

Aqui estão alguns exemplos de possíveis argumentos que podem ser usados com o método `add_argument()` em um contexto de API, como Flask-RESTful. Esses argumentos ajudam a definir e validar os parâmetros que sua API pode esperar nas solicitações HTTP.

1. Argumentos Simples

Esses são exemplos básicos de argumentos que podem ser adicionados para capturar e validar parâmetros simples.

python

 Copiar código

```
atributos.add_argument('nome', type=str, required=True, help="O campo 'nome' é obrigatório.")
atributos.add_argument('idade', type=int, required=False, help="Informe a idade do usuário.")
atributos.add_argument('ativo', type=bool, required=False, help="Indica se o usuário está ativo.")
```

- `type` : Define o tipo de dado esperado (por exemplo: `str`, `int`, `float`, `bool`).
- `required` : Indica se o campo é obrigatório (`True` ou `False`).
- `help` : Define a mensagem de erro exibida quando uma validação falha.

2. Valores Padrão

Se você quiser definir um valor padrão para um argumento, caso ele não seja fornecido na solicitação:

python

 Copiar código

```
atributos.add_argument('categoria', type=str, default='geral', help="Categoria do item.")
```

- `default` : Valor que será atribuído ao argumento caso ele não seja fornecido.

3. Escolha de Valores (Choices)

Se você quiser restringir o argumento a um conjunto de valores possíveis:

python

 Copiar código

```
atributos.add_argument('genero', type=str, choices=['masculino', 'feminino', 'outro'], help="Escolha um gênero válido.")
```

- `choices` : Lista de valores permitidos para o campo. Se o valor enviado não estiver nessa lista, o parser retornará um erro.

4. Intervalos de Números (min/max)

Para limitar o valor de um argumento numérico, você pode usar `type` junto com validações adicionais.

```
python
```

 Copiar código

```
atributos.add_argument('preco', type=float, required=True, help="Informe um preço válido.")
```

- Aqui você pode também usar lógica adicional para validar se o preço está dentro de um intervalo específico (por exemplo, maior que zero).

5. Lista de Valores

Se você quiser permitir que o argumento aceite uma lista de valores (por exemplo, uma lista de números):

```
python
```

 Copiar código

```
atributos.add_argument('ids', type=int, action='append', help="Lista de IDs.")
```

- `action='append'` : Faz com que o valor seja acumulado em uma lista, ao invés de substituir o valor existente. Útil para capturar vários valores de um mesmo argumento.

6. Multi-Tipo (via lambdas)

Se quiser processar um tipo de dado mais complexo, pode utilizar uma função `lambda` no `type`.

```
python
```

 Copiar código

```
atributos.add_argument('data', type=lambda x: datetime.strptime(x, '%Y-%m-%d'), help="Data no formato AAAA-MM-DD.")
```

- `lambda` : Permite personalizar como o dado será convertido, como no caso de uma data.

7. Local de Origem (onde o dado é esperado)

Por padrão, o parser procura argumentos tanto no corpo da requisição quanto na URL, mas você pode especificar de onde o dado deve vir.

```
python
```

 Copiar código

```
atributos.add_argument('token', type=str, location='headers', required=True, help="Token de autenticação faltando.")
```

- `location` : Especifica onde o argumento deve ser buscado ('json', 'args', 'form', 'headers', 'cookies', etc.).

8. Ocultar Argumentos Sensíveis

Para evitar a exposição de argumentos sensíveis em logs ou mensagens de erro, você pode usar a opção `dest`.

python

 Copiar código

```
atributos.add_argument('senha', type=str, required=True, dest='senha_hash',
help="Campo senha é obrigatório.")
```

- `dest` : Permite usar um nome diferente para o argumento internamente, sem alterar como ele é fornecido na solicitação.

9. Verificação de Tamanho de String (ou de Lista)

Para garantir que um argumento de texto tenha um tamanho mínimo ou máximo:

python

 Copiar código

```
atributos.add_argument('descricao', type=str, required=True, help="Descrição do item",
trim=True)
```

- `trim=True` : Remove espaços em branco à esquerda e à direita do valor fornecido.

Exemplos Completos:

python

 Copiar código

```
atributos.add_argument('nome', type=str, required=True, help="O nome é obrigatório.")
atributos.add_argument('email', type=str, required=True, help="O email é obrigatório.")
atributos.add_argument('idade', type=int, required=False, default=18, help="Idade do usuário.")
atributos.add_argument('genero', type=str, choices=['masculino', 'feminino', 'outro'], help="Gênero do usuário.")
atributos.add_argument('tags', type=str, action='append', help="Lista de tags associadas ao item.")
atributos.add_argument('preco', type=float, help="Preço do item.")
atributos.add_argument('token', type=str, location='headers', help="Token de autenticação.")
```