**Documentação da API**

**Visão Geral**

Esta API fornece endpoints para gerenciamento de usuários, profissionais, serviços, agendas e horários de profissionais. As operações básicas incluem CRUD (Criar, Ler, Atualizar e Excluir) para cada entidade, bem como funcionalidades adicionais como autenticação e verificação de disponibilidade.

**Endpoints**

**1. Usuários**

* **Obter todos os usuários**
  + **Endpoint:** /usuarios
  + **Método:** GET
  + **Autenticação:** validar\_token, verificar\_admin
  + **Resposta:**

json

Copiar código

{

"usuarios": [

{

"id\_usuario": 1,

"nome": "Nome do Usuário",

"email": "email@example.com",

"telefone": "1234567890"

},

...

]

}

* + **Erros:**
    - 500 Internal Server Error se ocorrer um erro no banco de dados
* **Obter um usuário específico**
  + **Endpoint:** /usuario/<int:id\_usuario>
  + **Método:** GET
  + **Parâmetros:** id\_usuario (int)
  + **Autenticação:** validar\_token
  + **Resposta:**

json

Copiar código

{

"id\_usuario": 1,

"nome": "Nome do Usuário",

"email": "email@example.com",

"telefone": "1234567890"

}

* + **Erros:**
    - 404 Not Found se o usuário não for encontrado
    - 500 Internal Server Error para outros erros
* **Criar um novo usuário**
  + **Endpoint:** /usuario
  + **Método:** POST
  + **Corpo da Requisição:**

json

Copiar código

{

"nome": "Nome do Usuário",

"email": "email@example.com",

"telefone": "1234567890",

"senha": "SenhaSegura123!",

"ativo": 1,

"tentativas\_login": 0,

"administrador": 0

}

* + **Resposta:**

json

Copiar código

{

"message": "Novo usuário criado com sucesso!"

}

* + **Erros:**
    - 400 Bad Request se a senha não atender aos requisitos
    - 500 Internal Server Error para outros erros
* **Atualizar um usuário**
  + **Endpoint:** /usuario/<int:id\_usuario>
  + **Método:** PUT
  + **Parâmetros:** id\_usuario (int)
  + **Corpo da Requisição:**

json

Copiar código

{

"nome": "Novo Nome",

"email": "novoemail@example.com",

"telefone": "0987654321",

"ativo": 1,

"administrador": 0

}

* + **Resposta:**

json

Copiar código

{

"message": "Usuário atualizado com sucesso!"

}

* + **Erros:**
    - 404 Not Found se o usuário não for encontrado
    - 500 Internal Server Error para outros erros
* **Login de usuário**
  + **Endpoint:** /usuario/login
  + **Método:** POST
  + **Corpo da Requisição:**

json

Copiar código

{

"email": "email@example.com",

"senha": "SenhaSegura123!"

}

* + **Resposta:**

json

Copiar código

{

"message": "Login bem-sucedido!",

"token": "jwt\_token\_aqui"

}

* + **Erros:**
    - 400 Bad Request se email ou senha não forem fornecidos
    - 401 Unauthorized se a senha estiver incorreta
    - 403 Forbidden se a conta estiver inativa ou o número máximo de tentativas de login for excedido
    - 404 Not Found se o email não for encontrado
    - 500 Internal Server Error para outros erros

**2. Profissionais**

* **Obter todos os profissionais**
  + **Endpoint:** /profissionais
  + **Método:** GET
  + **Autenticação:** validar\_token, verificar\_admin
  + **Resposta:**

json

Copiar código

{

"profissionais": [

{

"id\_profissional": 1,

"nome": "Nome do Profissional",

"telefone": "1234567890"

},

...

]

}

* + **Erros:**
    - 500 Internal Server Error se ocorrer um erro no banco de dados
* **Obter um profissional específico**
  + **Endpoint:** /profissional/<int:id\_profissional>
  + **Método:** GET
  + **Parâmetros:** id\_profissional (int)
  + **Autenticação:** validar\_token
  + **Resposta:**

json

Copiar código

{

"id\_profissional": 1,

"nome": "Nome do Profissional",

"telefone": "1234567890"

}

* + **Erros:**
    - 404 Not Found se o profissional não for encontrado
    - 500 Internal Server Error para outros erros
* **Verificar Profissionais Disponíveis**
  + **Endpoint:** /profissionais\_disponiveis?data\_hora=2024-07-23T10:00:00&id\_servico=1
  + **Método:** GET
  + **Parâmetros:** data\_hora, id\_servico
  + **Autenticação:** validar\_token
  + **Resposta:**
  + **200 OK**: Retorna a lista de profissionais disponíveis.

json

Copiar código

{

"profissionais\_disponiveis": [

{

"id\_profissional": 1,

"nome": "Nome do Profissional",

"telefone": "1234567890"

},

...

