



**FICHA NACIONAL  
DE REGISTRO  
DE HÓSPEDES**

## Documento de Integração

Versão 1.3 de 06/01/2025

Introdução .....	7
Visão Geral.....	7
Descrição .....	7
Versão mais recente .....	7
URL Base .....	8
Autenticação – Basic Authentication .....	8
Como funciona.....	8
Usuário e senha.....	8
Como codificar em Base64 .....	8
Linux:.....	8
Windows:.....	9
Exemplo prático .....	9
Observações .....	9
Exemplo de fluxo de uso dos endpoints .....	10
Endpoints.....	11
Orientações gerais - válidas para todos os endpoints .....	11
Domínios .....	11
Descrição .....	11
Endpoints .....	11
GET /dominios/fnrh/meios_transporte .....	12
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /dominios/fnrh/motivos_viagem .....	12
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /dominios/hospedes/situacoes .....	12
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /dominios/pessoas/generos .....	12
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /dominios/pessoas/opcao_deficiencia .....	12
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /dominios/pessoas/racas.....	12
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /dominios/pessoas/tipos_deficiencia .....	12

Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /dominios/pessoas/tipos_documento.....	12
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /dominios/pessoas/tipos_nacionalidade.....	12
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
🔍Hospedagem .....	13
Descrição .....	13
Endpoints .....	13
POST /hospedagem/registrar.....	13
Parâmetros de consulta (Query Parameters) .....	13
Body - Orientações .....	13
Exemplo de Body.....	<b>Erro! Indicador não definido.</b>
Descrição dos campos do Body .....	14
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
Hóspedes .....	16
Descrição .....	16
Endpoints .....	16
PATCH /hospedes/{hospede_id}/checkin .....	16
Parâmetros de rota (Path Parameters) .....	16
Exemplo de body .....	<b>Erro! Indicador não definido.</b>
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
PATCH /hospedes/{hospede_id}/checkout.....	17
Parâmetros de rota (Path Parameters) .....	17
Exemplo de body .....	<b>Erro! Indicador não definido.</b>
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
PATCH /hospedes/{hospede_id}/no-show.....	17
Parâmetros de rota (Path Parameters) .....	17
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
GET /hospedes/{hospede_id} .....	18
Parâmetros de rota (Path Parameters) .....	18
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
GET /hospedes/pre-checkins .....	18

Parâmetros de consulta (Query Parameters) .....	18
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
Pessoas .....	19
Descrição .....	19
Endpoints .....	19
GET /pessoas/{pessoa_id} .....	20
Parâmetros de rota (Path Parameters) .....	20
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
GET /pessoas/documento/{tipo_documento}/{numero_documento} .....	21
Parâmetros de rota (Path Parameters) .....	21
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
POST /pessoas .....	21
Parâmetros de rota (Path Parameters) .....	21
Body - Orientações .....	21
Exemplo de Body .....	<b>Erro! Indicador não definido.</b>
Exemplo de resposta .....	<b>Erro! Indicador não definido.</b>
Reservas .....	22
Descrição .....	22
Endpoints .....	22
GET /reservas .....	22
Parâmetros de rota (Path Parameters) .....	22
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
POST /reservas .....	23
Body - Orientações .....	23
Exemplo de Body .....	<b>Erro! Indicador não definido.</b>
Descrição dos campos do Body .....	24
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
GET /reservas/{reserva_id} .....	25
Parâmetros de rota (Path Parameters) .....	25
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
DEL /reservas/{reserva_id} .....	25
Parâmetros de rota (Path Parameters) .....	25

PUT /reservas/{reserva_id} .....	26
Parâmetros de rota (Path Parameters) .....	26
Exemplo de body .....	<b>Erro! Indicador não definido.</b>
POST /reservas/{reserva_id}/cancelar .....	26
Parâmetros de rota (Path Parameters) .....	26
POST /reservas/{reserva_id}/checkin .....	27
Parâmetros de rota (Path Parameters) .....	27
Exemplo de body .....	<b>Erro! Indicador não definido.</b>
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
POST /reservas/{reserva_id}/checkout .....	28
Parâmetros de rota (Path Parameters) .....	28
Exemplo de body .....	<b>Erro! Indicador não definido.</b>
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
GET /reservas/{reserva_id}/hospedes.....	29
Parâmetros de rota (Path Parameters) .....	29
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
POST /reservas/{reserva_id}/hospedes .....	29
Parâmetros de rota (Path Parameters) .....	29
Body - Orientações .....	29
Exemplo de Body.....	<b>Erro! Indicador não definido.</b>
POST /reservas/{reserva_id}/vincular-hospede/{hospede_id}.....	29
Parâmetros de rota (Path Parameters) .....	29
Exemplo de Resposta .....	<b>Erro! Indicador não definido.</b>
Erros Comuns.....	29
400 Bad Request.....	29
Exemplo: .....	30
401 Unauthorized .....	30
Exemplo: .....	30
403 Not Found .....	30
Exemplo: .....	30
404 Not Found .....	30
Exemplo: .....	30

