

@startuml

title ユースケース シーケンス図：社内ドキュメント RAG+Chat 回答、Word/PDF 出力

actor ユーザー as User

participant ブラウザ as Browser

participant "DocumentAsk ページ\n(ASP.NET WebForms)" as Server

participant "セッション\n(ASP.NET)" as Sess

database "SQL Server\n(DocumentEmbeddings)" as DB

participant "OpenAI API\nhttps://api.openai.com/v1" as OpenAI

note right of Server

Page_Init:

- HttpClient に Bearer {OpenAI_API_Key} を設定
- ConnectionLimit/Expect100Continue 設定
- AppSettings から ChatModel/EmbeddingModel/Temperature を読込

end note

User -> Browser: 質問を入力し「質問する」クリック

Browser -> Server: btnAsk_Click(PostBack)

activate Server

Server -> Server: 入力検証

alt 質問が空

Server -> Browser: 「質問内容を入力してください。」表示

else 質問あり

Server -> Server: useRAG = chkUseRAG.Checked

alt RAG を使用

group RAG コンテキスト構築 BuildRagContextAsync

note right

質問文から拡張子(.cs/.sql 等)を抽出して

Title LIKE フィルタを行う場合あり (ExtractTitleLike)

end note

Server -> OpenAI: POST /embeddings\n{ model=EmbeddingModel, input=質問 }

activate OpenAI

OpenAI --> Server: 質問の埋め込みベクトル

deactivate OpenAI

Server -> DB: SELECT Title, ChunkIndex, Chunk, Embedding\nFROM
DocumentEmbeddings\n[必要に応じて Title LIKE]

activate DB

DB --> Server: ベクトル付きチャンク一覧

deactivate DB

Server -> Server: コサイン類似度を計算→上位 topN 抽出\n同一 Title で±neighbor を連

結

Server -> Server: finalPrompt = 指示文 + コンテキスト + 質問

end group

else RAG を使用しない

Server -> Server: finalPrompt = 質問

end

group Chat Completions 呼び出し (PostWithRetryAsync)

loop 最大4回までリトライ

Server -> OpenAI: POST /chat/completions\n{ model=ChatModel, messages=[user: finalPrompt], temperature? }

activate OpenAI

OpenAI --> Server: 応答 または 429/5xx/タイムアウト

deactivate OpenAI

alt 成功

break

else 429/5xx/タイムアウト

Server -> Server: 指数バックオフで待機して再試行

end

end

end

alt 応答に本文あり

Server -> Server: reply を抽出（空なら例外）

Server -> Sess: Session["ChatAnswer"] = reply

Server -> Browser: 回答を表示（HTML エンコード） \n 出力ボタンを有効化

else 応答が空/エラー

Server -> Browser: エラーメッセージ表示

note right of Server

エラー内容を DBLogger.LogError に記録

end note

end

end

deactivate Server

== 回答のエクスポート ==

User -> Browser: 「Word 出力」または「PDF 出力」クリック

alt Word 出力

Browser -> Server: btnExportWord_Click

activate Server

Server -> Sess: Session["ChatAnswer"] を取得

alt 回答なし

Server -> Browser: 「出力する回答が見つかりません。」表示

else 回答あり

Server -> Server: OpenXML SDK で .docx 生成

Server -> Browser: ダウンロード応答\nContent-Type=application/vnd.openxmlformats-officedocument.wordprocessingml.document

note right of Server

Response.Flush → SuppressContent → CompleteRequest

end note

end

deactivate Server

else PDF 出力

Browser -> Server: btnExportPDF_Click

activate Server

Server -> Sess: Session["ChatAnswer"] を取得

alt 回答なし

Server -> Browser: 「出力する回答が見つかりません。」表示

else 回答あり

Server -> Server: iTextSharp で PDF 生成 (MSGothic フォント)

Server -> Browser: ダウンロード応答\nContent-Type=application/pdf

note right of Server

Response.Flush → SuppressContent → CompleteRequest

end note

end

deactivate Server

end

@enduml

ユースケースシーケンス図：社内ドキュメントRAG+Chat回答、Word/PDF出力

