

**INSTITUTO FEDERAL**

Paraná

Campus Paranaguá

# AdonisJs - Autenticação

Desenvolvimento Web

Prof. Diego Stiehl

# Autenticação

---

- O Adonis vem com suporte parcial para autenticação
- Model
  - `app/Models/User.js`
- Migration
  - `database/migrations/1503248427885_user.js`
  - `database/migrations/1503248427886_token.js`
- Apenas models, sem views ou controllers

# Adonis Auth Scaffold

---

- Há um pacote NPM que pode nos ajudar
- Adonis Auth Scaffold
  - <https://github.com/creatrixity/adonis-auth-scaffold>
- Ele é capaz de gerar:
  - Controller de autenticação
  - Migrações e alterações nos models
  - Rotas
  - Várias views (login, cadastro, ...)
  - E-mails no registro e recuperação de senha

# Instalação

---

- Instalar pacote do NPM

```
npm install adonis-auth-scaffold --save-dev
```

- Instalar dependências

```
adonis install @adonisjs/mail  
adonis install @adonisjs/persona  
adonis install @adonisjs/validator
```

# Configuração

---

- start/app.js

```
const providers = [  
  // ... Outros já existentes  
  '@adonisjs/validator/providers/ValidatorProvider',  
  '@adonisjs/mail/providers/MailProvider',  
  '@adonisjs/persona/providers/PersonaProvider'  
];  
const aceProviders = [  
  // ... Outros já existentes  
  'adonis-auth-scaffold/providers/AdonisAuthScaffoldProvider'  
];
```

- start/kernel.js

```
const globalMiddleware = [  
  // ... Outros já existentes  
  'App/Middleware/ViewHelper'  
];
```

# Gerar Recursos

Rotas e views para HTML



```
adonis make:auth --http-only --bootstrap
```



Template e views com Bootstrap

```
└─ adonis make:auth --http-only --bootstrap
Created Controllers/Http/AuthController.js
Created config/adonis-auth-scaffold.js
Successfully created public/auth/auth-styles.css
Created resources/views/auth/emails/password.edge
Created resources/views/auth/emails/welcome-mail.edge
Created resources/views/auth/partials/password-reset-request-form.edge
Created resources/views/auth/partials/password-change-form.edge
Create resources/views/auth/dashboard.edge
Create resources/views/auth/login.edge
Create resources/views/auth/register.edge
Create resources/views/auth/password-reset.edge
Created resources/views/layouts/auth.edge
Created start/authRoutes.js
Created start/authEvents.js
Created app/Middleware/ViewHelper.js
Created database/migrations/1571680044570_add_account_status_to_users_schema.js
```

# Testando

---

- Executar migrações

```
adonis migration:run
```

- Iniciar servidor

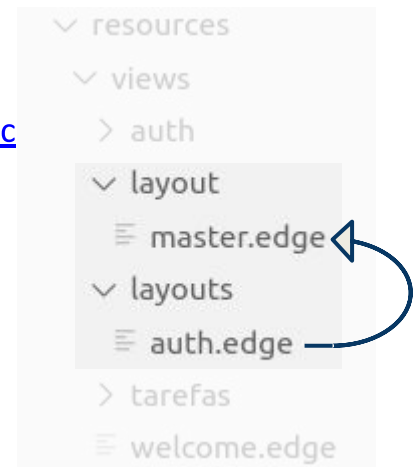
```
adonis serve --dev
```

- Testar autenticação

- <http://127.0.0.1:3333/login>

# Novo Layout

- Vamos reaproveitar template gerado
- Abra este link:
  - <https://gist.github.com/seccomiro/4d263bba07ca4c16e182ac647000445c>
    - Layout mais completo
- Copie o conteúdo para:
  - layout/master.edge
- Apague a pasta layouts





# View Helpers

---

- Template usa um view helper

```
@loggedIn
```

```
Mostra o HTML daqui somente se logado
```

```
@else
```

```
Mostra o HTML daqui somente se deslogado
```

```
@endloggedIn
```

- E uma seção de título customizável

```
<title>
```

```
@section('titulo')
```

```
IFList
```

```
@endsection
```

```
| IFList
```

```
</title>
```

# Acertar Views

- O novo layout muda algumas `@section`
  - Vai gerar problemas nas views
- Para consertar
  - **Ctrl + Shift + H** ← Find and Replace
- Buscar e substituir:
  - `@layout('layouts.auth')` → `@layout('layout.master')`
  - `@section('view-title')` → `@section('titulo')`
  - `@section('content')` → `@section('conteudo')`



# Corrigir Rota Raiz

---

- Abrir: **config/adonis-auth-scaffold.js**
  - Alterar (linha 35): `registrationSuccessRedirectTo: '/',`
- Abrir: **start/routes.js**
  - Apagar linha: `Route.on('/', ).render('welcome');`
  - Adicionar:

```
Route.any('/', ({ response }) => response.redirect('/tarefas'));
```

- Abrir: **start/authRoutes.js**
  - Apagar linha:

```
Route.get(registrationSuccessRoute, "AuthController.getAuthDashboard");
```

# Tratar Erros

---

- Melhorar mensagens de erro de validação
  - Deixar mais bonito
- Criar pasta/arquivo:
  - `app/Exceptions/Handler.js`
- Copiar conteúdo
  - <https://gist.github.com/seccomiro/c53bf148b55bab692bddfbbd8bd83d22>

# Associar Models

---

- Precisamos mapear a nossa relação entre User e Tarefa
  - Um User **tem muitas** Tarefas
  - Uma Tarefa **pertence a** um User
- A relação já deve existir no BD/Migration
  - Mapeamento facilita consultas