409 Conflict.....	30
Exemplo: .....	31
422 Unprocessable Entity.....	31
Exemplo: .....	31
500 Internal Server Error .....	31
Exemplo: .....	31
Perguntas Frequentes (FAQ) .....	31
Qual o funcionamento de "Pré-Check-ins"?.....	31
Qual a diferença entre "Get Pessoa" e "Get Hospede"? .....	31
Como devo proceder para registrar ou consultar dados de hóspedes sem a figura da reserva (ou seja, sem vincular a uma reserva imediatamente)? .....	32
Quais são os valores aceitos para o filtro situacao em GET /reservas?.....	32
Os endpoints são idempotentes? Posso reenviar a mesma requisição POST/PUT/PATCH em caso de falha de rede? .....	33
Qual o formato esperado para datas e horas (timestamps) nos campos da API? E qual fuso horário é utilizado? .....	33
Glossário.....	35

# Documentação de API REST

Documentação das API's de integração do FNRH

## Introdução

Este documento tem como objetivo apresentar a especificação técnica da API REST desenvolvida para o gerenciamento dos dados que geram as Fichas Nacionais de Registro de Hóspedes (FNRH).

Este documento descreve todos os endpoints disponíveis, seus respectivos fluxos de uso, exemplos de requisição e resposta, além de boas práticas de consumo da API.

## Visão Geral

### Descrição

A API foi projetada para ser simples, clara e eficiente, atendendo tanto aos sistemas internos quanto a integrações com parceiros externos (como PMS - *Property Management System*, ou Sistema de Gestão de propriedades Hoteleiras, em tradução livre). Através desta interface, é possível realizar ações como:

- Criar e gerenciar reservas;
- Registrar check-ins e check-outs de hóspedes;
- Consultar dados de pessoas e hóspedes;
- Cancelar reservas e registrar no-shows; e
- Executar fluxos completos de hospedagem em uma única chamada.

A autenticação com a API é feita via HTTP Basic Authentication, garantindo segurança e controle de acesso sobre os dados transacionados.

### Versão mais recente

A versão mais recente deste documento pode ser obtida no item MANUAIS do site do FNRH do Ministério do Turismo:

<https://www.gov.br/turismo/pt-br/acesso-a-informacao/acoes-e-programas/programas-projetos-acoes-obras-e-atividades/ficha-nacional-de-registro-de-hospedes>

## URL Base

Homologação: [https://hom-lowcode.serpro.gov.br/FNRH\\_API/rest/v1/](https://hom-lowcode.serpro.gov.br/FNRH_API/rest/v1/)

Produção: [https://fnrh.turismo.serpro.gov.br/FNRH\\_API/rest/v1/](https://fnrh.turismo.serpro.gov.br/FNRH_API/rest/v1/)

**Comentado [EA1]:** Adicionar essa URL quando o sistema for para produção

## Autenticação – Basic Authentication

A autenticação da API é feita utilizando o padrão HTTP Basic Authentication, que exige o envio de um par de credenciais (usuário e senha) codificados em Base64 no cabeçalho Authorization de cada requisição.

### Como funciona

O cliente da API deve enviar o cabeçalho Authorization no seguinte formato:

```
Authorization: Basic <credenciais_codificadas>
```

Onde <credenciais\_codificadas> é a string resultante da codificação em base64 do texto:

```
usuario:senha
```

### Usuário e senha

Para ter acesso ao usuário e senha, é necessário acessar o sistema FNRH\_SRH e navegar até o módulo Chave de Acesso API's.

Nessa tela são exibidos os dados de usuário e senha (chave).

Caso necessário, é possível gerar uma nova chave, de modo que ao realizar esse procedimento, a chave antiga será inutilizada.

