

Documentation for Using Talking Head API Endpoints

1 Overview

The Talking Head API provides several endpoints to manage and interact with an animated talking head. This includes loading characters, starting and stopping mouth animations, setting emotions, and streaming video frames. Additionally, there is a classification endpoint to analyze text and update the talking head's emotion accordingly.

2 API Endpoints

2.1 1. Load Talking Head Character

Endpoint: /api/talkinghead/load

Method: POST

Description: Loads the talking head sprite posted in the request. Resumes animation if the talking head module was paused.

Request Body: Form data containing the image file to load.

Example:

```
curl -X POST -F "file=@character.png" http://localhost:5100/api/talkinghead/load
```

Usage:

- **Localhost:** http://localhost:5100/api/talkinghead/load
- **Azure/AWS:** Replace localhost with your server's address and include the API key in headers if required.

2.2 2. Load Custom Emotion Templates

Endpoint: /api/talkinghead/load_emotion_templates

Method: POST

Description: Loads custom emotion templates for the talking head or resets to defaults.

Request Body: JSON object with emotion templates. Send a blank JSON to reset to defaults.

Example:

```
curl -X POST -H "Content-Type: application/json" -d '{}' http://localhost:5100/api/talkinghead/load_emotion_templates
```

Usage:

- **Localhost:** `http://localhost:5100/api/talkinghead/load_emotion_templates`
- **Azure/AWS:** Replace `localhost` with your server's address and include the API key in headers if required.

2.3 3. Load Custom Animator Settings

Endpoint: `/api/talkinghead/load_animator_settings`

Method: POST

Description: Loads custom settings for the talking head animator and postprocessor or resets to defaults.

Request Body: JSON object with animator settings. Send a blank JSON to reset to defaults.

Example:

```
curl -X POST -H "Content-Type: application/json" -d '{}' http://localhost:5100/api/talkinghead/load_animator_settings
```

Usage:

- **Localhost:** `http://localhost:5100/api/talkinghead/load_animator_settings`
- **Azure/AWS:** Replace `localhost` with your server's address and include the API key in headers if required.

2.4 4. Unload Talking Head

Endpoint: `/api/talkinghead/unload`

Method: GET

Description: Pauses the talking head module. To resume, load a character via `/api/talkinghead/load_character`

Example:

```
curl -X GET http://localhost:5100/api/talkinghead/unload
```

Usage:

- **Localhost:** `http://localhost:5100/api/talkinghead/unload`
- **Azure/AWS:** Replace `localhost` with your server's address and include the API key in headers if required.

2.5 5. Start Talking Animation

Endpoint: /api/talkinghead/start_talking

Method: GET

Description: Starts the mouth animation for talking.

Example:

```
curl -X GET http://localhost:5100/api/talkinghead/start_talking
```

Usage:

- **Localhost:** http://localhost:5100/api/talkinghead/start_talking
- **Azure/AWS:** Replace localhost with your server's address and include the API key in headers if required.

2.6 6. Stop Talking Animation

Endpoint: /api/talkinghead/stop_talking

Method: GET

Description: Stops the mouth animation for talking.

Example:

```
curl -X GET http://localhost:5100/api/talkinghead/stop_talking
```

Usage:

- **Localhost:** http://localhost:5100/api/talkinghead/stop_talking
- **Azure/AWS:** Replace localhost with your server's address and include the API key in headers if required.

2.7 7. Set Emotion

Endpoint: /api/talkinghead/set_emotion

Method: POST

Description: Sets the talking head character emotion.

Request Body: JSON object with the emotion name.

```
{
  "emotion_name": "joy"
}
```

Example:

```
curl -X POST -H "Content-Type: application/json" -d '{"emotion_name": "joy"}' http://localhost:5100/api/talkinghead/set_emotion
```

Usage:

- **Localhost:** `http://localhost:5100/api/talkinghead/set_emotion`
- **Azure/AWS:** Replace `localhost` with your server's address and include the API key in headers if required.

2.8 8. Stream Video Frames

Endpoint: `/api/talkinghead/result_feed`

Method: GET

Description: Streams live character output as a series of PNG encoded images.

Example:

```
curl -X GET http://localhost:5100/api/talkinghead/result_feed
```

Usage:

- **Localhost:** `http://localhost:5100/api/talkinghead/result_feed`
- **Azure/AWS:** Replace `localhost` with your server's address and include the API key in headers if required.

2.9 9. Classify Text

Endpoint: `/api/classify`

Method: POST

Description: Performs sentiment analysis on the text and returns the classification result. If the talking head module is enabled, it automatically updates its emotion based on the classification result.

Request Body: JSON object with the text to classify.

```
{
  "text": "I am very happy today!"
}
```

Example:

```
curl -X POST -H "Content-Type: application/json" -d '{"text": "I am very happy today!"}' http://localhost:5100/api/classify
```

Usage:

- **Localhost:** `http://localhost:5100/api/classify`
- **Azure/AWS:** Replace `localhost` with your server's address and include the API key in headers if required.

