

Umstieg von Jira

Übernimm Projekte, Vorgänge, Epics, Sprints, Kommentare, Zeiten und Dateien aus Jira Cloud oder Jira Data Center, prüfe vorab, was passieren würde, und mache es bei Bedarf rückgängig.

Überblick

Der Import ist Teil von Resourcecloud (**Admin → Datenimport**, für RC Admins). Du exportierst deine Jira-Daten in eine Datei, lädst sie hoch, legst fest, was jedes Jira-Ding wird, siehst dir an, was passieren *würde*, und startest. Der Import läuft im Hintergrund und berichtet, was er getan hat.

1. Export

Ein Skript liest die REST-API von Jira und schreibt ein Zip.

2. Hochladen

Beliebig groß; ein unterbrochener Upload macht dort weiter, wo er aufgehört hat.

3. Zuordnen

Typen, Status, Projekte und Personen, vorgeschlagen aus dem, was du schon hast.

4. Prüfen, starten

Ein Probelauf zeigt, was neu, geändert und unberührt bliebe.

- **Wiederholbar.** Jedes Jira-Objekt wird über seine ID wiedererkannt. Derselbe Export noch einmal ändert nichts; ein neuerer Export aktualisiert, was sich in Jira geändert hat, und ergänzt Neues. Nichts wird doppelt angelegt.
- **Umkehrbar.** Ein Import lässt sich rückgängig machen: Er entfernt, was er angelegt hat, und behält, woran seitdem gearbeitet wurde.
- **Leise.** Beim Import werden keine E-Mails und keine Benachrichtigungen verschickt.
- **Deins.** Die Daten gehen von Jira auf den Rechner, auf dem du das Skript ausführst, und von dort in dein eigenes Resourcecloud. Sie werden nirgendwohin sonst geschickt.

Vorbereitung

- **Ein RC-Admin-Konto** in Resourcecloud.
- **Teams.** Jedes Jira-Projekt braucht in Resourcecloud ein Team (wer die Aufgaben sieht und bearbeitet) und eine Lösung (wozu die Arbeit gehört). Vorhandene werden über den Namen zugeordnet; sonst wählst du sie oder lässt sie vom Import anlegen (siehe unten).
- **Personen.** Jira-Personen werden Resourcecloud-Benutzern zuerst über die E-Mail-Adresse zugeordnet, dann über den Benutzernamen. Lege die Konten, die bleiben sollen, vorher an, oder erlaube dem Import, fehlende als *inaktive* Konten anzulegen.
- **Stufen.** Hat das Team, das die Arbeit bekommt, schon ein Board mit Stufen, benenne sie wie deine Jira-Status (oder lege in der Zuordnung fest, welcher Jira-Status in welche Stufe geht).
- **Zugang zu Jira.** Ein Konto, das die Projekte sehen darf. Jira Cloud: deine E-Mail-Adresse und ein API-Token. Data Center: ein Personal Access Token (oder Benutzername und Passwort). Für die Liste aller Personen braucht das Konto außerdem das Recht, Benutzer zu sehen; sonst werden die Personen aus den Vorgängen genommen.
- **Python 3.9 oder neuer** auf dem Rechner, der das Export-Skript ausführt. Mehr Software ist nicht nötig.
- **Speicherplatz** für den Export (meist Anhänge) und für den Upload auf dem Resourcecloud-Server.

Begriffe Jira und Resourcecloud

In Jira	In Resourcecloud	Hinweis
Projekt	Team + Lösung	Das Team bestimmt, wer die Aufgaben sieht, die Lösung, wozu die Arbeit gehört. Gewählt pro Projekt in der Zuordnung.
Vorgang (Task, Bug, Story, ...)	Aufgabe	Der Aufgabentyp kommt aus der Zuordnung (Bug → Bugfix, ...). Eine Jira- Story wird zur Aufgabe vom Typ <i>Story</i> .
Sub-Task	Unteraufgabe	Gehört zu ihrer Aufgabe und übernimmt deren Typ, wenn du keinen anderen wählst.