### Como codificar em Base64

Para codificar em base64 é preciso seguir o passo a passo abaixo de acordo com o Sistema Operacional em uso:

#### Linux:

1. Abra o Terminal;
2. Separe o texto a ser codificado. Para esse exemplo utilizaremos "joao:senha123";
3. Digite o seguinte comando:

```
echo "joao:senha123" | base64
```

```
echo "joao:senha123" | base64
```



4. O terminal retornará o texto entre aspas codificado em base 64 e você obterá um retorno como o ilustrado a seguir:

```
Terminal Output:
linux-console@hp34:~$ echo joao:senha123 | base64
am9hbzpzZW5oYTEyMw==
```

#### Windows:

1. Abra o Prompt de Comando;
2. Separe o texto a ser codificado. Para esse exemplo utilizaremos “joao:senha123”;
3. Digite o seguinte comando:

```
[Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes("joao:senha123"))
[Convert]::ToBase64String([System.Text.Encoding]::UTF8.GetBytes("joao:senha123"))
```

4. O terminal retornará o texto entre aspas codificado em base 64 e você obterá um retorno como o ilustrado a seguir:

```
am9hbzpzZW5oYTEyMw==
```

### Exemplo prático

Suponha que o nome de usuário seja empresa123 e a senha seja senhaSegura!

Concatene os dados:

```
empresa123:senhaSegura!
```

Codifique em base64:

```
ZW1wcmVzYTEyMzpzZW5oYVNlZ3VyYSE=
```

Envie o cabeçalho HTTP:

```
Authorization: Basic ZW1wcmVzYTEyMzpzZW5oYVNlZ3VyYSE=
```

### Observações

**Governança:** As credenciais devem ser geradas e fornecidas pela equipe responsável do meio de hospedagem, através do Sistema de Registro de Hóspedes.

**Comunicação:** A comunicação com a API deve ser feita via HTTPS para garantir a segurança da autenticação.

**Segurança:** A chave de acesso (senha) não expira. Caso uma nova chave seja gerada no sistema, a anterior será invalidada permanentemente. Guarde suas credenciais com segurança.

## Exemplo de fluxo de uso dos endpoints

Existem diversos fluxos possíveis de endpoints para o envio das informações ao MTur. Aqui listamos apenas um deles, em sua forma mais direta, passo a passo:

1) Criar reserva

POST /reservas

2) Acrescentar +1 hóspede na reserva já criada

GET /pessoas/documento/{tipo}/{número} (para buscar o id da pessoa)

Se existe a pessoa, retorna o id

Se não existe a pessoa, ela tem que criada

POST /pessoas

POST reservas/{reserva id}/hospedes

3) VINCULAR o pré check-in de balcão na reserva

POST /reservas/{reserva\_id}/vincular-hospede/{hospede\_id}

4) Efetuar o check-in

4.1) Individual de cada hóspede:

PATCH /hospedes/{hospede\_id}/checkin

4.2) De todos os hóspedes da reserva (na mesma data/hora)

POST /reservas/{reserva\_id}/checkin

5) Efetuar o check-out

5.1) Individual de cada hóspede:

PATCH /hospedes/{hospede\_id}/checkout

5.2) De todos os hóspedes da reserva (na mesma data/hora)

POST /reservas/{reserva\_id}/checkout

Além do fluxo passo a passo apresentado acima, uma outra opção é informar todos esses passos em apenas um método:

POST /hospedagem/registrar

## Endpoints

### Orientações gerais - válidas para todos os endpoints

- Os scripts atualizados de cada endpoint podem ser consultados no respectivo swagger de cada ambiente ([aqui](#))
- Código de país:** definidos pelo IBGE conforme Registro de Referência de Países do órgão. A documentação do serviço pode ser acessada [aqui](#)

## Domínios

### Descrição

Esses endpoints fornecem listas de valores fixos (domínios) utilizados em outros endpoints da API, como tipos de documento, motivos de viagem etc. É uma boa prática consultar esses endpoints para garantir que os valores enviados em outras requisições sejam válidos.

### Endpoints

Método	URL	Descrição
GET	/dominios/fnrh/meios_transporte	Lista os meios de transporte.
GET	/dominios/fnrh/motivos_viagem	Lista os motivos de viagem.
GET	/dominios/hospedes/situacoes	Lista as possíveis situações de um hóspede.
GET	/dominios/pessoas/generos	Lista os gêneros.
GET	/dominios/pessoas/opcao_deficiencia	Lista as opções para declaração de deficiência.
GET	/dominios/pessoas/racas	Lista as raças.

