

Documentação da API - NatureLens

Documentação da API - Vertex AI API

1. Propósito e Funcionalidade

Descrição Geral:

- **Propósito:** *O propósito da Vertex AI API é fornecer uma plataforma poderosa para criar, treinar e implantar modelos de machine learning. A API ajuda a integrar recursos avançados de IA, como reconhecimento de imagens, dentro de aplicativos para facilitar a análise e automação de tarefas complexas, como a identificação de plantas e animais.*
 - **Funcionalidade:** *A Vertex AI oferece a funcionalidade de gerenciar pipelines de machine learning, treinar modelos com dados personalizados ou pré-treinados, e realizar inferências em tempo real. Ela resolve o problema de criar e escalar modelos de IA com infraestrutura gerenciada, garantindo precisão e performance.*
-

2. Especificação Técnica

Endpoints e Métodos:

- URL Base: <https://vertex.googleapis.com/v1>

Endpoints:

- Treinamento de Modelos
 - Endpoint:
/projects/{project-id}/locations/{location-id}/models
 - Método Suportado: *POST*

Parâmetros de Requisição:

- project-id (Tipo: String, Obrigatório): *Identificador do projeto no Google Cloud.*
- location-id (Tipo: String, Obrigatório): *Localização do recurso no Google Cloud.*
- Execução de Inferências
 - Endpoint:
/projects/{project-id}/locations/{location-id}/models/{model-id}
 - Método Suportado: *POST*

Parâmetros de Requisição:

- project-id (Tipo: String, Obrigatório): *Identificador do projeto.*
- location-id (Tipo: String, Obrigatório): *Localização do recurso.*
- model-id (Tipo: String, Obrigatório): *Identificador do modelo.*

Parâmetros na URL ou Corpo da Requisição:

- image (Tipo: Base64, Obrigatório): *A imagem a ser analisada pela IA.*

- **modelParameters** (Tipo: JSON, Opcional): *Parâmetros adicionais para a previsão.*

Formato de Dados:

- Entrada: *JSON*
- Saída: *JSON*

Exemplo de Requisição:

http

Copiar código

POST

`/projects/{project-id}/locations/{location-id}/models/{model-id}:predict HTTP/1.1`

`Host: vertex.googleapis.com`

`Authorization: Bearer {token}`

`Content-Type: application/json`

```
{
  "instances": [
    {
      "image": { "bytes":
"<base64-encoded-image>" }
    }
  ]
}
```

Exemplo de Resposta:

json

Copiar código

```
{
  "predictions": [
    {
      "displayName": "Cactus",
      "confidence": 0.98
    }
  ]
}
```

Códigos de Resposta e Descrições:

- 200 OK: *Sucesso na execução da operação.*
- 400 Bad Request: *Parâmetros inválidos ou mal formatados.*
- 401 Unauthorized: *Token de autenticação inválido ou ausente.*
- 404 Not Found: *Modelo ou recurso não encontrado.*
- 500 Internal Server Error: *Erro interno no servidor.*

Erros Comuns:

- Erro 400 - Parâmetro Inválido: *Verifique os parâmetros obrigatórios e o formato da requisição.*
 - Erro 401 - Token Inválido: *Certifique-se de que o token de autenticação está correto e não expirou.*
-

3. Segurança e Autorização

Autenticação:

- Método: *OAuth2 com token Bearer.*

Exemplo de Cabeçalho de Autenticação:

http

Copiar código

Authorization: Bearer {token}

Autorização:

- *Para acessar os endpoints da API, o cliente deve ter as permissões corretas definidas no Google Cloud IAM (Identity and Access Management).*

Medidas de Proteção de Dados:

- *Transmissão Segura: Todas as requisições devem ser feitas via HTTPS para garantir a segurança dos dados em trânsito.*
 - *Criptografia: Dados sensíveis são criptografados durante o armazenamento no Google Cloud.*
-

4. Monitoramento e Performance

Monitoramento:

- *Ferramentas Utilizadas: O Google Cloud Monitoring (anteriormente Stackdriver) pode ser utilizado para monitorar a performance da API.*
- *Métricas Monitoradas: Tempo de resposta, taxa de erro, latência e uso de recursos.*

Desempenho:

- **Limites de Taxa (Rate Limits):** *A API tem limites de requisições por minuto, dependendo do nível de serviço contratado.*
- **Otimização:** *Utilização de caching e balanceamento de carga para garantir respostas rápidas e distribuição de requisições.*

Escalabilidade:

- *A Vertex AI lida automaticamente com aumentos de carga, permitindo escalabilidade horizontal com balanceamento de carga integrado.*
-

5. Versionamento e Compatibilidade

Política de Versionamento:

- *A API segue uma política de versionamento para garantir que as alterações não quebrem funcionalidades existentes. Exemplo: /v1/*

Compatibilidade:

- *As mudanças são comunicadas por meio da documentação oficial e anúncios no Google Cloud. Normalmente, as versões anteriores da API permanecem disponíveis para garantir compatibilidade retroativa.*