EdGame – Interface Table v2.0

Document Version: 2.0
Status: Phase 1 Complete, Phase 2+ Marked
Last Updated: February 2026

This document specifies all system interfaces including data formats, timing requirements, and failure handling. **Phase 1 interfaces are required for MVP; Phase 2+ interfaces are planned but not implemented initially.**

---

1. Authentication APIs (Supabase-Managed)

Supabase Auth handles authentication. These endpoints are managed by Supabase — we configure, not implement.

Endpoint	Method	Phase	Description	Request	Response	·
`/auth/v1/signup`	POST	**P1**	Register new user	`{ email, password, data: { role, first_name, last_name } }`	`{ user, session }`	<
`/auth/v1/token?grant_type=password`	POST	**P1**	Login with password	`{ email, password }`	`{ access_token, refresh_token, user }`	<
`/auth/v1/token?grant_type=refresh_token`	POST	**P1**	Refresh access token	`{ refresh_token }`	`{ access_token, refresh_token }`	<
`/auth/v1/logout`	POST	**P1**	Logout user	`{ }` (requires auth header)	`{ }`	<
`/auth/v1/recover`	POST	**P1**	Request password reset	`{ email }`	`{ }`	<
`/auth/v1/otp`	POST	**P1**	Magic link login	`{ email }`	`{ }`	<
`/auth/v1/authorize?provider=google`	GET	**P1**	Google OAuth redirect	Query params	302 Redirect	<
`/auth/v1/authorize?provider=microsoft`	GET	**P2**	Microsoft OAuth redirect	Query params	302 Redirect	<

---

2. Game Environment APIs

2.1 Environment Management

Endpoint	Method	Phase	Description	Request	Re

`GET /api/environments`	GET	**P1**	List available environments	Query: `?subject=math&grade=6`	`{`
					`envi`
					`[...`

`GET /api/environments/[slug]`	GET	**P1**	Get environment details	Path param: slug	`{ id`
					`name`
					`subj`
					`conf`

`GET /api/environments/[slug]/config`	GET	**P1**	Get game configuration	Path param: slug	`{ le`
					`asse`
					`stan`
					`...`

`POST /api/environments`	POST	**P3**	Create custom environment (admin)	`{ name, subject, config }`	`{ id`

`PUT /api/environments/[slug]`	PUT	**P3**	Update environment (admin)	`{ name, config, is_active }`	`{ en`
					`}`

2.2 Game Sessions

Endpoint	Method	Phase	Description	Request	Response
`POST /api/sessions`	POST	**P1**	Start new game session	`{ environment_id, assignment_id? }`	`{ session_id, config, resume_state? }`
`PATCH /api/sessions/[id]`	PATCH	**P1**	Update session (events, completion)	`{ events: [...], completed?, score? }`	`{ status: 'ok', events_received }`

`POST /api/sessions/[id]/end`	POST	**P1**	End session, trigger metrics computation	`{ final_score, completed }`	`{ session_id, computed_metrics }`
`GET /api/sessions/[id]`	GET	**P1**	Get session details	Path param: id	`{ session, events, metrics }`
`GET /api/sessions`	GET	**P1**	List user's sessions	Query: `?limit=20&offset=0`	`{ sessions: [...], total }`

---

3. Assignment APIs

Endpoint	Method	Phase	Description	Request	Res
`GET /api/assignments`	GET	**P1**	List assignments (teacher: own, student: assigned)	Query: `?status=active`	`{ assign [...] }`
`POST /api/assignments`	POST	**P1**	Create assignment	`{ environment_id, class_id, title, due_at?, config? }`	`{ assign`
`GET /api/assignments/[id]`	GET	**P1**	Get assignment details	Path param: id	`{ assign submissi }`
`PUT /api/assignments/[id]`	PUT	**P1**	Update assignment	`{ title?, due_at?, config? }`	`{ assign`
`DELETE /api/assignments/[id]`	DELETE	**P1**	Delete assignment	Path param: id	`{ status }`
`GET /api/assignments/[id]/submissions`	GET	**P1**	Get all student submissions	Path param: id	`{ submis [...] }`
`POST /api/assignments/[id]/duplicate`	POST	**P2**	Duplicate assignment	`{ new_class_id? }`	`{ assign`

4. Analytics APIs

4.1 Teacher Analytics

Endpoint	Method	Phase	Description	Request
`GET /api/analytics/class/[id]`	GET	**P1**	Class-level analytics	Query: `?date_range=7d`
`GET /api/analytics/class/[id]/insights`	GET	**P1**	Top 3 insights per student	Path param: class_id
`GET /api/analytics/student/[id]`	GET	**P1**	Individual student analytics	Query: `?date_range=30d`
`GET /api/analytics/student/[id]/timeline`	GET	**P2**	Detailed activity timeline	Query: `?start&end`
`GET /api/analytics/environment/[slug]/class/[id]`	GET	**P2**	Class performance in specific environment	Path params

