

# Frontend Integration Guide - Complete API Documentation

---

## Table of Contents

- 1. [Overview](#)
  - 2. [Base URLs & Authentication](#)
  - 3. [User Journey Flow](#)
  - 4. [Phase 1: Initial Assessment \(Day 91\)](#)
  - 5. [Phase 2: Assessment Results & Dashboard Setup](#)
  - 6. [Phase 3: Daily Learning Dashboard](#)
  - 7. [Phase 4: Daily Tasks \(Day 1 onwards\)](#)
  - 8. [Error Handling](#)
  - 9. [Complete Code Examples](#)
- 

## Overview

This guide covers the complete integration flow for the Ziya LRG (Listening, Reading, Grammar) platform including Writing and Speaking modules.

### User Journey:

```
Sign Up → Initial Assessment (Day 91) → Assessment Report → Dashboard → Daily Tasks (Day 1→90)
```

---

## Base URLs & Authentication

### API Base URLs

```
const LRG_API_BASE = "http://localhost:8080/api/v1";
const WRITING_SPEAKING_API_BASE = "https://your-microservice-url.com"; // To be provided
```

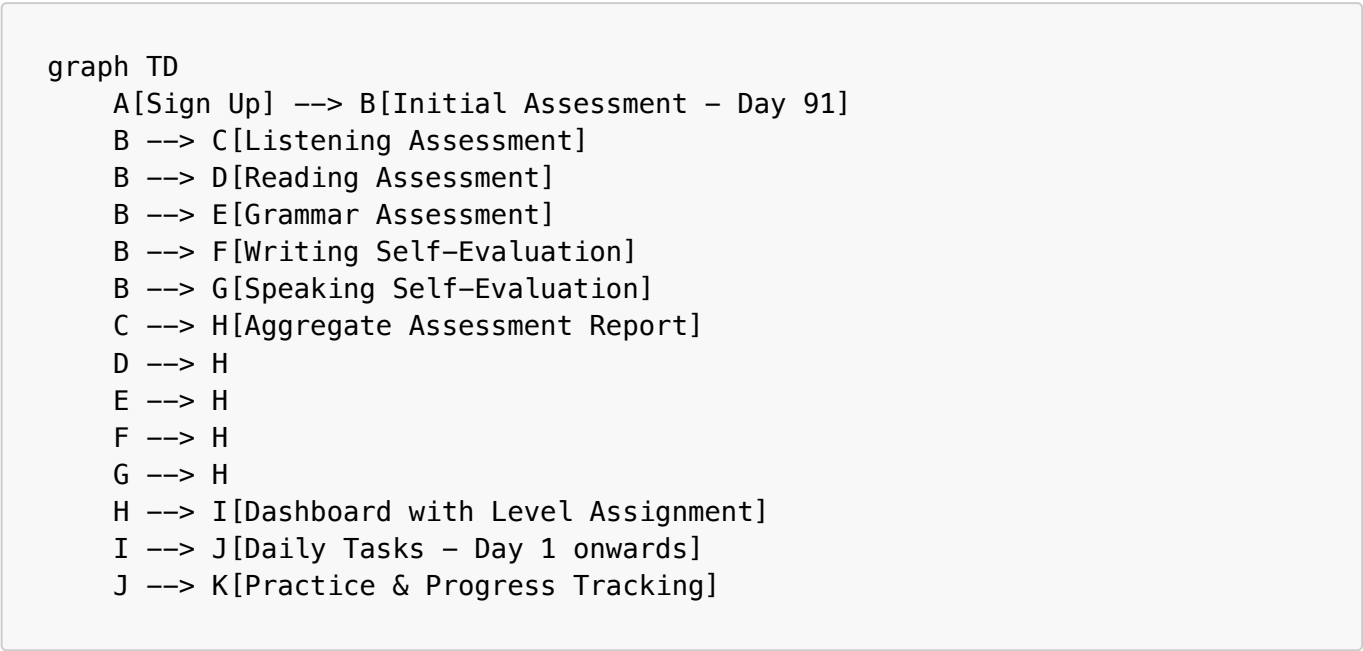
### Authentication

All endpoints require JWT authentication:

```
headers: {
  "Authorization": "Bearer <your-jwt-token>",
  "Content-Type": "application/json"
}
```

---

## User Journey Flow



## Phase 1: Initial Assessment (Day 91)

After sign-up, users complete a comprehensive assessment using **Day 91** content to determine their level (beginner, intermediate, advanced).

### 1.1 Listening Assessment

Step 1: Get Listening Content for Day 91

```
GET /api/v1/listening/day/day91
Authorization: Bearer <token>
```

Response:

```
{
  "listening_id": "uuid",
  "day_code": "day91",
  "title": "Assessment: Business Conversation",
  "audio_url": "https://storage.example.com/audio/day91.mp3",
  "audio_duration_sec": 180,
  "difficulty_level": "intermediate",
  "questions": [
    {
      "question_id": "q1",
      "question_text": "What is the main topic of the conversation?",
      "question_type": "multiple_choice",
      "options": ["A", "B", "C", "D"],
      "skill": "main_idea"
```

```
    },
    {
      "question_id": "q2",
      "question_text": "How many people are speaking?",
      "question_type": "fill_blank",
      "skill": "details"
    }
  ]
}
```

## Step 2: Start Listening Session

**POST** /api/v1/listening/sessions

**Authorization:** Bearer <token>

**Content-Type:** application/json

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "day_code": "day91",
  "audio_url": "https://storage.example.com/audio/day91.mp3"
}
```

## Response:

```
{
  "session_id": "750e8400-e29b-41d4-a716-446655440001",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "day_code": "day91",
  "audio_url": "https://storage.example.com/audio/day91.mp3",
  "started_at": "2025-01-15T10:30:00Z",
  "message": "Session started successfully"
}
```

## Step 3: Submit Listening Answers

**POST** /api/v1/listening/sessions/{session\_id}/submit

**Authorization:** Bearer <token>

**Content-Type:** application/json

```
{
  "answers": [
    {
      "item_id": "q1",
      "user_answer": "Business meeting",
      "correct_answer": "Business meeting",
      "is_correct": true,
      "time_spent_sec": 45,
      "skill": "main_idea",
    }
  ]
}
```

```
    "question_type": "multiple_choice"
  },
  {
    "item_id": "q2",
    "user_answer": "3",
    "correct_answer": "2",
    "is_correct": false,
    "time_spent_sec": 30,
    "skill": "details",
    "question_type": "fill_blank"
  }
],
"duration_sec": 300,
"score_pct": 75,
"xp_earned": 25,
"audio_replay_count": 2
}
```

**Response:**