GET	/dominios/pessoas/tipos_deficiencia	Lista os tipos de deficiência.
GET	/dominios/pessoas/tipos_documento	Lista os tipos de documentos aceitos.
GET	/dominios/pessoas/tipos_nacionalidade	Lista os tipos de nacionalidade.

[GET /dominios/fnrh/meios\\_transporte](#)

Retorna a lista de meios de transporte.

[GET /dominios/fnrh/motivos\\_viagem](#)

Retorna a lista de motivos de viagem.

[GET /dominios/hospedes/situacoes](#)

Retorna a lista de situações de hóspedes.

[GET /dominios/pessoas/generos](#)

Retorna a lista de gêneros de pessoas.

[GET /dominios/pessoas/opcao\\_deficiencia](#)

Retorna a lista de opções de deficiências de pessoas.

[GET /dominios/pessoas/racas](#)

Retorna a lista de raças de pessoas.

[GET /dominios/pessoas/tipos\\_deficiencia](#)

Retorna a lista de tipos de deficiências de pessoas.

[GET /dominios/pessoas/tipos\\_documento](#)

Retorna a lista de tipos de documentos de pessoas.

[GET /dominios/pessoas/tipos\\_nacionalidade](#)

Retorna a lista de tipos de nacionalidade de pessoas.

## 🔍 Hospedagem

### Descrição

Endpoint principal que permite registrar uma reserva e seus hóspedes em uma única chamada.

### Endpoints

Método	URL	Descrição
POST	/hospedagem/registrar	Cria a reserva e registra os hóspedes.

### POST /hospedagem/registrar

Cria a reserva e registra os hóspedes.

#### Parâmetros de consulta (Query Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
cpf_solicitante	string	Sim	CPF do usuário que está realizando o registro.

#### Body - Orientações

- origem\_reserva\_id – opções: “MEIOHOSPEDAGEM” ou “OTA”
- responsável - se aplica quando o hóspede é um dependente. Nesse caso, aqui deve ser informado o tipo e o número do documento do responsável pelo dependente
- situacao\_hospede – Uma destas opções:
  - "PRECHECKIN\_PENDENTE"
    - checkin\_em, checkout\_em e dados\_ficha não devem ser informados
  - "CHECKIN\_REALIZADO"
    - Nesse caso deve ser informado checkin\_em e dados\_ficha
  - "CHECKOUT\_REALIZADO"
    - Nesse caso devem ser informados checkin\_em, checkout\_em e dados\_ficha
  - "NOSHOW"
    - checkin\_em, checkout\_em e dados\_ficha não devem ser informados
- nacionalidade: Código IBGE do País de residência do hóspede
- estado\_id: Sigla IBGE do Estado de residência do hóspede

### Descrição dos campos do Body

Campo	Tipo	Obrigatório	Descrição
reserva	object	Sim	Objeto contendo os detalhes da reserva.
reserva.numero_reserva	string	Sim	Código único da reserva no sistema do hotel/PMS.
reserva.numero_reserva_ota	string	Não	Código da reserva na agência de viagens online (OTA), se aplicável.
reserva.data_entrada	date	Sim	Data de entrada no formato AAAA-MM-DD.
reserva.data_saida	date	Sim	Data de saída no formato AAAA-MM-DD.
reserva.quantidade_hospede_adulto	integer	Não	Número de adultos na reserva.
reserva.quantidade_hospede_menor	integer	Não	Número de menores de idade na reserva.
reserva.origem_reserva_id	string	Não	ID da origem da reserva. Ex: MEIOHOSPEDAGEM.
dados_hospede	array	Não	Lista de objetos, onde cada objeto representa um hóspede.
dados_hospede[].is_principal	boolean	Não	true se este for o hóspede principal/responsável pela reserva.
dados_hospede[].situacao_hospede	string	Não	Situação inicial do hóspede. Use CHECKIN_REALIZADO para registrar a entrada. Veja os valores em /dominios/hospedes/situacoes.
dados_hospede[].check_in_em	date	Opcional	Data e hora do check-in no formato ISO 8601 (AAAA-MM-DDTHH:mm:ss.sssZ). Obrigatório se situacao_hospede for CHECKIN_REALIZADO.
dados_hospede[].check_out_em	date	Não	Data e hora do check-out no formato ISO 8601.
dados_hospede[].dados_pessoais	object	Não	Objeto com as informações pessoais do hóspede.
...	...	Não	Os demais campos, como nome, documento_id, etc, devem ser preenchidos conforme a necessidade.