]

}

* + Erros:
    - **400 Bad Request**: Retorna um erro se os parâmetros data\_hora ou id\_servico estiverem ausentes.
    - **500 Internal Server Error**: Retorna um erro se houver uma exceção durante a execução da função verificar\_disponibilidade.
* **Criar um novo profissional**
  + **Endpoint:** /profissional
  + **Método:** POST
  + **Corpo da Requisição:**

json

Copiar código

{

"nome": "Nome do Profissional",

"telefone": "1234567890",

"ativo": 1

}

* + **Resposta:**

json

Copiar código

{

"message": "Novo profissional criado com sucesso!"

}

* + **Erros:**
    - 500 Internal Server Error para outros erros
* **Atualizar um profissional**
  + **Endpoint:** /profissional/<int:id\_profissional>
  + **Método:** PUT
  + **Parâmetros:** id\_profissional (int)
  + **Corpo da Requisição:**

json

Copiar código

{

"nome": "Nome Atualizado",

"telefone": "0987654321",

"ativo": 1

}

* + **Resposta:**

json

Copiar código

{

"message": "Profissional atualizado com sucesso!"

}

* + **Erros:**
    - 404 Not Found se o profissional não for encontrado
    - 500 Internal Server Error para outros erros
* **Excluir um profissional**
  + **Endpoint:** /profissional/<int:id\_profissional>
  + **Método:** DELETE
  + **Parâmetros:** id\_profissional (int)
  + **Autenticação:** validar\_token, verificar\_admin
  + **Resposta:**

json

Copiar código

{

"message": "Profissional excluído com sucesso!"

}

* + **Erros:**
    - 404 Not Found se o profissional não for encontrado
    - 500 Internal Server Error para outros erros

**3. Serviços**

* **Obter todos os serviços**
  + **Endpoint:** /servicos
  + **Método:** GET
  + **Autenticação:** validar\_token
  + **Resposta:**

json

Copiar código

{

"servicos": [

{

"id\_servico": 1,

"nome": "Nome do Serviço",

"descricao": "Descrição do Serviço",

"valor": 100.00,

"tempo": "01:00:00"

},

...

]

}

* + **Erros:**
    - 500 Internal Server Error se ocorrer um erro no banco de dados
* **Obter um serviço específico**
  + **Endpoint:** /servico/<int:id\_servico>
  + **Método:** GET
  + **Parâmetros:** id\_servico (int)
  + **Autenticação:** validar\_token
  + **Resposta:**

json

Copiar código

{

"id\_servico": 1,

"nome": "Nome do Serviço",

"descricao": "Descrição do Serviço",

"valor": 100.00,

"tempo": "01:00:00"

}

* + **Erros:**
    - 404 Not Found se o serviço não for encontrado
    - 500 Internal Server Error para outros erros
* **Criar um novo serviço**
  + **Endpoint:** /servico
  + **Método:** POST
  + **Corpo da Requisição:**

json

Copiar código

{

"nome": "Nome do Serviço",

"descricao": "Descrição do Serviço",

"valor": 100.00,

"tempo": "01:00:00",

"ativo": 1,

"file": (arquivo)

}

* + **Resposta:**

json

Copiar código

{

"message": "Serviço criado com sucesso!",

"file\_path": "/caminho/do/arquivo"

}

* + **Erros:**
    - 400 Bad Request se o arquivo não for enviado ou for do tipo não permitido
    - 500 Internal Server Error para outros erros
* **Atualizar um serviço**
  + **Endpoint:** /servico/<int:id\_servico>
  + **Método:** PUT
  + **Parâmetros:** id\_servico (int)
  + **Corpo da Requisição:**

json

Copiar código

{

"nome": "Nome Atualizado",

"descricao": "Descrição Atualizada",

"valor": 150.00,

"tempo": "01:30:00",

"ativo": 1

}

* + **Resposta:**

json

Copiar código

{

"message": "Serviço atualizado com sucesso!"

}

* + **Erros:**
    - 404 Not Found se o serviço não for encontrado
    - 500 Internal Server Error para outros erros
* **Excluir um serviço**
  + **Endpoint:** /servico/<int:id\_servico>
  + **Método:** DELETE
  + **Parâmetros:** id\_servico (int)
  + **Autenticação:** validar\_token, verificar\_admin
  + **Resposta:**

json

Copiar código

{

"message": "Serviço excluído com sucesso!"

}

* + **Erros:**
    - 404 Not Found se o serviço não for encontrado
    - 500 Internal Server Error para outros erros

**4. Agendas**

* **Obter todas as agendas**
  + **Endpoint:** /agendas
  + **Método:** GET
  + **Autenticação:** validar\_token
  + **Resposta:**

json

Copiar código

{

"agendas": [

{

"id\_agenda": 1,

"id\_profissional": 1,

"data": "2024-07-23",

"hora\_inicio": "09:00:00",

"hora\_fim": "17:00:00"

},

...

