
Jira Software, Jira Service Desk and Jira Core

Release #111805f, 2019-05-03

Matt Harasymczuk

2019-05-03

1	Introduction	3
1.1	Requirements	3
1.2	Jira Licensing	4
1.3	Conferences	4
2	End-User	5
2.1	Keyboard Shortcuts	5
2.2	Profile Configuration	7
2.3	Project	7
2.4	Issues	8
2.5	JQL - JIRA Query Language	11
2.6	Filtry	15
2.7	Dashboard	15
2.8	Jira Software (Agile)	16
3	Administration	23
3.1	Configuration	23
3.2	License	25
3.3	Issue Types	25
3.4	Workflow	26
3.5	Screens	29
3.6	Fields	29
3.7	Issue Features	30
3.8	Issue Attributes	30
3.9	User Management	32
3.10	Apps	33
3.11	Dirty hacks	34
4	System Administration	37
4.1	Instalacja	37
4.2	Upgrade	39
4.3	Backup	41
4.4	Migrating	42
4.5	Testing Environment	47
4.6	Performance Tuning	55
4.7	DB Performance	58
4.8	API and Scripting	60
5	Appendix	63
5.1	License	63
5.2	Bibliography	63

5.3 Glossary	63
Bibliography	65

Tip: This book is free and open source at <http://jira.astrotech.io>



author	Matt Harasymczuk
email	matt@astrotech.io
www	http://www.astrotech.io
github	https://github.com/astromatt
linkedin	https://linkedin.com/in/mattharasyuczuk
facebook	https://facebook.com/matt.harasyuczuk
slideshare	https://www.slideshare.net/astrotech/presentations

Table 1.: Other books from author

	Title
http://python.astrotech.io	Python 3: from None to Machine Learning
http://devops.astrotech.io	DevOps and CI/CD with Docker
http://jira.astrotech.io	Jira Software, Jira Service Desk and Jira Core
http://git.astrotech.io	GIT and GIT Flow
http://agile.astrotech.io	Agile, Scrum, Kanban, XP, Lean
http://arch.astrotech.io	Software Architecture, Cloud, Microservices and Processes
http://linux.astrotech.io	Linux and Shell Programming with Bash

Tip: If you're interested in training course on topics from this book, please email me at matt@astrotech.io

1.1 Requirements

1.1.1 End-User

1. Dostęp do Jiry
2. Przeglądarka internetowa, preferowane:
 - Chrome
 - Safari
 - Firefox

1.1.2 Administration

1. Dostęp do Jiry
2. Przeglądarka internetowa, preferowane:
 - Chrome
 - Safari
 - Firefox

1.1.3 System Administration

1. Dostęp do Jiry
2. Dostęp SSH do serwera Jiry
3. Przeglądarka internetowa, preferowane:
 - Chrome
 - Safari
 - Firefox

1.2 Jira Licensing

- Cloud vs. Server [1]
- Ilość użytkowników
- Długość trwania licencji
- Jira Core vs. Software vs. Service Desk
- Evaluation license

1.3 Conferences

1.3.1 Atlassian Camp

- For Developers
- <https://www.atlassian.com/company/events/summit-europe/programs/atlascamp?tab=build-add-ons>

1.3.2 Atlassian Summit

- For Business
- Summit US
- Summit Europe
- <https://www.atlassian.com/company/events/summit-europe>

1.3.3 Jira Day Poland

- <http://www.intenso.pl/jira-day/>

2.1 Keyboard Shortcuts

2.1.1 Most important

- gg or .
- c
- e
- a
- i

2.1.2 Global shortcuts

Table 2.1.: Global shortcuts

Key	Description
\	Starred and recent
/	Quick search
Esc	Close drawer
c	Create issue
?	Keyboard shortcuts
g then i	Find issues
g then p'	Browse to a project
[Toggle sidebar

2.1.3 Navigating issues

Table 2.2.: Global shortcuts

Key	Description
o or Enter	View selected issue
j	Next issue
k	Previous issue
z	Toggle issue fullscreen
[Dock/undock the filters panel
n	Next activity
p	Previous activity
f	Focus search field
u	Search for issues
t	Switch filter view
y	Detail view order by

2.1.4 Issue actions

Table 2.3.: Global shortcuts

Key	Description
e	Edit issue
a	Assign issue
m	Comment on issue
s	Share issue
l	Edit issue labels
,	Jump to fields for editing
i	Assign to me