```
{
  "session_id": "750e8400-e29b-41d4-a716-446655440001",
  "score_pct": 75,
  "xp_earned": 25,
  "skill_mastery_updated": true,
  "audio_replay_count": 2,
  "message": "Listening assessment completed"
}
```

---

## 1.2 Reading Assessment

### Step 1: Get Reading Content for Day 91

```
GET /api/v1/reading/day/day91
Authorization: Bearer <token>
```

**Response:**

```
{
  "reading_id": "uuid",
  "day_code": "day91",
  "title": "Assessment: Climate Change Article",
  "passage": "Full text of the reading passage...",
  "difficulty_level": "intermediate",
  "questions": [
    {
      "question_id": "r1",
```

```
    "question_text": "What is the author's main argument?",
    "question_type": "multiple_choice",
    "options": ["A", "B", "C", "D"],
    "skill": "comprehension"
  },
  {
    "question_id": "r2",
    "question_text": "What does the word 'mitigate' mean in context?",
    "question_type": "vocabulary",
    "skill": "vocabulary"
  }
]
```

## Step 2: Start Reading Session

```
POST /api/v1/sessions
Authorization: Bearer <token>
Content-Type: application/json

{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "modality": "reading",
  "day_code": "day91"
}
```

## Response:

```
{
  "session_id": "850e8400-e29b-41d4-a716-446655440002",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "modality": "reading",
  "day_code": "day91",
  "started_at": "2025-01-15T11:00:00Z",
  "message": "Session started successfully"
}
```

## Step 3: Submit Reading Answers

```
POST /api/v1/sessions/{session_id}/submit
Authorization: Bearer <token>
Content-Type: application/json

{
  "answers": [
    {
      "item_id": "r1",
      "user_answer": "Climate action needed",

```

```
    "correct_answer": "Climate action needed",
    "is_correct": true,
    "time_spent_sec": 90,
    "skill": "comprehension"
  },
  {
    "item_id": "r2",
    "user_answer": "reduce",
    "correct_answer": "reduce",
    "is_correct": true,
    "time_spent_sec": 60,
    "skill": "vocabulary"
  }
],
"duration_sec": 450,
"score_pct": 85,
"xp_earned": 30
}
```

**Response:**

```
{
  "session_id": "850e8400-e29b-41d4-a716-446655440002",
  "score_pct": 85,
  "xp_earned": 30,
  "badges_earned": [],
  "streak_updated": true,
  "current_streak": 1,
  "message": "Reading assessment completed"
}
```

---

## 1.3 Grammar Assessment

### Step 1: Get Grammar Content for Day 91

```
GET /api/v1/grammar/day/day91
Authorization: Bearer <token>
```

**Response:**

```
{
  "grammar_id": "uuid",
  "day_code": "day91",
  "topic": "Assessment: Mixed Grammar",
  "difficulty_level": "intermediate",
  "questions": [
    {
```

```
    "question_id": "g1",
    "question_text": "Choose the correct form: She ___ been working here
for 5 years.",
    "question_type": "multiple_choice",
    "options": ["has", "have", "had", "having"],
    "skill": "verb_tenses",
    "correct_answer": "has"
  },
  {
    "question_id": "g2",
    "question_text": "Identify the error in: 'He don't like coffee.'",
    "question_type": "error_identification",
    "skill": "subject_verb_agreement"
  }
]
```

## Step 2: Start Grammar Session

```
POST /api/v1/sessions
Authorization: Bearer <token>
Content-Type: application/json

{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "modality": "grammar",
  "day_code": "day91"
}
```

## Response:

```
{
  "session_id": "950e8400-e29b-41d4-a716-446655440003",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "modality": "grammar",
  "day_code": "day91",
  "started_at": "2025-01-15T12:00:00Z",
  "message": "Session started successfully"
}
```

## Step 3: Submit Grammar Answers

```
POST /api/v1/sessions/{session_id}/submit
Authorization: Bearer <token>
Content-Type: application/json

{
  "answers": [
```

```
{
  "item_id": "g1",
  "user_answer": "has",
  "correct_answer": "has",
  "is_correct": true,
  "time_spent_sec": 30,
  "skill": "verb_tenses"
},
{
  "item_id": "g2",
  "user_answer": "don't should be doesn't",
  "correct_answer": "doesn't",
  "is_correct": true,
  "time_spent_sec": 45,
  "skill": "subject_verb_agreement"
}
],
"duration_sec": 240,
"score_pct": 90,
"xp_earned": 35
}
```

**Response:**

```
{
  "session_id": "950e8400-e29b-41d4-a716-446655440003",
  "score_pct": 90,
  "xp_earned": 35,
  "badges_earned": ["grammar_ace"],
  "streak_updated": true,
  "current_streak": 1,
  "message": "Grammar assessment completed"
}
```

---

## 1.4 Writing Self-Evaluation

### Get Writing Prompt (Day 91)

```
GET {WRITING_SPEAKING_API_BASE}/api/v1/writing/day/day91
Authorization: Bearer <token>
```

**Response:**

```
{
  "writing_id": "uuid",
  "day_code": "day91",
  "prompt": "Write about your experience learning a new language.",
}
```



```
"writing_type": "general",
"word_limit": 200,
"difficulty_level": "intermediate"
}
```

## Submit Writing Self-Evaluation

POST {WRITING\_SPEAKING\_API\_BASE}/writing/evaluation/self-save  
Authorization: Bearer <token>  
Content-Type: application/json

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "scores": {
    "grammar": 75,
    "vocabulary": 70,
    "coherence": 80,
    "style": 65,
    "clarity": 78,
    "engagement": 72
  },
  "user_level": "intermediate",
  "evaluation_id": "optional-uuid",
  "created_at": "2025-01-15T13:00:00Z"
}
```

## Response:

```
{
  "evaluation_id": "a50e8400-e29b-41d4-a716-446655440004",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "overall_score": 73,
  "scores": {
    "grammar": 75,
    "vocabulary": 70,
    "coherence": 80,
    "style": 65,
    "clarity": 78,
    "engagement": 72
  },
  "user_level": "intermediate",
  "created_at": "2025-01-15T13:00:00Z",
  "message": "Writing self-evaluation saved successfully"
}
```

---

## 1.5 Speaking Self-Evaluation

### Get Speaking Topic (Day 91)

```
GET {WRITING_SPEAKING_API_BASE}/api/v1/speaking/day/day91
Authorization: Bearer <token>
```

**Response:**

```
{
  "speaking_id": "uuid",
  "day_code": "day91",
  "topic": "Describe your daily routine",
  "teaching_mode_code": "conversation",
  "difficulty_level": "intermediate",
  "duration_minutes": 5
}
```