4.2 Student Analytics (Self-View)

Endpoint	Method	Phase	Description	Request	Respor
`GET /api/me/progress`	GET	**P1**	Student's own progress	Query: `?environment=math-arena`	`{ mastery, recent_ses achievemen`
`GET /api/me/achievements`	GET	**P2**	Student badges/achievements	None	`{ achievem [...] }`

4.3 Admin Analytics

Endpoint	Method	Phase	Description	Request	
`GET /api/admin/school/[id]/analytics`	GET	**P2**	School-wide analytics	Query: `?date_range=30d`	
`GET /api/admin/school/[id]/talent`	GET	**P3**	Top performers	Query: `?subject=math&count=20`	

			for competitions		
`GET /api/admin/district/[id]/analytics`	GET	**P3**	District-level analytics	Query: `?schools=[...]`	

4.4 Parent Analytics

Endpoint	Method	Phase	Description	Request	Respons
`GET /api/parent/child/[id]/summary`	GET	**P2**	Child's learning summary	Query: `?date_range=7d`	`{ summary, highlights, concerns }`
`GET /api/parent/child/[id]/behavior`	GET	**P2**	Behavioral insights	Query: `?date_range=30d`	`{ collaborati persistence engagement`

---

5. Class/Roster Management APIs

Endpoint	Method	Phase	Description	Request	Respons
`GET /api/classes`	GET	**P1**	List teacher's classes	None	`{ classe [...] }`
`POST /api/classes`	POST	**P1**	Create a class	`{ name, grade, subject }`	`{ class, join_cod }`
`GET /api/classes/[id]`	GET	**P1**	Get class details	Path param: id	`{ class, students stats }`
`PUT /api/classes/[id]`	PUT	**P1**	Update class	`{ name?, archived? }`	`{ class`
`DELETE /api/classes/[id]`	DELETE	**P1**	Archive class	Path param: id	`{ status 'archive }`
`POST /api/classes/[id]/students`	POST	**P1**	Add student to class	`{ email }` or `{ join_code }`	`{ studen }`

`DELETE /api/classes/[id]/students/[student_id]`	DELETE	**P1**	Remove student from class	Path params	`{ status 'removed }`
`POST /api/classes/join`	POST	**P1**	Student joins via code	`{ join_code }`	`{ class`

6. User Management APIs

Endpoint	Method	Phase	Description	Request	Response
`GET /api/me`	GET	**P1**	Get current user profile	None	`{ user, school?, subscription? }`
`PUT /api/me`	PUT	**P1**	Update profile	`{ first_name?, last_name?, preferences? }`	`{ user }`
`GET /api/me/subscription`	GET	**P1**	Get subscription status	None	`{ plan, expires_at, features }`
`DELETE /api/me`	DELETE	**P1**	Delete account (GDPR)	`{ confirm: true }`	`{ status: 'scheduled' }`
`POST /api/me/export`	POST	**P2**	Request data export (GDPR)	None	`{ job_id, estimated_completion }`

7. School/Organization APIs

Endpoint	Method	Phase	Description	Request	Response	Timing
`GET /api/schools/[id]`	GET	**P1**	Get school details	Path param: id	`{ school, license, stats }`	<150ms

`PUT /api/schools/[id]`	PUT	**P2**	Update school info (admin)	`{ name?, config? }`	`{ school }`	<200ms
`GET /api/schools/[id]/teachers`	GET	**P2**	List school teachers	Path param: id	`{ teachers: [...] }`	<200ms
`POST /api/schools/[id]/invite`	POST	**P2**	Invite teacher to school	`{ email, role }`	`{ invitation }`	<300ms

---

8. Integration APIs

8.1 LTI 1.3 (Google Classroom, Canvas, Schoology)

Endpoint	Method	Phase	Description	Request	Response	Timing	Fa Ha
`GET /api/lti/jwks`	GET	**P1**	LTI public key set	None	`{ keys: [...] }`	<50ms	200
`POST /api/lti/login`	POST	**P1**	LTI OIDC login initiation	LTI login params	302 Redirect	<100ms	400 para
`POST /api/lti/launch`	POST	**P1**	LTI resource launch	LTI launch params	302 Redirect to game	<300ms	400 JWT, Unau
`POST /api/lti/deep-link`	POST	**P1**	LTI deep linking (assignment creation)	LTI params	Deep link response	<200ms	400 para
`POST /api/lti/grades`	POST	**P1**	Submit grade back to LMS	`{ session_id }`	`{ status }`	<500ms	400 cont LMS

8.2 Clever Integration

Endpoint	Method	Phase	Description	Request	Response
`GET /api/integrations/clever/callback`	GET	**P2**	Clever OAuth callback	Query params	302 Redirect
`POST /api/integrations/clever/sync`	POST	**P2**	Sync rosters from Clever	`{ school_id }`	`{ synced_count, errors }`

8.3 Webhooks (Outbound)

Event	Phase	Payload	Target
`session.completed`	**P2**	`{ session_id, student_id, score, metrics }`	School webhook URL
`assignment.due`	**P2**	`{ assignment_id, incomplete_students }`	Teacher notification
`student.struggling`	**P3**	`{ student_id, concept, recommendation }`	Parent notification

