📋 Plágium Ellenőrző Rendszer - Áttekintés

# Robot Framework Plágium Ellenőrző Rendszer

![Robot Framework](https://robotframework.org/)

![Python](https://www.python.org/)

![SQLite](https://sqlite.org/)

![License](LICENSE)

> 🔍 Automatizált DOCX dokumentum plágium ellenőrzés hash-alapú algoritmussal, Excel jelentéskészítéssel és automatikus email értesítéssel.

## Főbb funkciók

* 🔍 Automatikus plágium detektálás SHA-256 hash algoritmussal
* 📊 Excel jelentések színkódolt státusz kategóriákkal
* 📧 Automatikus email küldés Outlook COM integrációval
* 🗄️ SQLite adatbázis teljes előzmény nyilvántartással
* 🎯 Háromszintű kategorizálás: Rendben / Gyanús / Másolt
* 🔄 Batch feldolgozás több dokumentum egyidejű kezelésére

## Gyors indítás

### 1️⃣ Telepítés

# Virtuális környezet aktiválása  
rf\_env\Scripts\activate  
  
# Vagy egyszerűen  
.\telepito.bat

### 2️⃣ Konfiguráció

# Plagium.config szerkesztése  
email=your-email@company.com  
input\_folder=C:\Documents\ToCheck  
output\_folder=C:\Reports

### 3️⃣ Futtatás

# Robot Framework teszt futtatása  
rf\_env\Scripts\robot.exe PLG-00-main.robot  
  
# Vagy batch fájlon keresztül  
start.bat

## Eredmény példa

### Konsol kimenet

=== DOCX FÁJLOK KERESÉSE ===  
Talált DOCX fájlok száma: 6  
  
🏆 VÉGSŐ ÖSSZESÍTÉS:  
📊 Összes dokumentum: 45  
🟢 Rendben: 41 dokumentum (91.1%)  
🟡 Gyanús: 2 dokumentum (4.4%)  
🔴 Másolt: 2 dokumentum (4.4%)  
  
✅ Excel fájl: plagium\_eredmenyek\_20250825\_040223.xlsx  
🎯 PLAGIUM ELLENORZÉS TELJES FOLYAMATA BEFEJEZVE! ✅

### Automatikus email

* HTML formátumú jelentés színkódolt státuszokkal
* Excel melléklet részletes adatokkal
* Összesítő statisztikák százalékos megoszlással

## Rendszer architektúra

📁 PlagiumEllenorzes/  
├── 🤖 PLG-00-main.robot # Fő Robot Framework teszt  
├── ⚙️ Plagium.config # Konfigurációs fájl  
├── 📚 libraries/ # Python modulok  
│ ├── 🐍 DocxReader.py # DOCX olvasó library  
│ ├── 📧 send\_email.py # Email küldő rendszer  
│ ├── 📊 excel\_export\_simple.py # Excel export engine  
│ └── ⚙️ get\_config.py # Konfiguráció betöltő  
├── 📂 resources/ # Robot Framework erőforrások  
│ ├── 🔑 keywords.robot # Kulcsszó definíciók  
│ └── 🔢 variables.robot # Változó definíciók  
├── 🗃️ test\_database.db # SQLite adatbázis  
└── 🐍 rf\_env/ # Python virtuális környezet

## Plágium algoritmus

### Hash-alapú összehasonlítás

1. 📄 DOCX beolvasás: Szöveges tartalom kinyerése
2. 🔐 SHA-256 hash: Minden sorhoz egyedi hash generálás
3. 🔍 Összehasonlítás: Hash értékek összevetése adatbázisban
4. 📊 Kategorizálás: Redundancia hossz alapján értékelés

### Kategorizálási szabályok

* 🟢 Rendben: < 300 karakter redundancia
* 🟡 Gyanús: 300-1200 karakter redundancia
* 🔴 Másolt: > 1200 karakter redundancia

## Email automatizálás

### Háromszintű küldési stratégia

1. 🎯 Outlook COM: Automatikus mail.Send() hívás
2. 💾 Piszkozat: Fallback mentés Drafts mappába
3. 📝 Hibanapló: Részletes hibajelentés

# Outlook COM automatikus küldés  
outlook = win32com.client.Dispatch("Outlook.Application")  
mail = outlook.CreateItem(0)  
mail.Send() # 🚀 AUTOMATIKUS KÜLDÉS

## Adatbázis struktúra

### Redundancia tábla

* status: Kategorizálás (Rendben/Gyanús/Másolt)
* file\_name: Dokumentum neve
* maxismeteltkarakterszam: Legnagyobb redundancia
* record\_date: Feldolgozás dátuma

### HashCodes tábla

* hash\_value: SHA-256 hash (PRIMARY KEY)
* filename, filepath: Fájl információk
* line\_content: Eredeti szöveg tartalom

### Repeat tábla

* repeated\_line: Ismétlődő szövegrészek
* block\_id: Ismétlési blokk azonosító
* sumlinelength: Összesített redundancia hossz