**Submit Speaking Self-Evaluation**

```
POST {WRITING_SPEAKING_API_BASE}/speaking/evaluation/self-save
Authorization: Bearer <token>
Content-Type: application/json
```

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "session_id": "b60e8400-e29b-41d4-a716-446655440005",
  "scores": {
    "fluency": 78,
    "pronunciation": 82,
    "vocabulary": 75,
    "grammar": 80,
    "focus": 85,
    "understanding": 88
  },
  "user_level": "intermediate",
  "evaluation_id": "optional-uuid",
  "created_at": "2025-01-15T14:00:00Z"
}
```

**Response:**

```
{
  "evaluation_id": "c70e8400-e29b-41d4-a716-446655440006",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "session_id": "b60e8400-e29b-41d4-a716-446655440005",
  "overall_score": 81,
  "scores": {
    "fluency": 78,
    "pronunciation": 82,
    "vocabulary": 75,

```

```
    "grammar": 80,
    "focus": 85,
    "understanding": 88
  },
  "user_level": "intermediate",
  "created_at": "2025-01-15T14:00:00Z",
  "message": "Speaking self-evaluation saved successfully"
}
```

---

## Phase 2: Assessment Results & Dashboard Setup

---

After completing all 5 assessments, retrieve the aggregate report to determine user level and display initial dashboard.

### 2.1 Get Aggregate Assessment Report

Get Overall Mastery Overview

```
GET /api/v1/users/{user_id}/mastery-overview
Authorization: Bearer <token>
```

**Response:**

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "listening": {
    "overall_mastery_pct": 75,
    "skills": {
      "main_idea": 80,
      "details": 70,
      "inference": 75,
      "vocabulary": 78
    }
  },
  "reading": {
    "overall_mastery_pct": 85,
    "skills": {
      "comprehension": 88,
      "vocabulary": 82,
      "inference": 85
    }
  },
  "grammar": {
    "overall_mastery_pct": 90,
    "skills": {
      "verb_tenses": 92,
      "subject_verb_agreement": 88,
      "articles": 90
    }
  }
}
```

```
}  
}  
}
```

## Get Writing Assessment Results

```
GET {WRITING_SPEAKING_API_BASE}/writing/competencies?user_id={user_id}&days=1  
Authorization: Bearer <token>
```

### Response:

```
{  
  "user_id": "550e8400-e29b-41d4-a716-446655440000",  
  "days_covered": 1,  
  "overall_score": 73,  
  "daily_scores": [  
    {  
      "date": "2025-01-15",  
      "day_code": "day91",  
      "scores": {  
        "grammar": 75,  
        "vocabulary": 70,  
        "coherence": 80,  
        "style": 65,  
        "clarity": 78,  
        "engagement": 72  
      },  
      "overall_score": 73  
    }  
  ]  
}
```

## Get Speaking Assessment Results

```
GET {WRITING_SPEAKING_API_BASE}/speaking/competencies?user_id=  
{user_id}&days=1  
Authorization: Bearer <token>
```

### Response:

```
{  
  "user_id": "550e8400-e29b-41d4-a716-446655440000",  
  "days_covered": 1,  
  "overall_score": 81,  
  "daily_scores": [  
    {  

```

```
    "date": "2025-01-15",
    "day_code": "day91",
    "scores": {
      "fluency": 78,
      "pronunciation": 82,
      "vocabulary": 75,
      "grammar": 80,
      "focus": 85,
      "understanding": 88
    },
    "overall_score": 81
  }
]
```

## 2.2 Assessment Summary Display

Display the following to the user:

 ASSESSMENT RESULTS

---

Listening:	75% (Proficient)	✓
Reading:	85% (Proficient)	✓
Grammar:	90% (Advanced)	✓
Writing:	73% (Developing)	✗
Speaking:	81% (Proficient)	✓

---

Overall Level: INTERMEDIATE

You're ready to start your learning journey!

# Phase 3: Daily Learning Dashboard

After initial assessment, users see the dashboard on each login. They can select days and view competencies.

## 3.1 Dashboard Summary

Get Dashboard Overview

```
GET /api/v1/dashboard/summary?window=7d
Authorization: Bearer <token>
```

Response:

```
{
  "streak_days": 5,
  "xp": {
    "total": 1250,
    "today": 85,
    "level": 3,
    "next_level_at": 1500
  },
  "weekly_activity": {
    "sessions": 15,
    "time_sec": 4500,
    "days_active": 5
  },
  "accuracy_trend": [
    {"date": "2025-01-15", "accuracy": 85},
    {"date": "2025-01-16", "accuracy": 87}
  ],
  "recent_results": [
    {
      "modality": "listening",
      "day_code": "day3",
      "score_pct": 88,
      "completed_at": "2025-01-16T10:35:00Z"
    }
  ],
  "badges": [
    {
      "badge_key": "first_week",
      "title": "First Week Complete",
      "earned_at": "2025-01-16T10:35:00Z"
    }
  ],
  "last_activity": "2025-01-16T10:35:00Z"
}
```

## Get Streak Information

```
GET /api/v1/users/{user_id}/streak
Authorization: Bearer <token>
```

## Response:

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "current_streak": 7,
  "longest_streak": 15,
  "last_active_date": "2025-01-16",
  "is_active_today": true,
  "streak_status": "active"
}
```

## 3.2 Competencies by Day

Select Day and Get Listening Competencies

```
GET /api/v1/users/{user_id}/competencies/listening/day/day3
Authorization: Bearer <token>
```

**Response:**

```
{
  "modality": "listening",
  "date_range": "day3",
  "skills": [
    {
      "skill": "main_idea",
      "sessions_practiced": 2,
      "total_questions": 6,
      "correct_answers": 5,
      "overall_mastery_pct": 83,
      "mastery_level": "proficient",
      "trend": "stable",
      "avg_time_per_question": 38.7
    },
    {
      "skill": "vocabulary",
      "sessions_practiced": 1,
      "total_questions": 4,
      "correct_answers": 4,
      "overall_mastery_pct": 100,
      "mastery_level": "advanced",
      "trend": "stable",
      "avg_time_per_question": 25.0
    }
  ]
}
```

Get Reading Competencies

```
GET /api/v1/users/{user_id}/competencies/reading/day/day5
Authorization: Bearer <token>
```

**Response:**

```
{
  "modality": "reading",
  "date_range": "day5",
  "skills": [
    {
      "skill": "comprehension",
      "sessions_practiced": 3,
      "total_questions": 15,
      "correct_answers": 12,
      "overall_mastery_pct": 80,
      "mastery_level": "proficient",
      "trend": "stable",
      "avg_time_per_question": 62.3
    }
  ]
}
```

### Get Grammar Competencies

