
}
```

```
public function disciplina() {  
    return $this->belongsTo('App\Models\Disciplina');  
}  
}  
...  
Arquivo de Modelo "Matricula" - Definindo a Relação "Muitos-Para-Muitos" / belongsTo()  


---

  
(Arquivo-fonte: 08-ORM/Models/Disciplina.php)
```

### Utilizando os Recursos: *"belongsToMany()" / "withPivot()"*

As implementações a seguir serão feitas dentro do arquivo de rotas *web.php* apenas para facilitar o entendimento com relação ao uso dos recursos apresentados. Contudo, para construção de uma aplicação web, e pensando em boas práticas de programação, tal codificação estaria dentro de uma classe de controle.

(Arquivo: *"/app/Http/Middleware/VerifyCsrfToken.php"*)

```
...  
protected $except = [  
    'pessoas/add',  
    'cursos/add',  
    'disciplinas/add',  
    'alunos/add',  
    'matriculas/add',  
];  
...  
Definindo uma Exceção para o uso do Token CSRF / Rotas 'alunos/add' e 'matriculas/add'  


---

  
(Arquivo-fonte: 08-ORM/Middleware/VerifyCsrfToken.php)
```

## Inserindo Novos Alunos e Matrículas

(Arquivo: "/route/web.php")

```
...  
  
use App\Models\Aluno;  
use App\Models\Matricula;  
  
...  
  
Route::post('alunos/add', function (Request $request) {  
  
    $obj_aluno = new Aluno();  
  
    $obj_aluno->nome = mb_strtoupper($request->nome, 'UTF-8');  
  
    $obj_aluno->save();  
  
    return "<h1>Aluno Cadastrado com Sucesso!</h1>";  
  
});  
  
Route::post('matriculas/add', function (Request $request) {  
  
    $obj_aluno = Aluno::find($request->aluno_id);  
  
    $obj_disciplina = Disciplina::find($request->disciplina_id);  
  
    if(isset($obj_aluno) && isset($obj_disciplina)) {  
  
        $obj_matricula = new Matricula();  
  
        $obj_matricula->descricao = mb_strtoupper($request->descricao, 'UTF-8');  
  
        $obj_matricula->aluno()->associate($obj_aluno);  
  
        $obj_matricula->disciplina()->associate($obj_disciplina);  
  
        $obj_matricula->save();  
  
        return "<h1>Matrícula Cadastrada com Sucesso!</h1>";  
  
    }  
  
    return "<h1>[ERRO] Aluno e/ou Disciplina NÃO ENCONTRADO(S)!</h1>";  
  
});  
  
...
```



Arquivo de Rotas - Definindo Novas Rotas: "alunos/add" e "matriculas/add"

(Arquivo-fonte: 08-ORM/routes/web.php)

```
POST http://localhost:8000/alunos/add Send
JSON Auth Query Header 1 Docs
1 {
2   "nome" : "Joãozinho da Silva"
3 }
```

```
POST http://localhost:8000/alunos/add Send
JSON Auth Query Header 1 Docs
1 {
2   "nome" : "Mariazinha da Silva"
3 }
```

### Alunos

id	nome
1	JOÃOZINHO DA SILVA
2	MARIAZINHA DA SILVA

*(Execução da codificação anterior)*

```
POST http://localhost:8000/matriculas/add Send
JSON Auth Query Header 1 Docs
1 {
2   "aluno_id" : "1",
3   "disciplina_id" : "2",
4   "descricao" : "Período da Manhã"
5 }
```

