



**FICHA NACIONAL
DE REGISTRO
DE HÓSPEDES**

Documento de Integração

Versão 2.1 de 09/03/2026

Documentação da API FNRH - Versão 2 (v2)



Índice

- [Informações Gerais](#)
- [Autenticação e Segurança](#)
- [Endpoints](#)
 - [Domínios](#)
 - [Reservas](#)
 - [Hóspedes](#)
 - [Pessoas](#)
 - [Hospedagem](#)
 - [Fichas](#)
- [Códigos de Status](#)
- [Códigos de Países \(ISO 3166-1 alpha-2\)](#)
- [Códigos de Cidades \(IBGE\)](#)
- [Notas de Migração v1 → v2](#)
- [Changelog](#)

Informações Gerais

Visão Geral

A API FNRH v2 fornece endpoints RESTful para gerenciar reservas, hóspedes e fichas de registro no sistema da Ficha Nacional de Registro de Hóspedes.

URLs Base

Ambiente	URL
Produção	https://fnrh.turismo.serpro.gov.br/FNRH_API/rest/v2
Homologação	https://hom-lowcode.serpro.gov.br/FNRH_API/rest/v2

Especificações Técnicas

- **Protocolo:** HTTPS (TLS 1.2+) - **OBRIGATÓRIO**
- **Formato de Dados:** JSON (UTF-8)
- **Versionamento:** URL-based ([/v2](#))
- **Autenticação:** Basic Authentication (HTTP 401)

Headers Padrão

Todas as requisições devem incluir:

```
Content-Type: application/json
Authorization: Basic {credenciais_base64}
```

Para gerar as credenciais em Base64, concatenar "usuário:senha" e codifique.

Autenticação e Segurança

Método de Autenticação: Basic Authentication

A API utiliza HTTP Basic Authentication conforme definido em [RFC 7617](#).

Passo a Passo para Autenticar

1. Preparar Credenciais

usuário: seu_usuario
senha: sua_senha_segura

2. Concatenar com Dois-Pontos

seu_usuario:sua_senha_segura

3. Codificar em Base64

Base64(seu_usuario:sua_senha_segura) =
c2V1X3VzdWFyaW86c3VhX3N1bmh1cmE=

4. Incluir no Header Authorization

Authorization: Basic c2V1X3VzdWFyaW86c3VhX3N1bmh1cmE=

Exemplo Prático com cURL

Usando credenciais diretas

```
curl -X GET \  
  
"https://api.fnrh.gov.br/FNRH_API/rest/v2/fichas?page_number=0" \  
-H "Authorization: Basic c2V1X3VzdWFyaW86c3VhX3N1bmh1cmE=" \  
-H "Content-Type: application/json"
```

Ou deixando o cURL fazer o encoding

```
curl -X GET \  
"https://api.fnrh.gov.br/FNRH_API/rest/v2/fichas?page_number=0" \  
-u seu_usuario:sua_senha_segura \  
-H "Content-Type: application/json"
```

Exemplo com JavaScript/Node.js

```
const axios = require('axios');  
const credentials =  
Buffer.from('seu_usuario:sua_senha_segura').toString('base64');  
const response = await axios.get(  
  'https://api.fnrh.gov.br/FNRH_API/rest/v2/fichas?page_number=0',  
  {  
    headers: {  
      'Authorization': `Basic ${credentials}`,
```

```
        'Content-Type': 'application/json'
    }
}
);
```

```
console.log(response.data);
```

Exemplo com Python

```
import requests
from requests.auth import HTTPBasicAuth
response = requests.get(
    'https://api.fnrh.gov.br/FNRH_API/rest/v2/fichas?page_number=0',
    auth=HTTPBasicAuth('seu_usuario', 'sua_senha_segura'),
    headers={'Content-Type': 'application/json'})
print(response.json())
```

Tratamento de Erros de Autenticação

Resposta 401 Unauthorized:

```
{
  "erro": "Unauthorized",
  "mensagem": "Credenciais inválidas ou ausentes",
  "codigo": 401
}
```

Possíveis Causas:

- Credenciais ausentes ou malformadas
- Usuário/senha incorretos
- Token expirado
- Conta desativada

Boas Práticas de Segurança

1. **Nunca** exponha credenciais no código-fonte ou versionamento
2. Use variáveis de ambiente para armazenar credenciais
3. Implementar rate limiting no cliente
4. Usar HTTPS em todas as requisições
5. Validar certificados SSL/TLS
6. Rotar credenciais regularmente
7. Manter logs de acesso para auditoria

Endpoints

Domínios

Os endpoints de domínio retornam listas de valores válidos para campos específicos.

1. Meios de Transporte

GET /dominios/fnrh/meios_transporte

Headers:

Accept: application/json

- **Descrição:** Lista todos os meios de transporte disponíveis para preenchimento na Ficha Nacional de Registro de Hóspedes. Este endpoint retorna os 8 meios de transporte validados pelo sistema, que devem ser utilizados ao registrar o meio de transporte utilizado pelo hóspede para chegar ao hotel.
- **Caso de Uso:** Ao preencher a ficha de hospedagem, o sistema solicita o meio de transporte utilizado. Este endpoint fornece a lista de opções válidas aceitas pelo sistema.

Response (200 OK):

```
{
  "dados": [
    {
      "id": "ONIBUS",
      "label": "Ônibus"
    },
    {
      "id": "AUTOMOVEL",
      "label": "Automóvel"
    },
    {
      "id": "AVIAO",
      "label": "Avião"
    },
    {
      "id": "MOTO",
      "label": "Moto"
    },
    {
      "id": "NAVIO_BARCO",
      "label": "Navio/Barco"
    },
    {
      "id": "PE",
```

```
    "label": "A pé"
  },
  {
    "id": "TREM",
    "label": "Trem"
  },
  {
    "id": "BICICLETA",
    "label": "Bicicleta"
  }
]
```

Campos da Resposta:

- **id**: Identificador único do meio de transporte (deve ser utilizado em requisições)
- **label**: Descrição legível do meio de transporte

2. Motivos de Viagem

GET /dominios/fnrh/motivos_viagem

Headers:

Accept: application/json

- **Descrição**: Retorna a lista completa de motivos de viagem aceitos pelo sistema FNRH. Estes motivos categorizam a principal razão da viagem do hóspede e são utilizados para compilar estatísticas sobre o turismo no Brasil. A classificação engloba desde lazer até saúde, comércio e religião.
- **Caso de Uso**: Ao preencher a ficha de hospedagem, o sistema solicita o motivo principal da viagem. Este endpoint fornece as 8 categorias válidas que devem ser selecionadas.
- **Importância Estatística**: O motivo da viagem é crucial para análise do perfil de turistas e fluxo de turismo nos municípios brasileiros.

Response (200 OK):

```
{
  "dados": [
    {
      "id": "COMPRAS",
```

```
    "label": "Compras"
  },
  {
    "id": "CONGRESSO_FEIRA",
    "label": "Congresso/Feira"
  },
  {
    "id": "ESTUDOS_CURSOS",
    "label": "Estudos/Cursos"
  },
  {
    "id": "LAZER_FERIAS",
    "label": "Lazer/Férias"
  },
  {
    "id": "NEGOCIOS",
    "label": "Negócios"
  },
  {
    "id": "PARENTES_AMIGOS",
    "label": "Parentes/Amigos"
  },
  {
    "id": "RELIGIAO",
    "label": "Religião"
  },
  {
    "id": "SAUDE",
    "label": "Saúde"
  }
]
}
```

Campos da Resposta:

- **id**: Identificador único do motivo de viagem (use este valor em requisições)
- **label**: Descrição em português do motivo de viagem

3. Situações de Hóspedes

GET /dominios/hospedes/situacoes

Headers:

Accept: application/json

- **Descrição:** Lista todos os estados possíveis de um hóspede no ciclo de vida da hospedagem. Estes estados representam o fluxo desde o registro inicial até a saída do hotel.
- **Importância no Fluxo:** Este domínio é fundamental para mapear o progresso da hospedagem. Cada transição de estado representa uma ação registrada no sistema, permitindo auditoria completa do processo.
- **Caso de Uso:** A recepção utiliza estes valores para consultar e atualizar o status de hóspede durante as operações do dia. Filtros de listagem frequentemente permitem segmentar hóspedes por situação para operações em lote.