```
GET /api/v1/users/{user_id}/competencies/grammar/day/day7
Authorization: Bearer <token>
```

### Response:

```
{
  "modality": "grammar",
  "date_range": "day7",
  "skills": [
    {
      "skill": "verb_tenses",
      "sessions_practiced": 2,
      "total_questions": 12,
      "correct_answers": 11,
      "overall_mastery_pct": 92,
      "mastery_level": "advanced",
      "trend": "stable",
      "avg_time_per_question": 31.5
    }
  ]
}
```

### Get Writing Competencies

```
GET {WRITING_SPEAKING_API_BASE}/writing/competencies?user_id=
{user_id}&days=30
Authorization: Bearer <token>
```



**Response:**

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "days_covered": 30,
  "overall_score": 78,
  "daily_scores": [
    {
      "date": "2025-01-16",
      "day_code": "day5",
      "scores": {
        "grammar": 80,
        "vocabulary": 75,
        "coherence": 82,
        "style": 70,
        "clarity": 78,
        "engagement": 75
      },
      "overall_score": 77
    }
  ],
  "skill_averages": {
    "grammar": 80,
    "vocabulary": 75,
    "coherence": 82,
    "style": 70,
    "clarity": 78,
    "engagement": 75
  }
}
```

**Get Speaking Competencies**

```
GET {WRITING_SPEAKING_API_BASE}/speaking/competencies?user_id=
{user_id}&days=30
Authorization: Bearer <token>
```

**Response:**

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "days_covered": 30,
  "overall_score": 83,
  "daily_scores": [
    {
      "date": "2025-01-16",
      "day_code": "day5",
      "scores": {
        "fluency": 82,
```

```
    "pronunciation": 85,
    "vocabulary": 80,
    "grammar": 83,
    "focus": 87,
    "understanding": 88
  },
  "overall_score": 84
}
],
"skill_averages": {
  "fluency": 82,
  "pronunciation": 85,
  "vocabulary": 80,
  "grammar": 83,
  "focus": 87,
  "understanding": 88
}
}
```

---

### 3.3 Progress Tracking (Day 1 to Current Day)

#### Get Listening Progress

```
GET /api/v1/listening/users/{user_id}/progress?from_day=day1&to_day=day10
Authorization: Bearer <token>
```

#### Response:

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "modality": "listening",
  "date_range": "day1 - day10",
  "overall_mastery_pct": 78,
  "total_sessions": 15,
  "total_audio_replay_count": 35,
  "avg_audio_replays": 2.3,
  "skills": [
    {
      "skill": "main_idea",
      "sessions_practiced": 10,
      "total_questions": 50,
      "correct_answers": 42,
      "overall_mastery_pct": 84,
      "mastery_level": "proficient",
      "trend": "improving",
      "avg_time_per_question": 40.5
    },
    {
      "skill": "details",
```

```
    "sessions_practiced": 10,  
    "total_questions": 60,  
    "correct_answers": 45,  
    "overall_mastery_pct": 75,  
    "mastery_level": "proficient",  
    "trend": "stable",  
    "avg_time_per_question": 48.2  
  }  
]  
}
```

## Get Reading Progress

```
GET /api/v1/users/{user_id}/skills/progress?  
modality=reading&from_day=day1&to_day=day10  
Authorization: Bearer <token>
```

### Response:

```
{  
  "user_id": "550e8400-e29b-41d4-a716-446655440000",  
  "modality": "reading",  
  "date_range": "day1 - day10",  
  "skills": [  
    {  
      "skill": "comprehension",  
      "sessions_practiced": 10,  
      "total_questions": 75,  
      "correct_answers": 65,  
      "overall_mastery_pct": 87,  
      "mastery_level": "proficient",  
      "trend": "improving",  
      "avg_time_per_question": 60.5  
    }  
  ]  
}
```

## Get Grammar Progress

```
GET /api/v1/users/{user_id}/skills/progress?  
modality=grammar&from_day=day1&to_day=day10  
Authorization: Bearer <token>
```

### Response:

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "modality": "grammar",
  "date_range": "day1 - day10",
  "skills": [
    {
      "skill": "verb_tenses",
      "sessions_practiced": 10,
      "total_questions": 80,
      "correct_answers": 72,
      "overall_mastery_pct": 90,
      "mastery_level": "advanced",
      "trend": "improving",
      "avg_time_per_question": 32.1
    }
  ]
}
```

## Get Writing Progress

```
GET {WRITING_SPEAKING_API_BASE}/writing/progress?user_id={user_id}&days=30
Authorization: Bearer <token>
```

## Response:

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "days": 30,
  "total_evaluations": 15,
  "overall_progress": {
    "current_avg": 78,
    "starting_avg": 70,
    "improvement": 8
  },
  "skill_progress": {
    "grammar": {
      "current": 80,
      "starting": 72,
      "trend": "improving"
    },
    "vocabulary": {
      "current": 76,
      "starting": 68,
      "trend": "improving"
    },
    "coherence": {
      "current": 82,
      "starting": 75,
      "trend": "improving"
    }
  },
}
```

```
    "style": {
      "current": 72,
      "starting": 68,
      "trend": "stable"
    },
    "clarity": {
      "current": 79,
      "starting": 72,
      "trend": "improving"
    },
    "engagement": {
      "current": 75,
      "starting": 68,
      "trend": "improving"
    }
  }
}
```

## Get Speaking Progress

```
GET {WRITING_SPEAKING_API_BASE}/speaking/progress?user_id={user_id}&days=30
Authorization: Bearer <token>
```

### Response:

```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "days": 30,
  "total_sessions": 12,
  "overall_progress": {
    "current_avg": 84,
    "starting_avg": 78,
    "improvement": 6
  },
  "skill_progress": {
    "fluency": {
      "current": 85,
      "starting": 78,
      "trend": "improving"
    },
    "pronunciation": {
      "current": 87,
      "starting": 82,
      "trend": "improving"
    },
    "vocabulary": {
      "current": 82,
      "starting": 75,
      "trend": "improving"
    },
    "grammar": {
```

```
    "current": 84,  
    "starting": 80,  
    "trend": "stable"  
  },  
  "focus": {  
    "current": 88,  
    "starting": 85,  
    "trend": "stable"  
  },  
  "understanding": {  
    "current": 89,  
    "starting": 88,  
    "trend": "stable"  
  }  
}  
}
```

---

## Phase 4: Daily Tasks (Day 1 onwards)

---

Users complete daily tasks for each modality. Each task follows a similar flow: Get content → Start session → Complete task → Submit → Get gamified results.