]

}

* + **Erros:**
    - 500 Internal Server Error se ocorrer um erro no banco de dados
* **Obter uma agenda específica**
  + **Endpoint:** /agenda/<int:id\_agenda>
  + **Método:** GET
  + **Parâmetros:** id\_agenda (int)
  + **Autenticação:** validar\_token
  + **Resposta:**

json

Copiar código

{

"id\_agenda": 1,

"id\_profissional": 1,

"data": "2024-07-23",

"hora\_inicio": "09:00:00",

"hora\_fim": "17:00:00"

}

* + **Erros:**
    - 404 Not Found se a agenda não for encontrada
    - 500 Internal Server Error para outros erros
* **Criar uma nova agenda**
  + **Endpoint:** /agenda
  + **Método:** POST
  + **Corpo da Requisição:**

json

Copiar código

{

"id\_profissional": 1,

"data": "2024-07-23",

"hora\_inicio": "09:00:00",

"hora\_fim": "17:00:00"

}

* + **Resposta:**

json

Copiar código

{

"message": "Nova agenda criada com sucesso!"

}

* + **Erros:**
    - 400 Bad Request se a data ou hora não forem válidas
    - 500 Internal Server Error para outros erros
* **Atualizar uma agenda**
  + **Endpoint:** /agenda/<int:id\_agenda>
  + **Método:** PUT
  + **Parâmetros:** id\_agenda (int)
  + **Corpo da Requisição:**

json

Copiar código

{

"data": "2024-07-24",

"hora\_inicio": "10:00:00",

"hora\_fim": "18:00:00"

}

* + **Resposta:**

json

Copiar código

{

"message": "Agenda atualizada com sucesso!"

}

* + **Erros:**
    - 404 Not Found se a agenda não for encontrada
    - 500 Internal Server Error para outros erros
* **Excluir uma agenda**
  + **Endpoint:** /agenda/<int:id\_agenda>
  + **Método:** DELETE
  + **Parâmetros:** id\_agenda (int)
  + **Autenticação:** validar\_token, verificar\_admin
  + **Resposta:**

json

Copiar código

{

"message": "Agenda excluída com sucesso!"

}

* + **Erros:**
    - 404 Not Found se a agenda não for encontrada
    - 500 Internal Server Error para outros erros

**5. Horários de Profissionais**

* **Obter todos os horários de um profissional**
  + **Endpoint:** /horarios/<int:id\_profissional>
  + **Método:** GET
  + **Parâmetros:** id\_profissional (int)
  + **Autenticação:** validar\_token
  + **Resposta:**

json

Copiar código

{

"horarios": [

{

"id\_horario": 1,

"id\_profissional": 1,

"data": "2024-07-23",

"hora\_inicio": "09:00:00",

"hora\_fim": "17:00:00"

},

...

]

}

* + **Erros:**
    - 404 Not Found se o profissional não for encontrado
    - 500 Internal Server Error para outros erros
* **Adicionar um horário para um profissional**
  + **Endpoint:** /horario
  + **Método:** POST
  + **Corpo da Requisição:**

json

Copiar código

{

"id\_profissional": 1,

"data": "2024-07-23",

"hora\_inicio": "09:00:00",

"hora\_fim": "17:00:00"

}

* + **Resposta:**

json

Copiar código

{

"message": "Horário adicionado com sucesso!"

}

* + **Erros:**
    - 400 Bad Request se a data ou hora não forem válidas
    - 500 Internal Server Error para outros erros
* **Atualizar um horário de um profissional**
  + **Endpoint:** /horario/<int:id\_horario>
  + **Método:** PUT
  + **Parâmetros:** id\_horario (int)
  + **Corpo da Requisição:**

json

Copiar código

{

"data": "2024-07-24",

"hora\_inicio": "10:00:00",

"hora\_fim": "18:00:00"

}

* + **Resposta:**

json

Copiar código

{

"message": "Horário atualizado com sucesso!"

}

* + **Erros:**
    - 404 Not Found se o horário não for encontrado
    - 500 Internal Server Error para outros erros
* **Excluir um horário**
  + **Endpoint:** /horario/<int:id\_horario>
  + **Método:** DELETE
  + **Parâmetros:** id\_horario (int)
  + **Autenticação:** validar\_token, verificar\_admin
  + **Resposta:**

json

Copiar código

{

"message": "Horário excluído com sucesso!"

}

* + **Erros:**
    - 404 Not Found se o horário não for encontrado
    - 500 Internal Server Error para outros erros

**Autenticação**

* **Token JWT**
  + Todos os endpoints protegidos exigem um token JWT válido para autenticação.
  + O token deve ser incluído no cabeçalho Authorization da requisição: Authorization: Bearer <token>.

**Erros Comuns**

* **400 Bad Request**: Solicitação inválida, geralmente devido a dados de entrada incorretos.
* **401 Unauthorized**: Falha de autenticação devido a credenciais inválidas.
* **403 Forbidden**: Acesso negado, geralmente devido a permissões insuficientes.
* **404 Not Found**: Recurso solicitado não encontrado.
* **500 Internal Server Error**: Erro interno do servidor.