Estados do Ciclo de Vida:

- ***PRECHECKIN_NAOVINCULADO*:** Hóspede que fez pré-check-in online, mas ainda não foi vinculado a uma reserva.
- ***PRECHECKIN_PENDENTE*:** Hóspede vinculado a uma reserva, aguardando a realização do pré-check-in online.
- ***PRECHECKIN_REALIZADO*:** Hóspede que fez pré-check-in e encontra-se vinculado a uma reserva.
- ***CHECKIN_REALIZADO*:** Hóspede confirmou chegada e foi registrado na recepção.
- ***CHECKOUT_REALIZADO*:** Hóspede saiu do hotel e foi registrado o checkout.
- ***NOSHOW*:** Hóspede não compareceu na data da reserva.
- ***CANCELADO*:** Hóspede cancelado devido ao cancelamento da Reserva ao qual estava vinculado.

Response (200 OK):

```
{
  "dados": [
    {
      "id": "CANCELADO",
      "label": "Cancelado"
    },
    {
      "id": "CHECKIN_REALIZADO",
      "label": "Check-in realizado"
    },
    {
      "id": "CHECKOUT_REALIZADO",
      "label": "Check-out realizado"
    }
  ]
}
```

```

    },
    {
      "id": "NOSHOW",
      "label": "No-Show"
    },
    {
      "id": "PRECHECKIN_NAOVINCULADO",
      "label": "Pré-check-in não vinculado"
    },
    {
      "id": "PRECHECKIN_PENDENTE",
      "label": "Pré-check-in pendente"
    },
    {
      "id": "PRECHECKIN_REALIZADO",
      "label": "Pré-check-in realizado"
    }
  ]
}

```

Campos da Resposta:

- **id**: Identificador único da situação do hóspede
- **label**: Descrição legível da situação do hóspede

4. Gêneros

GET /dominios/pessoas/generos

Headers:

Accept: application/json

- **Descrição**: Lista todos os gêneros possíveis de um hóspede.
- **Importância no Fluxo**: Este domínio é fundamental para mapear os gêneros dos hóspedes, registrados no sistema de hospedagem turística nacional.
- **Caso de Uso**: O Meio de Hospedagem utiliza estes valores para consultar e atualizar os campos de gêneros dos hóspedes.

Response (200 OK):

```

{
  "dados": [
    {
      "id": "HOMEM",
      "label": "Masculino"
    }
  ]
}

```

```

    },
    {
      "id": "MULHER",
      "label": "Feminino"
    },
    {
      "id": "OUTRO",
      "label": "Outros"
    },
    {
      "id": "NAOINFORMADO",
      "label": "Prefiro não informar"
    }
  ]
}

```

Campos da Resposta:

- **id**: Identificador único do gênero do hóspede (deve ser utilizado em requisições)
- **label**: Descrição legível do gênero

5. Opção de Deficiência

GET /dominios/pessoas/opcao_deficiencia

Headers:

Accept: application/json

- **Descrição**: Lista as opções de deficiências de um hóspede.
- **Importância no Fluxo**: Este domínio é fundamental para mapear a incidência de hóspedes que possuem ou não deficiências, registrados no sistema de hospedagem turística nacional.
- **Caso de Uso**: O Meio de Hospedagem utiliza estes valores para consultar e atualizar os campos de opções de deficiências dos hóspedes.

Response (200 OK):

```

{
  "dados": [
    {
      "id": "SIM",
      "label": "Sim"
    }
  ]
}

```

```
{
  {
    "id": "NAO",
    "label": "Não"
  },
  {
    "id": "NAOINFORMAR",
    "label": "Prefiro não informar"
  }
]
```

Campos da Resposta:

- **id**: Identificador único das opções de deficiência (deve ser utilizado em requisições)
- **label**: Descrição legível da opção de deficiência

6. Raças/Etnias

GET /dominios/pessoas/racas

Headers:

Accept: application/json

- **Descrição**: Lista as opções de raças/etnias de um hóspede.
- **Importância no Fluxo**: Este domínio é fundamental para mapear as raças/etnias de hóspedes, registradas no sistema de hospedagem turística nacional.
- **Caso de Uso**: A recepção utiliza estes valores para consultar e atualizar os dados das raças/etnias dos hóspedes.

Response (200 OK):

```
{
  "dados": [
    {
      "id": "AMARELA",
      "label": "Amarelo"
    },
    {
      "id": "BRANCA",
      "label": "Branco"
    },
    {
      "id": "INDIGENA",
```

```

        "label": "Indígena"
      },
      {
        "id": "PARDA",
        "label": "Pardo"
      },
      {
        "id": "PRETA",
        "label": "Preto"
      },
      {
        "id": "NAOINFORMAR",
        "label": "Prefiro não informar"
      }
    ]
  }
}

```

Campos da Resposta:

- **id**: Identificador único da raça/etnia do hóspede (deve ser utilizado em requisições)
- **label**: Descrição legível da raça/etnia

7. Tipos de Deficiência

GET /dominios/pessoas/tipos_deficiencia

Headers:

Accept: application/json

- **Descrição**: Lista os tipos de deficiências de um hóspede.
- **Importância no Fluxo**: Este domínio é fundamental para mapear a incidência dos tipos de deficiências dos hóspedes, registrados no sistema de hospedagem turística nacional.
- **Caso de Uso**: O Meio de Hospedagem utiliza estes valores para consultar e atualizar os campos de tipos de deficiências dos hóspedes.

Response (200 OK):

```

{
  "dados": [
    {
      "id": "FISICA",
      "label": "Deficiência Física"
    },
  ],
}

```

```
{
  {
    "id": "AUDITIVA_SURDEZ",
    "label": "Deficiência Auditiva/Surdez"
  },
  {
    "id": "VISUAL",
    "label": "Deficiência Visual"
  },
  {
    "id": "INTELECTUAL",
    "label": "Deficiência Intelectual"
  },
  {
    "id": "MULTIPLA",
    "label": "Deficiência Múltipla"
  }
]
```

Campos da Resposta:

- **id**: Identificador único dos tipos de deficiências (deve ser utilizado em requisições)
- **label**: Descrição legível do tipo de deficiência

8. Tipos de Documento

GET /dominios/pessoas/tipos_documento

Headers:

Accept: application/json

- **Descrição**: Lista as opções de tipo de documentos de um hóspede.
- **Importância no Fluxo**: Este domínio é fundamental para a padronização dos dados a serem informados pelo Meio de Hospedagem.
- **Caso de Uso**: O Meio de Hospedagem utiliza estes valores para enviar, consultar e atualizar os dados de um hóspede.

Response (200 OK):

```
{
  "dados": [
    {
      "id": "CPF",
```

```
    "label": "CPF"
  },
  {
    "id": "PASSAPORTE",
    "label": "Passaporte"
  }
]
```

Campos da Resposta:

- **id**: Identificador único do tipo de documento (deve ser utilizado em requisições)
- **label**: Descrição legível do tipo de documento

9. Situações de Reserva

GET /dominios/reservas/situacoes

Headers:

Accept: application/json

- **Descrição**: Lista todos os estados possíveis de uma reserva no ciclo de vida da hospedagem. Estes estados representam o fluxo desde o registro de uma reserva até a sua conclusão.
- **Importância no Fluxo**: Este domínio é fundamental para mapear o progresso da reserva em uma hospedagem. Cada transição de estado representa uma ação registrada no sistema, permitindo auditoria completa do processo.
- **Caso de Uso**: A recepção utiliza estes valores para consultar e atualizar o status de uma reserva durante as operações do sistema. Filtros de listagem frequentemente permitem segmentar as reservas por situação para operações em lote.

Estados do Ciclo de Vida:

- **CRIADA**: Reserva criada pelo Meio de Hospedagem, possuindo ou não hóspedes aguardando check-in.
- **EM_ANDAMENTO**: Reserva que possui no mínimo 1 (um) hóspede com status "check-in realizado".
- **CONCLUÍDA**: Reserva sem hóspedes com pendências de check-out.
- **NOSHOW**: Reserva onde todos os hóspedes não compareceram na data de entrada.
- **CANCELADA**: Reserva cancelada (só é possível cancelar reservas com situação "Criada").

