Analysis Output Format

Overview

This document describes the standardized JSON output format for swing analysis results. This format is used by:
- UI components (Dashboard, Video Detail, Progress)
- Coach Rick AI for personalized explanations
- API endpoints for third-party integrations

JSON Structure

```json
{
  "videoId": "string",
  "athlete": {
    "name": "string",
    "level": "MLB | College | HS | Youth",
    "age": 0,
    "bats": "R | L | S",
    "throws": "R | L | S"
  },
  "scores": {
    "momentumTransfer": {
      "score": 0,
      "goatyBand": -3,
      "goatyLabel": "Elite | Advanced | Above Average | Average | Below Average | Needs Work",
      "confidence": 0.0
    },
    "anchor": {
      "score": 0,
      "label": "Ground → Hips",
      "leakSeverity": "none | mild | moderate | severe"
    },
    "engine": {
      "score": 0,
      "label": "Hips → Torso",
      "leakSeverity": "none | mild | moderate | severe"
    },
    "whip": {
      "score": 0,
      "label": "Torso → Barrel",
      "leakSeverity": "none | mild | moderate | severe"
    }
  },
  "timing": {
    "abRatio": 0.0,
    "loadDurationMs": 0,
    "swingDurationMs": 0,
    "sequenceOrder": ["pelvis", "torso", "hands", "bat"],
    "segmentGapsMs": {
      "pelvisToTorso": 0,
      "torsoToHands": 0,
      "handsToBat": 0
    }
  },
  "flags": {
    "mainLeak": "anchor | engine | whip | none",
    "secondaryLeak": "anchor | engine | whip | null",
    "sequenceBroken": false
  },
  "coachSummary": {
    "overall": "string – high-level meaning of the momentum transfer score.",
    "leak": "string – where the main leak is (Anchor / Engine / Whip).",
    "nextStep": "string – one simple next focus, no drill names yet."
  }
}
```

Field Descriptions

`videoId`

- **Type**: `string`
- **Description**: Unique identifier for the analyzed video
- **Example**: `"cm48xj2k...."`

`athlete`

- **Type**: `object`
- **Description**: Information about the player
- **Fields**:
- `name`: Player's full name
- `level`: Competitive level (MLB, College, HS, Youth)
- `age`: Current age in years
- `bats`: Batting handedness (R=Right, L=Left, S=Switch)
- `throws`: Throwing handedness

`scores`

`momentumTransfer`

- **Type**: `object`
- **Description**: The master metric (60% of final score)
- **Fields**:
- `score`: 0-100 momentum transfer quality
- `goatyBand`: -3 to +3 band mapping
- `goatyLabel`: Human-readable band label
- `confidence`: 0.0-1.0 confidence in the analysis

GOATY Band Mapping:
- `+3` or `+2`: Elite (92-100)
- `+1`: Advanced (85-91)
- `0`: Above Average (75-84)
- `-1`: Average (60-74)
- `-2`: Below Average (45-59)
- `-3`: Needs Work (0-44)

`anchor` / `engine` / `whip`

- **Type**: `object`
- **Description**: Sub-scores for energy transfer stages (40% total weight)
- **Fields**:
- `score`: 0-100 quality score
- `label`: Description of what this stage measures
- `leakSeverity`: How much energy is lost at this stage

Leak Severity:
- `none`: Score within 5 points of momentum transfer score
- `mild`: 5-9 point gap
- `moderate`: 10-14 point gap
- `severe`: 15+ point gap

`timing`

- **Type**: `object`
- **Description**: Detailed timing metrics
- **Fields**:
- `abRatio` : Acceleration-to-Brake ratio (ideal: 1.2-1.5)
- `loadDurationMs` : Time from start to launch (ms)
- `swingDurationMs` : Time from launch to impact (ms)
- `sequenceOrder` : Array showing segment firing order
- `segmentGapsMs` : Timing gaps between body segments

Ideal Timing Gaps:
- `pelvisToTorso` : 30-50ms
- `torsoToHands` : 20-40ms
- `handsToBat` : 10-30ms

`flags`

- **Type**: `object`
- **Description**: Quick-reference indicators
- **Fields**:
- `mainLeak` : Primary area losing energy
- `secondaryLeak` : Secondary area (if significant)
- `sequenceBroken` : True if kinematic sequence is out of order

`coachSummary`

- **Type**: `object`
- **Description**: AI-generated coaching text
- **Fields**:
- `overall` : 1-2 sentences about the momentum transfer score
- `leak` : 1-2 sentences identifying where energy is lost
- `nextStep` : 1 sentence with a simple feel cue

Example:

```
"coachSummary": {
  "overall": "Your momentum transfer is 78, which is above average. The pattern is
solid — now we're chasing tiny efficiency gains.",
  "leak":
"Your lower body is late or unstable, so the hips can't pass clean energy up the
chain.",
  "nextStep": "Next step: Learn to load into the ground and hold it so your hips can
fire at the right time."
}
```

API Endpoints

Get Analysis Summary

Endpoint: `GET /api/videos/[id]/analysis-summary`

Authentication: Required (session-based)

Response: Returns the complete analysis output in the format above

Example:

```
curl -X GET \
  http://localhost:3000/api/videos/cm48xj2k.../analysis-summary \
  -H 'Cookie: next-auth.session-token=...'
```

