

Datasheet — Jira Issue Extractor

Présentation

Jira Issue Extractor est une application web d'archivage et d'export d'issues Jira, déployable on-premises ou en cloud privé. Elle permet aux équipes de constituer des archives structurées de leurs projets Jira, de planifier des exports récurrents et d'explorer les données hors ligne — sans dépendance permanente à l'instance Jira source.

Cas d'usage typiques

Cas d'usage	Description
Archivage réglementaire	Constitution d'archives conformes pour audit ou obligation légale de conservation
Migration Jira	Export complet d'un projet avant migration vers une nouvelle instance
Consultation hors ligne	Accès aux issues et pièces jointes sans connexion à Jira
Reporting & intégration	Export CSV/JSON pour alimentation de tableaux de bord ou ETL
Continuité d'activité	Snapshot périodique des projets critiques en cas d'indisponibilité Jira

Fonctionnalités clés

Connectivité Jira

- Compatible **Jira Cloud** et **Jira On-Premises** (Server / Data Center)
- Authentification **Basic** (identifiant + mot de passe ou API token), **PAT** (Personal Access Token) et **OAuth2** (authorization code)
- Connexions nommées et réutilisables entre exports et planifications
- Test de connexion en un clic avec affichage des informations serveur

Export

- Sélection des issues par **requête JQL** libre avec validateur intégré
- Aperçu des 10 premières issues avant lancement
- **4 formats de sortie** : XML, HTML, JSON, CSV — avec option d'empaquetage ZIP
- Inclusion optionnelle des **pièces jointes** (téléchargement binaire)
- Limite d'issues configurable
- Suivi en **temps réel** de la progression (barre + terminal de logs)

Planification

- Exports **automatiques** sans intervention : unique, quotidien, hebdomadaire
- Activation/désactivation sans suppression
- Déclenchement manuel immédiat ("Lancer maintenant")

Stockage et navigation

- Stockage des exports sur **MinIO** (S3-compatible) ou volume local
- **Navigateur de fichiers** intégré : exploration de l'arborescence dans le navigateur
- Prévisualisation HTML sécurisée avec mécanisme de token à usage unique
- Téléchargement direct des archives ZIP

Multi-utilisateurs

- 3 rôles : **admin**, **editor**, **visitor**
- Gestion des comptes par invitation email
- Réinitialisation de mot de passe par email
- Préférences utilisateur sauvegardées entre sessions

Formats d'export détaillés

Format	Contenu	Granularité
XML	Tous les champs, commentaires, liens, historique	1 fichier par issue
HTML	Rendu navigateur : description, commentaires, pièces jointes	1 fichier par issue + index
JSON	Payload brut API Jira (champs personnalisés inclus)	1 fichier par issue
CSV	Tableau synthétique : clé, résumé, statut, assigné, champs custom	1 fichier global

Format	Contenu	Granularité
ZIP _(option)_	Archive compressée de tous les fichiers produits — nécessite au moins un format de contenu	1 fichier global

Architecture et déploiement

Stack technique

Composant	Technologie
Backend	Python 3.11 / FastAPI / Uvicorn
Frontend	React 18 / Vite / Tailwind CSS
Base de données	PostgreSQL 16
Bus d'événements	Redis 7 (optionnel)
Stockage objet	MinIO (optionnel, S3-compatible)
Conteneurisation	Docker / Docker Compose

Modes de déploiement

Mode	Conteneurs	Usage recommandé
Complet	6 (frontend, backend, scheduler, postgres, redis, minio)	Production, multi-utilisateurs
Simplifié	3 (frontend, backend, postgres)	Usage interne mono-instance
Développement	Services locaux	Développement et tests

Prérequis matériels (mode complet)

Ressource	Minimum	Recommandé
CPU	2 vCPU	4 vCPU
RAM	2 Go	4 Go
Stockage	20 Go	50 Go+

Sécurité

Mécanisme	Détail
Authentification	JWT Bearer (HS256, 8h)
Mots de passe	bcrypt, facteur 12, politique de complexité
Tokens email	UUID haché SHA-256, usage unique, TTL court
Prévisualisation	Token usage unique → cookie HttpOnly SameSite=Strict
Contrôle d'accès	RBAC 3 niveaux (admin / editor / visitor)
Transport	HTTPS recommandé via reverse proxy (nginx, Traefik)

Intégrations

Système	Type
Jira Cloud	REST API v3
Jira Server / Data Center	REST API v2
MinIO / AWS S3	API S3-compatible
Serveur SMTP	Envoi d'emails transactionnels (invitations, réinitialisation)

Modes dégradés (résilience)

Service absent	Comportement
Redis	SSE en mémoire (single-node)
MinIO	Exports sur volume local
SMTP	Invitations non envoyées (URL loguée en console)
Scheduler désactivé	Planifications non exécutées automatiquement

Licence et distribution

- Application open-source déployée en interne
- Code source versionné sur Bitbucket (SmartView)

- Déploiement par image Docker ou build local
- Pas de dépendance à un service cloud tiers obligatoire