### 4.1 Listening Task Flow

Step 1: Get Listening Content for Day N

```
GET /api/v1/listening/day/day{n}  
Authorization: Bearer <token>
```

**Example: Day 3**

```
GET /api/v1/listening/day/day3
```

**Response:**

```
{  
  "listening_id": "uuid",  
  "day_code": "day3",  
  "title": "Workplace Communication",  
  "audio_url": "https://storage.example.com/audio/day3.mp3",  
  "audio_duration_sec": 150,  
  "difficulty_level": "intermediate",  
  "questions": [  
    {  
      "question_id": "q1",  
      "question_text": "What is the meeting about?",  
    }  
  ]  
}
```

```

        "question_type": "multiple_choice",
        "options": ["Project update", "Team building", "Budget review",
"Hiring"],
        "skill": "main_idea"
    }
]
}

```

## Step 2: Start Session & Present Audio

```

POST /api/v1/listening/sessions
Authorization: Bearer <token>

{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "day_code": "day3",
  "audio_url": "https://storage.example.com/audio/day3.mp3"
}

```

## Step 3: After Audio, Show Questions & Collect Answers

## Step 4: Submit Answers & Get Gamified Results

```

POST /api/v1/listening/sessions/{session_id}/submit
Authorization: Bearer <token>

{
  "answers": [...],
  "duration_sec": 300,
  "score_pct": 88,
  "xp_earned": 30,
  "audio_replay_count": 1
}

```

## Response (Gamified):

```

{
  "session_id": "uuid",
  "score_pct": 88,
  "xp_earned": 30,
  "skill_mastery_updated": true,
  "badges_earned": [
    {
      "badge_key": "listening_streak_3",
      "title": "3-Day Listening Streak!",
      "earned_at": "2025-01-16T10:35:00Z"
    }
  ],
}

```

```
"streak_updated": true,  
"current_streak": 3,  
"level_up": false,  
"message": "Great work! Keep it up! 🎉"  
}
```

### Display Gamified UI:

🎧 LISTENING COMPLETE!

Score: 88% ★★★★★

XP Earned: +30 XP

Streak: 🔥 3 days

🏆 NEW BADGE UNLOCKED!  
"3-Day Listening Streak"

Keep practicing to maintain your streak!

---

## 4.2 Reading Task Flow

### Step 1: Get Reading Content

```
GET /api/v1/reading/day/day5  
Authorization: Bearer <token>
```

### Response:

```
{  
  "reading_id": "uuid",  
  "day_code": "day5",  
  "title": "Technology and Society",  
  "passage": "Full reading passage text...",  
  "difficulty_level": "intermediate",  
  "questions": [...]  
}
```

### Step 2: Start Session

```
POST /api/v1/sessions  
Authorization: Bearer <token>  
  
{  
  "user_id": "550e8400-e29b-41d4-a716-446655440000",  
  "modality": "reading",  
}
```



```
"day_code": "day5"
}
```

### Step 3: Submit & Get Gamified Results

```
POST /api/v1/sessions/{session_id}/submit
Authorization: Bearer <token>

{
  "answers": [...],
  "duration_sec": 420,
  "score_pct": 92,
  "xp_earned": 35
}
```

### Response (Gamified):

```
{
  "session_id": "uuid",
  "score_pct": 92,
  "xp_earned": 35,
  "badges_earned": [
    {
      "badge_key": "reading_master",
      "title": "Reading Master",
      "earned_at": "2025-01-16T11:00:00Z"
    }
  ],
  "streak_updated": true,
  "current_streak": 5,
  "level_up": true,
  "new_level": 4,
  "message": "Outstanding! You leveled up! 🎉"
}
```

---

## 4.3 Grammar Task Flow

### Step 1: Get Grammar Content

```
GET /api/v1/grammar/day/day7
Authorization: Bearer <token>
```

### Step 2: Start Session

```
POST /api/v1/sessions
Authorization: Bearer <token>

{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "modality": "grammar",
  "day_code": "day7"
}
```

### Step 3: Submit & Get Gamified Results

```
POST /api/v1/sessions/{session_id}/submit
Authorization: Bearer <token>

{
  "answers": [...],
  "duration_sec": 180,
  "score_pct": 95,
  "xp_earned": 40
}
```

#### Response (Gamified):

```
{
  "session_id": "uuid",
  "score_pct": 95,
  "xp_earned": 40,
  "badges_earned": [],
  "streak_updated": true,
  "current_streak": 7,
  "message": "Excellent grammar skills! 📝 "
}
```

---

## 4.4 Speaking Task Flow

### Step 1: Get Speaking Topic

```
GET {WRITING_SPEAKING_API_BASE}/api/v1/speaking/day/day1
Authorization: Bearer <token>
```

#### Response:

```
{
  "speaking_id": "uuid",
```

```

    "day_code": "day1",
    "topic": "Introduce yourself and talk about your hobbies",
    "teaching_mode_code": "conversation",
    "difficulty_level": "beginner",
    "duration_minutes": 5,
    "prompts": [
      "What's your name?",
      "Where are you from?",
      "What are your hobbies?"
    ]
  }
}

```

## Step 2: User Completes Speaking Exercise

## Step 3: Evaluate Speaking Performance

```

POST {WRITING_SPEAKING_API_BASE}/speaking/evaluate
Authorization: Bearer <token>
Content-Type: application/json

{
  "session_id": "speaking-session-uuid",
  "language": "english",
  "user_level": "intermediate",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "save_evaluation": true
}

```

## Response (Gamified):


```

{
  "evaluation_id": "uuid",
  "session_id": "speaking-session-uuid",
  "total_turns": 25,
  "overall_score": 85,
  "scores": {
    "fluency": 82,
    "pronunciation": 88,
    "vocabulary": 83,
    "grammar": 84,
    "coherence": 87,
    "comprehension": 86
  },
  "strengths": [
    "Excellent conversational flow",
    "Clear pronunciation",
    "Good vocabulary range"
  ],
  "improvements": [
    "Work on complex sentence structures",





```

```
    "Expand idiomatic expressions"
  ],
  "suggestions": [
    "Practice with native speakers daily",
    "Listen to English podcasts",
    "Record and review your speaking"
  ],
  "conversation_summary": "Discussed daily routines and hobbies. Great engagement!",
  "feedback_summary": "Excellent progress! Keep practicing complex structures.",
  "fluency_level": "intermediate",
  "vocabulary_range": "good",
  "created_at": "2025-01-16T14:00:00Z",
  "xp_earned": 35,
  "badges": [
    {
      "badge_key": "speaker_pro",
      "title": "Speaker Pro"
    }
  ]
}
```