Response (200 OK):

```
{
  "dados": [
    {
      "id": "CRIADA",
      "label": "Criada"
    },
    {
      "id": "EM_ANDAMENTO",
      "label": "Em Andamento"
    },
    {
      "id": "CONCLUIDA",
      "label": "Concluída"
    },
    {
      "id": "CANCELADA",
      "label": "Cancelada"
    },
    {
      "id": "NOSHOW",
      "label": "No-Show"
    }
  ]
}
```

Campos da Resposta:

- **id**: Identificador único da situação da reserva (deve ser utilizado em requisições)
 - **label**: Descrição legível da situação da reserva
-

Reservas

1. Listar Reservas

GET /reservas

- **Descrição**: Lista as reservas de acordo com os filtros aplicados. Retorna no máximo 20 reservas por requisição, com suporte a paginação. Este endpoint é essencial para consultas em lote, permitindo filtrar reservas por status, datas e número.

- **Paginação:** As respostas são limitadas a 20 itens por página. Use o parâmetro `page_number` para navegar. A primeira página é sempre 1.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para consultar todas as reservas de um período específico, filtrar por situação (reservas abertas, confirmadas, canceladas).

Autenticação: ☒ Obrigatória

Paginação: ☒ Sim (máximo 500 por página)

Parâmetros Query:

Parâmetro	Tipo	Obrigatório	Descrição
<code>page_number</code>	integer	☒	Número da página (começa em 1)
<code>situacao</code>	string	☐	Filtro por situação
<code>codigo_reserva</code>	string	☐	Número da reserva
<code>data_entrada</code>	date	☐	Data de entrada (YYYY-MM-DD)
<code>data_saida</code>	date	☐	Data de saída (YYYY-MM-DD)

Exemplo de Request:

Accept: application/json

```
GET /v2/reservas?page_number=1&situacao=CRIADA
```

Response (200 OK):

```
{
  "dados": [
    {
      "reserva_id": "3a749fc0-bf2b-41cc-af00-fbdefca71b8c",
      "numero_reserva": "RESERVA002",
      "situacao_reserva_id": "CRIADA",
      "situacao_reserva_label": "Criada",
      "origem_reserva_id": "MEIOHOSPEDAGEM",

```

```

        "origem_reserva_label": "Meio Hospedagem",
        "data_entrada": "2025-10-20",
        "data_saida": "2025-10-21",
        "quantidade_hospede_adulto": 1,
        "link_precheckin": "https://hom-
lowcode.serpro.gov.br/FNRH_Hospede/QRCodeFNRH?Source=..."
    },
    {
        "reserva_id": "1f4e94f3-f350-4dc7-a2a2-f42b31239a4b",
        "numero_reserva": "RESERVA001",
        "situacao_reserva_id": "CRIADA",
        "situacao_reserva_label": "Criada",
        "origem_reserva_id": "MEIOHOSPEDAGEM",
        "origem_reserva_label": "Meio Hospedagem",
        "data_entrada": "2025-12-16",
        "data_saida": "2025-12-17",
        "quantidade_hospede_adulto": 1,
        "link_precheckin": "https://hom-
lowcode.serpro.gov.br/FNRH_Hospede/QRCodeFNRH?Source=..."
    }
]
}

```

2. Criar Reserva

POST /reservas

- **Descrição:** Cria uma reserva no sistema. Este é o primeiro passo do fluxo de hospedagem. Uma vez criada, a reserva pode receber hóspedes. O sistema gera automaticamente um **link_precheckin** que pode ser enviado para o hóspede realizar o pré-check-in online.
- **Origem da Reserva:** O campo **origem_reserva_id** indica se a reserva foi feita através do próprio sistema (**MEIOHOSPEDAGEM**) ou (**OTA**).
- **Caso de Uso:** Quando um hóspede faz uma nova reserva, a recepção cria o registro usando este endpoint.

Parâmetros Body:

Parâmetro	Tipo	Obrigatório	Descrição
-----------	------	-------------	-----------

numero_reserva	String	✓	Descrição do código da reserva do Meio de Hospedagem (UNIQUE)
numero_reserva_OTA	String	✗	Descrição do código da reserva OTA
data_entrada	Date	✓	Data de entrada da reserva (YYYY-MM-DD)
data_saida	Date	✓	Data de saída da reserva (YYYY-MM-DD)
quantidade_hospede_adulto	Integer	✗	quantidade de hóspedes adultos na reserva
quantidade_hospede_menor	Integer	✗	quantidade de hóspedes menores na reserva
origem_reserva_id	String	✓	Tipo da reserva (MEIODEHOSPEDAGEM ou OTA)

Exemplo Request Body:

```
{
  "numero_reserva": "RESERVA003",
  "numero_reserva_ota": "",
  "data_entrada": "2025-04-10",
  "data_saida": "2025-04-20",
  "quantidade_hospede_adulto": 2,
  "quantidade_hospede_menor": 0,
  "origem_reserva_id": "MEIOHOSPEDAGEM"
}
```

Response (200 OK):

```
{
  "reserva": {
    "reserva_id": "634eecb8-5973-47fd-a8a6-718e3c9d717f",
    "numero_reserva": "RESERVA003",
    "numero_reserva_ota": "",
    "situacao_reserva_id": "CRIADA",
    "origem_reserva_id": "MEIOHOSPEDAGEM",
    "data_entrada": "2025-04-10",
    "data_saida": "2025-04-20",
    "quantidade_hospede_adulto": 2,
    "quantidade_hospede_menor": 0,
    "link_precheckin": "https://fnrh.turismo.gov.br/precheckin/634eecb8..."
  }
}
```

3. Obter Reserva por ID

GET /reservas/{id}

- **Descrição:** Retorna todos os detalhes de uma reserva específica usando seu UUID. Este endpoint fornece a visão completa do estado atual da reserva, incluindo datas, quantidade de hóspedes e link para pré-check-in.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para recuperar os dados de uma reserva específica. Também é útil para validar se uma reserva existe antes de adicionar hóspedes.

Parâmetros Path:

Parâmetro	Tipo	Descrição
id	UUID	Identificador único da reserva

Exemplo de Request:

```
GET /v2/reservas/7c6a71T3-e915-4b9d-b138-f3ad6335d19e
```

Response (200 OK):

```
{
  "reserva": {
    "reserva_id": "7c6a71T3-e915-4b9d-b138-f3ad6335d19e",
    "numero_reserva": "RESERVA004",
```

```
"situacao_reserva_id": "CRIADA",
"situacao_reserva_label": "Criada",
"origem_reserva_id": "MEIOHOSPEDAGEM",
"origem_reserva_label": "Meio Hospedagem",
"data_entrada": "2026-01-01",
"data_saida": "2026-01-07",
"quantidade_hospede_adulto": 1,
"link_precheckin":
"https://fnrh.turismo.serpro.gov.br/FNRH_Hospede/QRCodeFNRH?Source=..."
}
}
```

4. Atualizar Reserva

PUT /reservas/{id}

- **Descrição:** Atualiza os dados de uma reserva específica usando seu UUID. Este endpoint possibilita a atualização dos dados de uma reserva.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para atualizar os dados de uma reserva específica modificando as informações da reserva.

Parâmetros Path:

Parâmetro	Tipo	Descrição
id	UUID	Identificador único da reserva

Parâmetros Body:

Parâmetro	Tipo	Obrigatório	Descrição
numero_reserva	String	✓	Descrição do código da reserva do Meio de Hospedagem (UNIQUE)
numero_reserva_OTA	String	✗	Descrição do código da reserva OTA

data_entrada	Date	✓	Data de entrada da reserva (YYYY-MM-DD)
data_saida	Date	✓	Data de saída da reserva (YYYY-MM-DD)
quantidade_hospede_adulto	Integer	✗	quantidade de hóspedes adultos na reserva
quantidade_hospede_menor	Integer	✗	quantidade de hóspedes menores na reserva

Exemplo Request Body:

```
{
  "numero_reserva": "RES202501001-ALTERADA",
  "numero_reserva_ota": "",
  "data_entrada": "2025-12-17",
  "data_saida": "2025-12-20",
  "quantidade_hospede_adulto": 3,
  "quantidade_hospede_menor": 1
}
```

Response (200 OK): Corpo vazio

5. Excluir Reserva

DELETE /reservas/{id}

- **Descrição:** Remove uma reserva específica usando seu UUID.

- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para excluir os dados de uma reserva específica.

Restrições:

- Apenas reservas no estado **CRIADA** podem ser deletadas
- Não é possível deletar reserva com check-in realizado

Parâmetros Path:

Parâmetro	Tipo	Descrição
id	UUID	Identificador único da reserva

Exemplo de Request:

```
DELETE /v2/reservas/7c6a71T3-e915-4b9d-b138-f3ad6335d19e
```

Response (200 OK): Corpo vazio

6. Cancelar Reserva

POST /reservas/{reserva_id}/cancelar

- **Descrição:** Cancela uma reserva específica usando seu UUID. Altera a situação da reserva e dos hóspedes vinculados à **CANCELADA**.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para cancelar uma reserva específica e seus hóspedes.

Parâmetros Path:

Parâmetro	Tipo	Descrição
reserva_id	UUID	Identificador único da reserva

Exemplo de Request:

```
POST /v2/reservas/7c6a71T3-e915-4b9d-b138-f3ad6335d19e/cancelar
```

Response (200 OK): *Corpo vazio*

7. Listar Hóspedes da Reserva

GET /reservas/{reserva_id}/hospedes

- **Descrição:** Lista todos os hóspedes presentes em uma reserva específica usando seu UUID.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para listar a relação de hóspedes presentes em uma determinada reserva e suas situações.

Parâmetros Path:

Parâmetro	Tipo	Descrição
reserva_id	UUID	Identificador único da reserva

Exemplo de Request:

```
GET /v2/reservas/7c6a71f3-e963-4b9d-b138-f3516335d19e/hospedes
```

Response (200 OK):

```
{
  "dados": [
    {
      "pessoa": {
        "pessoa_id": "df5a6ba6-e673-4e04-9777-d9466937c23c",
        "nome": "JOÃO DA SILVA",
        "nome_social": "",
        "PaisNacionalidade_id": "BR",
        "genero_id": "HOMEM",
        "genero": "Homem",
        "GeneroDescricao": "",
        "data_nascimento": "1985-05-15",
        "numero": "12345678901",
        "tipo_documento_id": "CPF",
        "tipo_documento": "CPF"
      },
      "hospede": {
        "hospede_id": "8349f016-7763-4264-89b7-4e57580ff7f2",
        "reserva_id": "7c6a71f3-e963-4b9d-b138-f3516335d19e",
        "pessoa_id": "df5a6ba6-e673-4e04-9777-d9466937c23c",
        "hospedagem_id": 12345,

```

```
    "situacao_hospede_id": "CHECKIN_REALIZADO",
    "situacao_hospede": "Check-in realizado",
    "situacao_cor": "#FFA500",
    "responsavel_quarto": true,
    "nome_responsavel": "",
    "checkin_em": null,
    "checkout_em": null
  }
}
```

8. Adicionar Pessoa na Reserva

POST /reservas/{reserva_id}/hospedes

✦ Uma **pessoa** se torna **Hóspede** ao ser adicionada a uma reserva.

- **Descrição:** Adiciona uma pessoa em uma reserva específica utilizando seu UUID, mudando seu papel para hóspede.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para adicionar uma pessoa a uma reserva, elevando seu papel para hóspede.

Parâmetros Path:

Parâmetro	Tipo	Descrição
reserva_id	UUID	Identificador único da reserva

Parâmetros Body:

Parâmetro	Tipo	Obrigatório	Descrição
Pessoa_id	UUID	✓	Identificador único da pessoa
is_principal	Booleano	✗	Indicação se a pessoa é o responsável pela reserva

situacao_hospede_id	String	✓	Situação do hóspede (conforme domínio)
checkin_em	DateTime	situacao_hospede = "CHECKIN_REALIZADO" ou "CHECKOUT_REALIZADO"	Data e hora check-in(ISO 8601 com timezone UTC (Z))
checkout_em	DateTime	situacao_hospede = "CHECKOUT_REALIZADO"	Data e hora check-out (ISO 8601 com timezone UTC (Z))
motivo_viagem_id (fnrh)	String	situacao_hospede = "CHECKIN_REALIZADO" ou "CHECKOUT_REALIZADO"	Identificador do Motivo da viagem (conforme domínio)
meio_transporte_id (fnrh)	String	situacao_hospede = "CHECKIN_REALIZADO" ou "CHECKOUT_REALIZADO"	Identificador do Meio de transporte (conforme domínio)
responsavel_id	UUID	Pessoa menor de 18 anos	Identificador do hospede responsavel

Exemplo Request Body:

```
{
```

```
"pessoa_id": "df5a6ba6-e673-4e04-9777-d9466937c23c",
"is_principal": false,
"situacao_hospede_id": "PRECHECKIN_PENDENTE",
"checkin_em": null,
"checkout_em": null,
"fnrh": {
  "motivo_viagem_id": "LAZER_FERIAS",
  "meio_transporte_id": "AVIAO"
},
"responsavel_id": "8349f016-7763-4264-89b7-4e57580ff7f2"
}
```

Response (200 OK): *Corpo vazio*

Observação Importante:

- A resposta não inclui o hospede_id criado. Para obter o identificador é necessário utilizar o GET /reservas/{reserva_id}/hospedes

9. Check-in da Reserva (em lote)

POST /reservas/{reserva_id}/checkin

- **Descrição:** Efetua o check-in de todos os hóspedes registrados na reserva.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para realizar o check-in em lote, quando todos os hóspedes da reserva realizam a entrada ao mesmo momento.

Parâmetros Path:

Parâmetro	Tipo	Descrição
reserva_id	UUID	Identificador único da reserva

Exemplo de Request:

```
POST /v2/reservas/063ff52e-10d8-4ed0-a632-e602a33a908a/checkin
```

Request Body (opcional, `text/plain`):

```
2026-02-10T15:00:00.000Z
```

Response (200 OK):

```
{
  "dados": [
    {
      "hospede_id": "550e8400-e29b-41d4-a716-446655440010",
      "situacao_id": "CHECKIN_REALIZADO",
      "data_hora": "2026-02-10T15:00:00.000Z"
    },
    {
      "hospede_id": "550e8400-e29b-41d4-a716-446655440011",
      "situacao_id": "CHECKIN_REALIZADO",
      "data_hora": "2026-02-10T15:00:00.000Z"
    }
  ]
}
```

10. Check-out da Reserva (em lote)

POST /reservas/{reserva_id}/checkout

- **Descrição:** Efetua o check-out de todos os hóspedes registrados na reserva.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para realizar o check-out em lote, quando todos os hóspedes da reserva realizam a saída no mesmo momento.

Parâmetros Path:

Parâmetro	Tipo	Descrição
reserva_id	UUID	Identificador único da reserva

Request Body (opcional, text/plain):

2026-02-15T11:00:00.000Z

Response (200 OK):

```
{
  "dados": [
    {
      "hospede_id": "550e8400-e29b-41d4-a716-446655440010",
      "situacao_id": "CHECKOUT_REALIZADO",
      "data_hora": "2026-02-15T11:00:00.000Z"
    }
  ]
}
```

11. No-Show da Reserva (em lote)

POST /reservas/{reserva_id}/noshow

- **Descrição:** Efetua o no-show de todos os hóspedes registrados na reserva.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint quando todos os hóspedes da reserva não compareceram na data de entrada, utilizado para tarefas em lote.

Parâmetros Path:

Parâmetro	Tipo	Descrição
reserva_id	UUID	Identificador único da reserva

Exemplo de Request:

```
POST /v2/reservas/063ff52e-10d8-4ed0-a632-e602a33a908a/noshow
```

Response (200 OK):

```
{
  "dados": [
    {
      "hospede_id": "550e8400-e29b-41d4-a716-446655440010",
      "situacao_id": "NOSHOW",
      "data_hora": "2026-02-10T00:00:00.000Z"
    }
  ]
}
```

12. Vincular Hóspede à Reserva

POST /reservas/{reserva_id}/vincular-hospede/{hospede_id}

- **Descrição:** Vincula um hóspede com pré-check-in não vinculado a uma reserva criada.

- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para realizar a vinculação de um hóspede (pessoa que já efetuou um pré-check-in de balcão) a uma reserva criada, atualizando a situação do hóspede para **PRECHECKIN_REALIZADO**.

Parâmetros Path:

Parâmetro	Tipo	Descrição
reserva_id	UUID	Identificador único da reserva
hospede_id	UUID	Identificador único do hóspede

Exemplo de Request:

```
POST /v2/reservas/063ff52e-10d8-4ed0-a632-e602a33a908a/vincular-hospede/07a82947-0ed5-4b2a-bb98-ec7f8711c1c5
```

Response (200 OK):

```
{
  "vinculo": {
    "reserva_id": "063ff52e-10d8-4ed0-a632-e602a33a908a",
    "hospede_id": "07a82947-0ed5-4b2a-bb98-ec7f8711c1c5",
    "situacao": "PRECHECKIN_PENDENTE"
  }
}
```

Hóspedes

1. Obter Hóspede por ID

GET /hospedes/{id}

- **Descrição:** Retorna os dados completos de um hóspede específico, incluindo todas as informações pessoais, dados de contato, informações da reserva vinculada e situação atual. Este é o endpoint mais completo para obtenção de informações de um hóspede, retornando dados estruturados em dois objetos principais: **Hospede** e **Pessoa**.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para consultar os dados do hóspede.

Parâmetros Path:

Parâmetro	Tipo	Descrição
id	UUID	Identificador único do hóspede

Exemplo de Request:

```
GET /v2/hospedes/8349f016-7763-4264-89b7-4e57580ff7f2
```

Response (200 OK):

```
{
  "dados": {
    "Hospede": {
      "hospede_id": "8349f916-7763-4264-80b7-4e5Gt80ff7f2",
      "reserva": {
        "reserva_id": "7c6a71T3-e903-4b9d-b138-f3516G34d19e",
        "numero_reserva": "RESERVATESTES",
        "situacao": {
          "id": "CRIADA",
          "label": "Criada"
        },
        "origem": {
          "id": "MEIOHOSPEDAGEM",
          "label": "Meio Hospedagem"
        },
        "data_entrada": "2025-12-16",
        "data_saida": "2025-12-17",
        "quantidade_hospede_adulto": 1
      },
      "situacao": {
        "id": "PRECHECKIN_PENDENTE",
        "label": "Pré-check-in pendente"
      },
      "is_principal": true
    },
    "Pessoa": {
      "id": "e07d8a49-c7Ad-4171-8Ta5-3cb849fc22d7",
      "documento": {
        "numero_documento": "1234567899",
        "tipo_documento": {
          "id": "CPF",
          "label": "CPF"
        }
      },
      "dado_pessoal": {
        "nome": "JOAO MARCOS SILVA",

```

```
        "nome_social": "NOME SOCIAL",
        "genero": {
            "id": "OUTRO",
            "label": "Outros"
        },
        "GeneroDescricao": "TESTE GENERO",
        "data_nascimento": "1970-10-01",
        "raca_id": {
            "id": "PARDA",
            "label": "Pardo"
        },
        "deficiencia": {
            "possui_deficiencia": {
                "id": "SIM",
                "label": "Sim"
            },
            "tipo_deficiencia": {
                "id": "FISICA",
                "label": "Deficiência Física"
            }
        },
        "PaisNacionalidade_id": "ZA"
    },
    "contato": {
        "email": "abc@email.com.br",
        "telefone": "1234567890",
        "endereco": {
            "id": "e07d8a49-c7ad-4171-8ba5-3cb849fc22d7",
            "PaisResidencia_id": "AL"
        }
    }
}
}
```

2. Listar Pré-Check-ins

GET /hospedes/pre-checkins

Parâmetros Query:

Parâmetro	Tipo	Descrição
exibir_vinculado	Booleano	Exibir pré-check-ins já vinculados
data_inicio	Date	Data inicial do período da realização do pre-check-in (YYYY-MM-DD)

Parâmetro	Tipo	Descrição
<code>data_fim</code>	Date	Data final do período da realização do pre-check-in (YYYY-MM-DD)
<code>tipo_documento</code>	String	Identificador do tipo de documento (CPF ou PASSAPORTE)
<code>numero_documento</code>	String	Número do documento para filtro

Exemplo:

`GET https://fnrh.turismo.serpro.gov.br/FNRH_API/rest/v2/hospedes/pre-checkins?exibir_vinculado=false&data_inicio=2025-12-01&data_fim=2025-12-31`

Response (200 OK):

```
{
  "dados": [
    {
      "hospede_id": "24f3a6d9-d1c4-4619-8e21-81474d070b17",
      "pessoa_id": "17f841cd-38e1-41de-9fe1-ea1820c1d33d",
      "nome": "CLAUDIA SILVA",
      "PaisNacionalidade_id": "BR",
      "genero": "Feminino",
      "data_nascimento": "1971-10-09",
      "numero_documento": "123456789",
      "tipo_documento": "CPF",
      "situacao_hospede_id": "PRECHECKIN_NAOVINCULADO"
    },
    {
      "hospede_id": "d5f34888-2e56-491f-a9bf-7f6b69b134be",
      "pessoa_id": "22b1ffbf-c5b9-4cb0-93c1-047505fecfaf",
      "nome": "AUGUSTO SILVA",
      "nome_social": "AD",
      "PaisNacionalidade_id": "AL",
      "genero": "Outros",
      "data_nascimento": "2025-10-01",
      "numero_documento": "2",
      "tipo_documento": "Passaporte",
      "responsavel_id": "ecce35af-961c-42cb-bb1c-bda7e54666d0",
      "situacao_hospede_id": "PRECHECKIN_NAOVINCULADO"
    },
    {
      "hospede_id": "1090b74d-0f15-489d-bee6-0e6faf62936f",
      "pessoa_id": "77f3ead3-41a7-4917-95a5-95a949bf5bd6",
      "nome": "FELIPE SILVA",
```

```
    "PaisNacionalidade_id": "ZA",
    "genero": "Prefiro não informar",
    "data_nascimento": "1985-08-20",
    "numero_documento": "12345678988",
    "tipo_documento": "CPF",
    "situacao_hospede_id": "PRECHECKIN_NAOVINCULADO"
  }
]
```

3. Check-in Individual do Hóspede

PATCH /hospedes/{hospede_id}/checkin

- **Descrição:** Registra o check-in de um hóspede específico. Registra a data e hora exata de chegada. Este endpoint é usado para check-ins individuais, quando cada hóspede chega em horários diferentes.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint quando cada hóspede chega e confirma sua entrada. Para grupos que chegam juntos, pode-se usar o check-in em lote via `/reservas/{reserva_id}/checkin`.

Parâmetros Path:

Parâmetro	Tipo	Descrição
<code>hospede_id</code>	UUID	Identificador único do hóspede

Exemplo de Request:

```
PATCH /v2/hospedes/d414bc8b-de66-4147-8be3-2e2585a458fb/checkin
```

Request Body (text/plain):

```
2025-11-14T10:00:00.938Z
```

Response (200 OK):

```
{
  "dados": {
    "hospede_id": "d414bc8b-de66-4147-8be3-2e2585a458fb",
```

```
"situacao_id": "CHECKIN_REALIZADO",
"data_hora": "2025-11-14T10:00:00.938Z"
}
}
```

4. Check-out Individual do Hóspede

PATCH /hospedes/{hospede_id}/checkout

- **Descrição:** Registra o check-out de um hóspede específico. Registra a data e hora exata de saída quando cada hóspede sai em horários diferentes. Este é o último evento do ciclo de hospedagem.
- **Diferença do Check-in:** Enquanto o check-in marca a chegada, o checkout marca a saída.
- **Caso de Uso:** Quando um hóspede deixa o hotel, a recepção executa este endpoint para registrar a saída.