```
POST http://localhost:8000/matriculas/add Send
JSON Auth Query Header 1 Docs
1 {
2   "aluno_id" : "2",
3   "disciplina_id" : "1",
4   "descricao" : "Período da Manhã"
5 }
```

### Matriculas

aluno_id	disciplina_id	descricao
1	2	PERÍODO DA MANHÃ
2	1	PERÍODO DA MANHÃ

*(Execução da codificação anterior)*

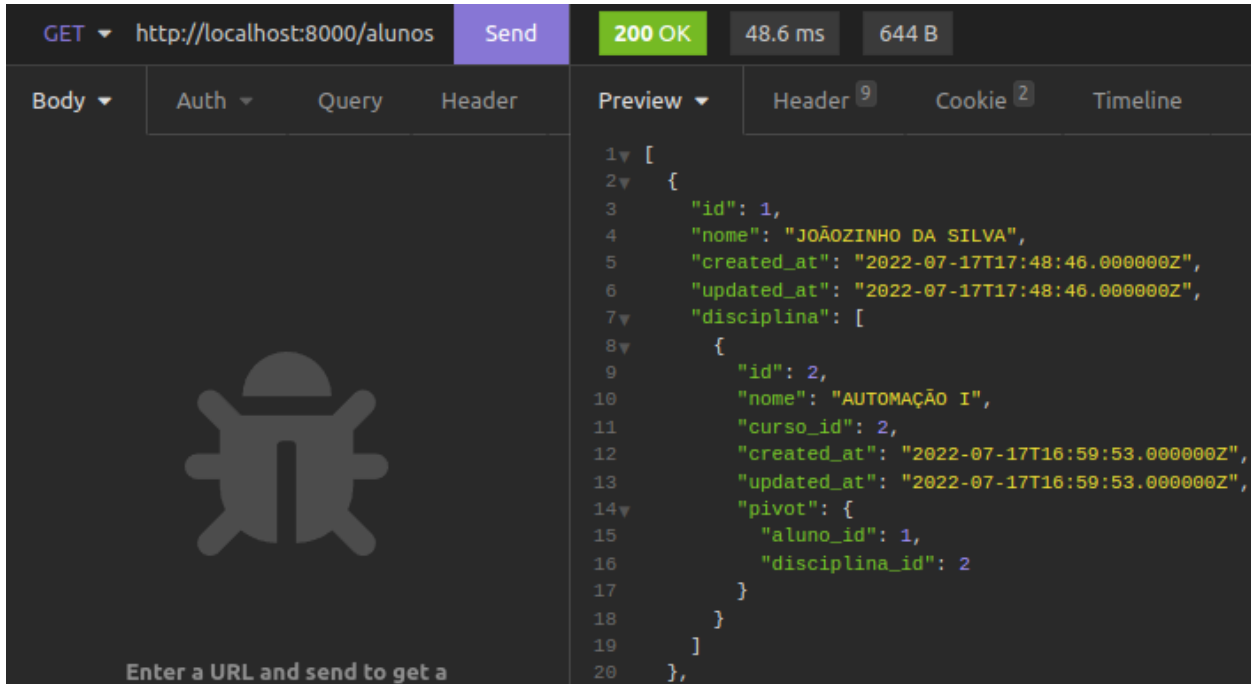
### Selecionando os Dados de Alunos

(Arquivo: "/route/web.php")

```
...
Route::get('/alunos', function () {
    $a = Aluno::with(['disciplina'])->get();
    return $a->toJson();
});
...
```

Arquivo de Rotas - Definindo uma Nova Rota "/alunos"

(Arquivo-fonte: 08-ORM/routes/web.php)



GET http://localhost:8000/alunos Send 200 OK 48.6 ms 644 B

Body Auth Query Header Preview Header 9 Cookie 2 Timeline

