

```
import Foundation
import Foundation
import SwiftUI
```

```
class NetworkManager {
```

```
    func fetchTTSDData(text: String, completion: @escaping (Data?) -> Void) {
        let url = URL(string: "https://api.elevenlabs.io/v1/text-to-speech/onwK4e9ZLuTAKqWW03F9/stream")!

        var request = URLRequest(url: url)
        request.httpMethod = "POST"
        request.setValue("audio/mpeg", forHTTPHeaderField: "Accept")
        request.setValue("application/json", forHTTPHeaderField: "Content-Type")
        request.setValue("c3d42b5eb3515a478fd1d31fe4ac6f70", forHTTPHeaderField: "xi-api-key")
        // Atualizado para a nova chave de API

        let requestBody: [String: Any] = [
            "text": text,
            "model_id": "eleven_multilingual_v2",
            "voice_settings": [
                "stability": 0.5,
                "similarity_boost": 0.5,
                "style": 0.0,
                "use_speaker_boost": true
            ]
        ]

        do {
            request.httpBody = try JSONSerialization.data(withJSONObject: requestBody)
        } catch {
            print("Erro ao serializar JSON: \(error.localizedDescription)")
            completion(nil)
            return
        }

        URLSession.shared.dataTask(with: request) { data, response, error in
            if let error = error {
                print("Erro na URLSession: \(error.localizedDescription)")
                completion(nil)
                return
            }

            // Verificação do status HTTP:
            if let httpResponse = response as? HTTPURLResponse, !
(200...299).contains(httpResponse.statusCode) {
                if let data = data, let errorMsg = String(data: data, encoding: .utf8) {
                    print("Detalhes do erro: \(errorMsg)")
                }
                print("Erro no código de status HTTP: \(httpResponse.statusCode)")
                completion(nil)
                return
            }

            guard let data = data else {
                print("Dados não recebidos.")
                completion(nil)
            }
        }
    }
}
```

return

}

completion(data)

}.resume()

}

}

--	--