Parâmetros Path:

Parâmetro	Tipo	Descrição
<code>hospede_id</code>	UUID	Identificador único do hóspede

Exemplo de Request:

```
PATCH /v2/hospedes/2eb49cfd-2a23-43bc-9c68-8c325fe7e194/checkout
```

Request Body (text/plain):

```
2025-10-07T16:33:59.938Z
```

Response (200 OK):

```
{
  "dados": {
    "hospede_id": "2eb49cfd-2a23-43bc-9c68-8c325fe7e194",
    "situacao_id": "CHECKOUT_REALIZADO",
    "data_hora": "2025-10-07T16:33:59.938Z"
  }
}
```

```
}
```

5. Informar No-Show do Hóspede

PATCH `/hospedes/{hospede_id}/noshow`

⚠ **Nota:** Padronizado como `noshow` (sem hífen) para consistência com `/reservas/{reserva_id}/noshow`.

- **Descrição:** Registra que o hóspede não compareceu (no-show).
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint quando um hóspede específico não compareceu na data de entrada.

Parâmetros Path:

Parâmetro	Tipo	Descrição
<code>hospede_id</code>	UUID	Identificador único do hóspede

Exemplo de Request:

```
PATCH /v2/hospedes/d414bc8b-de66-4147-8be3-2e2585a458fb/noshow
```

Response (200 OK):

```
{
  "dados": {
    "hospede_id": "d414bc8b-de66-4147-8be3-2e2585a458fb",
    "situacao_id": "NOSHOW"
  }
}
```

Pessoas

1. Registrar Pessoa

POST /pessoas

⚠ **Nota:** Enviar no Header da requisição um CPF válido na variável "cpf_solicitante".

Headers:

Content-Type: application/json

cpf_solicitante: 12*****25

- **Descrição:** Registra os dados de uma nova pessoa no sistema.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para cadastrar uma nova pessoa, os dados são validados pelo CPF informado da pessoa .

Parâmetros Body:

Parâmetro	Tipo	Obrigatório	Descrição
nome	String	✓	Nome completo da pessoa
nome_social	String	✗	Nome social da pessoa
PaisNacionalidade_id	String	✓	País de nacionalidade da pessoa (ISO 3166-1 alpha-2)
genero_id	String	✓	Identificador do Gênero da pessoa (conforme domínio)
data_nascimento	Date	✓	Data de nascimento da pessoa (YYYY-MM-DD)

<code>raca_id</code>	String	✓	Identificador da Raça/etnia da pessoa (conforme domínio)
<code>deficiencia_id</code>	String	✓	Identificador da opção de deficiência (conforme domínio)
<code>tipo_deficiencia_id</code>	String	"deficiencia_id" = "SIM"	Identificador do tipo de deficiência (conforme domínio)
<code>numero_documento</code>	String	✓	Número de documento da pessoa (sem caracteres especiais)
<code>tipo_documento_id</code>	String	✓	Identificador do tipo de documento (conforme domínio)
<code>email</code>	String	✗	Email da pessoa
<code>telefone</code>	String	✗	Telefone da pessoa
<code>PaisResidencia_id</code>	String	✓	País de residência da pessoa (ISO 3166-1 alpha-2)
<code>cep</code>	String	✗	CEP no formato de 8 dígitos (sem caracteres especiais)
<code>logradouro</code>	String	cep ≠ vazio	Logradouro da residência pessoa
<code>numero</code>	String	cep ≠ vazio	Número da residência da pessoa
<code>complemento</code>	String	✗	Dados complementares da residência da pessoa
<code>bairro</code>	String	cep ≠ vazio	Descrição do bairro

cidade_id	Integer	PaisResidencia_id = "BR"	Código do município no formato de 7 dígitos (conforme IBGE)
estado_id	String	PaisResidencia_id = "BR"	Sigla do estado Formato 2 caracteres (Conforme IBGE)

Request Body:

```
{
  "pessoa": {
    "nome": "CARLOS SILVA",
    "nome_social": "",
    "PaisNacionalidade_id": "DE",
    "genero_id": "HOMEM",
    "data_nascimento": "1970-08-14",
    "raca_id": "BRANCA",
    "deficiencia_id": "NAO",
    "tipo_deficiencia_id": "",
    "documento_id": {
      "numero_documento": "P8993HH45",
      "tipo_documento_id": "PASSAPORTE"
    },
  },
  "contato": {
    "email": "abc@email.com.br",
    "telefone": "55934567890",
    "cep": "",
    "logradouro": "",
    "numero": "",
    "complemento": "",
    "bairro": "",
    "PaisResidencia_id": "AR",
    "cidade_id": 0,
    "estado_id": ""
  }
}
```

Response (200 OK):

```
{
  "pessoa_id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890"
}
```

2. Obter Pessoa por ID

GET /pessoas/{id}

- **Descrição:** Retorna os dados pessoais consultados pelo ID da pessoa.
- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint recuperar os dados de uma respectiva pessoa.

Parâmetros Path:

Parâmetro	Tipo	Descrição
id	UUID	Identificador único da pessoa

Exemplo de Request:

```
GET /v2/pessoas/df5a6ba6-e673-4e04-9777-d9466937c23c
```

Response (200 OK):

```
{
  "dados": {
    "id": "df5a6ba6-e673-4e04-9777-d9466937c23c",
    "documento": {
      "numero_documento": "12345678901",
      "tipo_documento": {
        "id": "CPF",
        "label": "CPF"
      }
    }
  },
  "dado_pessoal": {
    "nome": "JOÃO DA SILVA",
    "nome_social": "",
    "genero": {
      "id": "HOMEM",
      "label": "Homem"
    },
    "GeneroDescricao": "",
    "data_nascimento": "1985-05-15",
    "raca_id": {
      "id": "BRANCA",
      "label": "Branca"
    },
    "deficiencia": {
      "possui_deficiencia": {
        "id": "NAO",
        "label": "Não"
      }
    }
  }
}
```


- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint quando o hóspede chega. verificando se a pessoa já está cadastrada em sua base, seus dados são recuperados automaticamente.

Parâmetros Path:

Parâmetro	Tipo	Descrição
tipo	String	Tipo do documento (CPF ou PASSAPORTE)
numero	String	Número do documento

Exemplo de Request:

```
GET /v2/pessoas/documento/CPF/13427265706
```

Headers:

cpf_solicitante: 12345678900

Response (200 OK):

```
{
  "dados": {
    "documento": {
      "numero_documento": "13427265706",
      "tipo_documento": {
        "id": "CPF",
        "label": "CPF"
      }
    },
    "dado_pessoal": {
      "nome": "SAULO BOMFIM BRITO",
      "data_nascimento": "1989-08-16"
    },
    "contato": {
      "endereco": {
        "cep": "22760400",
        "logradouro": "GEREMARIO DANTAS",
        "numero": "1137",
        "complemento": "BLOCO 1 APT 501",
        "bairro": "FREGUESIA",
        "cidade": {
          "id": 3304557,
          "estado": {
            "uf": "RJ"
          }
        }
      }
    }
  }
}
```

```
}
}
}
}
```

Hospedagem

Registrar Hospedagem Completa

POST /hospedagem/registrar

⚠ **Nota:** Enviar no Header da requisição um CPF válido na variável "cpf_solicitante".

Headers:

cpf_solicitante: 12345678900

- **Descrição:** Cria a reserva e registra os hóspedes em uma única operação. Este é o endpoint mais completo do sistema, consolidando a criação de uma reserva com o registro de todas as informações dos hóspedes. Suporta múltiplos hóspedes em uma única requisição, facilitando o fluxo operacional.
- **Caso de Uso:** Os Meios de hospedagem (hotéis, pousadas) utilizam este endpoint para registrar uma nova reserva com todos os hóspedes em uma única chamada. (Impacta rastreabilidade das situações dos hóspedes e ocupação em tempo real)

Parâmetros Body

Parâmetro	Tipo	Obrigatório	Descrição
numero_reserva	String	✓	Descrição do código da reserva do Meio de Hospedagem (UNIQUE)
numero_reserva_OTA	String	✗	Descrição do código da reserva OTA

