Moral Decisions API Documentation

LocalHost URL: http://localhost:3001 **Public URL:** http://115.146.86.210:3001 **Public Survey URL:**
http://115.146.86.210:8080

Table of Contents

Overview

The Moral Decisions API provides endpoints for managing surveys, searching posts, and retrieving moral
decision-related content.

API Endpoints

Survey Endpoints

1. Get Survey Question

GET `/survey/question`

Retrieve a question for a specific study.

Query Parameters:

- `studyId` (required, number): The study ID

Responses:

- `200`: Return a question
- `400`: Invalid Parameters, Failed to get the question

Example Request:

```
GET /survey/question?studyId=1
```

2. Submit Survey Answer

POST `/survey/answer`

Submit answers to survey questions.

Request Body: Answer

Responses:

- `201`: Return the answer id has been created
- `400`: Invalid Parameters, Failed to get the question

Example Request:

```
POST /survey/answer
Content-Type: application/json

{
  "prolificId": "prolific-Xinlong",
  "studyId": 1,
  "answerDetail": [...],
  "decisionMaking": [1,2,3,4,5,...],
  "personalityChoice": [1,2,3,4,5,...],
  "time": 123456789
}
```

---

3. Get Answer by ID

GET `/survey/answer`

Retrieve a specific answer by its ID.

Query Parameters:

- `answerId` (required, string): The ID of the answer you want to retrieve
  - Example: `66f4cc713d9f96013d0b4516`

Responses:

- `201`: Find the answer by id
- `400`: Invalid Parameters, Failed to get the question

Example Request:

```
GET /survey/answer?answerId=66f4cc713d9f96013d0b4516
```

---

4. Create Prolific User

POST `/survey/prolific`

Create a new Prolific user record.

Request Body: Prolific

Responses:

- `201`: Return created prolific
- `400`: Invalid Parameters, Failed to create the prolific

Example Request:

```
POST /survey/prolific
Content-Type: application/json

{
  "prolificId": "prolific-Xinlong",
  "age": 20,
  "country": "China",
  "frequentUser": true,
  "language": "English",
  "takenBefore": [true, false, false, false, false],
  "visitSubreddit": true
}
```

---

5. Get Prolific User by ID

GET `/survey/prolific`

Retrieve a Prolific user record by ID.

Query Parameters:

- `prolificId` (required, string): The ID of the prolific you want to retrieve
  - Example: `prolific-Xinlong`

Responses:

- `201`: Return corresponding prolific

Example Request:

```
GET /survey/prolific?prolificId=prolific-Xinlong
```

Search Endpoints

6. Search Posts

GET `/search`

Search for posts by topic and keywords with pagination.

Query Parameters:

- `topic` (optional, string): The topic to search for
    - Example: `Death`
- `keywords` (optional, string): Keywords to search for
    - Example: `Death Life Afterlife`
- `pageSize` (optional, number): The limit of the search (default: 10)
- `page` (optional, number): The page number (default: 0)

Responses:

- `200`: Return a list of search results (array of PostSummary)
- `400`: Invalid Parameters, Failed to get the question
- `503`: Server is rebuilding searching cache, please waiting

Example Request:

```
GET /search?
topic=Death&keywords=Death%20Life%20Afterlife&pageSize=10&page=0
```

Example Response:

```json
[
  {
    "id": "123",
    "title": "Post Title",
    "selftext": "Post content...",
    "verdict": "YTA",
    "YTA": 150,
    "NTA": 50,
    "commentCount": 200,
    "topics": ["道德", "生活"]
  }
]
```

Post Endpoints

7. Get Topic List

GET `/post/topics`

Retrieve a list of all available topics.

Responses:

- **200**: Success

Example Request:

```
GET /post/topics
```

---

8. Get Hot Posts

GET /post/hotPosts

Retrieve hot/popular posts with pagination.

Query Parameters:

- `pageSize` (optional, number): The number of posts per page (default: 10)
- `page` (optional, number): The page number (default: 0)

Responses:

- **200**: Success

Example Request:

```
GET /post/hotPosts?pageSize=10&page=0
```

---

9. Get Post by ID

GET /post/{id}

Retrieve a specific post by its ID.