2.10 10. Get Classification Labels

Endpoint: /api/classify/labels

Method: GET

Description: Returns the available classifier labels for text sentiment (character emotion).

Example:

```
curl -X GET http://localhost:5100/api/classify/labels
```

Usage:

- **Localhost:** http://localhost:5100/api/classify/labels
- **Azure/AWS:** Replace localhost with your server's address and include the API key in headers if required.

3 Example Scenario

3.1 Scenario 1: Streaming Live Video with Expression Changes

1. Load Character:

```
curl -X POST -F "file=@character.png" http://localhost:5100/api/talkinghead/load
```

2. Start Talking:

```
curl -X GET http://localhost:5100/api/talkinghead/start_talking
```

3. Change Expression:

```
curl -X POST -H "Content-Type: application/json" -d '{"emotion_name": "joy"}' http://localhost:5100/api/talkinghead/set_emotion
```

4. Stream Video Frames:

```
curl -X GET http://localhost:5100/api/talkinghead/result_feed
```

5. **Toggle Talking and Change Expression:** Set up a script or use the provided JavaScript to toggle talking and change expressions every 5 seconds.

3.2 Scenario 2: Classify Text and Update Emotion

1. Classify Text:

```
curl -X POST -H "Content-Type: application/json" -d '{"text": "I am very happy today!"}' http://localhost:5100/api/classify
```

2. **Set Emotion Based on Classification:** The emotion will be automatically set if the talking head module is enabled.

4 Considerations for Azure or AWS Deployment

- **API Endpoint URL:** Replace localhost with your server's address.
- **API Keys:** Include API keys in headers if required.

```
curl -X POST -H "Content-Type: application/json" -H "Authorization: Bearer YOUR_API_KEY" -d '{"text": "I am very happy today!"}' http://your-server-address/api/classify
```

5 Conclusion

This documentation covers all the essential endpoints related to the Talking Head functionality, providing detailed usage instructions and scenarios.

6 Example HTML and JavaScript

You can look at the following HTML and JavaScript codes for better understanding.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Talking Head</title>
</head>
<body>
  <input type="file" id="fileInput">
  <div id="expression-holder" style="display: none;">
    <img id="expression-image" class="expression" src="" alt="Character Expression">
  </div>
  <script>
    document.addEventListener('DOMContentLoaded', () => {
      const serverAddress = 'http://localhost:5100';
      const imgElement = document.getElementById('expression-image');
      ;
      const expressionHolder = document.getElementById('expression-holder');
      let talking = false;
      const expressions = ["anger", "joy", "grief", "fear", "surprise"];
      let currentExpressionIndex = 0;

      function loadCharacter() {
        const formData = new FormData();
```

```

const imageFile = document.getElementById('fileInput').
  files[0];
formData.append('file', imageFile);

fetch(`${serverAddress}/api/talkinghead/load`, {
  method: 'POST',
  body: formData,
})
.then(response => {
  if (!response.ok) {
    throw new Error('Failed to load character');
  }
  displayTalkingHead();
  toggleTalkingAndExpression();
})
.catch(error => console.error('Error loading character:',
  error));
}

function startTalking() {
  fetch(`${serverAddress}/api/talkinghead/start_talking`, {
    method: 'GET' })
    .then(response => {
      if (!response.ok) {
        throw new Error('Failed to start talking');
      }
      talking = true;
    })
    .catch(error => console.error('Error starting talking
      :', error));
}

function stopTalking() {
  fetch(`${serverAddress}/api/talkinghead/stop_talking`, {
    method: 'GET' })
    .then(response => {
      if (!response.ok) {
        throw new Error('Failed to stop talking');
      }
      talking = false;
    })
    .catch(error => console.error('Error stopping talking
      :', error));
}

function toggleTalkingAndExpression() {
  setInterval(() => {
    if (talking) {

```

```

        stopTalking();
    } else {
        startTalking();
    }
    changeExpression();
}, 5000);
}

function changeExpression() {
    const currentExpression = expressions[
        currentExpressionIndex];
    currentExpressionIndex = (currentExpressionIndex + 1) %
        expressions.length;

    fetch(`${serverAddress}/api/talkinghead/set_emotion`, {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({ emotion_name: currentExpression
            }),
    })
    .then(response => {
        if (!response.ok) {
            throw new Error('Failed to set emotion');
        }
    })
    .catch(error => console.error('Error setting emotion:',
        error));
}

function displayTalkingHead() {
    const talkingheadResultFeedSrc = `${serverAddress}/api/
        talkinghead/result_feed`;

    expressionHolder.style.display = 'block';
    if (imgElement.src !== talkingheadResultFeedSrc) {
        imgElement.src = talkingheadResultFeedSrc;
    }
}

document.getElementById('fileInput').addEventListener('change
    ', loadCharacter);
});
</script>
</body>
</html>

```