

ワールドカップ映像分析 連携ガイド

Pegasus 1.5 Time-Based Metadata API · リクエスト/レスポンス形式および連携ガイド

本ドキュメントは、TwelveLabs Pegasus 1.5 時間ベースメタデータ（TBM）APIを用いたワールドカップ放送映像分析の連携ガイドです。デモと同じリクエストを送信すると同一形式のJSONレスポンスが返却されます。放送局側でこれをパースし、任意のUIに適用していただけます。

デモ デモ: tbm-demo-alpha.vercel.app · 下のリクエスト形式・レスポンス形式セクションから、実際のリクエスト・レスポンスJSONをダウンロードできます。（デモ画面右上のボタンからも可能）

01 API 概要

項目	値
Base URL	<code>https://api.twelvelabs.io/v1.3</code>
分析リクエスト（非同期）	<code>POST /analyze/tasks</code>
結果取得	<code>GET /analyze/tasks/{task_id}</code> — status: ready になるまでポーリング
認証	<code>x-api-key: <API_KEY></code>
モデル	<code>pegasus1.5</code>
分析モード	<code>analysis_mode: "time_based_metadata"</code>
公式ドキュメント	<code>docs.twelvelabs.io</code>

分析結果はタスクの `result.data` にJSON文字列として格納されます。クライアント側でもう一度パースすると、以下のレスポンス構造が得られます。

02 リクエスト形式

抽出する項目を `segment_definitions` で定義します。各定義はセグメント分割の指示 (`description`) と抽出フィールド (`fields`) のリストで構成され、自然言語で設計できます。以下は本デモ生成に使用した実際のリクエストです (3レイヤー: ショット境界 / サッカーイベント / 放送構成)。

```
{
  "model_name": "pegasus1.5",
  "video": {
    "type": "asset_id",
    "asset_id": "6a170c7dc6d6bda7c4c9b08d"
  },
  "analysis_mode": "time_based_metadata",
  "response_format": {
    "type": "segment_definitions",
    "segment_definitions": [
      {
        "id": "L0_shot_boundaries",
        "description": "Detect all shot boundaries in this soccer broadcast.",
        "fields": [
          { "name": "shot_type", "type": "string",
            "enum": ["WIDE_PITCH", "PLAYER_CLOSE_UP", "CROWD", "BENCH", "OTHER"] },
          { "name": "description", "type": "string" },
          { "name": "main_actions", "type": "array", "items": {"type": "string"} },
          { "name": "emotional_elements", "type": "string" },
          { "name": "player_names", "type": "array", "items": {"type": "string"} }
        ]
      },
      {
        "id": "H16_soccer_events",
        "description": "Identify soccer gameplay events, live or replayed.",
        "fields": [
          { "name": "event_type", "type": "string",
            "enum": ["GOAL", "SHOT_ON_TARGET", "SHOT_OFF_TARGET", "HEADER",
              "SAVE", "FREE_KICK", "CORNER_KICK", "PENALTY_KICK",
              "FOUL", "YELLOW_CARD", "RED_CARD", "VAR_REVIEW", "CELEBRATION"] },
          { "name": "playback_status", "type": "string",
            "enum": ["LIVE_GAMEPLAY", "REPLAY"] },
          { "name": "match_start_time", "type": "string" },
          { "name": "match_end_time", "type": "string" },
          { "name": "description", "type": "string" },
          { "name": "player_names", "type": "array", "items": {"type": "string"} }
        ]
      },
      {
        "id": "H16_broadcast_structure",
        "description": "Classify non-live inserts: replays, halftime, interviews.",
        "fields": [
          { "name": "segment_type", "type": "string",
            "enum": ["REPLAY_ANALYSIS", "HALFTIME", "POST_MATCH",
              "PLAYER_SPOTLIGHT", "CROWD_REACTION", "INTERVIEW"] },
          { "name": "description", "type": "string" },
          { "name": "player_names", "type": "array", "items": {"type": "string"} },
          { "name": "emotional_state", "type": "string" }
        ]
      }
    ]
  }
}
```

フィールドごとの詳細な指示を含む完全なdescription文は、デモのAPI Requestボタンからダウンロードできるファイルに収録されています。

03 レスポンス形式

レスポンスは各 `segment_definition` の `id` をキーとするセグメント配列です。各セグメントは `start_time / end_time` (秒) と、定義したフィールドを含む `metadata` を持ちます。

```
{
  "H16_soccer_events": [
    {
      "start_time": 53.0,
      "end_time": 84.0,
      "metadata": {
        "event_type": "GOAL",
        "playback_status": "LIVE_GAMEPLAY",
        "match_start_time": "00:53",
        "match_end_time": "00:58",
        "description": "Pereira scores for Fulham with a chip
          over the goalkeeper.",
        "player_names": ["Andreas Pereira", "Aaron Ramsdale"]
      }
    }
  ]
}
```

レイヤー別フィールド

レイヤー (ID)	主要フィールド
L0_shot_boundaries	shot_type, description, main_actions[], emotional_elements, player_names[]
H16_soccer_events	event_type, playback_status (LIVE/REPLAY), match_start_time, match_end_time, description, player_names[]
H16_broadcast_structure	segment_type, description, player_names[], emotional_state

全レスポンスのサンプル (3レイヤーすべて) はデモのResponse JSONボタンからダウンロードできます。

04 分析構成 (3つの方式)

放送局のオンデバイス/オフライン優先環境を考慮し、方式AまたはBから始めることを推奨します。

方式	構成	インターネット	適合用途
A - 優先	Pegasus単独: イベントJSONを抽出してテキスト検索	抽出のみクラウド	放送リアルタイム (オフライン)
B - 次に	Pegasus + Bedrock Claude: JSONベースの連続イベント分析/自然言語クエリ	Bedrock (事前計算可能)	ゴール直前のビルドアップ等の深度分析
C - 将来	Jockey + Bedrock Claude: マルチモーダルエージェント検索	必要	プレショーのオープンエンド検索

方式Bは試合JSONをClaudeプロンプトにそのまま渡すため、別途ベクターDBなしで実装可能です。

05 連携パイプライン

最もシンプルな形態（方式A）の連携フローです。パイプラインの実装と最終的なUI決定は放送局側で行っていただきます。AWSインフラ（S3、signed URL等）は必要に応じてガイドいたします。

- 1 （放送局）メディアファイルをS3にアップロード
- 2 （放送局）署名付きURL（signed URL）を生成
- 3 Pegasus 1.5 TBM API を呼び出す — POST /analyze/tasks
- 4 結果JSONをS3に保存
- 5 （放送局）JSONをパースしてフロントエンド/UIに適用

方式Bでは、ステップ4の後に「Bedrock ClaudeでJSONを分析」するステップが追加されます。