```

1 [
2   {
3     "id": 1,
4     "nome": "JOÃOZINHO DA SILVA",
5     "created_at": "2022-07-17T17:48:46.000000Z",
6     "updated_at": "2022-07-17T17:48:46.000000Z",
7     "disciplina": [
8       {
9         "id": 2,
10        "nome": "AUTOMAÇÃO I",
11        "curso_id": 2,
12        "created_at": "2022-07-17T16:59:53.000000Z",
13        "updated_at": "2022-07-17T16:59:53.000000Z",
14        "pivot": {
15          "aluno_id": 1,
16          "disciplina_id": 2
17        }
18      }
19    ]
20  },

```

Enter a URL and send to get a

**OBS.:** Observe que junto aos dados das disciplinas temos um array denominado **“pivot”**. O **“pivot”** contém as chaves primárias das tabelas alunos e disciplinas. Contudo, observe que o campo descrição não foi trazido juntamente com esses dados.

### Utilizando a cláusula **“withPivot”**

(Arquivo: `“/app/Models/Aluno.php”`)

```

...

class Aluno extends Model {

    use HasFactory;

    public function disciplina() {

        return $this->belongsToMany('\App\Models\Disciplina', 'matriculas')

            ->withPivot('descricao');

    }

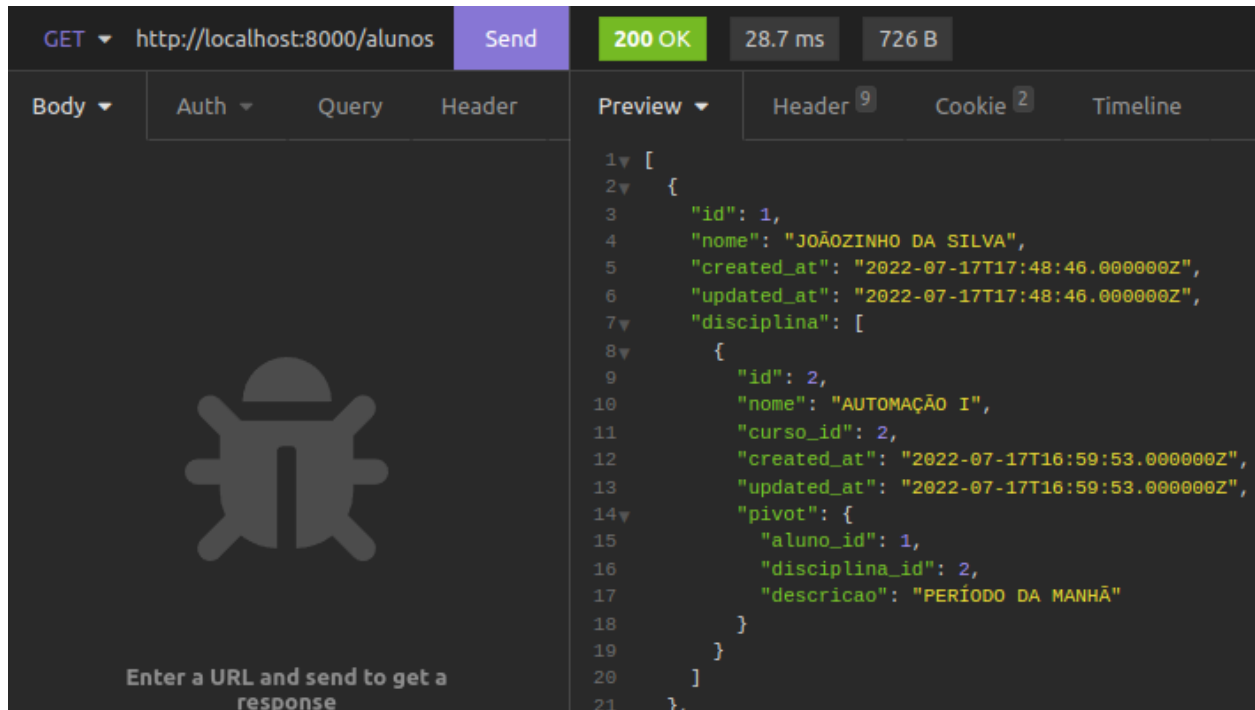
}

...

```

Arquivo de Modelo “Aluno” - Utilizando a cláusula “withPivot()”

(Arquivo-fonte: 08-ORM/app/Models/Aluno.php)



GET http://localhost:8000/alunos Send 200 OK 28.7 ms 726 B

Body Auth Query Header Preview Header 9 Cookie 2 Timeline