# Hóspedes

## Descrição

Endpoints para gerenciar o ciclo de vida de um hóspede individualmente (check-in, check-out, etc) e consultar seus dados.

## Endpoints

Método	URL	Descrição
PATCH	/hospedes/{hospede_id}/checkin	Realiza o check-in de um hóspede.
PATCH	/hospedes/{hospede_id}/checkout	Realiza o check-out de um hóspede.
PATCH	/hospedes/{hospede_id}/no-show	Marca um hóspede como “no-show” (não comparecimento.)
GET	/hospedes/{hospede_id}	Consulta os dados detalhados de um hóspede específico.
GET	/hospedes/pre-checkins	Lista informações de pré-check-in de hóspede.

### PATCH /hospedes/{hospede\_id}/checkin

Realiza o check-in de um hóspede, atualizando sua situação e a data/hora do evento.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
hospede_id	string	Sim	O ID único do hóspede a ser atualizado.



### PATCH /hospedes/{hospede\_id}/checkout

Realiza o check-out de um hóspede, atualizando sua situação e a data/hora do evento.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
hospede_id	string	Sim	O ID único do hóspede a ser atualizado.

### PATCH /hospedes/{hospede\_id}/no-show

Marca o hóspede como “no-show” (não comparecimento), atualizando sua situação.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
hospede_id	string	Sim	O ID único do hóspede a ser atualizado.

### GET /hospedes/{hospede\_id}

Retorna as informações de um hóspede de acordo com seu id.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
hospede_id	string	Sim	O ID único do hóspede a ser atualizado.

### GET /hospedes/pre-checkins

Retorna as informações de um hóspede em determinado período, de acordo com seu documento. Somente caso sua situacao\_id atual seja PRECHECKIN\_REALIZADO

#### Parâmetros de consulta (Query Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
exibir_vinculado	boolean	Sim	true se deseja exibir vinculado.
data_inicio	date	Sim	Data de entrada no formato AAAA-MM-DD.
data_fim	date	Sim	Data de saída no formato AAAA-MM-DD.
tipo_documento	string	Sim	Tipo de documento do hóspede (CPF ou PASSAPORTE).
numero_documento	string	Sim	Número de documento do hóspede (apenas dígitos).

## Pessoas

### Descrição

Endpoints para criar e consultar registros de pessoas. Uma "pessoa" pode se tornar um "hóspede" ao ser associada a uma reserva.

### Endpoints

Método	URL	Descrição
GET	/pessoas/{pessoa_id}	Consulta os dados de uma pessoa pelo seu pessoa_id.
GET	/pessoas/documento/{tipo_documento}/{numero_documento}	Consulta os dados de uma pessoa pelo seu documento.
POST	/pessoas	Cria um novo tipo de pessoa.

## GET /pessoas/{pessoa\_id}

Retorna os dados de uma pessoa com base em seu pessoa\_id.

### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Descrição
pessoa_id	string	O ID único da pessoa a ser consultada.

## GET /pessoas/documento/{tipo\_documento}/{numero\_documento}

Retorna os dados de uma pessoa com base em seu documento.

### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
tipo_documento	string	Sim	Tipo de documento do hóspede (CPF ou PASSAPORTE).
numero_documento	string	Sim	Número de documento do hóspede (apenas dígitos).

## POST /pessoas

Registra os dados de uma nova pessoa no sistema.

### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
cpf_solicitante	string	Sim	CPF do usuário que está realizando o registro.

### Body - Orientações

- nacionalidade: Código IBGE do País de residência do hóspede
- cidade\_id: Código IBGE da Cidade de residência do hóspede
- estado\_id: Sigla IBGE do Estado de residência do hóspede

## Reservas

### Descrição

Endpoints utilizados para o ciclo de vida completo de uma reserva, incluindo consulta de reservas em locais para hospedagem, bem como dados de situação, código da reserva, data de entrada, data de saída, quantidade de hóspedes e outros, conforme visto no exemplo de resposta esperada.