Parâmetro	Tipo	Obrigatório	Descrição
data_entrada	Date	✓	Data de entrada (YYYY-MM-DD)
data_saida	Date	✓	Data de saída (YYYY-MM-DD)
quantidade_hospede_adulto	Integer	✗	Quantidade de pessoa adulto na reserva
quantidade_hospede_menor	Integer	✗	Quantidade de pessoa menor na reserva
origem_reserva_id	String	✓	Identificador do tipo da reserva (MEIODEHOSPEDAGEM ou OTA)
is_principal	Booleano	✗	Indica se o hóspede é responsável pela reserva
situacao_hospede	String	✓	Identificador da situação do hóspede (conforme domínio)
check_in_em	DateTime	situacao_hospede = "CHECKIN_REALIZADO"	Data e hora check-in (ISO 8601 com timezone UTC (Z))
check_out_em	DateTime	situacao_hospede = "CHECKOUT_REALIZADO"	Data e hora check-out (ISO 8601 com timezone UTC (Z))
nome	String	✓	Nome completo da pessoa
nome_social	String	✗	Nome social da pessoa
PaisNacionalidade_id	String	✓	País de nacionalidade da

Parâmetro	Tipo	Obrigatório	Descrição
			pessoa (ISO 3166-1 alpha-2)
genero_id	String	✓	Identificador do Gênero da pessoa (conforme domínio)
data_nascimento	Date	✓	Data de nascimento da pessoa (YYYY-MM-DD)
raca_id	String	✓	Identificador da Raça/etnia da pessoa (conforme domínio)
deficiencia_id	String	✓	Identificador da opção de deficiência (conforme domínio)
tipo_deficiencia_id	String	"deficiencia_id" = "SIM"	Identificador do tipo de deficiência (conforme domínio)
numero_documento	String	✓	Número de documento da pessoa (sem caracteres especiais)
tipo_documento_id	String	✓	Identificador do tipo de documento (conforme domínio)
email	String	✗	Email da pessoa
telefone	String	✗	Telefone da pessoa
PaisResidencia_id	String	✓	País de residência da pessoa (ISO 3166-1 alpha-2)
cep	String	✗	CEP no formato de 8 dígitos(sem caracteres especiais)

Parâmetro	Tipo	Obrigatório	Descrição
logradouro	String	cep ≠ vazio	logradouro da residência pessoa
numero	String	cep ≠ vazio	Número da residência da pessoa
complemento	String	✗	Dados complementares da residência da pessoa
bairro	String	cep ≠ vazio	Descrição do bairro
cidade_id	Integer	PaisResidencia_id = "BR"	Código do município no formato de 7 dígitos (conforme IBGE)
estado_id	String	PaisResidencia_id = "BR"	Sigla do estado Formato 2 caracteres (Conforme IBGE)
numero_documento (responsavel)	String	data_nascimento do hóspede < 18 anos	Número de documento do responsável pelo hóspede (sem caracteres especiais)
tipo_documento (responsavel)	String	data_nascimento do hóspede < 18 anos	Identificador do tipo de documento (conforme domínio)
motivo_viagem_id (dados_ficha)	String	✓	Identificador do motivo da viagem (conforme domínio)
meio_transporte_id (dados_ficha)	String	✓	Identificador do meio de transporte (conforme domínio)

Exemplo Request Body:

```

{
  "reserva": {
    "numero_reserva": "RESERVA005",
    "numero_reserva_ota": "",
    "data_entrada": "2025-12-16",
    "data_saida": "2025-12-17",
    "quantidade_hospede_adulto": 1,
    "quantidade_hospede_menor": 0,
    "origem_reserva_id": "MEIOHOSPEDAGEM"
  },
  "dados_hospede": [
    {
      "is_principal": true,
      "situacao_hospede": "PRECHECKIN_PENDENTE",
      "check_in_em": "",
      "check_out_em": "",
      "dados_pessoais": {
        "nome": "MARCOS SILVA",
        "nome_social": "",
        "PaisNacionalidade_id": "TT",
        "genero_id": "HOMEM",
        "GeneroDescricao": "",
        "data_nascimento": "1970-08-14",
        "raca_id": "INDIGENA",
        "deficiencia_id": "NAO",
        "tipo_deficiencia_id": "",
        "documento_id": {
          "numero_documento": "12345678991",
          "tipo_documento_id": "CPF"
        },
        "contato": {
          "email": "abc@email.com.br",
          "telefone": "1234567890",
          "cep": "1256844",
          "logradouro": "",
          "numero": "",
          "complemento": "",
          "bairro": "CENTRO",
          "PaisResidencia_id": "BR",
          "cidade_id": 4303509,
          "estado_id": "RS"
        }
      },
      "responsavel": {
        "numero_documento": "",
        "tipo_documento_id": ""
      },
      "dados_ficha": {
        "motivo_viagem_id": "",
        "meio_transporte_id": ""
      }
    }
  ]
}

```

Response (200 OK):

```
{
  "dados": {
    "reserva": {
      "reserva_id": "1f4e94f3-f350-4dc7-a2a2-f42b31239a4b",
      "numero_reserva": "RESERVA005",
      "situacao_reserva_id": "CRIADA",
      "origem_reserva_id": "MEIOHOSPEDAGEM",
      "data_entrada": "2025-12-16",
      "data_saida": "2025-12-17",
      "quantidade_hospede_adulto": 1,
      "link_precheckin": "https://hom-
lowcode.serpro.gov.br/FNRH_Hospede/QRCodeFNRH?Source=..."
    },
    "dados_hospedes": [
      {
        "hospede_id": "8a18f4c0-3946-4a33-ae73-f085a7242d67",
        "hospede": {
          "is_principal": true,
          "pessoa_id": "962fef0f-b71f-4841-b50a-1bee60a70491",
          "reserva_id": "1f4e94f3-f350-4dc7-a2a2-f42b31239a4b",
          "situacao_hospede_id": "PRECHECKIN_PENDENTE"
        }
      }
    ]
  }
}
```

Observações Importantes:

- A resposta inclui o **link_precheckin** que pode ser enviado para o hóspede via email ou SMS
- O hóspede no exemplo é de Trinidad e Tobago (TT) e reside no Brasil (BR), demonstrando suporte v2

Fichas

Listar Fichas

GET /fichas

- **Descrição:** Busca as fichas cadastradas no sistema pelos filtros definidos.

- **Caso de Uso:** O Meio de Hospedagem utiliza este endpoint para buscar as fichas informadas ao Ministério do Turismo, baseadas em suas hospedagens conforme parâmetros pesquisados.

Parâmetros Query:

Parâmetro	Tipo	Obrigatório	Descrição
page_number	integer	✓	Número da página (começa em 0)
status	string	✗	Identificador da situação da ficha (conforme domínio)
tipo_documento	string	✗	Identificador do tipo de documento (conforme domínio)
numero_documento	string	✗	Número do documento
nome	string	✗	Nome do hóspede
data_inicial	date	✗	Data inicial (YYYY-MM-DD)
data_final	date	✗	Data final (YYYY-MM-DD)

Exemplo de Request:

```
GET /v2/fichas?page_number=0&status=AGUARDANDO_CHECKIN
```

Response (200 OK):

```
{
  "dados": [
    {
      "Id": "ficha-uuid-1",
      "ResponsavelId": "responsavel-uuid",
      "Status": "AGUARDANDO_CHECKIN",
      "MeioHospedagemId": "meio-hospedagem-uuid",
      "NumeroDocumento": "12345678901",
      "TipoDocumentoId": "CPF",
      "Nome": "JOÃO DA SILVA",
    }
  ]
}
```

```
"PaisNacionalidade_id": "BR",
"GeneroId": "HOMEM",
"DataNascimento": "1985-05-15",
"RacaId": "BRANCA",
"TipoDeficienciaId": "",
"DeficienciaId": "NAO",
"MotivoViagemId": "LAZER",
"MeioTransporteId": "AEREO",
"EntradaPrevistaEm": "2025-12-16",
"SaidaPrevistaEm": "2025-12-17",
"PreCheckinEm": "2025-12-01T10:00:00.000Z",
"CheckinEm": null,
"CheckoutEm": null,
"CriadaEm": "2025-12-01T09:00:00.000Z",
"UltimaAtualizacaoEm": "2025-12-01T10:00:00.000Z",
"StatusFichaId": "AGUARDANDO_CHECKIN",
"StatusFichaLabel": "Aguardando Check-in",
"PaisResidencia_id": "BR",
"UF": "SP",
"Cidade": "São Paulo",
"UltimaAlteracaoEm": "2025-12-01T10:00:00.000Z"
}
]
}
```

Códigos de Status

Código	Descrição
200	Sucesso
400	Requisição inválida
401	Não autenticado
403	Não autorizado
404	Recurso não encontrado
500	Erro interno do servidor

Códigos de Países (ISO 3166-1 alpha-2)

Os campos **PaisNacionalidade_id** e **PaisResidencia_id** utilizam códigos de país padronizados pela ISO 3166-1 alpha-2. Esta padronização internacional garante compatibilidade com sistemas de outros órgãos governamentais e base de dados internacionais.