## Konfiguráció

### Plagium.config beállítások

|  |  |  |
| --- | --- | --- |
| **Paraméter** | **Leírás** | **Példa** |
| email | Címzett email | manager@company.hu |
| input\_folder | DOCX forrás könyvtár | C:\Documents\Check |
| output\_folder | Excel cél könyvtár | C:\Reports\Output |
| email\_subject | Email tárgy sablon | Plágium Jelentés |
| excel\_prefix | Excel fájl prefix | plagium\_report |

## Rendszerkövetelmények

* OS: Windows 10/11
* Python: 3.8+ (python.org)
* Outlook: Microsoft Outlook telepítve
* Memória: Min. 4GB RAM
* Tárhely: Min. 1GB szabad hely

## Függőségek

robotframework==6.1.1  
robotframework-databaselibrary==1.2.4  
python-docx==0.8.11  
openpyxl==3.1.2  
pywin32==306

## Telepítés részletesen

### 1️⃣ Python telepítés

# Python letöltés: https://python.org  
python --version # Ellenőrzés

### 2️⃣ Projekt klónozás

git clone https://github.com/your-repo/plagium-checker.git  
cd plagium-checker

### 3️⃣ Virtuális környezet

python -m venv rf\_env  
rf\_env\Scripts\activate  
pip install -r requirements.txt

### 4️⃣ Konfiguráció

notepad Plagium.config # Email és könyvtárak beállítása

## Tesztelés

### Unit tesztek

# Hash generálás teszt  
rf\_env\Scripts\python.exe -c "from libraries.DocxReader import \*; print('Hash test OK')"  
  
# Email teszt  
rf\_env\Scripts\python.exe libraries\send\_email.py  
  
# Excel export teszt  
rf\_env\Scripts\python.exe libraries\excel\_export\_simple.py test.xlsx

### Integrációs teszt

# Teljes folyamat dry-run  
rf\_env\Scripts\robot.exe --dryrun PLG-00-main.robot

## Teljesítmény

### Feldolgozási sebesség

* Kis fájl (<1MB): ~10-15 másodperc
* Közepes fájl (1-5MB): ~30-60 másodperc
* Nagy fájl (>5MB): ~2-5 perc

### Skálázhatóság

* Batch méret: 50+ fájl egyidejűleg
* Adatbázis: 10,000+ dokumentum
* Hash tábla: 1M+ bejegyzés

## Hibaelhárítás

### Gyakori problémák

# Outlook újraindítás  
taskkill /F /IM outlook.exe  
start outlook.exe

# Fájl integritás ellenőrzés  
Get-FileHash dokumentum.docx

# Folyamatok leállítása  
taskkill /F /IM python.exe

### Debug mód

# Plagium.config  
debug\_mode=true

## Dokumentáció

* 📋 Teljes dokumentáció: Részletes technikai leírás
* ⚡ Gyors referencia: 5 perces útmutató
* 🔧 Technikai áttekintés: Fejlesztői információk
* 📞 Telepítési útmutató: Lépésről lépésre

## Közreműködés

### Pull Request folyamat

1. Fork-old a repository-t
2. Hozz létre feature branch-et (git checkout -b feature/amazing-feature)
3. Commitold a változásokat (git commit -m 'Add amazing feature')
4. Push-old a branch-et (git push origin feature/amazing-feature)
5. Nyiss Pull Request-et

### Fejlesztői irányelvek

* Python: PEP 8 kódstílus
* Robot Framework: Hivatalos style guide
* Commit üzenetek: Conventional Commits formátum
* Tesztelés: Minden új funkcióhoz unit teszt

## Changelog

### v2.1.0 (2025-08-25)

* ✅ ÚJ: Háromszintű email küldési rendszer
* ✅ JAVÍTÁS: Windows path escape karakterek
* ✅ OPTIMALIZÁLÁS: Libraries könyvtár refaktoring
* ✅ FEJLESZTÉS: Tisztított logging rendszer

### v2.0.0 (2025-08-20)

* ✅ ÚJ: Outlook COM automatikus email
* ✅ ÚJ: Excel export funkció
* ✅ JAVÍTÁS: SQLite optimalizálás

## Licenc

Ez a projekt MIT licenc alatt áll. Lásd a LICENSE fájlt a részletekért.

## Támogatás

* 📧 Email: support@plagium-checker.com
* 💬 Issues: GitHub Issues
* 📖 Wiki: Projekt Wiki

## Köszönetnyilvánítás

* Robot Framework csapat a fantasztikus automatizálási keretrendszerért
* Python-docx fejlesztők a DOCX támogatásért
* OpenPyXL közösség az Excel integrációért
* Microsoft a Outlook COM API-ért

---

⭐ Ha hasznos volt a projekt, adj egy csillagot a GitHub-on!

🤖 Robot Framework Plágium Ellenőrző v2.1.0

📅 Utolsó frissítés: 2025. augusztus 25.