2.1.5 Board shortcuts

Table 2.4.: Global shortcuts

Key	Description
1	Backlog
2	Active sprints/Kanban board
3	Reports
z	Hide/show menus
n	Next column
p	Previous column
t	Hide/show detail view
-	Toggle all swimlanes
s then t	Send to top
s then b	Send to bottom

2.1.6 Service desk

Table 2.5.: Global shortcuts

Key	Description
o or Enter	View selected issue
j	Next issue
k	Previous issue

2.2 Profile Configuration

- Język
- Avatar (gravatar)
- Powiadamianie mailami
- Dobre praktyki filtrów na maila

2.3 Project

- Project Lead
- Categories
 - Department
 - Team
 - Project / Product
- Project vs. Boards
- Issue Collector

2.3.1 Issue key

- krótki i zwięzły
- łatwy do zapamiętania
- 2-10 liter

2.3.2 Project Configuration

- Versions
- Components
- Roles and Permissions
- Application Links

2.3.3 Assignments

Projekt

- Stwórz projekt
- Dodaj użytkownika `jira-administrator` do roli `Developers`
- Dodaj użytkownika `jira-administrator` do roli `Administrators`

2.4 Issues

2.4.1 Issue Types

- Bug
- Task
- User Story
- Epic
- Sub-task

2.4.2 Issue Fields

- Components
 - Component Leaders
- Labels
- Links
- Assignee
- Reporter

2.4.3 Custom Fields

- kilka - kilkanaście
- Team Assigned
- Start Date (and Due Date)

2.4.4 Versions

- Roadmap
- Releases (with Bamboo)
- Konwencja nazewnictwa YYYY-MM (2017-01, 2017-02, 2017-03)
- Time Tracking Report by Version
- `affectsVersion` vs. `fixVersion`

2.4.5 Priorities

- Lower, Low, Medium, High, Highest
- Low, Medium, High, Highest, Blocker
- Urgent, Important, Standard
- MoSCoW (Must, Should, Could)
- Important, Normal
- Expedite, Standard
- Top, Normal, Bottom
- Important, Normal, Someday/Maybe
- DEFCON-1, DEFCON-2, DEFCON-3

2.4.6 Statusy

- To Do
- In Progress
- Done
- In Review
- Waiting / Blocked
- In Test

2.4.7 Resolutions

- Fixed
- Won't Fix
- Duplicate
- Cannot Reproduce
- Incomplete
- [Jira Agile] -> Done

2.4.8 Issue Actions

- Workflow Actions (Open, In Progress, Done)
- Voting
- Watching
- Add Attachments
- Clone
- Move
- Create subtask
- Delete (kiedy?)
- Log Work
- Keyboard Shortcuts

- Comment
 - Mentions
 - Rich Text Editing
 - Tworzenie tabel
 - Używanie formatowania

2.4.9 Time Reporting

- Original Time Estimate
- Remaining Time
- Log Work
- Reports

2.4.10 Assignments

Tworzenie issues

1. Ustaw ekran tworzenia zadania
 - Issue Type
 - Summary
 - Description
 - Priority
 - Attachment
 - Linked Issues
 - Assignee
 - Time Tracking
2. Załóż zadanie w projekcie
3. Do zadania dodaj załącznik
 - obrazek PNG lub JPEG
 - archiwum .zip z przynajmniej dwoma plikami tekstowymi
4. Zmień priorytet na Highest
5. Zmień Issue Type na Task
6. Powiąż dwa zadania linkami jako “is blocked by”/”blocks”
7. Sklonuj zadanie
 - z attachmentami
 - ze sprintem
8. Zadanie ma mieć trzy sub-taski
 - status pierwszego: To Do
 - status drugiego: In Progress
 - status trzeciego: Done
 - wyceń każde z zadań na 10m

9. Przenieś zadanie z projektu do innego projektu
 - nie wysyłaj informacji mailem o zmianach

2.5 JQL - JIRA Query Language

2.5.1 Where to find?

- Issues -> Search for Issues
 - Basic -> Advanced
 - Detail View -> List View

2.5.2 Where is used?

- Searching Issues
- Board Configuration
- Filters for Dashboard
- Filters for Subscriptions
- Bulk edit (to change limit: `echo 'jira.bulk.edit.limit.issue.count = 1000' >> $JIRA_HOME/jira-config.properties`)
- `jira.issue.editable = true` dla statusu Done (Workflow)