Contexto de Utilização:

- **PaisNacionalidade_id**: Código do país onde a pessoa é cidadã/tem nacionalidade
- **PaisResidencia_id**: Código do país onde a pessoa atualmente reside

Estes campos são essenciais para:

- Compilação de estatísticas de turismo estrangeiro
- Controle de imigração e segurança
- Análise de fluxos de viajantes internacionais
- Relatórios para órgãos governamentais internacionais

Exemplos de Códigos Utilizados na Documentação:

Código	País	Utilização nos Exemplos
BR	Brasil	Nacionalidade mais comum (hóspedes nacionais)
AR	Argentina	Turista da América do Sul
DE	Alemanha	Turista europeu
US	Estados Unidos	Turista norte-americano
PT	Portugal	Turista europeu (CPLP)
ES	Espanha	Turista europeu
FR	França	Turista europeu
IT	Itália	Turista europeu
TT	Trinidad e Tobago	Exemplo em hospedagem registrada (hóspede estrangeiro)

Código	País	Utilização nos Exemplos
AL	Albânia	Exemplo em pré-check-ins (hóspede com passaporte)
ZA	África do Sul	Exemplo em pré-check-ins e dados de hóspede
UY	Uruguai	Turista da América do Sul
PY	Paraguai	Turista da América do Sul
CL	Chile	Turista da América do Sul
CO	Colômbia	Turista da América do Sul
PE	Peru	Turista da América do Sul
LU	Luxemburgo	Exemplo em pré-check-ins
EE	Estônia	Exemplo em pré-check-ins
KE	Quênia	Exemplo em pré-check-ins
AD	Andorra	Exemplo em pré-check-ins
SB	Ilhas Salomão	Exemplo em pré-check-ins
AF	Afeganistão	Exemplo em pré-check-ins

Nota Importante: A lista acima contém apenas os códigos utilizados nos exemplos desta documentação. O sistema suporta todos os códigos ISO 3166-1 alpha-2 para todos os países reconhecidos internacionalmente. Para uma lista completa, consulte a documentação oficial da ISO em www.iso.org.

Códigos de Cidades (IBGE)

O campo `cidade_id` utiliza o código numérico do IBGE para identificar cidades brasileiras de forma padronizada. Este código garante uniformidade com a base de dados oficial do Instituto Brasileiro de Geografia e Estatística.

Estrutura do Código IBGE:

- **Formato:** Número inteiro de 7 dígitos
- **Primeiros 2 dígitos:** Código do estado
- **Dígitos seguintes:** Identificação única da cidade dentro do estado
- **Função:** Vinculação com sistemas de geolocalização, estatísticas e planejamento urbano

Exemplos Utilizados na Documentação:

Código	Cidade	Estado	Utilização nos Exemplos
3550308	São Paulo	SP	Exemplo padrão em respostas
3304557	Rio de Janeiro	RJ	Consulta por documento (CEP 22760400)
4303509	Cachoeira do Sul	RS	Registro de hospedagem
5300108	Brasília	DF	Exemplo adicional

Notas Importantes:

- Os códigos IBGE são oficiais e podem ser consultados na base de dados do IBGE
- O campo `cidade_id` é sempre numérico
- Geralmente acompanhado pelo campo `estado_id` (sigla de 2 letras como "SP", "RJ", "RS")
- Essencial para análise geográfica de fluxos turísticos
- Alguns sistemas utilizam também o CEP (Código de Endereçamento Postal) como referência adicional

Notas de Migração v1 → v2

Pontos de Atenção

1. Substituição de campos de nacionalidade:

Os campos `tipo_nacionalidade_id` e `nacionalidade` foram substituídos por `PaisNacionalidade_id`.

- **v1:** Utilizava um campo texto livre `nacionalidade` que permitia valores arbitrários
- **v2:** Utiliza `PaisNacionalidade_id` com código ISO 3166-1 alpha-2, garantindo padronização
- **Impacto:** Integrações legadas que enviavam `"nacionalidade": "Italiano"` devem enviar `"PaisNacionalidade_id": "IT"`

2. Novo campo obrigatório para estrangeiros:

O campo `PaisResidencia_id` é importante para hóspedes que residem em país diferente da nacionalidade.

- **Necessidade:** Hóspedes estrangeiros precisam agora registrar tanto país de nacionalidade quanto país de residência
- **Exemplo:** Alemão (DE) residindo no Brasil (BR): `"PaisNacionalidade_id": "DE"` e `"PaisResidencia_id": "BR"`
- **Diferença em relação à v1:** Sistema anterior não diferenciava nacionalidade de residência

3. Compatibilidade de endpoints:

Os endpoints mantêm a mesma estrutura de URL, apenas os campos foram atualizados.

- **URLs:** Todas as URLs permanecem iguais (`/reservas`, `/hospedes`, `/pessoas`, etc.)
- **Métodos HTTP:** Métodos HTTP não mudaram (GET, POST, PUT, DELETE, PATCH)
- **Autenticação:** Basic Authentication continua sendo o padrão

4. Formato de data:

Continua utilizando ISO 8601:

- **Data:** `YYYY-MM-DD` (exemplo: `2025-12-16`)
- **DateTime:** `YYYY-MM-DDTHH:mm:ss.sssZ` (exemplo: `2025-12-16T14:30:00.000Z`)
- **Timezone:** Sempre em UTC (Z no final)

5. Validação mais rigorosa:

O sistema v2 valida mais rigorosamente os valores de país.

- **v1:** Aceitava valores livres

- **v2:** Aceita apenas códigos ISO válidos
- **Consequência:** Requisições com valores inválidos retornarão erro 400

Guia de Migração

Passo 1: Atualizar campos de nacionalidade

// v1 (DESATUALIZADO)

```
{
  "tipo_nacionalidade_id": "BRASILEIRO",
  "nacionalidade": "Brasil"
}
```

// v2 (CORRETO)

```
{
  "PaisNacionalidade_id": "BR"
}
```

Passo 2: Adicionar residência para hóspedes

// v1 - Não diferenciava

```
{
  "pais": "Brasil"
}
```

// v2 - Diferencia nacionalidade e residência

```
{
  "PaisNacionalidade_id": "DE",
  "PaisResidencia_id": "BR"
}
```

Passo 3: Atualizar validações

- Validar que **PaisNacionalidade_id** é sempre um código ISO válido
- Não permitir mais valores livres de texto para nacionalidade
- Validar que **PaisResidencia_id** também é um código ISO válido quando preenchido

☒ Benefícios da Migração

Benefício	Descrição
Padronização Internacional	Compatível com sistemas de outros países
Melhor Análise	Códigos padronizados permitem análises mais precisas
Conformidade	Alinhado com padrões de dados do Ministério do Turismo
Segurança	Validação rigorosa reduz erros de entrada
Interoperabilidade	Integração com sistemas de imigração e segurança

Changelog - Mudanças da v1 para v2

☒ Novos Campos Adicionados na v2

Campo	Descrição	Localização
PaisNacionalidade_id	Código ISO do país de nacionalidade (ex: "BR", "DE", "AR")	Pessoa, Hóspede, Ficha
PaisResidencia_id	Código ISO do país de residência	Contato/Endereço

☒ Campos Substituídos/Alterados

v1	v2	Descrição
tipo_nacionalidade_id	PaisNacionalidade_id	Mudança de conceito: de tipo para país específico
nacionalidade (string livre)	PaisNacionalidade_id (código ISO)	Padronização usando código de país

✕ Campos Removidos na v2

Campo	Motivo
<code>tipo_nacionalidade_id</code>	Substituído por <code>PaisNacionalidade_id</code>
<code>nacionalidade</code> (texto livre)	Padronizado para código de país
Referências a <code>Nacionalidade_ViewModel</code>	Substituído por campo direto de país