Epic	Epic	In Resourcecloud enthält ein Epic <i>Stories</i> und eine Story Aufgaben; deshalb bekommt jedes Epic eine gleichnamige Story, die seine Aufgaben enthält.
Story (Vorgangstyp)	Aufgabe vom Typ Story oder Resourcecloud-Story	Standard ist eine Aufgabe. In der Zuordnung kannst du einen Vorgangstyp stattdessen zur Resourcecloud-Story machen (ein Behälter für Aufgaben).
Board	Kanban-Board und Sprintplanung des Teams	Boards selbst werden nicht kopiert; sie bestimmen, zu welchem Team ein Sprint gehört.
Workflow, Status	Stufen des Team-Boards	Kategorie <i>To Do</i> → Zugewiesen (nicht begonnen), <i>Done</i> → Aufgabe geschlossen, <i>In Progress</i> → eine Stufe deiner Wahl (vorgeschlagen: Working).
Sprint	Sprint	Name, Ziel, Daten und ob er abgeschlossen, aktiv oder künftig ist. Aufgaben kommen in die Sprints, in denen sie waren.
Story Points	Story Points	Auf ganze Zahlen gerundet.
Ursprüngliche Schätzung	Geschätzte Zeit	Aufgaben ohne Schätzung bekommen 10 Minuten, wie in der API.
Worklog	Zeiteintrag	Gebucht auf die Person, die ihn erfasst hat, an ihrem Tag. Braucht einen passenden Benutzer (siehe unten).
Priorität	Priorität: normal oder hoch	Highest, High, Blocker, Critical, Major, Urgent → hoch. Pro Priorität änderbar.
Kommentar	Kommentar	Mit ursprünglichem Autor und Datum.
Anhang	Datei	Aus dem Zip.
Melder, Bearbeiter	Erstellt von, zugewiesen an	Zugeordnet über E-Mail, dann Benutzername.
Gelöst, Lösungsdatum	Erledigt, Erledigt-Datum	Eine Aufgabe, deren Status auf „Erledigt“ zeigt, wird an ihrem Lösungsdatum geschlossen.

Beschreibung	Beschreibung	In formatierten Text umgewandelt. Unsicheres Markup wird entfernt.
--------------	--------------	--

1. Export aus Jira

Jira hat keinen Knopf „alles exportieren“, nimm deshalb das Skript, das Resourcecloud mitbringt: `tools/jira_export.py`. Es fragt die REST-API von Jira (es liest nur) und schreibt ein Zip mit Projekten, Vorgängen, Kommentaren, Worklogs, Anhängen, Boards, Sprints und Personen.

1 Jira Cloud:

```
python3 jira_export.py --url https://acme.atlassian.net \
  --user du@acme.com --token "$JIRA_TOKEN" --projects ABC,DEF
```

2 Jira Data Center / Server:

```
python3 jira_export.py --url https://jira.acme.com \
  --token "$JIRA_TOKEN" --projects ABC,DEF
```

(ein Personal Access Token; oder `--user` und `--password`)

3 Ohne `--projects` werden alle Projekte exportiert, die das Konto sieht. Gib Zugangsdaten über die Umgebung an (`JIRA_USER` , `JIRA_TOKEN` , `JIRA_PASSWORD`), damit sie nicht in der Shell-Historie landen.

4 Ergebnis ist ein Ordner `jira-export/` und `jira-export.zip`. Eine `README.txt` darin sagt, was exportiert wurde.

Nützliche Optionen. `--no-attachments` lässt Dateien weg. `--max-attachment-mb 200` überspringt größere Dateien. `--delay 0.5` macht Pausen zwischen Anfragen bei einem ausgelasteten Jira. `--cafile` vertraut einem internen Zertifikat. Mit demselben `--out` erneut gestartet, macht das Skript weiter: bereits geladene Dateien werden nicht noch einmal geholt.

Das Skript wartet, wenn Jira um Langsamkeit bittet, ergänzt die Kommentar- und Worklog-Listen, die Jiras Suche abschneidet, schickt deine Zugangsdaten nie an einen anderen Host (Jira Cloud liefert Dateien von einem anderen aus) und bricht mit einer klaren Meldung ab, wenn Zugangsdaten oder Projektschlüssel nicht stimmen.

Eigener Export. Der Import liest, was die REST-API von Jira liefert: Seiten von Suchergebnissen, Sprint- und Board-Seiten, Benutzerlisten, als `.json`, `.jsonl`, `.gz` oder als `.zip` daraus. Hast du solche Dateien schon, lade sie unverändert hoch. Anhänge gehören ins Zip als `attachments/<ID>/<Dateiname>`; der Import findet sie über Jiras Anhangs-ID.

