Lecture 40: Audio Speech Model TTS (Text to Speech)

Introduction:

- After transcription (Speech → Text), the next is **Text to Speech (TTS)**.
- TTS converts written **text** into **spoken audio**.
- In Spring AI, this is implemented using the **OpenAiAudioSpeechModel**.

Controller Setup:

- A new endpoint is created for TTS.
    - **Mapping**: **POST /api/tts**
    - **Accepts**: Text as a request parameter.
    - **Returns**: Byte array (speech output).

Code Implementation:

```java
@RestController
public class AudioGenController {

    private OpenAiAudioTranscriptionModel audioModel;
    private OpenAiAudioSpeechModel audioSpeechModel;

    public AudioGenController(OpenAiAudioTranscriptionModel audioModel, OpenAiAudioSpeechModel audioSpeechModel) {
        this.audioModel = audioModel;
        this.audioSpeechModel = audioSpeechModel;
    }

    @PostMapping("api/tts")
    public byte[] tts(@RequestParam String text) {
        return audioSpeechModel.call(text);
    }
}
```

Testing the API:

- Send a **POST request** to http://localhost:8080/api/tts.
- Add text as a request parameter (text=Hello World).
- The response will be a **raw byte array**.
- Save/export as .mp3 to play the audio.

Key Points:

- **OpenAiAudioSpeechModel** is used for TTS.
- Output is returned as a **byte array**.
- REST clients like Insomnia return raw data, which can be saved as an audio file.

Summary:

- TTS converts text into speech.
- Implemented via audioSpeechModel.call(text).
- Returns byte array → save/play as, for example, .mp4.