Google Summer of Code 2025 Proposal

1. Project Title

Convert endpoints to new Pattern

2. Student Information

- **Name**: Ahmed Nasser
- **Email**: ahmed.n.abdeltwab@gmail.com
- **GitHub**: @ahmed-n-abdeltwab
- **Relevant PR**:
    - #PR 35680 (Refactored type definitions)
    - #PR 35692 (Add OpenAPI Support to stats api)

3. Project Abstract

Rocket.Chat currently faces challenges with API documentation fragmentation, inconsistent validation, and manual maintenance of specifications. This project aims to address these issues by:

1. Converting all API endpoints to support OpenAPI 3.1.
2. Implementing AJV-based request and response validation.
3. Unifying API documentation at /api-docs with interactive Swagger UI.

This enhancement will improve developer experience, ensure type safety, and create a single source of truth for API documentation.

4. Technical Details

4.1 Current Problems

- **Fragmented Documentation:** APIs are spread across /docs/json and /api-docs, leading to inconsistencies.
- **Lack of Standardization:** Some endpoints lack proper documentation.
- **Manual API Specification Maintenance:** Keeping documentation up-to-date requires manual updates.
- **Inconsistent Validation:** Lack of unified request/response validation leads to potential runtime errors.

4.2 Proposed Solution

To address these challenges, the project will introduce:

1. **Type-Safe Endpoint Conversion**
    - Adopting a **chained method pattern** for better readability
    - Expanding definition.ts to include all API types.
    - Annotating routes with OpenAPI metadata
2. **Unified Validation Layer**
    - Using **AJV schemas** to validate all API requests and responses.
    - Ensuring real-time validation to prevent incorrect data from propagating.
3. **Automated Documentation**
    - Replacing /docs/json and /api-docs with a single **OpenAPI-driven documentation portal**.
    - Integrating **Swagger UI** for interactive API exploration and testing.
4. **Backward Compatibility**
    - Ensuring **zero breaking changes** for existing API consumers.
    - Implementing a **deprecation strategy** for old endpoints.

Example of OpenAPI Integration (*Aligned with Federation API*)

```
api.v1.get('federation/matrixIds.verify')
  .validate(federationVerifyMatrixIdSchema)
  .handle(getFederationVerifyMatrixIdHandler);
```

Proposed System Architecture (*Mermaid.js diagram*)

```
graph TD;
      Client -->|Request| API_Endpoint;
      API_Endpoint -->|Validate Request| AJV_Validator;
      AJV_Validator -->|Check Schema| OpenAPI_Spec;
      OpenAPI_Spec -->|Pass| API_Logic;
      API_Logic -->|Response| Client;

      subgraph "New OpenAPI-based System"
      API_Endpoint
      AJV_Validator
      OpenAPI_Spec
      API_Logic
      end
```

.

4.3 Implementation Plan

Phase 1: Core Integration (Weeks 1-3)

- Convert 20% of high-priority endpoints.

- Implement AJV validation for these endpoints.
- Establish OpenAPI documentation infrastructure.

Phase 2: Full Conversion (Weeks 4-8)

- Migrate remaining endpoints.
- Ensure complete validation coverage with AJV.
- Document missing endpoints.

Phase 3: Finalization (Weeks 9-10)

- Improve developer documentation.
- Optimize API performance.
- Conduct final mentor reviews and refinements.

5. Deliverables

5.1 Code Deliverables

- OpenAPI 3.1 compliant API specification
- 100% endpoint coverage in /api-docs
- AJV validation for all endpoints

5.2 Documentation

- Schema-based validation reference
- Interactive API documentation with Swagger UI

5.3 Metrics for Success

- 100% API documentation coverage.
- Validation overhead below 100ms.
- Zero breaking changes.

6. Timeline

Period	Tasks
Community Bonding	Finalize endpoint priority list, discuss with mentors
Week 1-3	Implement OpenAPI spec generator, convert 20% of endpoints
Week 4-8	Complete endpoint migration, add missing documentation
Week 9-10	Testing, documentation, and performance optimizations
Buffer Week	Mentor review, final adjustments, submission

7. About Me

7.1 Qualifications

- **Rocket.Chat Contributor:**
  - Refactored Type Definitions (PR #35680)
  - Added OpenAPI Support to stats API (PR #35692)
- **Backend Expertise:**
  - Built a real-time chat application for *100+* users.
  - Developed an **AI-based malware detection system** with *97.1%* accuracy.

7.2 Availability

- Full-time commitment (*40+ hrs/week*)
- Active participation in Rocket.Chat channels

8. References

1. Rocket.Chat API Docs Issue
2. PR #35680 - Type System Refactor
3. PR #35692 - OpenAPI Stats API
4. OpenAPI Specification
5. AJV Validation Library