### Endpoints

Método	URL	Descrição
GET	/reservas	Lista reservas com base em filtros.
POST	/reservas	Cria uma nova reserva.
GET	/reservas/{reserva_id}	Consulta os dados de uma reserva específica.
DEL	/reservas/{reserva_id}	Exclui uma reserva.
PUT	/reservas/{reserva_id}	Atualiza todos os dados de uma reserva.
POST	/reservas/{reserva_id}/checkin	
POST	/reservas/{reserva_id}/checkout	
POST	/reservas/{reserva_id}/cancelar	Cancela uma reserva.
GET	/reservas/{reserva_id}/hospedes	Lista os hóspedes associados a uma reserva.
POST	/reservas/{reserva_id}/hospedes	Adiciona um hóspede a uma reserva existente.
POST	/reservas/{reserva_id}/vincular-hospede/{hospede_id}	Vincula um hóspede (de um pré-check-in) a uma reserva.

### GET /reservas

Lista as reservas com paginação. Retorna no máximo 20 reservas por requisição.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
page_number	integer	Sim	O número da página a ser retornada.
situacao	string	Não	Filtra por situação da reserva (valores aceitos: CRIADA, EM_ANDAMENTO, CONCLUIDA, CANCELADA e NOSHOW).
codigo_reserva	string	Não	Filtra pelo número da reserva
data_entrada	date	Não	Filtra por data de entrada no formato AAAA-MM-DD.
data_saida	date	Não	Filtra por data de saída no formato AAAA-MM-DD.

## POST /reservas

Cria uma reserva no sistema. Requer informações como datas e quantidade de hóspedes.

### *Body - Orientações*

- origem\_reserva\_id – opções: “MEIOHOSPEDAGEM” ou “OTA”

#### Descrição dos campos do Body

Campo	Tipo	Obrigatório	Descrição
numero_reserva	string	Sim	Código único da reserva no sistema do hotel/PMS.
numero_reserva_ota	string	Não	Código da reserva na agência de viagens online (OTA), se aplicável.
data_entrada	date	Sim	Data de entrada no formato AAAA-MM-DD.
data_saida	date	Sim	Data de saída no formato AAAA-MM-DD.
quantidade_hospede_adulto	integer	Não	Número de adultos na reserva.
quantidade_hospede_menor	integer	Não	Número de menores de idade na reserva.
origem_reserva_id	string	Não	ID da origem da reserva. Ex: MEIOHOSPEDAGEM.



### GET /reservas/{reserva\_id}

Retorna os dados de uma reserva específica, incluindo status e datas.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.

### DEL /reservas/{reserva\_id}

Remove uma reserva. Essa ação é limitada a reservas que ainda não possuem hóspedes com check-in realizado.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.

### PUT /reservas/{reserva\_id}

Atualiza completamente os dados de uma reserva existente. Todos os campos do corpo da requisição devem ser enviados, pois a operação substitui o recurso existente.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.

### POST /reservas/{reserva\_id}/cancelar

Cancela uma reserva. Altera o status para "CANCELADA". Pode depender de regras como ausência de check-in.

#### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.

## POST /reservas/{reserva\_id}/checkin

Efetua o check-in dos hóspedes registrados na reserva

### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.

## POST /reservas/{reserva\_id}/checkout

Efetua o check-out dos hóspedes registrados na reserva

### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.

## GET /reservas/{reserva\_id}/hospedes

Retorna a listagem de todos os hóspedes associados a uma reserva específica.

### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.

## POST /reservas/{reserva\_id}/hospedes

Adiciona hóspedes a uma reserva existente.

### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.

### Body - Orientações

- responsavel\_id - se aplica quando o hóspede é um dependente. Nesse caso, aqui deve ser informado o id do hóspede responsável pelo dependente

## POST /reservas/{reserva\_id}/vincular-hospede/{hospede\_id}

Vincula um hóspede existente a uma reserva. Utilizado para vincular pré-check-ins não vinculados.

### Parâmetros de rota (Path Parameters)

Parâmetro	Tipo	Obrigatório	Descrição
reserva_id	string	Sim	ID da reserva.
hospede_id	string	Sim	ID do hóspede.

## Erros Comuns

A API pode retornar os seguintes códigos de erro em respostas malformadas ou inválidas:

### 400 Bad Request

A requisição possui sintaxe incorreta ou parâmetros inválidos.

*Exemplo:*

```
{
  "error": "Bad Request",
  "message": "O campo 'nome' é obrigatório."
}
```

## 401 Unauthorized

A autenticação falhou ou o token de acesso é inválido/ausente.

*Exemplo:*

```
{
  "error": "Unauthorized",
  "message": "Token inválido ou ausente."
}
```

## 403 Not Found

A requisição foi compreendida, mas o usuário não tem permissão para executá-la.