# Associar Models

- Adicionar à **app/Models/Tarefa.js**

```
class Tarefa extends Model {  
  // ... Resto da classe  
  user() {  
    return this.belongsTo('App/Models/User');  
  }  
}
```

- Adicionar à **app/Models/User.js**

```
class User extends Model {  
  // ... Resto da classe  
  tarefas() {  
    return this.hasMany('App/Models/Tarefa');  
  }  
}
```

# Proteger Rotas

---

- A autenticação do Adonis fornece um middleware de proteção de rotas
- Mudar rotas em **start/routes.js**:

```
Route.resource('tarefas', 'TarefaController').middleware('auth');  
Route.get('/tarefas/:id/concluida', 'TarefaController.concluida')  
  .as('tarefas.concluida')  
  .middleware('auth');
```

- Todas rotas de **/tarefas** precisam de login
  - Se tentar acessar deslogado, redireciona o usuários para **/login**

# Problema de Segurança

---

- Usuários ainda conseguem ver tarefas criadas por outros usuários
  - Pela URL
  - Teste
- Solução
  - Criar um middleware
  - Aplicar middleware em todas as rotas que usam o ID da tarefa
    - **/tarefas/:id** ← show, edit, update, destroy e concluida



# Criar Middleware

- Execute

```
adonis make:middleware ProtegeTarefa --type=http
```

- Conteúdo de **app/Middleware/ProtegeTarefa.js**

```
'use strict';  
class ProtegeTarefa {  
  async handle(ctx, next) {  
    const { params, auth } = ctx;  
    const tarefa = await auth.user.tarefas().where('id', params.id).first();  
    ctx.tarefa = tarefa;  
    await next();  
  }  
}  
  
module.exports = ProtegeTarefa;
```

← Injeta variável **tarefa** no controller

# Aplicar Middleware


- Registrar middleware em **start/kernel.js**

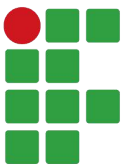
```
const namedMiddleware = {  
  // .. Outros que já existiam  
  protegeTarefa: 'App/Middleware/ProtegeTarefa'  
};
```

- Proteger rotas em **start/routes.js**

```
Route.resource('tarefas', 'TarefaController')  
  .middleware('auth')  
  .middleware(  
    new Map([['show', 'edit', 'update', 'destroy'], ['protegeTarefa']])  
  );  
Route.get('/tarefas/:id/concluida', 'TarefaController.concluida')  
  .as('tarefas.concluida')  
  .middleware(['auth', 'protegeTarefa']);
```

Somente nestas ações do Controller





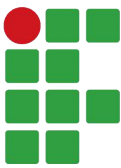
# Ação index

- Abrir **TarefaController.js**
  - Alterar a ação **index**

```
async index({ request, response, view, auth }) {  
  const tarefas = (await auth.user.tarefas()  
    .orderBy('concluida')  
    .orderBy('updated_at', 'desc')  
    .fetch()).rows;  
  return view.render('tarefas.index', { tarefas });  
}
```

Variável injetada pelo  
middleware de autenticação

Busca tarefas somente do usuário  
logado (**auth.user**)



# Ação store

- Abrir **TarefaController.js**
  - Alterar a ação **store**

Variável injetada pelo  
middleware de autenticação

```
async store({ request, response, auth }) {  
  const tarefaData = request.only(['titulo', 'descricao']);  
  tarefaData.user_id = auth.user.id;  
  const tarefa = await Tarefa.create(tarefaData);  
  response.route('tarefas.show', { id: tarefa.id });  
}
```

Vincular usuário logado (**auth.user**)  
à nova tarefa

# Ações no Controller

- Abrir **TarefaController.js**
  - Alterar a ação **show**

Variável injetada pelo middleware

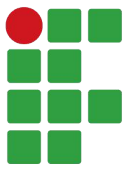
```
async show({ params, request, response, view, tarefa }) {  
const tarefa = await Tarefa.find(params.id);  
return view.render('tarefas.show', { tarefa });  
}
```

Linha desnecessária  
(remover)  
Tarefa já chega pronta

# Prática

---

- Altere as demais ações de TarefaController
  - edit
  - update
  - destroy
  - concluida



# Busca

---

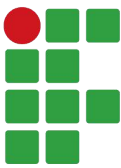
- Nosso template tem um formulário de busca que ainda não funciona
- Ideia
  - Fazer requisição para:
    - <http://127.0.0.1:3333/tarefas?busca=título>
  - Resultado
    - Página principal de tarefas, mas filtrada

# TarefaController

- Modificar ação **index**

```
async index({ request, response, view, auth }) {  
  const busca = request.input('busca') || '';  
  const tarefas = (await auth.user  
    .tarefas()  
    .where('titulo', 'like', `%${busca}%`)  
    .orderBy('concluida')  
    .orderBy('updated_at', 'desc')  
    .fetch()).rows;  
  return view.render('tarefas.index', { tarefas });  
}
```





- Modificar formulário no template

```
<form
  class="form-inline my-2 my-lg-0 "
  action="{{ route('tarefas.index') }}"
  method="get">
  <input
    name="busca"
    value="{{ request.input('busca') || '' }}"
    class="form-control mr-sm-2 " type="search"
    placeholder="Buscar Tarefa" aria-label="Buscar">
  <button class="btn btn-outline-success my-2 my-sm-0 "
type="submit">
  Buscar
</button>
</form>
```