

Transcription & Teaser API Documentation

Transcription Endpoint

- **URL:** `/api/public/transcribe`
- **Method:** POST
- **Content-Type:** `multipart/form-data`
- **Response:** Transcribed text in Dutch

```
async function transcribeAudio(audioFile) {
  const formData = new FormData();
  formData.append("file", audioFile);

  const response = await fetch("/api/public/transcribe", {
    method: "POST",
    body: formData,
  });

  const { text } = await response.json();
  return text;
}
```

Teaser Endpoint

- **URL:** `/api/public/teaser`
- **Method:** POST
- **Content-Type:** `application/json`
- **Body:**

```
interface TeaserRequest {
  text: string;
  image?: string; // base64 encoded image
}
```

- **Response:** AI-generated teaser text

```
async function generateTeaser(text, imageBase64) {
  const response = await fetch("/api/public/teaser", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({
      text,
      image
    }),
  });
}
```

```
    const { teaser } = await response.json();
    return teaser;
}

// Example with image
async function transcribeAndTeaser(audioFile, imageFile) {
  // Convert image to base64
  const imageBase64 = await new Promise((resolve) => {
    const reader = new FileReader();
    reader.onloadend = () => resolve(reader.result);
    reader.readAsDataURL(imageFile);
  });

  const transcription = await transcribeAudio(audioFile);
  const teaser = await generateTeaser(transcription, imageBase64);
  return { transcription, teaser };
}
```