```
1 [
2   {
3     "id": 1,
4     "nome": "JOÃOZINHO DA SILVA",
5     "created_at": "2022-07-17T17:48:46.000000Z",
6     "updated_at": "2022-07-17T17:48:46.000000Z",
7     "disciplina": [
8       {
9         "id": 2,
10        "nome": "AUTOMAÇÃO I",
11        "curso_id": 2,
12        "created_at": "2022-07-17T16:59:53.000000Z",
13        "updated_at": "2022-07-17T16:59:53.000000Z",
14        "pivot": {
15          "aluno_id": 1,
16          "disciplina_id": 2,
17          "descricao": "PERÍODO DA MANHÃ"
18        }
19      }
20    ]
21  },
```

Enter a URL and send to get a response

(Execução da codificação anterior)

**OBS.:** Observe que agora, junto aos dados das disciplinas, no array “**pivot**”, temos também o campo descrição.

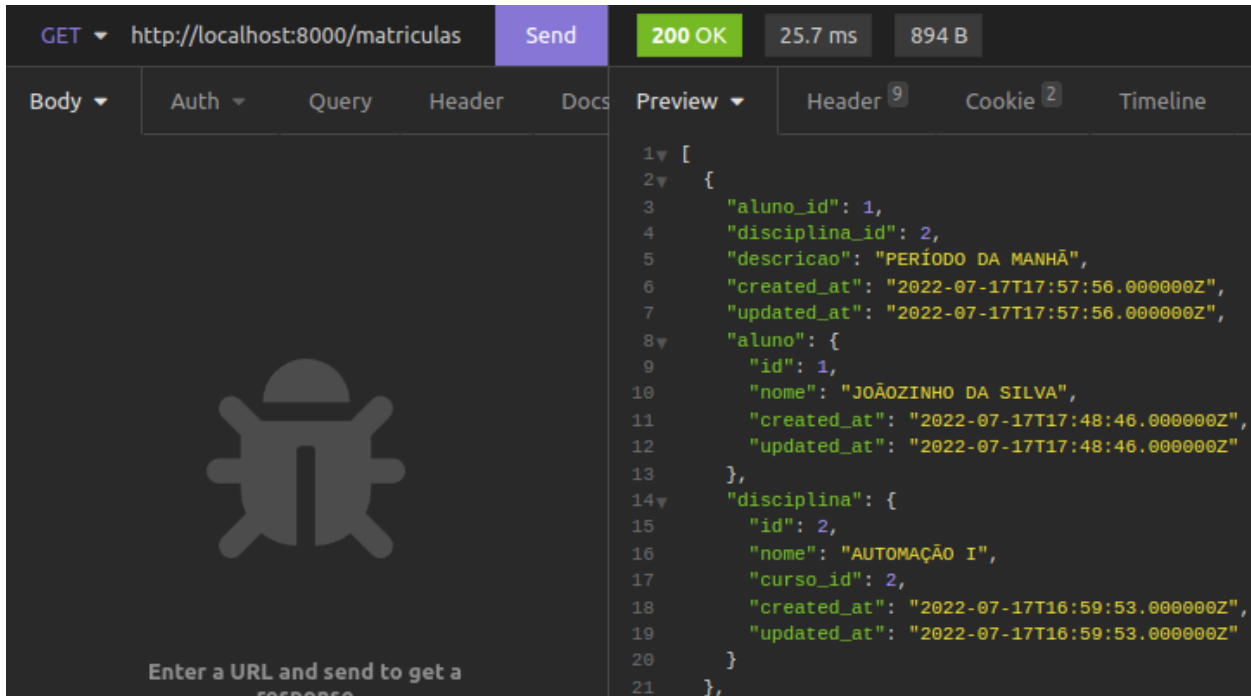
## Selecionando os Dados de Matrículas

(Arquivo: “/route/web.php”)