Display Gamified UI:

 **SPEAKING COMPLETE!**


---

Overall Score: 85%    


XP Earned: +35 XP

 **STRENGTHS:**


- Excellent conversational flow
- Clear pronunciation
- Good vocabulary range

 **IMPROVEMENTS:**

- Work on complex sentences
- Expand idiomatic expressions

 **SUGGESTIONS:**

- Practice with native speakers
- Listen to English podcasts
- Record and review your speaking

 **BADGE EARNED: Speaker Pro!**

4.5 Writing Task Flow

Step 1: Get Writing Prompt

```
GET {WRITING_SPEAKING_API_BASE}/api/v1/writing/day/day1
Authorization: Bearer <token>
```

**Response:**

```
{
  "writing_id": "uuid",
  "day_code": "day1",
  "prompt": "Describe your favorite place and why you love it.",
  "writing_type": "descriptive",
  "word_limit": 200,
  "difficulty_level": "intermediate"
}
```

**Step 2: User Writes Response****Step 3: Evaluate Writing**

```
POST {WRITING_SPEAKING_API_BASE}/writing/evaluate
Authorization: Bearer <token>
Content-Type: application/json

{
  "text": "User's written text here...",
  "language": "english",
  "writing_type": "descriptive",
  "user_level": "intermediate",
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "save_evaluation": true
}
```

**Response (Gamified):**

```
{
  "evaluation_id": "uuid",
  "original_text": "User's written text...",
  "scores": {
    "grammar": 88,
    "vocabulary": 82,
    "coherence": 85,
    "style": 78,
    "clarity": 90,
    "engagement": 84
  },
  "overall_score": 85,
  "strengths": [
    "Clear sentence structure",
  ]
}
```

```
    "Rich descriptive vocabulary",
    "Strong opening statement"
  ],
  "improvements": [
    "Use more transitional phrases",
    "Vary sentence length",
    "Strengthen conclusion"
  ],
  "suggestions": [
    "Practice using connectors",
    "Read examples of strong conclusions",
    "Use active voice more"
  ],
  "improved_version": "Enhanced version of the text...",
  "feedback_summary": "Great descriptive writing! Work on transitions.",
  "created_at": "2025-01-16T13:00:00Z",
  "xp_earned": 35,
  "badges": [
    {
      "badge_key": "writer_star",
      "title": "Writing Star"
    }
  ]
}
```

### Display Gamified UI:

✍️ WRITING COMPLETE!

Overall Score: 85% ★★★★★

XP Earned: +35 XP

#### 💪 STRENGTHS:

- Clear sentence structure
- Rich descriptive vocabulary
- Strong opening statement

#### 📈 IMPROVEMENTS:

- Use more transitional phrases
- Vary sentence length
- Strengthen conclusion

#### 📝 IMPROVED VERSION:

[Show enhanced text]

🏆 BADGE EARNED: Writing Star!

---

## 4.6 Get Aggregate Session Results

After completing all tasks for a day, show the aggregate summary:


Get Daily Summary

```
GET /api/v1/users/{user_id}/daily-progress
Authorization: Bearer <token>
```

Response:


```
{
  "user_id": "550e8400-e29b-41d4-a716-446655440000",
  "date": "2025-01-16",
  "xp_earned": 175,
  "sessions_completed": 5,
  "time_spent_sec": 2100,
  "perfect_day": true,
  "goals_met": ["all_modalities", "daily_xp_goal"],
  "modality_scores": {
    "listening": 88,
    "reading": 92,
    "grammar": 95,
    "writing": 85,
    "speaking": 85
  },
  "overall_average": 89
}
```

Display Aggregate UI:

 TODAY'S SUMMARY

---

Date: January 16, 2025  
Time Spent: 35 minutes

 SCORES:

Listening: 88%

★★★★★

Reading: 92%

★★★★★

Grammar: 95%

★★★★★

Writing: 85%


★★★★★


Speaking: 85%


★★★★★

---

Overall Average: 89%

 Total XP: +175 XP

 Streak: 7 days

 PERFECT DAY! All modalities completed!

---

# Error Handling

---

## Common Error Responses

### 400 Bad Request

```
{
  "detail": "Invalid day code format. Use 'dayN' (e.g., 'day1', 'day10')"
}
```

### 401 Unauthorized

```
{
  "detail": "Not authenticated. Please provide a valid token."
}
```

### 403 Forbidden

```
{
  "detail": "Cannot access another user's data"
}
```

### 404 Not Found

```
{
  "detail": "Content not found for day91"
}
```

### 500 Internal Server Error

```
{
  "detail": "An error occurred while processing your request"
}
```

## Error Handling Best Practices

```
async function fetchData(url, options) {
  try {
    const response = await fetch(url, options);
```



```
if (!response.ok) {
  const error = await response.json();

  switch (response.status) {
    case 400:
      throw new Error(`Validation Error: ${error.detail}`);
    case 401:
      // Redirect to login
      window.location.href = '/login';
      break;
    case 403:
      throw new Error('Access Denied');
    case 404:
      throw new Error('Resource Not Found');
    case 500:
      throw new Error('Server Error. Please try again later.');
```

---

```
    default:
      throw new Error(`Error: ${error.detail}`);
  }
}

return await response.json();
} catch (error) {
  console.error('API Error:', error);
  // Show user-friendly error message
  showErrorNotification(error.message);
  throw error;
}
```

