**MICOMI FETCHING API**

Add this in the .env: REFRESH\_SECRET="way\_clue\_kay\_secret"

<http://localhost:3000/map> - fetch all map

<http://localhost:3000/map/select-map/:map_id> – select map by id

<http://localhost:3000/map/select-map/:map_id/select-level/:level_id> – select level in the map

User APIs

<http://localhost:3000/player/login> - user login (email and password)

<http://localhost:3000/admin/login> - admin log-in (email and password)

<http://localhost:3000/player/register> - user create account (username, email, and password)

Game Logic

<http://localhost:3000/game/select-map/:playerId/:mapId> – map selection

<http://localhost:3000/game/select-character/:playerId/:characterId> – character selection

<http://localhost:3000/game/entryLevel/:playerId/:levelId> – level selection

Answering the challenge

<http://localhost:3000/game/submit-challenge/:playerId/:levelId/:challengeId>

Body:

{

"answer": "" – player's answer

}

Delete the answer to the challenge – For testing purposes only (Due to lack of data)

<http://localhost:3000/game/delete-challenge-answer/:playerId/:levelId/:challengeId>

Shop

Character Creation:

<http://localhost:3000/shop/create-character>

Body:

{

"character\_id": 3,

"character\_price": 0

}

Buy Character:

<http://localhost:3000/game/buy-character/:playerId>

Body:

{

"characterShopId": 1

}

GET all characters in the shop

<http://localhost:3000/shop/shop-characters/> -getting all characters in the shop

Sample:

<http://localhost:3000/game/entryLevel/11/1>

<http://localhost:3000/game/submit-challenge/11/1/10>

Sample response when entering a level:  
{

"success": true,

"message": "Entered level",

"data": {

"level": {

"level\_id": 1,

"level\_number": 1,

"level\_difficulty": "easy"

},

"enemy": {

"enemy\_id": 3,

"enemy\_health": 50

},

"currentChallenge": {

"challenge\_id": 11,

"level\_id": 1,

"challenge\_type": "fill in the blank",

"title": "Complete the HTML Paragraph",

"description": "Fill in the blanks to complete the HTML paragraph structure:\n\n<p> \_\_\_\_ </p>",

"correct\_answer": [

"This is a paragraph."

],

"hint": "Remember: Paragraphs in HTML start with <p> and end with </p>.",

"points\_reward": 5,

"coins\_reward": 10,

"guide": null,

"test\_cases": [

{

"input": "<p>This is a paragraph.</p>",

"expected\_output": true

}

],

"timeLimit": 10,

"timeRemaining": 10,

"timer": "00:10"

}

}

}

Sample response when submitting an answer:  
  
{

"success": true,

"message": "Challenge successfully submitted.",

"data": {

"isCorrect": true,

"attempts": 1,

"fightResult": {

"status": "in\_progress",

"charHealth": 200,

"enemyHealth": 100,

"message": "The enemy attacked! New question awaits.",

"timer": "00:10"

},

"message": "Time's up! Enemy attacked you.",

"nextChallenge": {

"challenge\_id": 11,

"level\_id": 1,

"challenge\_type": "fill in the blank",

"title": "Complete the HTML Paragraph",

"description": "Fill in the blanks to complete the HTML paragraph structure:\n\n<p> \_\_\_\_ </p>",

"correct\_answer": [

"This is a paragraph."

],

"hint": "Remember: Paragraphs in HTML start with <p> and end with </p>.",

"points\_reward": 5,

"coins\_reward": 10,

"guide": null,

"test\_cases": [

{

"input": "<p>This is a paragraph.</p>",

"expected\_output": true

}

],

"timeLimit": 10,

"timeRemaining": 0,

"timer": "00:00"

},

"levelStatus": {

"isCompleted": false,

"battleWon": false,

"battleLost": false,

"canProceed": false,

"showFeedback": false,

"playerHealth": 200,

"enemyHealth": null

}

}

}