Roteiro de Testes de Autorização por Perfil - Projeto cuidar+

Este roteiro fornece um guia passo a passo para testar as regras de permissão da API, simulando o fluxo de trabalho dos perfis **ADMIN**, **CLIENTE** e **PROFISSIONAL**.

Observação: O valor `<TOKEN_ADMIN`, `<TOKEN_CLIENTE`, `<ID_CLIENTE`, etc., deve ser substituído pelo valor real obtido na etapa anterior.

1. Configuração Inicial e Preparação

1.1. Criar Usuário ADMIN

Ação	Requisição	JSON	Resultado Esperado	Observações
Criar Admin	`POST /auth/register - admin`	(N/A)	`200 OK` (Mensagem de sucesso)	Cria o usuário `admin` com senha `123456`.
Login Admin	`POST /auth/login`	`{"email": "admin", "senha": "123456"}`	`200 OK` (Retorna `<TOKEN_ADMIN>`)	Salve o token para uso nos testes.

1.2. Criar Recursos Base (Usando `<TOKEN_ADMIN`)

Ação	Requisição	JSON	Resultado Esperado	Observações
Criar Profissional	`POST /profissionais`	`json¥n{¥n "nome": "Profissional Teste",¥n "email": "prof@teste.com",¥n "telefone": "11912345678",¥n "especialidade": "Massagista"¥n}¥n`	`201 Created` (Retorna `<ID_PROFISSIONAL>`)	Salve o ID para uso nos testes.
Criar Serviço	`POST /servicos`	`json¥n{¥n "nome": "Massagem Relaxante",¥n "descricao": "Sessão de 60 minutos.",¥n "duracaoEmMinutos": 60,¥n "preco": 120.00¥n}¥n`	`201 Created` (Retorna `<ID_SERVICO>`)	Salve o ID para uso nos testes.

---

2. Testes com o Perfil ADMIN

Pré-condição: Usar o `<TOKEN_ADMIN` em todas as requisições.

Ação	Requisição	JSON	Resultado Esperado	Justificativa
Listar Clientes	`GET /clientes`	(N/A)	`200 OK`	`hasRole('ADMIN')`
Buscar Cliente	`GET /clientes/<ID_CLIENTE>`	(N/A)	`200 OK`	`hasRole('ADMIN')`
Atualizar Cliente	`PUT /clientes/<ID_CLIENTE>`	`json¥n{¥n "nome": "Cliente Admin Update"¥n}¥n`	`200 OK`	`hasRole('ADMIN')`
Listar Agendamentos	`GET /agendamentos`	(N/A)	`200 OK`	Requer autenticação.
Buscar Agendamento	`GET /agendamentos/<ID_AGENDAMENTO>`	(N/A)	`200 OK`	`hasRole('ADMIN')`
Atualizar Agendamento	`PUT /agendamentos/<ID_AGENDAMENTO>`	`json¥n{¥n "status": "CONFIRMADO"¥n}¥n`	`200 OK`	`hasRole('ADMIN')`
Excluir Cliente	`DELETE /clientes/<ID_CLIENTE>`	(N/A)	`204 No Content`	`hasRole('ADMIN')`
Excluir Profissional	`DELETE /profissionais/<ID_PROFISSIONAL>`	(N/A)	`204 No Content`	`hasRole('ADMIN')`
Excluir Serviço	`DELETE /servicos/<ID_SERVICO>`	(N/A)	`204 No Content`	Requer autenticação (assumido como Admin).

3. Testes com o Perfil CLIENTE

3.1. Preparação do CLIENTE

Ação	Requisição	JSON	Resultado Esperado	Observações
Criar Cliente	POST /clientes	json¥n{¥n "nome": "Cliente de Teste",¥n "email": "cliente@teste.com",¥n "telefone": "11999999999",¥n "senha": "senha123"¥n}¥n	201 Created (Retorna <ID_CLIENTE>)	Salve o ID para uso nos testes.
Login Cliente	POST /auth/login	{"email": "cliente@teste.com", "senha": "senha123"}	200 OK (Retorna <TOKEN_CLIENTE>)	Salve o token para uso nos testes.
Criar Agendamento	POST /agendamentos	json¥n{¥n "clienteId": <ID_CLIENTE>,¥n "profissionalId": <ID_PROFISSIONAL>,¥n "servicoId": <ID_SERVICO>,¥n "dataHoraInicio": "2025-12-01T10:00:00"¥n}¥n	201 Created (Retorna <ID_AGENDAMENTO>)	Salve o ID para uso nos testes.