---

9. WebSocket / Real-time APIs

9.1 Supabase Realtime (Phase 1)

Channel	Phase	Events	Use Case
`class:{id}:activity`	**P1**	`{ student_id, action, timestamp }`	Live teacher dashboard
`session:{id}:progress`	**P1**	`{ progress, score }`	Parent watching child play
`assignment:{id}:submissions`	**P1**	`{ student_id, completed, score }`	Teacher sees submissions

9.2 SpacetimeDB (Phase 2 - Multiplayer)

Table/Reducer	Phase	Description	Sync Rate
`PlayerState`	**P2**	Real-time player position, score, status	60 fps
`GameRoom`	**P2**	Room state, players, game phase	On change
`ChatMessage`	**P2**	In-game chat messages	Immediate
`move_player`	**P2**	Reducer: player movement	Client-triggered
`submit_answer`	**P2**	Reducer: answer submission	Client-triggered
`use_powerup`	**P2**	Reducer: power-up activation	Client-triggered

---

10. Event Schema (Game Telemetry)

10.1 Base Event Structure

```
interface GameEvent {
  id: string;                    // UUID
  session_id: string;            // Reference to session
  timestamp: number;             // Unix ms
  type: EventType;
  data: Record<string, unknown>;
}
```

```
type EventType =
  | 'session_start'
  | 'session_end'
  | 'question_presented'
  | 'question_answered'
  | 'hint_requested'
  | 'hint_viewed'
  | 'level_started'
  | 'level_completed'
  | 'achievement_earned'
  | 'error_made'
  | 'retry_attempted'
  | 'idle_detected'
  | 'focus_lost'
  | 'focus_regained';
```

10.2 Specific Event Examples

```
// Question answered
{
  id: "evt_abc123",
  session_id: "sess_xyz789",
  timestamp: 1708444800000,
  type: "question_answered",
  data: {
    question_id: "q_001",
    concept: "fractions",
    standard: "CCSS.MATH.4.NF.A.1",
    presented_at: 1708444795000,
    answered_at: 1708444800000,
    response_time_ms: 5000,
    answer: "3/4",
    correct_answer: "3/4",
    is_correct: true,
    attempt_number: 1,
    hints_used: 0
  }
}

// Level completed
{
  id: "evt_def456",
  session_id: "sess_xyz789",
  timestamp: 1708445400000,
  type: "level_completed",
  data: {
    level_id: "level_3",
    duration_seconds: 180,
    questions_total: 10,
    questions_correct: 8,
    accuracy: 0.8,
    hints_used: 2,
    retries: 1,
    score: 850
  }
}
```

11. Error Response Format

All API errors follow consistent format:

```
interface APIError {
  error: {
    code: string;           // Machine-readable error code
    message: string;        // Human-readable message
    details?: unknown;      // Additional context
    request_id: string;     // For support/debugging
  }
}
```

11.1 Standard Error Codes

HTTP Status	Code	Description
400	`validation_error`	Request body/params invalid
401	`unauthorized`	Missing or invalid auth token
403	`forbidden`	User lacks permission
404	`not_found`	Resource doesn't exist
409	`conflict`	Resource already exists (duplicate)
422	`unprocessable`	Valid syntax but semantically wrong
429	`rate_limited`	Too many requests
500	`internal_error`	Server error (retry safe)
502	`upstream_error`	Third-party service failed
503	`service_unavailable`	Temporarily unavailable

12. Rate Limits

Endpoint Category	Limit	Window	Authenticated Multiplier
Authentication	10	1 min	N/A
Read (GET)	100	1 min	2x

Write (POST/PUT/DELETE)	30	1 min	2x
Analytics	20	1 min	3x
Game events	60	1 min	N/A (batched)
Webhooks (inbound)	100	1 min	N/A

Rate limit headers returned:

- `X-RateLimit-Limit` : Max requests in window
- `X-RateLimit-Remaining` : Requests remaining
- `X-RateLimit-Reset` : Unix timestamp when window resets

Summary: Phase 1 Endpoint Count

Category	P1 Endpoints	P2+ Endpoints
Authentication	8	1
Game Environments	3	2
Game Sessions	5	0
Assignments	6	1
Analytics (Teacher)	3	2
Analytics (Student)	1	1
Analytics (Admin)	0	3
Analytics (Parent)	0	2
Classes/Roster	8	0
User Management	4	1
School Management	1	3
LTI Integration	5	0
Clever Integration	0	2
Total	**44**	**18**

Note: Phase 1 includes ~44 endpoints. This is intentionally focused — enough for a fully functional MVP without over-engineering. Phase 2+ adds ~18 endpoints for multiplayer, parent portal, admin analytics, and additional integrations.

This interface table reflects the phased architecture. Phase 1 endpoints are required for MVP launch. Phase 2+ endpoints are documented for planning but not implemented until usage patterns justify the investment.