Response (200 OK):

```json
{
  "videoId": "cm48xj2k...",
  "athlete": {
    "name": "John Smith",
    "level": "HS",
    "age": 16,
    "bats": "R",
    "throws": "R"
  },
  "scores": {
    "momentumTransfer": {
      "score": 78,
      "goatyBand": 0,
      "goatyLabel": "Above Average",
      "confidence": 0.87
    },
    "anchor": {
      "score": 72,
      "label": "Ground → Hips",
      "leakSeverity": "mild"
    },
    "engine": {
      "score": 82,
      "label": "Hips → Torso",
      "leakSeverity": "none"
    },
    "whip": {
      "score": 75,
      "label": "Torso → Barrel",
      "leakSeverity": "mild"
    }
  },
  "timing": {
    "abRatio": 1.35,
    "loadDurationMs": 450,
    "swingDurationMs": 180,
    "sequenceOrder": ["pelvis", "torso", "hands", "bat"],
    "segmentGapsMs": {
      "pelvisToTorso": 42,
      "torsoToHands": 35,
      "handsToBat": 25
    }
  },
  "flags": {
    "mainLeak": "anchor",
    "secondaryLeak": null,
    "sequenceBroken": false
  },
  "coachSummary": {
    "overall": "Your momentum transfer is 78, which is above average. You're creating
good flow through the body, but the lower half is still a little late, so the hips
can't pass clean energy up the chain.",
    "leak": "Your lower body is late or unstable, so the hips can't pass clean energy
up the chain.",
    "nextStep":
"Next step: Learn to load into the ground and hold it so your hips can fire at the
right time."
  }
}
```

Error Responses:
- `401 Unauthorized` : No valid session
- `403 Forbidden` : Video belongs to another user
- `404 Not Found` : Video doesn’t exist
- `400 Bad Request` : Video hasn’t been analyzed yet

Usage Examples

React Component

```
import { AnalysisOutput } from '@/lib/scoring/analysis-output';
import { MomentumTransferCard } from '@/components/momentum-transfer-card';

function VideoAnalysis({ videoId }: { videoId: string }) {
  const [analysis, setAnalysis] = useState<AnalysisOutput ▯ null>(null);

  useEffect(() => {
    fetch(`/api/videos/${videoId}/analysis-summary`)
      .then(res => res.json())
      .then(data => setAnalysis(data));
  }, [videoId]);

  if (!analysis) return <div>Loading...</div>;

  return (
    <div>
      <MomentumTransferCard
        momentumTransferScore={analysis.scores.momentumTransfer.score}
        anchorScore={analysis.scores.anchor.score}
        engineScore={analysis.scores.engine.score}
        whipScore={analysis.scores.whip.score}
        goatyBand={analysis.scores.momentumTransfer.goatyBand}
        goatyBandLabel={analysis.scores.momentumTransfer.goatyLabel}
        showCoaching={true}
      />
    </div>
  );
}
```

Coach Rick Integration

Coach Rick automatically receives this data in the system prompt when a user asks about their swing:

```
const response = await fetch('/api/coach-rick', {
  method: 'POST',
  body: JSON.stringify({
    message: 'What should I work on?',
    videoId: 'cm48xj2k...', // Coach Rick fetches analysis summary
  }),
});
```

Coach Rick will use the `coachSummary` fields and raw scores to provide personalized feedback.

Implementation Notes

Data Flow

1. **Video Upload** → Stored in database
2. **Analysis** → Scoring engine runs (`/api/videos/[id]/analyze`)
3. **Storage** → Results saved in `Video.newScoringBreakdown`
4. **Retrieval** → `/api/videos/[id]/analysis-summary` formats output
5. **Display** → UI components consume standardized format

Type Safety

All types are defined in `lib/scoring/analysis-output.ts` :

```
import {
  AnalysisOutput,
  AthleteInfo,
  Scores,
  TimingData,
  Flags,
  CoachSummary
} from '@/lib/scoring/analysis-output';
```

Backward Compatibility

The new format coexists with legacy fields:
- Legacy fields: `overallScore` , `anchor` , `engine` , `whip`
- New fields: `newScoringBreakdown` , `goatyBand`

UI components can gracefully fall back to legacy data if `newScoringBreakdown` is null.

Future Enhancements

Phase 2: Drill Recommendations

When drill mapping is implemented, add:

```
"recommendations": {
  "primaryDrill": {
    "name": "Kwon Step-In",
    "category": "anchor",
    "description": "...",
    "videoUrl": "..."
  },
  "alternativeDrills": [ ... ]
}
```

Phase 3: Historical Comparison

Add a `comparison` object:

```
"comparison": {
  "previousScore": 72,
  "delta": +6,
  "trend": "improving",
  "sessionsAnalyzed": 8
}
```

Testing

To test the analysis output:

1. **Upload and analyze a video**
2. **Get the video ID** from the database or UI
3. **Call the API**:
   `bash`
   `   curl http://localhost:3000/api/videos/{videoId}/analysis-summary`
4. **Verify the structure** matches the format above

Summary

✅ **Standardized format** for all swing analysis results
✅ **Type-safe** with TypeScript definitions
✅ **Coach Rick ready** – AI can consume and explain
✅ **UI ready** – Components can render immediately
✅ **Extensible** – Easy to add drill recommendations later

This format is the **single source of truth** for how the app communicates swing analysis data.