ParisNeo / lollms Public

Code Issues 6 Pull requests Actions Projects Security and quality

Commit a6625dc

ParisNeo committed on Jan 1

```
feat: add authentication dependencies to file endpoints and enhance image upload validation

- Import PIL Image for image verification.
- Require `current_user` via `get_current_active_user` in:
  - `/extract-text`
  - `/export-markdown`
  - `/export-content`
- Implement strict MIME type whitelist and content verification using PIL in `upload_chat_image`.
- Improve filename handling with UUID prefix to avoid collisions.
```

1 parent 89bbd18 commit a6625dc

1 file changed +30 -6

Top

Filter files…

backend/routers

files.py

Search within code

backend/routers/files.py

```
@@ -25,7 +25,7 @@
25 25  from docx.oxml import parse_xml
26 26  from docx.oxml.ns import qn
27 27  from latex2mathml.converter import convert as latex2mathml
28    -
```

```
      28  + from PIL import Image
29    29
30    30    # Try to import optional document parsing libraries
31    31    try:
            @@ -230,7 +230,8 @@ def extract_text_from_file_bytes(file_bytes: bytes,
            filename: str, extract_image
230   230
231   231   @files_router.post("/extract-text")
232   232   async def extract_text_from_file(
233       - -     file: UploadFile = File(...)
      233   +     file: UploadFile = File(...),
      234   +     current_user: UserAuthDetails = Depends(get_current_active_user)
234   235   ):
235   236       """
236   237       Extracts text content from a single uploaded file.
            @@ -247,7 +248,8 @@ async def extract_text_from_file(
247   248   @files_router.post("/export-markdown")
248   249   async def export_as_markdown(
249   250       content: str = Form(...),
250       - -     filename: str = Form("export.md")
      251   +     filename: str = Form("export.md"),
      252   +     current_user: UserAuthDetails = Depends(get_current_active_user)
251   253   ):
252   254       """
253   255       Accepts text content and returns it as a downloadable Markdown file.
            @@ -637,7 +639,10 @@ def html_wrapper(html_body: str, title: str = "Export")
            -> bytes:
637   639       return f"<html><head><meta charset='utf-8'><title>{title}</title></head>
          <body>{html_body}</body></html>".encode("utf-8")
638   640
639   641   @files_router.post("/export-content")
640       - async def export_content(payload: ContentExportRequest):
      642   + async def export_content(
      643   +     payload: ContentExportRequest,
      644   +     current_user: UserAuthDetails = Depends(get_current_active_user)
      645   + ):
641   646       export_format = payload.format.lower()
642   647       setting_key_format = 'markdown' if export_format == 'md' else export_format
643   648       setting_key = f"export_to_{setting_key_format}_enabled"
            @@ -810,10 +815,29 @@ async def upload_chat_image(
```

```diff
     """Handles image uploads specifically for attaching to a chat message before sending."""
     temp_path = get_user_temp_uploads_path(current_user.username)
     uploaded_files = []
+
+    # Allowed mime types set for quick lookup
+    ALLOWED_MIME_TYPES = {"image/jpeg", "image/png", "image/gif", "image/webp", "image/bmp", "image/tiff"}
+
     for file in files:
-        if not file.content_type.startswith("image/"):
-            raise HTTPException(status_code=400, detail="Only image files are allowed.")
+        # 1. Content-Type Header Check
+        if file.content_type not in ALLOWED_MIME_TYPES:
+             raise HTTPException(status_code=400, detail=f"Invalid file type: {file.content_type}. Only images are allowed.")
 
+        # 2. Content Verification using PIL
+        try:
+            file.file.seek(0)
+            img = Image.open(file.file)
+            img.verify() # Verify integrity
+
+            # Check format (normalize to upper case)
+            if img.format.upper() not in ["JPEG", "PNG", "GIF", "WEBP", "BMP", "TIFF"]:
+                 raise HTTPException(status_code=400, detail="Unsupported image format.")
+
+            file.file.seek(0) # Reset pointer
+        except Exception:
+            raise HTTPException(status_code=400, detail="Invalid or corrupted image file.")
+
         s_filename = secure_filename(file.filename)
         # Use a more unique name to avoid collisions
         unique_filename = f"{uuid.uuid4().hex[:8]}_{s_filename}"
```

Comments 0