Path Parameters:

- `id` (required, string): The post ID

Responses:

- **200**: 成功获取帖子详情 (Successfully retrieved post details)
- **400**: 无效的帖子ID (Invalid post ID)
- **404**: 帖子不存在 (Post does not exist)
- **500**: 服务器内部错误 (Server internal error)

Example Request:

```
GET /post/123
```

Example Response (200):

```
{
  "id": "123",
  "title": "帖子标题",
  "selftext": "帖子内容",
  "verdict": "YTA",
  "YTA": 150,
  "NTA": 50,
  "commentCount": 200,
  "topics": ["道德", "生活"]
}
```

Example Error Response (404):

```
{
  "status": 404,
  "error": "帖子不存在",
  "message": "未找到ID为 123 的帖子"
}
```

---

10. Test Connection

GET `/post/test`

Test the database/API connection.

Responses:

- `200`: Success

Example Request:

```
GET /post/test
```

Cache Endpoints

11. Clear Cache

GET `/post/clearCache`

Clear the server cache.

Responses:

- 200: Success

Example Request:

```
GET /post/clearCache
```

Data Models

Answer Model

```
{
  "prolificId": string,        // The prolific id of the user (required)
  "studyId": number,           // The study id (required)
  "answerDetail": AnswerItem[], // Array of answer items (required)
  "comment": string,           // The comments of the user
  "decisionMaking": string[],  // List of answers for decision making
(25 Items, required)
  "personalityChoice": string[], // List of answers for personality (15
Items, required)
  "time": number               // The time stamp taken to complete the
survey (required)
}
```

Example:

```
{
  "prolificId": "prolific-Xinlong",
  "studyId": 1,
  "answerDetail": [...],
  "comment": "This is a comment",
  "decisionMaking": [1,2,3,4,5,1,2,3,4,5,1,2,3,4,5,1,2,3,4,5,1,2,3,4,5],
  "personalityChoice": [1,2,3,4,5,1,2,3,4,5,1,2,3,4,5],
  "time": 123456789
}
```

AnswerItem Model

```
{
  "questionId": string,          // The question ID (required)
  "individualAnswer": _AnswerItem, // The individual answer (required)
```

```
  "groupAnswer": _AnswerItem,        // The group answer (required)
  "comment": string                  // The user comments
}
```

Example:

```
{
  "questionId": "atcfwx",
  "individualAnswer": {
    "isAsshole": true,
    "rating": 4
  },
  "groupAnswer": {
    "isAsshole": false,
    "rating": 2
  },
  "comment": "This is a comment"
}
```

_AnswerItem Model

```
{
  "isAsshole": boolean,  // Whether user thinks it is an asshole or not
(required)
  "rating": number       // Rate the asshole level from 1 to 5 (required,
enum: 1,2,3,4,5)
}
```

Example:

```
{
  "isAsshole": true,
  "rating": 4
}
```

Prolific Model

```
{
  "prolificId": string,        // The prolific id (required)
  "age": number,               // The age of the user
  "country": string,           // The country of the user
  "frequentUser": boolean,     // Whether user is a frequent user
  "language": string,          // The language of the user
```

```
  "takenBefore": boolean[],       // Whether user has taken the survey
before
  "visitSubreddit": boolean       // Whether user has visited the subreddit
before
}
```

Example:

```
{
  "prolificId": "prolific-Xinlong",
  "age": 20,
  "country": "China",
  "frequentUser": true,
  "language": "English",
  "takenBefore": [true, false, false, false, false],
  "visitSubreddit": true
}
```

PostSummary Model

```
{
  "id": string,
  "title": string,
  "selftext": string,
  "verdict": string,
  "YTA": number,
  "NTA": number,
  "commentCount": number,
  "topics": string[]
}
```

Example:

```
{
  "id": "123",
  "title": "帖子标题",
  "selftext": "帖子内容",
  "verdict": "YTA",
  "YTA": 150,
  "NTA": 50,
  "commentCount": 200,
  "topics": ["道德", "生活"]
}
```

Notes

- **Base URL**: All endpoints are relative to `http://localhost:3001`
- **Authentication**: Not specified in the current API specification
- **Content-Type**: `application/json` for all POST requests
- **CORS**: Enabled on the server

Error Responses

Common error response format:

```
{
  "status": number,
  "error": string,
  "message": string
}
```

Common Status Codes:

- `200`: Success
- `201`: Created
- `400`: Bad Request
- `404`: Not Found
- `500`: Internal Server Error
- `503`: Service Unavailable