*Exemplo:*

```
{
  "error": "Forbidden",
  "message": "Você não tem permissão para acessar este recurso."
}
```

## 404 Not Found

O recurso solicitado não foi encontrado.

*Exemplo:*

```
{
  "error": "Not Found",
  "message": "Usuário não encontrado."
}
```

## 409 Conflict

A solicitação não pôde ser concluída devido a um conflito com o estado atual do recurso.

*Exemplo:*

```
{
  "error": "Conflict",
  "message": "E-mail já cadastrado."
}
```

## 422 Unprocessable Entity

Os dados fornecidos são semanticamente inválidos.

*Exemplo:*

```
{
  "error": "Unprocessable Entity",
  "message": "Formato do CPF inválido."
}
```

## 500 Internal Server Error

Erro inesperado no servidor.

*Exemplo:*

```
{
  "error": "Internal Server Error",
  "message": "Erro ao conectar ao banco de dados."
}
```

## Perguntas Frequentes (FAQ)

Esta seção aborda algumas das dúvidas mais comuns sobre a utilização da API.

### Qual o funcionamento de "Pré-Check-ins"?

O conceito de "Ficha" (FNRH) está intrinsecamente ligado aos dados do hóspede e da reserva. O endpoint GET /hospedes/pre-checkins é utilizado para listar hóspedes que realizaram o pré-check-in e ainda não foram vinculados a uma reserva formal ou estão em um estado de pré-registro. Esses registros de pré-check-in já contêm os dados da FNRH.

### Qual a diferença entre "Get Pessoa" e "Get Hospede"?

**GET** /pessoas/{id} **ou** **GET** /pessoas/documento/{tipo\_documento}/{numero\_documento}: Estes endpoints

são para consultar os dados de uma Pessoa no sistema. Uma Pessoa é uma entidade genérica que representa um indivíduo, independentemente de estar ou não hospedado.

**GET /hospedes/{id}:** Este endpoint é para consultar os dados de um Hóspede. Um Hóspede é uma Pessoa que está associada a uma reserva ou a um processo de hospedagem (como um pré-check-in). O registro de Hóspede contém informações específicas da estadia, como situação (check-in, check-out), dados da FNRH, etc.

### Como devo proceder para registrar ou consultar dados de hóspedes sem a figura da reserva (ou seja, sem vincular a uma reserva imediatamente)?

Se o objetivo é registrar uma pessoa no sistema sem que ela esteja imediatamente vinculada a uma reserva, você deve utilizar o endpoint POST /pessoas. Para consultar os dados dessa pessoa posteriormente, utilize GET /pessoas/{id} ou GET /pessoas/documento/{tipo\_documento}/{numero\_documento}.

O endpoint GET /hospedes/pre-checkins lista registros de hóspedes que já iniciaram um processo de pré-check-in, mas que podem ainda não estar formalmente vinculados a uma reserva. Esses registros já são considerados "hóspedes" no contexto do FNRH, mesmo que a reserva ainda não esteja completa ou vinculada.

Para fluxos onde a reserva não é criada de imediato, mas o hóspede precisa ser registrado para fins de FNRH (como um pré-check-in), o processo geralmente envolve:

1. **Registro da Pessoa:** POST /pessoas (se a pessoa ainda não estiver cadastrada);
2. **Criação do Pré-Check-in:** O sistema interno do meio de hospedagem deve iniciar o processo de pré-check-in, que pode gerar um registro de hóspede com status PRECHECKIN\_NAOVINCULADO, PRECHECKIN\_PENDENTE ou PRECHECKIN\_REALIZADO;
3. **Consulta de Pré-Check-ins:** GET /hospedes/pre-checkins para listar esses hóspedes pendentes;
4. **Vínculo à Reserva:** Quando a reserva for criada ou identificada, utilize POST /reservas/{reserva\_id}/vincular-hospede/{hospede\_id} para associar o hóspede do pré-check-in à reserva.

### Quais são os valores aceitos para o filtro situacao em GET /reservas?

A API possui diferentes conjuntos de "situações" para diferentes entidades:

**Situações de Hóspedes:** Referem-se ao estado de um hóspede individual. A lista completa de valores válidos para hóspedes pode ser obtida no endpoint GET /dominios/hospedes/situacoes.