2. Hochladen

- 1 Öffne **Admin → Datenimport** und ziehe das Zip auf das Feld (oder klicke darauf und wähle die Datei).
- 2 Die Datei wird in Stücken von 4 MB mit Fortschrittsanzeige gesendet. Die Größe spielt keine Rolle. Bricht die Verbindung ab, versucht die Seite es selbst erneut; schließt du die Seite, öffne den Import in der Liste, wähle **dieselbe Datei** und er macht dort weiter, wo der Server stehen geblieben ist.
- 3 Ist das letzte Stück da, liest Resourcecloud die Datei im Hintergrund („Export wird eingelesen...“) und die Zuordnung öffnet sich.

Steht Resourcecloud hinter einem Reverse-Proxy, der die Größe von Anfragen begrenzt (manche voreingestellt auf 1 MB), werden Stücke von 4 MB abgelehnt. Erhöhe das Limit für `/backend/import_upload` auf mindestens 8 MB.

3. Zuordnen

Die Zuordnung listet alles auf, was eine Entscheidung braucht, mit der Häufigkeit. Alles ist schon aus dem vorgeschlagen, was in Resourcecloud existiert; lies es durch, ändere, was nicht stimmt, und drücke **Prüfen**. **Zurück zum Vorschlag** verwirft deine Änderungen.

Tab	Du entscheidest
Vorgangstypen	Was jeder Jira-Vorgangstyp wird: Aufgabe, Unteraufgabe, Resourcecloud-Story oder Epic, und der Aufgabentyp. Fehlende Typen (wie <i>Story</i>) werden angelegt, wenn du es unter Optionen erlaubst.
Status	Die Stufe für jeden Jira-Status: <i>Zugewiesen</i> (nicht begonnen), <i>Erledigt</i> (geschlossen) oder eine Stufe des Teams. Hat das Team die Stufe nicht, kommen die Aufgaben nach <i>Zugewiesen</i> , außer du erlaubst das Ergänzen von Stufen.
Prioritäten	Normal oder hoch.
Projekte	Team, Lösung und (bei neuer Lösung) Firma pro Jira-Projekt. Wenn möglich über den Namen zugeordnet.
Personen	Welcher Resourcecloud-Benutzer jede Jira-Person ist. Filtern, nur nicht zugeordnete Personen zeigen, einen Benutzer suchen, <i>Ich</i> wählen (Kommentare werden dir zugeschrieben), <i>Niemand</i> oder <i>Benutzer anlegen</i> .
Optionen	Was angelegt werden darf und was importiert wird (Kommentare, Worklogs, Anhänge, Sprints).

Personen ohne Zuordnung

- Ihre **Kommentare** bleiben erhalten. Sie laufen unter dem importierenden Admin, und jeder sagt „Kommentar von <Name> in Jira (<Datum>)“.
- Ihre **Worklogs werden nicht importiert**. Zeit wird auf eine Person gebucht; sie auf den zu buchen, der den Import ausführt, würde Stunden in dessen Zeiterfassung legen, die er nie gearbeitet hat. Ordne solche Personen einem Benutzer zu oder erlaube das Anlegen von Benutzern, und importiere erneut.
- Aufgaben, die ihnen zugewiesen waren, bleiben **nicht zugewiesen**.

Benutzer und Teams anlegen

Standardmäßig aus. Ein RC Admin kann es mit dem Schalter auf der Übersichtsseite des Imports erlauben; der Import bietet dann *Benutzer anlegen* und *Virtuelles Team anlegen* an. Angelegte Benutzer sind **inaktiv**, können sich erst anmelden, wenn du sie aktivierst, und zählen bis dahin nicht

als lizenzierte Plätze. Angelegte Teams sind virtuelle Teams, die die Stufen bekommen, die ihre Vorgänge brauchen, und die Personen, die darin arbeiten, als Mitglieder. Der Schalter wird beim Lauf des Imports noch einmal geprüft.