---

## Complete Code Examples

---

### JavaScript Integration Example

```
class ZiyaLRGClient {
  constructor(baseUrl, writingSpeakingUrl, token) {
    this.lrgBase = baseUrl;
    this.wsBase = writingSpeakingUrl;
    this.token = token;
  }

  async request(url, options = {}) {
    const defaultOptions = {
      headers: {
        'Authorization': `Bearer ${this.token}`,
        'Content-Type': 'application/json',
        ...options.headers
      }
    };
  };
}
```

```
    const response = await fetch(url, { ...defaultOptions, ...options });

    if (!response.ok) {
      const error = await response.json();
      throw new Error(error.detail || 'Request failed');
    }

    return await response.json();
  }

  // ===== INITIAL ASSESSMENT (Day 91) =====

  async getListeningAssessment() {
    return this.request(`${this.lrgBase}/listening/day/day91`);
  }

  async startListeningSession(userId, dayCode, audioUrl) {
    return this.request(`${this.lrgBase}/listening/sessions`, {
      method: 'POST',
      body: JSON.stringify({ user_id: userId, day_code: dayCode, audio_url:
audioUrl })
    });
  }

  async submitListeningSession(sessionId, answers, duration, score, xp,
replayCount) {
    return
this.request(`${this.lrgBase}/listening/sessions/${sessionId}/submit`, {
      method: 'POST',
      body: JSON.stringify({
        answers,
        duration_sec: duration,
        score_pct: score,
        xp_earned: xp,
        audio_replay_count: replayCount
      })
    });
  }

  async getReadingAssessment() {
    return this.request(`${this.lrgBase}/reading/day/day91`);
  }

  async startReadingSession(userId, dayCode) {
    return this.request(`${this.lrgBase}/sessions`, {
      method: 'POST',
      body: JSON.stringify({ user_id: userId, modality: 'reading', day_code:
dayCode })
    });
  }

  async submitSession(sessionId, answers, duration, score, xp) {
    return this.request(`${this.lrgBase}/sessions/${sessionId}/submit`, {
      method: 'POST',
      body: JSON.stringify({
```

```
        answers,
        duration_sec: duration,
        score_pct: score,
        xp_earned: xp
    })
    });
}

async getGrammarAssessment() {
    return this.request(`${this.lrgBase}/grammar/day/day91`);
}

async startGrammarSession(userId, dayCode) {
    return this.request(`${this.lrgBase}/sessions`, {
        method: 'POST',
        body: JSON.stringify({ user_id: userId, modality: 'grammar', day_code:
dayCode })
    });
}

async saveWritingSelfEvaluation(userId, scores, userLevel = 'intermediate')
{
    return this.request(`${this.wsBase}/writing/evaluation/self-save`, {
        method: 'POST',
        body: JSON.stringify({ user_id: userId, scores, user_level: userLevel
})
    });
}

async saveSpeakingSelfEvaluation(userId, sessionId, scores, userLevel =
'intermediate') {
    return this.request(`${this.wsBase}/speaking/evaluation/self-save`, {
        method: 'POST',
        body: JSON.stringify({ user_id: userId, session_id: sessionId, scores,
user_level: userLevel })
    });
}

// ===== ASSESSMENT RESULTS =====

async getMasteryOverview(userId) {
    return this.request(`${this.lrgBase}/users/${userId}/mastery-overview`);
}

async getWritingCompetencies(userId, days = 1) {
    return this.request(`${this.wsBase}/writing/competencies?
user_id=${userId}&days=${days}`);
}

async getSpeakingCompetencies(userId, days = 1) {
    return this.request(`${this.wsBase}/speaking/competencies?
user_id=${userId}&days=${days}`);
}

// ===== DASHBOARD =====
```

```
    async getDashboardSummary(window = '7d') {
        return this.request(`${this.lrgBase}/dashboard/summary?
window=${window}`);
    }

    async getStreak(userId) {
        return this.request(`${this.lrgBase}/users/${userId}/streak`);
    }

    // ===== COMPETENCIES BY DAY =====

    async getListeningCompetencies(userId, dayCode) {
        return
this.request(`${this.lrgBase}/users/${userId}/competencies/listening/day/${da
yCode}`);
    }

    async getReadingCompetencies(userId, dayCode) {
        return
this.request(`${this.lrgBase}/users/${userId}/competencies/reading/day/${dayC
ode}`);
    }

    async getGrammarCompetencies(userId, dayCode) {
        return
this.request(`${this.lrgBase}/users/${userId}/competencies/grammar/day/${dayC
ode}`);
    }

    // ===== PROGRESS TRACKING =====

    async getListeningProgress(userId, fromDay, toDay) {
        return this.request(`${this.lrgBase}/listening/users/${userId}/progress?
from_day=${fromDay}&to_day=${toDay}`);
    }

    async getReadingProgress(userId, fromDay, toDay) {
        return this.request(`${this.lrgBase}/users/${userId}/skills/progress?
modality=reading&from_day=${fromDay}&to_day=${toDay}`);
    }

    async getGrammarProgress(userId, fromDay, toDay) {
        return this.request(`${this.lrgBase}/users/${userId}/skills/progress?
modality=grammar&from_day=${fromDay}&to_day=${toDay}`);
    }

    async getWritingProgress(userId, days = 30) {
        return this.request(`${this.wsBase}/writing/progress?
user_id=${userId}&days=${days}`);
    }

    async getSpeakingProgress(userId, days = 30) {
        return this.request(`${this.wsBase}/speaking/progress?
user_id=${userId}&days=${days}`);
    }
}
```

```
}

// ===== DAILY TASKS =====

async getListeningContent(dayCode) {
  return this.request(`${this.lrgBase}/listening/day/${dayCode}`);
}

async getReadingContent(dayCode) {
  return this.request(`${this.lrgBase}/reading/day/${dayCode}`);
}

async getGrammarContent(dayCode) {
  return this.request(`${this.lrgBase}/grammar/day/${dayCode}`);
}

async getSpeakingTopic(dayCode) {
  return this.request(`${this.wsBase}/api/v1/speaking/day/${dayCode}`);
}

async getWritingPrompt(dayCode) {
  return this.request(`${this.wsBase}/api/v1/writing/day/${dayCode}`);
}

async evaluateSpeaking(sessionId, language = 'english', userLevel =
'intermediate', userId, save = true) {
  return this.request(`${this.wsBase}/speaking/evaluate`, {
    method: 'POST',
    body: JSON.stringify({
      session_id: sessionId,
      language,
      user_level: userLevel,
      user_id: userId,
      save_evaluation: save
    })
  });
}

async evaluateWriting(text, writingType = 'general', userLevel =
'intermediate', userId, save = true) {
  return this.request(`${this.wsBase}/writing/evaluate`, {
    method: 'POST',
    body: JSON.stringify({
      text,
      language: 'english',
      writing_type: writingType,
      user_level: userLevel,
      user_id: userId,
      save_evaluation: save
    })
  });
}

async getDailyProgress(userId) {
  return this.request(`${this.lrgBase}/users/${userId}/daily-progress`);
}
```

```
}  
}  
  
// Usage Example  
const client = new ZiyaLRGClient(  
  'http://localhost:8080/api/v1',  
  'https://writing-speaking-api.example.com',  
  'your-jwt-token'  
);  
  
// Initial Assessment Flow  
async function runInitialAssessment(userId) {  
  try {  
    // 1. Listening  
    const listeningContent = await client.getListeningAssessment();  
    const listeningSession = await client.startListeningSession(userId,  
'day91', listeningContent.audio_url);  
    // ... user completes task ...  
    const listeningResult = await client.submitListeningSession(  
      listeningSession.session_id,  
      answers,  
      300,  
      85,  
      30,  
      2  
    );  
  
    // 2. Reading  
    const readingContent = await client.getReadingAssessment();  
    const readingSession = await client.startReadingSession(userId, 'day91');  
    // ... user completes task ...  
    const readingResult = await  
client.submitSession(readingSession.session_id, answers, 420, 88, 35);  
  
    // 3. Grammar  
    const grammarContent = await client.getGrammarAssessment();  
    const grammarSession = await client.startGrammarSession(userId, 'day91');  
    // ... user completes task ...  
    const grammarResult = await  
client.submitSession(grammarSession.session_id, answers, 240, 90, 38);  
  
    // 4. Writing Self-Evaluation  
    const writingScores = {  
      grammar: 75,  
      vocabulary: 70,  
      coherence: 80,  
      style: 65,  
      clarity: 78,  
      engagement: 72  
    };  
    const writingResult = await client.saveWritingSelfEvaluation(userId,  
writingScores);  
  
    // 5. Speaking Self-Evaluation  
    const speakingScores = {
```

```

        fluency: 78,
        pronunciation: 82,
        vocabulary: 75,
        grammar: 80,
        focus: 85,
        understanding: 88
    };
    const speakingResult = await client.saveSpeakingSelfEvaluation(userId,
sessionId, speakingScores);

    // 6. Get Aggregate Report
    const masteryOverview = await client.getMasteryOverview(userId);
    const writingComp = await client.getWritingCompetencies(userId, 1);
    const speakingComp = await client.getSpeakingCompetencies(userId, 1);

    // Display results
    displayAssessmentResults({
        listening: masteryOverview.listening.overall_mastery_pct,
        reading: masteryOverview.reading.overall_mastery_pct,
        grammar: masteryOverview.grammar.overall_mastery_pct,
        writing: writingComp.overall_score,
        speaking: speakingComp.overall_score
    });

} catch (error) {
    console.error('Assessment error:', error);
    showErrorMessage(error.message);
}
}