```
...
Route::get('/matriculas', function () {
    $obj_matricula = Matricula::with(['aluno', 'disciplina'])->get();
    return $obj_matricula->toJson();
});
...
```

Arquivo de Rotas - Definindo uma Nova Rota “/matriculas”

(Arquivo-fonte: aula09/routes/web.php)



GET http://localhost:8000/matriculas Send 200 OK 25.7 ms 894 B

Body Auth Query Header Docs Preview Header 9 Cookie 2 Timeline

```
1 [
2   {
3     "aluno_id": 1,
4     "disciplina_id": 2,
5     "descricao": "PERÍODO DA MANHÃ",
6     "created_at": "2022-07-17T17:57:56.000000Z",
7     "updated_at": "2022-07-17T17:57:56.000000Z",
8     "aluno": {
9       "id": 1,
10      "nome": "JOÃOZINHO DA SILVA",
11      "created_at": "2022-07-17T17:48:46.000000Z",
12      "updated_at": "2022-07-17T17:48:46.000000Z"
13    },
14    "disciplina": {
15      "id": 2,
16      "nome": "AUTOMAÇÃO I",
17      "curso_id": 2,
18      "created_at": "2022-07-17T16:59:53.000000Z",
19      "updated_at": "2022-07-17T16:59:53.000000Z"
20    }
21  },
22 ]
```

(Execução da codificação anterior)

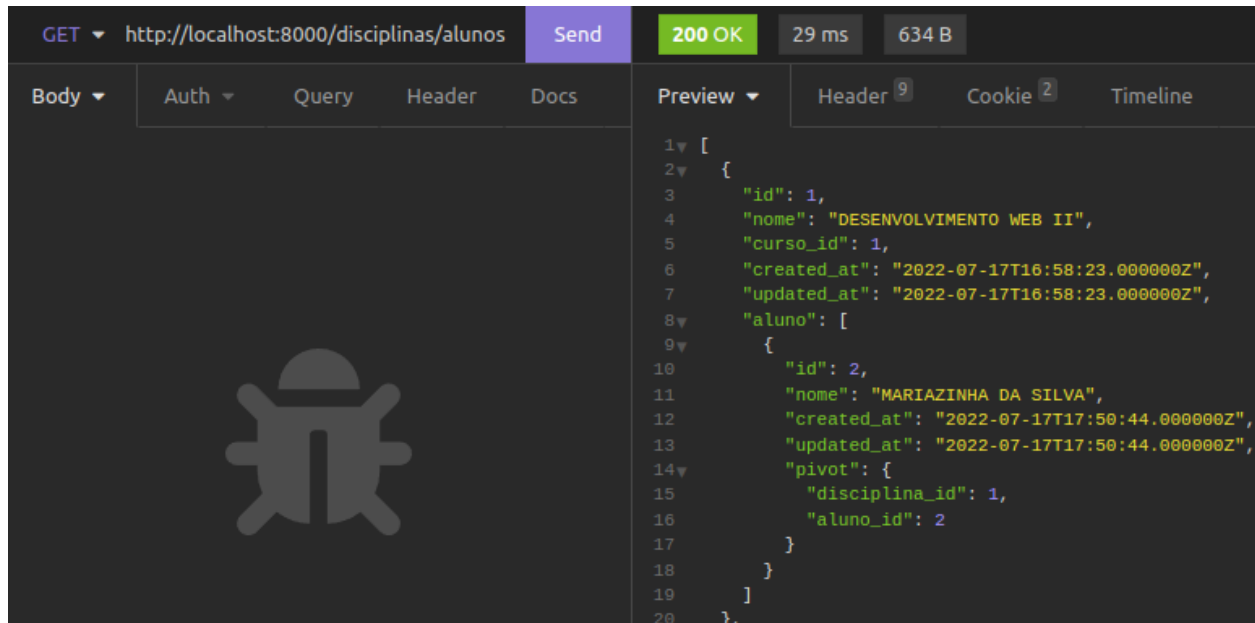
## Selecionando os Disciplinas e Alunos

(Arquivo: “/route/web.php”)

