

Rapport de faisabilite - Alfred Lecteur Hebdomadaire

Date : 23 fevrier 2026

Projet : Alfred - Majordome Numerique

Phase : Cerveau (Julia) / Coeur (Elixir)

1. Resume du besoin

Alfred doit lire 1 livre par semaine selon ce cycle :

- ? Jours 1-6 : lecture repartie (pages/jour calculees)
 - ? Jour 7 : rapport de lecture a Monsieur
 - ? Chaque jour : integration des valeurs + jugement critique
 - ? Memoire cumulative : chaque jour, Alfred se souvient de ce qu'il a lu + ses jugements
 - ? Fin de lecture : 3 jugements/valeurs cles + 3 grands axes/idees du livre
-

2. Source de livres : Project Gutenberg via Gutendex

Pourquoi Gutenberg ?

- ? 76 000+ livres du domaine public
- ? API REST JSON (Gutendex) sans authentication
- ? Format texte brut UTF-8 (le plus simple a traiter)
- ? Livres en francais disponibles (filtre languages=fr)
- ? Gratuit, legal, illimite

Workflow de telechargement

1. Recherche : GET <https://gutendex.com/books?languages=fr&sort=popular>
2. Selection aleatoire ou par genre/theme
3. Telechargement : GET <https://www.gutenberg.org/ebooks/{id}.txt.utf-8>
4. Stockage : ~/.alfred/data/library/{id}_{titre}.txt

Formats supportes

Format | Outil extraction | Dependance
.txt | File.read (Elixir) | Aucune
.epub | :zip (Erlang stdlib) + Regex | Aucune
.pdf | pdftotext (poppler-utils) | Deja installe

3. Architecture technique

3.1 Modules a creer

lib/alfred/library/
downloader.ex - Telechargement depuis Gutenberg
reader.ex - Extraction texte (txt/epub/pdf)
paginator.ex - Decoupage en pages (~250 mots)
scheduler.ex - Planning de lecture (pages/jour)

analyzer.ex - Analyse Mistral (jugements, valeurs)
memory.ex - Memoire de lecture persistante
reporter.ex - Generation du rapport J7

3.2 Cycle de lecture (7 jours)

Jour 0 (dimanche soir) :

- Telecharger un nouveau livre
- Compter les pages
- Calculer `pages_par_jour = ceil(total_pages / 6)`
- Stocker le planning dans `reading_state.json`

Jours 1-6 (quotidien, declenche par daemon) :

- Lire les pages du jour
- Envoyer a Mistral pour analyse/jugement
- Stocker jugements + resume du jour
- Accumuler la memoire de lecture

Jour 7 (rapport) :

- Synthetiser les 6 jours de lecture
- Selectionner 3 jugements/valeurs cles
- Identifier 3 grands axes/idees
- Generer rapport pour Monsieur
- Notifier via SimpleX

3.3 Analyse quotidienne (Mistral)

Chaque jour, Alfred envoie le texte du jour a Mistral avec ce prompt :

```
Tu es Alfred, majordome numerique. Tu lis un livre pour ton maitre.  
Livre : {titre} par {auteur}  
Pages lues aujourd'hui : {pages_debut} a {pages_fin}  
Ce que tu as deja lu et retenu : {memoire_jours_precedents}  
Texte du jour :  
{texte}  
Analyse ce passage. Pour chaque idee importante, donne ton jugement :  
- Bien / Mal  
- Interessant / Passionnant / Banal  
- Devrait etre applique dans la vie reelle  
- Utopique (beau mais irreal)  
- C'est juste une fiction  
- C'est une fiction mais ca serait bien que ca existe  
- Pertinent pour Monsieur  
Retourne un JSON :  
{  
  "resume": "resume du passage (3-5 phrases)",  
  "jugements": [  
    {"idee": "...", "jugement": "...", "force": 0.0-1.0}  
  ],  
  "valeurs_detectees": ["valeur1", "valeur2"],  
  "axes_emergents": ["axe1"]  
}
```

3.4 Memoire de lecture

Fichier : `~/alfred/data/library/current_reading.json`

```
{  
  "book": {  
    "id": 1342,  
    "title": "Orgueil et Prejuges",  
    "author": "Jane Austen",  
    "total_pages": 284,  
  },  
}
```

```

"pages_per_day": 48,
"started_at": "2026-02-24",
"source": "gutenberg"
},
"progress": {
"day": 3,
"pages_read": 144,
"pages_total": 284
},
"daily_logs": [
{
"day": 1,
"pages": "1-48",
"resume": "...",
"jugements": [...],
"valeurs": [...],
"axes": [...]
}
],
"accumulated": {
"all_jugements": [...],
"all_valeurs": [...],
"all_axes": [...],
"running_summary": "Ce que j'ai retenu jusqu'ici..."
}
}

