**PiazzaApi Documentation**

**Packages Details**

**Introduction:**

The "piazza-api" is a Node.js application designed to serve as the backend for a Piazza platform. It provides functionalities for user authentication, data validation, and interaction with a MongoDB database. This application is built using the Express.js framework, making it easy to create robust and scalable web APIs.

**Packages Used:**

1. **Cookie-parser**:

**Description**: Middleware for parsing cookies attached to the client's request.

**Purpose**: Used for handling HTTP cookies, often used for user authentication.

1. **Cors**:

**Description**: Middleware for enabling Cross-Origin Resource Sharing (CORS).

**Purpose**: Allows or restricts requested resources on a web server based on where the HTTP request was initiated.

3. **Dotenv**:

**Description**: Zero-dependency module that loads environment variables from a .env file.

**Purpose**: Simplifies environment variable management, particularly useful for configuration settings.

4. **Express**:

**Description**: Web application framework for Node.js.

**Purpose**: Used to build scalable and maintainable web applications by providing a set of features for web and mobile applications.

5. **Express-validator**:

**Description**: A set of Express.js middlewares that wraps validator.js validator and sanitizer functions.

**Purpose**: Simplifies the validation and sanitization of incoming requests.

6. **Google-Auth-library**:

**Description**: Library for handling Google's OAuth 2.0 authentication.

**Purpose**: Allows the application to integrate Google Sign-In for user authentication.

7. **JsonWebToken**:

**Description**: Library for creating and verifying JSON Web Tokens (JWT).

**Purpose**: Used for creating secure tokens for user authentication and authorization.

8. **Moment**:

**Description**: A library for parsing, validating, manipulating, and formatting dates.

**Purpose**: Simplifies working with dates and times in the application.

9. **MongoDB**:

**Description**: The official MongoDB driver for Node.js.

**Purpose**: Enables the application to interact with a MongoDB database, performing operations like CRUD (Create, Read, Update, Delete).

10. **Mongoose**:

**Description**: Elegant MongoDB object modeling for Node.js.

**Purpose**: Provides a schema-based solution to model the application's data and interact with MongoDB.

11. **Self signed**:

**Description**: Generates self-signed certificates.

**Purpose**: Useful for local development where HTTPS is required.

These packages collectively empower the "piazza-api" to handle user authentication, data validation, and database operations effectively.