4. Prüfen und starten

- 1 **Prüfen** speichert die Zuordnung und macht einen Probelauf. Er schreibt nichts. Er zeigt pro Art, wie viele Objekte *neu*, *geändert* oder *unverändert* wären, was außer der Arbeit angelegt würde (Aufgabentypen, Lösungen, Teams, Benutzer, Stufen) und was einen Blick braucht: Personen ohne Zuordnung, Stufen, die ein Team nicht hat, übergeordnete Vorgänge oder Epics, die nicht im Export sind.
 - 2 Fehler (etwa ein Projekt ohne Team) werden pro Zeile gezeigt und müssen behoben werden. Warnungen kannst du akzeptieren.
 - 3 **Import starten** fragt noch einmal nach. Der Import läuft im Hintergrund; du kannst die Seite schließen.
-

5. Bericht und Ergebnis

Während er läuft, siehst du Fortschrittsbalken und Protokoll. Danach nennt der Bericht pro Art, wie viele Objekte angelegt, aktualisiert, unverändert, übersprungen und fehlgeschlagen sind, jedes fehlgeschlagene Objekt mit Grund, und das Protokoll (Warnungen wie Personen ohne Zuordnung, Dateien, die nicht im Zip waren). Ein fehlgeschlagenes Objekt hält die anderen nicht auf.

Sieh dir das Ergebnis dann dort an, wo deine Leute arbeiten: Backlog und Kanban-Board jedes Teams (offene Arbeit landet im Backlog oder in der Stufe, die du zugeordnet hast), die Sprintplanung, Epics und Stories, Kommentare und Zeit einer Aufgabe, ihre Dateien. **Aufräumen** entfernt die hochgeladene Datei und die eingelesenen Daten vom Server; der Bericht und das Verzeichnis dessen, was importiert wurde, bleiben.

Geschwindigkeit. Auf einem Entwicklungsrechner schrieb der Import etwa 1.000 Objekte (Aufgaben, Kommentare, Worklogs) pro Sekunde; 50.000 Vorgänge mit 150.000 Kommentaren brauchen also einige Minuten. Das Einlesen der Datei ist noch schneller. Langsam ist der Export aus Jira, und er hängt von Jira ab.

Erneut importieren

Du kannst dasselbe Projekt beliebig oft importieren: zuerst zum Ansehen, noch einmal als letzter Lauf kurz vor dem Umstieg.

- Ein Objekt, das Jira seit dem letzten Mal nicht geändert hat, wird **in Ruhe gelassen**. Ein neues wird angelegt. Ein geändertes wird **an Ort und Stelle aktualisiert**.
- Eine Aktualisierung schreibt Jiras Version über das, was in Resourcecloud steht, auch über Änderungen, die inzwischen hier gemacht wurden (Titel, Beschreibung, Bearbeiter, Stufe, Daten). Der Probelauf warnt dich mit der Zahl der betroffenen Objekte. Arbeiten die Leute schon in Resourcecloud, mache den letzten Import, bevor sie anfangen, oder nimm in Kauf, dass diese Aufgaben überschrieben werden.
- Kommentare, Worklogs und Dateien, die in Jira neu sind, werden ergänzt; vorhandene nicht verdoppelt.

Import rückgängig machen

Im Bericht eines abgeschlossenen Imports zeigt **Import rückgängig machen** eine Vorschau und läuft nach Bestätigung im Hintergrund.

- Es entfernt, was **dieser Import angelegt hat**: Aufgaben mit ihren Kommentaren, Zeiteinträgen, Dateien und Verknüpfungen, Stories, Epics, Sprints und die Benutzer, Teams, Lösungen, Aufgabentypen und Stufen, die er angelegt hat – die letzten nur, wenn sie nirgends mehr verwendet werden.

- Es nimmt **nicht** Aktualisierungen zurück, die der Import an Dingen gemacht hat, die ein früherer Import angelegt hatte.
 - Eine Aufgabe, an der **seitdem gearbeitet wurde** (Kommentare, Dateien oder Chat in Resourcecloud ergänzt), **bleibt** vollständig erhalten, und der Bericht nennt den Grund. Mit *Auch Aufgaben entfernen, an denen seitdem gearbeitet wurde* werden auch diese entfernt.
 - Eine Aufgabe, von der etwas anderes abhängt – eine Rechnung, Urlaub, ein Lead, ein Deployment, ein Meeting, eine abgeleitete Aufgabe oder eine Unteraufgabe von anderswo – wird **nie entfernt**.
 - Was bleibt, behält seinen Vermerk; ein erneuter Import des Exports aktualisiert es, statt eine zweite Kopie anzulegen.
 - Es wird abgelehnt, solange ein anderer Import läuft und solange ein späterer Import aktualisiert hat, was dieser angelegt hat (die Meldung nennt ihn; mache zuerst jenen rückgängig).
-