```

3.5 Rapport final (Jour 7)

Mistral recoit l'ensemble des notes accumulees et genere :

1. ****3 jugements/valeurs cles**** -- les plus marquants, les plus forts
2. ****3 grands axes/idees**** -- les themes centraux du livre
3. ****Resume general**** -- 10-15 phrases
4. ****Recommandation**** -- Alfred dit s'il recommande ce livre a Monsieur

Le rapport est :

- ? Envoye via SimpleX
- ? Stocke dans `~/alfred/data/library/reports/`
- ? Les 3 valeurs deviennent des convictions (systeme existant)

3.6 Integration daemon

Nouveau slot daemon (1x/jour a 6h du matin = check #360 offset) :

- Si jour 0 : telecharger nouveau livre
- Si jours 1-6 : lire portion + analyser
- Si jour 7 : generer rapport

4. Estimation des couts API Mistral

Par livre (semaine)

Operation	Tokens entree	Tokens sortie	Cout
Analyse J1-J6 (6x)	~20K x 6 = 120K	~2K x 6 = 12K	~0.016
Rapport J7	~10K	~4K	~0.002
Total/livre	**~130K**	**~16K**	**~0.018 EUR**

Cout annuel : ~52 livres x 0.018 EUR = ~0.94 EUR/an

Contrainte : taille des chunks

- ? Mistral Small : 128K tokens de contexte
 - ? Un livre de 300 pages = ~97K tokens (texte seul)
 - ? Strategie : envoyer par journee de lecture (~50 pages = ~16K tokens)
 - ? Largement dans les limites
-

5. Estimation du volume de code

Module | Lignes estimees
downloader.ex | ~80
reader.ex | ~120 (txt + epub + pdf)
paginator.ex | ~40
scheduler.ex | ~100
analyzer.ex | ~150
memory.ex | ~100
reporter.ex | ~120
Tests | ~200
Modifs daemon | ~20
Total | **~930 lignes**

6. Faisabilite

FAISABLE a 100%

Critere | Statut | Detail
Source de livres | OK | Gutenberg API, gratuit, 76K+ livres
Lecture txt/epub | OK | Zero dependances (Erlang :zip)
Lecture PDF | OK | pdftotext deja installe
Analyse IA | OK | Mistral Small, 128K contexte
Cout API | OK | < 1 EUR/an
Memoire lecture | OK | JSON local (pattern existant)
Integration daemon | OK | Slot quotidien
Rapport SimpleX | OK | Bridge existant
Convictions | OK | Systeme deja en place

Points d'attention

1. Livres scannes (PDF image) : pdftotext ne peut pas les lire -> privilegier txt/epub
 2. Qualite OCR Gutenberg : certains vieux textes ont des artefacts
 3. Livres tres longs (>500 pages) : augmenter a 7-8 jours ou sauter
 4. Choix du livre : automatique (populaire) ou sur demande de Monsieur ?
-

7. Questions ouvertes

1. **Langue des livres** : uniquement francais ? Ou aussi anglais/autres ?

2. **Genre/theme** : preference ? (philosophie, science-fiction, classiques, economie, technique...)
 3. **Choix du livre** : Alfred choisit seul (aleatoire/populaire) ou Monsieur suggere ?
 4. **Rapport** : format souhaite ? (message SimpleX, fichier PDF, les deux ?)
 5. **Convictions** : les valeurs extraites des livres doivent-elles devenir des convictions d'Alfred ?
 6. **Bibliotheque locale** : garder les livres telecharges ? Ou supprimer apres lecture ?
 7. **Historique** : Alfred doit-il se souvenir de TOUS les livres lus, ou juste les 3 valeurs + 3 axes ?
-

8. Planning d'implementation

Etape | Duree estimee

1. Downloader + Reader (txt/epub) | Premiere session
 2. Paginator + Scheduler | Meme session
 3. Analyzer (prompts Mistral) | Deuxieme session
 4. Memory + Reporter | Meme session
 5. Integration daemon + tests | Troisieme session
 6. Test reel avec 1er livre | Validation
-

Rapport genere par Claude Opus 4.6 pour le projet Alfred