3.2. Testes de Permissão do CLIENTE

Pré-condição: Usar o <TOKEN_CLIENTE em todas as requisições

Ação	Requisição	JSON	Resultado Esperado	Justificativa
Listar Clientes	`GET /clientes`	(N/A)	`403 Forbidden`	Cliente não pode listar todos.
Buscar Próprio Cliente	`GET /clientes/<ID_CLIENTE>`	(N/A)	`200 OK`	`#id == authentication.principal.id`
Buscar Outro Cliente	`GET /clientes/<ID_OUTRO_CLIENTE>`	(N/A)	`403 Forbidden`	Não é o ID principal.
Atualizar Próprio Cliente	`PUT /clientes/<ID_CLIENTE>`	`json¥n{¥n "nome": "Meu Nome Atualizado"¥n}¥n`	`200 OK`	`#id == authentication.principal.id`
Atualizar Outro Cliente	`PUT /clientes/<ID_OUTRO_CLIENTE>`	`json¥n{¥n "nome": "Tentativa Invalida"¥n}¥n`	`403 Forbidden`	Não é o ID principal.
Excluir Próprio Cliente	`DELETE /clientes/<ID_CLIENTE>`	(N/A)	`204 No Content`	`#id == authentication.principal.id`
Listar Profissionais	`GET /profissionais`	(N/A)	`200 OK`	`permitAll()`
Criar Agendamento	`POST /agendamentos`	(JSON de criação)	`201 Created`	`hasAnyRole('CLIENTE', 'ADMIN')`
Buscar Próprio Agendamento	`GET /agendamentos/<ID_AGENDAMENTO>`	(N/A)	`200 OK`	`isAgendamentoOwner`
Buscar Agendamento de Outro	`GET /agendamentos/<ID_AGENDAMENTO_OUTRO>`	(N/A)	`403 Forbidden`	Não é o dono.
Atualizar Agendamento	`PUT /agendamentos/<ID_AGENDAMENTO`	`json¥n{¥n "status": "CANCELADO"¥n}¥`	`403 Forbidden`	Cliente não pode atualizar, apenas cancelar

	>	n		(DELETE).
Cancelar Próprio Agendamento	`DELETE /agendamentos/<ID_AGENDAMENTO>`	(N/A)	`204` No Content	`isAgendamentoOwner`
Excluir Serviço	`DELETE /servicos/<ID_SERVICO>`	(N/A)	`403` Forbidden	Cliente não pode gerenciar serviços.

4. Testes com o Perfil PROFISSIONAL

4.1. Preparação do PROFISSIONAL

Pré-condição: O profissional já foi criado na Seção 1.2.

Ação	Requisição	JSON	Resultado Esperado	Observações
Login Profissional	POST /auth/login	{"email": "prof@teste.com", "senha": "senha123"}	200 OK (Retorna <TOKEN_PROFISSIONAL>)	**Nota:** O Profissional precisa ter uma senha definida no serviço ou no banco de dados para poder logar. Assumimos que a senha foi definida como senha123 para este teste.
Criar Agendamento	POST /agendamentos	json¥n{¥n "clienteId": <ID_CLIENTE>,¥n "profissionalId": <ID_PROFISSIONAL>,¥n "servicoId": <ID_SERVICO>,¥n "dataHoraInicio": "2025-12-01T11:00:00"¥n}¥n	201 Created (Retorna <ID_AGENDAMENTO_PROF>)	Agendamento onde o profissional é o relacionado.

4.2. Testes de Permissão do PROFISSIONAL

Pré-condição: Usar o <TOKEN_PROFISSIONAL em todas as requisições.

Ação	Requisição	JSON	Resultado Esperado	Justificativa
Listar Clientes	`GET /clientes`	(N/A)	`403 Forbidden`	Profissional não pode listar todos.
Buscar Próprio Profissional	`GET /profissionais/<ID_PROFISSIONAL>`	(N/A)	`200 OK`	`#id == authentication.principal.id`
Buscar Outro Profissional	`GET /profissionais/<ID_OUTRO_PROFISSIONAL>`	(N/A)	`403 Forbidden`	Não é o ID principal.
Atualizar Próprio Profissional	`PUT /profissionais/<ID_PROFISSIONAL>`	`json¥n{¥n "telefone": "11988887777"¥n}¥n`	`200 OK`	`#id == authentication.principal.id`
Atualizar Outro Profissional	`PUT /profissionais/<ID_OUTRO_PROFISSIONAL>`	(JSON de atualização)	`403 Forbidden`	Não é o ID principal.
Excluir Próprio Profissional	`DELETE /profissionais/<ID_PROFISSIONAL>`	(N/A)	`204 No Content`	`#id == authentication.principal.id`
Criar Agendamento	`POST /agendamentos`	(JSON de criação)	`403 Forbidden`	Profissional não tem a role `CLIENTE` ou `ADMIN`.
Buscar Agendamento Relacionado	`GET /agendamentos/<ID_AGENDAMENTO_PROF>`	(N/A)	`200 OK`	`isAgendamentoForProfessional`
Buscar Agendamento Não Relacionado	`GET /agendamentos/<ID_AGENDAMENTO_OUTRO>`	(N/A)	`403 Forbidden`	Não é o profissional relacionado.

Atualizar Agendamento Relacionado	`PUT /agendamentos/<ID_AGENDAMENTO_PROF>`	`json¥n{¥n "status": "CONCLUIDO"¥n}¥n`	`200 OK`	`isAgendamentoForProfessional`
Atualizar Agendamento Não Relacionado	`PUT /agendamentos/<ID_AGENDAMENTO_OUTRO>`	(JSON de atualização)	`403 Forbidden`	Não é o profissional relacionado.
Excluir Serviço	`DELETE /servicos/<ID_SERVICO>`	(N/A)	`403 Forbidden`	Profissional não pode gerenciar serviços.

5. Testes de Acesso Público (`permitAll()`)

Ação	Requisição	JSON	Resultado Esperado	Justificativa
Listar Profissionais	`GET /profissionais`	(N/A)	`200 OK`	`permitAll()` no SecurityConfig.
Listar Serviços	`GET /servicos`	(N/A)	`200 OK`	`permitAll()` no SecurityConfig.
Criar Cliente	`POST /clientes`	(JSON de criação)	`201 Created`	`permitAll()` no SecurityConfig.
Acesso a endpoint autenticado	`GET /agendamentos`	(N/A)	`401 Unauthorized`	`anyRequest().authenticated()`

Este roteiro cobre o fluxo completo de testes de autorização para as três entidades principais. Certifique-se de seguir a ordem e substituir os IDs e tokens conforme necessário.