```
...
Route::get('/disciplinas/alunos', function () {
    $obj_disciplina = Disciplina::with(['aluno'])->get();
    return $obj_disciplina->toJson();
});
...
```

Arquivo de Rotas - Definindo uma Nova Rota “/disciplinas/alunos”

(Arquivo-fonte: 08-ORM/routes/web.php)



GET http://localhost:8000/disciplinas/alunos Send 200 OK 29 ms 634 B

Body Auth Query Header Docs Preview Header 9 Cookie 2 Timeline

```
1 [
2 {
3   "id": 1,
4   "nome": "DESENVOLVIMENTO WEB II",
5   "curso_id": 1,
6   "created_at": "2022-07-17T16:58:23.000000Z",
7   "updated_at": "2022-07-17T16:58:23.000000Z",
8   "aluno": [
9     {
10      "id": 2,
11      "nome": "MARIAZINHA DA SILVA",
12      "created_at": "2022-07-17T17:50:44.000000Z",
13      "updated_at": "2022-07-17T17:50:44.000000Z",
14      "pivot": {
15        "disciplina_id": 1,
16        "aluno_id": 2
17      }
18    }
19  ]
20 },
```

(Execução da codificação anterior)

## Removendo Todas as Matrículas e Efetuando Novamente

(Arquivo: "/app/Http/Middleware/VerifyCsrfToken.php")

```
...
protected $except = [
    'pessoas/add',
    'cursos/add',
    'disciplinas/add',
    'alunos/add',
    'matriculas/add',
    'matriculas/aluno',
];
...
```

Definindo uma Exceção para o uso do Token CSRF / Rota 'matriculas/aluno'

(Arquivo-fonte: 08-ORM/Middleware/VerifyCsrfToken.php)

(Arquivo: “/route/web.php”)

```
...  
Route::post('/matriculas/aluno', function (Request $request) {  
    $obj_aluno = Aluno::find($request->aluno_id);  
    if(isset($obj_aluno)) {  
        $obj_aluno->disciplina()->detach();  
        $disciplinas = $request->input('disciplinas');  
        foreach($disciplinas as $item) {  
            $obj_disciplina = Disciplina::find($item['disciplina_id']);  
            if(isset($obj_disciplina)){  
                $obj_matricula = new Matricula();  
                $obj_matricula->descricao = mb_strtoupper($item['descricao'], 'UTF-8');  
                $obj_matricula->aluno()->associate($obj_aluno);  
                $obj_matricula->disciplina()->associate($obj_disciplina);  
                $obj_matricula->save();  
            }  
        }  
        return "<h1>Todas as Matrículas Efetuadas com Sucesso!</h1>";  
    }  
    return "<h1>[ERRO] Aluno e/ou Disciplina NÃO ENCONTRADO(S)!</h1>";  
});  
...
```

Arquivo de Rotas - Definindo uma Nova Rota “/matriculas/aluno”

(Arquivo-fonte: 08-ORM/routes/web.php)

```
POST http://localhost:8000/matriculas/aluno Send
JSON Auth Query Header 1 Docs
1 {
2   "aluno_id" : "2",
3   "disciplinas" : [
4     {
5       "disciplina_id" : "1",
6       "descricao" : "Período da Manhã"
7     },
8     {
9       "disciplina_id" : "2",
10      "descricao" : "Período da Tarde"
11    }
12  ]
13 }
```

### Matriculas

aluno_id	disciplina_id	descricao
1	2	PERÍODO DA MANHÃ
2	1	PERÍODO DA MANHÃ
2	2	PERÍODO DA TARDE

*(Execução da codificação anterior)*