Os seguintes valores pertencem a esse contexto:



- CANCELADO
- CHECKIN\_REALIZADO
- CHECKOUT\_REALIZADO
- NOSHOW
- PRECHECKIN\_NAOVINCULADO
- PRECHECKIN\_PENDENTE
- PRECHECKIN\_REALIZADO

**Situações de Reservas:** Referem-se ao estado da reserva como um todo. O endpoint GET /reservas filtra por esta situação. Os valores são diferentes dos de hóspedes.

Os valores corretos para filtrar as reservas são:

- CRIADA
- EM\_ANDAMENTO
- CONCLUIDA
- CANCELADA
- NOSHOW

## Os endpoints são idempotentes? Posso reenviar a mesma requisição POST/PUT/PATCH em caso de falha de rede?

**Requisições GET e DELETE:** São naturalmente idempotentes. Reenviá-las múltiplas vezes não causará efeitos colaterais adicionais.

**Requisições PUT:** São projetadas para serem idempotentes. Reenviar uma requisição PUT com o mesmo corpo e para o mesmo recurso resultará no mesmo estado final do recurso.

**Requisições POST:** Por padrão, requisições POST não são idempotentes, o que significa que reenviá-las pode criar recursos duplicados (ex: duas reservas idênticas). No entanto, a API FNRH implementa mecanismos para garantir a idempotência em endpoints POST (como POST /hospedagem/registrar e POST /reservas) através de validações internas que impedem a duplicação de registros com base em chaves de negócio (ex: numero\_reserva).

## Qual o formato esperado para datas e horas (timestamps) nos campos da API? E qual fuso horário é utilizado?

Todos os campos de data e hora na API (ex: data\_entrada, check\_in\_em) devem seguir o padrão ISO 8601.

**Datas:** AAAA-MM-DD (ex: 2024-12-31)

**Datas e Horas:** AAAA-MM-DDTHH:mm:ss.sssZ (ex: 2024-12-31T23:59:59.938Z)

O sufixo Z indica que a data/hora está em UTC (Coordinated Universal Time). É responsabilidade do sistema cliente converter as datas/horas para o fuso horário local, se necessário, antes de enviar ou após receber os dados da API.

## Glossário

Esta seção define os termos técnicos e específicos utilizados ao longo da documentação da API.

- **API (Application Programming Interface):** Conjunto de definições e protocolos que permite a comunicação e interação entre diferentes softwares.
- **Base64:** Um esquema de codificação que representa dados binários em um formato de texto ASCII, comumente usado para transmitir dados de forma segura em ambientes de texto.
- **Check-in:** O processo de registro da entrada de um hóspede no meio de hospedagem, marcando o início da sua estadia.
- **Check-out:** O processo de registro da saída de um hóspede do meio de hospedagem, marcando o fim da sua estadia.**Domínio:** Conjunto de valores pré-definidos e aceitos para um determinado campo na API (ex: tipos de documento, motivos de viagem, situações de hóspede).
- **Endpoint:** Um URL específico que representa um recurso na API e permite a interação com ele (ex: /reservas, /pessoas/{id}).
- **FNRH (Ficha Nacional de Registro de Hóspedes):** Documento obrigatório para o registro de hóspedes em meios de hospedagem no Brasil, coletando dados para fins estatísticos e de segurança.
- **Hóspede:** Uma pessoa que está ou esteve associada a um processo de hospedagem (reserva, check-in, pré-check-in...) no contexto do FNRH.
- **HTTP Basic Authentication:** Método de autenticação simples onde o nome de usuário e a senha são enviados codificados em Base64 no cabeçalho Authorization de cada requisição HTTP.
- **No-Show:** Situação em que um hóspede com reserva confirmada não comparece para o check-in na data prevista.
- **Pessoa:** Uma entidade genérica que representa um indivíduo cadastrado no sistema, independentemente de sua relação com uma hospedagem. Pode se tornar um hóspede.
- **PMS (Property Management System):** Sistema de Gestão de Propriedades Hoteleiras, software utilizado para gerenciar operações diárias de hotéis, como reservas, check-in/out, faturamento, manutenção, etc.
- **Pré-check-in:** Processo de registro antecipado de dados do hóspede antes da chegada física ao meio de hospedagem, agilizando o processo de check-in.**Reserva:** Um registro de uma futura ou atual estadia de um ou mais hóspedes em um meio de hospedagem.
- **REST (Representational State Transfer):** Estilo arquitetural para sistemas distribuídos, que define um conjunto de princípios para a comunicação entre

cliente e servidor, utilizando operações HTTP padrão (GET, POST, PUT, DELETE, PATCH).