```

---

## Python Integration Example

```

import requests
from typing import Dict, List, Optional

class ZiyaLRGClient:
    def __init__(self, lrg_base_url: str, ws_base_url: str, token: str):
        self.lrg_base = lrg_base_url
        self.ws_base = ws_base_url
        self.headers = {
            'Authorization': f'Bearer {token}',
            'Content-Type': 'application/json'
        }

    def _request(self, url: str, method: str = 'GET', data: Optional[Dict] =
None) -> Dict:
        """Make HTTP request"""
        try:
            if method == 'GET':
                response = requests.get(url, headers=self.headers)
            elif method == 'POST':
                response = requests.post(url, headers=self.headers,

```

```
json=data)

        response.raise_for_status()
        return response.json()
    except requests.exceptions.HTTPError as e:
        print(f"HTTP Error: {e}")
        raise
    except Exception as e:
        print(f"Error: {e}")
        raise

# Assessment Methods
def get_listening_assessment(self) -> Dict:
    return self._request(f"{self.lrg_base}/listening/day/day91")

def start_listening_session(self, user_id: str, day_code: str, audio_url:
str) -> Dict:
    return self._request(
        f"{self.lrg_base}/listening/sessions",
        method='POST',
        data={'user_id': user_id, 'day_code': day_code, 'audio_url':
audio_url}
    )

def submit_listening_session(self, session_id: str, answers: List[Dict],
duration: int, score: int, xp: int,
replay_count: int) -> Dict:
    return self._request(
        f"{self.lrg_base}/listening/sessions/{session_id}/submit",
        method='POST',
        data={
            'answers': answers,
            'duration_sec': duration,
            'score_pct': score,
            'xp_earned': xp,
            'audio_replay_count': replay_count
        }
    )

def get_mastery_overview(self, user_id: str) -> Dict:
    return self._request(f"{self.lrg_base}/users/{user_id}/mastery-
overview")

def get_dashboard_summary(self, window: str = '7d') -> Dict:
    return self._request(f"{self.lrg_base}/dashboard/summary?window=
{window}")

def get_listening_competencies(self, user_id: str, day_code: str) ->
Dict:
    return self._request(
        f"
{self.lrg_base}/users/{user_id}/competencies/listening/day/{day_code}"
    )

def get_daily_progress(self, user_id: str) -> Dict:
```



```
        return self._request(f"{self.lrg_base}/users/{user_id}/daily-
progress")

# Usage
client = ZiyaLRGClient(
    'http://localhost:8080/api/v1',
    'https://writing-speaking-api.example.com',
    'your-jwt-token'
)

# Get dashboard
dashboard = client.get_dashboard_summary()
print(f"Streak: {dashboard['streak_days']} days")
print(f"Total XP: {dashboard['xp']['total']}")
```

Summary Table: All Endpoints

Phase	Endpoint	Method	Purpose
Assessment	/listening/day/day91	GET	Get listening assessment content
	/listening/sessions	POST	Start listening session
	/listening/sessions/{id}/submit	POST	Submit listening answers
	/reading/day/day91	GET	Get reading assessment content
	/sessions	POST	Start reading/grammar session
	/sessions/{id}/submit	POST	Submit reading/grammar answers
	/grammar/day/day91	GET	Get grammar assessment content
	/writing/evaluation/self-save	POST	Save writing self-evaluation
	/speaking/evaluation/self-save	POST	Save speaking self-evaluation

Phase	Endpoint	Method	Purpose
Results	/users/{id}/mastery-overview	GET	Get LRG mastery overview
	/writing/competencies	GET	Get writing competencies
	/speaking/competencies	GET	Get speaking competencies
Dashboard	/dashboard/summary	GET	Get dashboard overview
	/users/{id}/streak	GET	Get streak information
Competencies	/users/{id}/competencies/{modality}/day/{day}	GET	Get competencies for specific day
Progress	/listening/users/{id}/progress	GET	Get listening progress
	/users/{id}/skills/progress	GET	Get reading/grammar progress
	/writing/progress	GET	Get writing progress
	/speaking/progress	GET	Get speaking progress
Daily Tasks	/listening/day/{dayN}	GET	Get listening content
	/reading/day/{dayN}	GET	Get reading content
	/grammar/day/{dayN}	GET	Get grammar content
	/speaking/day/{dayN}	GET	Get speaking topic
	/writing/day/{dayN}	GET	Get writing prompt
	/speaking/evaluate	POST	Evaluate speaking performance
	/writing/evaluate	POST	Evaluate writing performance

Phase	Endpoint	Method	Purpose
	/users/{id}/daily-progress	GET	Get daily summary

End of Documentation

For support or questions, contact the development team.