Was nicht importiert wird

- Labels und Komponenten als Felder: Aufgaben in Resourcecloud haben keine, deshalb stehen sie als Zeile am Ende der Beschreibung (durchsuchbar, aber ohne Filter).
- Fix-Versionen an allem außer Epics: Ein Epic kommt in ein Release mit dem Namen der ersten Fix-Version (weitere werden gemeldet, nicht übernommen); Aufgaben nicht. Release-Termine werden nicht importiert.
- Vorgangsverknüpfungen außer „Blocks“ (hängt zusammen mit, Duplikat, ...) sowie Beobachter und Stimmen. Eine „Blocks“-Verknüpfung zwischen zwei importierten Aufgaben wird zur Abhängigkeit: Die blockierte Aufgabe wartet auf die andere. Eine in Jira später entfernte Verknüpfung wird durch eine Aktualisierung hier nicht entfernt.
- Benutzerdefinierte Felder außer Story Points, Epic-Link und Sprints.
- Der Änderungsverlauf der Vorgänge, Jira-Workflows und ihre Regeln, Bildschirme, Berechtigungsschemata, Benachrichtigungsschemata, Automatisierungen.
- Dashboards, Filter, gespeicherte Ansichten und Daten von Marketplace-Apps.
- Jiras Sicherheitsstufen für Vorgänge: Importierte Aufgaben sind für ihr Team sichtbar wie jede andere Aufgabe.
- Kommentare und Worklogs an einem *Epic* (ein Epic hat in Resourcecloud keine Kommentare); der Bericht listet sie auf.
- Anhänge, die nicht im Zip sind (der Vorgang wird importiert; der Bericht nennt die fehlende Datei).

Lege die Regeln, die du brauchst, auf der Resourcecloud-Seite neu an: Stufen und Arbeitsmodell jedes Teams, Rollen, Benachrichtigungen.

Umstiegsplan

- 1 **Probe.** Exportiere ein Projekt, importiere es in ein Testteam, sieh dich um, mache es rückgängig. Verbessere die Zuordnung, bis sie so ist, wie du sie willst.
 - 2 **Alles importieren,** ein bis zwei Wochen vor dem Umstieg. Lass die Teammitglieder ihre Aufgaben ansehen; korrigiere Namen und Stufen, die sie überraschen.
 - 3 **Jira einfrieren** am Tag selbst: schreibgeschützt machen oder alle bitten aufzuhören. Noch einmal exportieren und importieren; angefasst wird nur, was sich geändert hat.
 - 4 **Umstieg.** Ab jetzt arbeiten alle in Resourcecloud. Halte Jira so lange schreibgeschützt, wie deine Aufbewahrungspflichten es verlangen; das Export-Zip ist eine vollständige Kopie dessen, was importiert wurde.
 - 5 Führe am Ende **Aufräumen** für die Importe aus, damit die Uploads nicht auf dem Server bleiben.
-

Fragen

- ▶ Warum ist eine Jira-Story eine Aufgabe und keine Resourcecloud-Story?
- ▶ Einige Personen wurden nicht zugeordnet. Was nun?
- ▶ Worklogs fehlen.
- ▶ Ein Anhang wurde als fehlend gemeldet.

► Emoji fehlen in Titeln.

► Ein Titel hat Text zwischen < und > verloren.

► Kann ich in ein Team importieren, das schon ein Board hat?

► Kann ich zwei Jira-Installationen importieren?

► Verschickt der Import E-Mails?

► Der Upload bricht jedes Mal an derselben Stelle ab.

Sicherheit

- Nur ein RC Admin kann den Import benutzen; jede Aktion wird auf dem Server geprüft, und jede Änderung braucht das übliche Sicherheits-Token.
- Die hochgeladene Datei liegt in einem Ordner, den der Webserver nicht ausliefert, und wird durch **Aufräumen** entfernt.
- Was ein Export enthält, gilt als nicht vertrauenswürdig: Markup und Skripte werden aus Titeln, Namen und formatiertem Text entfernt; Dateinamen werden bereinigt; ein Anhang, der Gigabytes behauptet, wird abgelehnt, bevor er gelesen wird.
- Zeit wird nur für Personen gebucht und Kommentare nur Personen zugeschrieben, die es in Resourcecloud gibt. Jedes importierte Objekt lässt sich dem Import, dem Admin und der Zeit zuordnen.