FACERE

Consult Integration Technical Documentation

Request Formats

All requests are `https` requests unless specified otherwise. All request bodies and responses are `application/json` unless specified otherwise.

Authentication

All endpoints are secured using Bearer Authorization. We will provide you with the token. For example if the token is `ABC1234` then your header should look like:

```
Unset
Authorization: Bearer ABC1234
```

FACERE

Recording a Consult

For each unique consult session a new `uuid` needs to be generated. This same uuid should be used to retrieve the summary and letter for the session

We recommend that the audio be sampled at 44.1khz, 16 bits depth and encoded as LPCM.

At regular intervals samples of the audio stream should be sent to the endpoint, we recommend an interval of 2 seconds.

We will assume all bytes sent to us are in Big endian. (Note: If Little endian requires less effort, we can change the way we process the bytes on our end)

Request

```
Unset
POST /api/consult/{uuid}

Request Body:
{
   audioData: String
   sampleRate: Long // Long is the equivalent of a signed int64
   sampleDepth: Long
   sampleLength: Long
   psuedoTimestamp: Long
}
```

audioData base64 encoded raw audio bytes
sampleRate is the original sample rate of the audio i.e. 44100
sampleDepth is the original depth of the audio i.e. 16
sampleLength represents the length of the audio bytes. i.e. if we audio take every 2 seconds from the audio stream, then sampleLength should have the value 2000
pseudoTimestamp represents the start time of audio data relative to the first sample in milliseconds

For example if we use an interval of 2 seconds:

```
Unset
First request body
{
```

```
   audioData: 67ef......,
   sampleRate: 44100,
   sampleDepth: 16,
   sampleLength: 2000,
   pseudoTimestamp: 0
}

Second Request body
{
   audioData: 8c6d......,
   sampleRate: 44100,
   sampleDepth: 16,
   sampleLength: 2000,
   pseudoTimestamp: 2000
}

Third Request body
{
   audioData: 91u7......,
   sampleRate: 44100,
   sampleDepth: 16,
   sampleLength: 2000,
   pseudoTimestamp: 4000,
}
```

Response

If successful

```
Unset
200 OK
```

else;

```
Unset
{
   error: String // Represents the error code e.g. BAD_REQUEST
   errorMessage: String // Human readable error message
}
```

Requesting a Summary

Use the same `uuid` you generated earlier for the consult recording to make the request.

Request

```
Unset
POST /api/consult/{uuid}/summary
```

Response

If successful

```
Unset
200 OK
{
  summary: String
}
```

else;

```
Unset
{
   error: String // Represents the error code e.g. BAD_REQUEST
   errorMessage: String // Human readable error message
}
```

Requesting a Letter

Use the same `uuid` you generated earlier for the consult recording to make the request.

Request

```
Unset
POST /api/consult/{uuid}/letter
```

Response

If successful

```
Unset
{
  letter: String
}
```

else;

```
Unset
{
   error: String // Represents the error code e.g. BAD_REQUEST
   errorMessage: String // Human readable error message
}
```

FACERE

Tech stack

Application
- Springboot (6)
    - Spring web mvc
    - Spring security
    - Spring data jpa
- Kotlin
- JPA/Hibernate
    - Springboot integration
- Junit

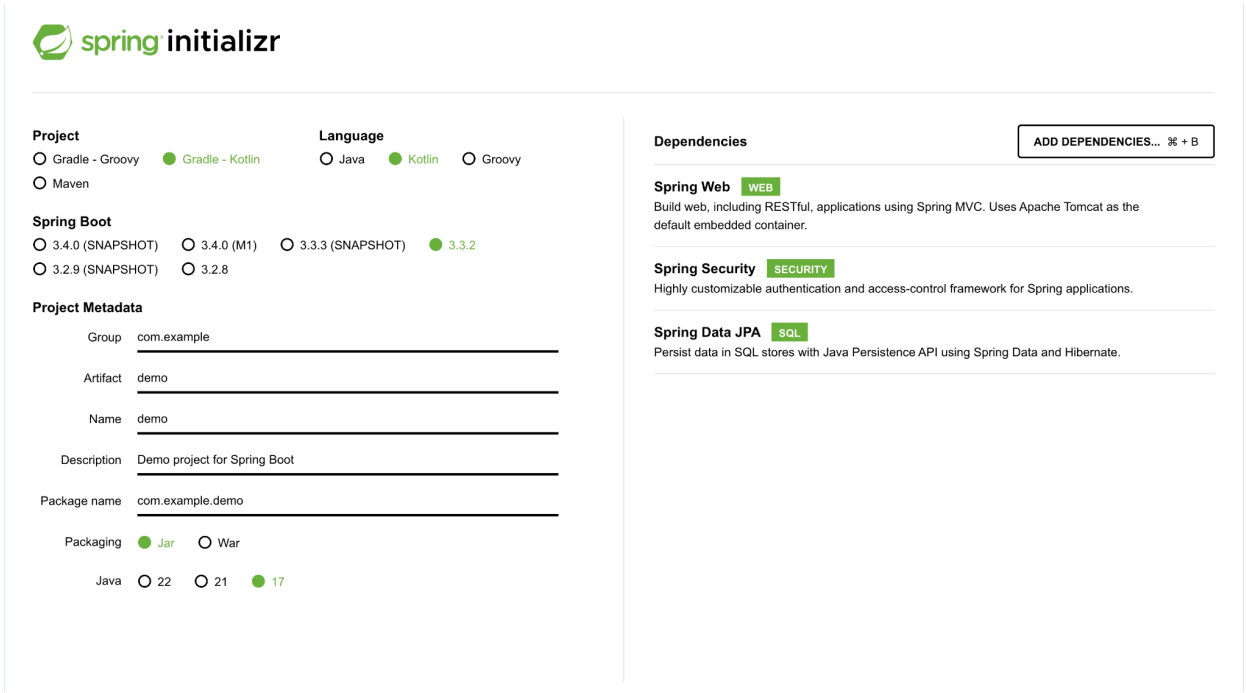
Database
- Postgresql
- Liquibase
    - Springboot integration

Docker
- containerisation

K3S
- Infrastructure

AWS
- EC2

Misc
- FFMPEG



https://start.spring.io/