

```

import os
from fastapi import FastAPI, HTTPException
from azure.storage.blob import BlobServiceClient
from datetime import datetime

app = FastAPI()

# --- Configuration (Read at runtime) ---
# It is safer to pull these inside the functions or ensure they exist
BLOB_CONNECTION_STRING = os.getenv("BLOB_CONNECTION_STRING")
BLOB_CONTAINER = os.getenv("BLOB_CONTAINER", "payloads")

@app.get("/")
def health():
    """Basic health check to ensure the app is live."""
    return {"status": "ok", "timestamp": datetime.utcnow().isoformat()}

@app.get("/save/{loadid}")
async def save_load_id(loadid: str):
    """
    Queries data (placeholder for DB) and saves a file to Azure Blob Storage.
    """

    # 1. Validate Connection String exists
    if not BLOB_CONNECTION_STRING:
        raise HTTPException(status_code=500, detail="BLOB_CONNECTION_STRING not configured in Azure.")

    try:
        # 2. Initialize the client INSIDE the call to prevent startup hangs
        blob_service_client =
            BlobServiceClient.from_connection_string(BLOB_CONNECTION_STRING)
        container_client = blob_service_client.get_container_client(BLOB_CONTAINER)

        # 3. Prepare the content
        # Note: In the next step, we will replace this with your Database Query result
        content = f"LOADID: {loadid}\nSaved at {datetime.utcnow()} UTC"
        blob_name = f"{loadid}.txt"

        # 4. Upload to Blob
        container_client.upload_blob(
            name=blob_name,
            data=content,
            overwrite=True
        )
    
```

```
return {  
    "status": "success",  
    "container": BLOB_CONTAINER,  
    "blob": blob_name,  
    "loadid": loadid  
}  
  
except Exception as e:  
    # This catches things like Authentication errors or Container not found  
    raise HTTPException(status_code=500, detail=f"Storage Error: {str(e)}")  
  
if __name__ == "__main__":  
    import uvicorn  
    # Azure App Service usually handles the port, but for local testing:  
    uvicorn.run(app, host="0.0.0.0", port=8000)
```