2.5.3 Operators

Table 2.6.: Operators

Operator	Description
!=	Not equal (is different than)
==	Equals
=	Equals
>	Greater than
<	Less than
<=	Greater or equal
>=	Less or equal

2.5.4 View

- Konfiguracja Kolumn wyszukiwania
- Import / Export CSV
 - All fields
 - current fields
- Limit wyświetlania wyników dla JQL (change: General Configuration -> Advanced Settings -> `jira.search.views.default.max`)

2.5.5 JQL Examples

- Operators capital letter

Select issues

```
project = DEMO
```

```
status = "To Do"
```

```
assignee = admin
```

```
reporter = my.username
```

```
summary ~ "Hello"
```

```
assignee != my.username
```

```
statusCategory = "To Do"
```

```
statusCategory != "Done"
```

```
Flagged IS NOT EMPTY
```

Complex queries

```
status = "To Do" OR status = "In Progress"
```

```
status IN ("To Do", "In Progress")
```

```
status NOT IN ("To Do", "In Progress")
```

```
statusCategory NOT IN ("To Do", "In Progress")
```

```
project = DEMO  
AND status = "To Do"
```

```
project = DEMO  
AND resolution NOT IN (Fixed, "Won't Fix")
```

Ordering

```
project = DEMO  
ORDER BY priority DESC
```

```
project = DEMO  
ORDER BY priority DESC, key ASC
```

Functions

```
currentLogin()  
lastLogin()  
now()  
startOfDay()
```

(continues on next page)

(continued from previous page)

```
startOfWeek()
startOfMonth()
startOfYear()
endOfDay()
endOfWeek()
endOfMonth()
endOfYear()
```

```
assignee = currentUser()
```

```
Sprint IN closedSprints()
```

```
Sprint IN openSprints()
```

```
Sprint IN futureSprints()
```

Queries in History

```
project = DEMO
  AND status WAS Done
  AND status != Done
```

```
project = DEMO
  AND status WAS Done
  AND status != Done
  AND updated > -1d
```

```
due >= 2017-03-01 AND due <= 2017-03-31
```

```
due >= startOfMonth() AND due <= endOfMonth()
```

```
due <= now()
  AND statusCategory != Done
```

```
status WAS NOT "In Progress" BEFORE "2011/02/02"
status WAS NOT IN ("Resolved", "In Progress") BEFORE "2011/02/02"
status WAS IN ("Resolved", "In Progress")
status WAS "Resolved" BY jsmith DURING ("2010/01/01", "2011/01/01")
status WAS "Resolved" BY jsmith BEFORE "2011/02/02"
```

```
status CHANGED BY currentUser()
```

```
AFTER "date"
BEFORE "date"
BY "username"
DURING ("date1", "date2")
ON "date"
FROM "oldvalue"
TO "newvalue"
```

```
assignee CHANGED
```

```
priority CHANGED BY freddo BEFORE endOfWeek() AFTER startOfWeek()
```

```
status CHANGED FROM "In Progress" TO "Open"
```

2.5.6 Useful Queries

My issues To Do

```
assignee = currentUser()  
  AND statusCategory = "To Do"
```

```
assignee = currentUser()  
  AND statusCategory = "To Do"  
  ORDER BY priority DESC, key ASC
```

```
project = DEMO  
  AND sprint IN openSprints()  
  AND assignee = currentUser()
```

Tracking reported issues

```
reporter = currentUser()  
  AND statusCategory != Done  
  AND assignee != currentUser()
```

Tracking team members work

```
statusCategory NOT IN (Done, "In Progress")  
  AND assignee IN membersOf("jira-administrators")
```

```
project = DEMO  
  AND updated >= -7d  
  AND assignee IN membersOf("jira-administrators")
```

```
assignee IN membersOf("jira-administrators")  
  AND updated >= startOfWeek(-7d)  
  AND updated <= endOfWeek(-7d)
```

Daily

```
project = DEMO  
  AND sprint IN openSprints()  
  AND (statusCategory = "In Progress" OR Flagged is not EMPTY)  
  AND updated >= -1d
```

2.5.7 More info

- <https://confluence.atlassian.com/jira064/advanced-searching-720416661.html>

2.5.8 Assignments

JQL i Wyszukiwanie zadań

1. wyszukaj wszystkie zadania, które są w statusie “In Progress”
2. wyszukaj zadania, które zostały zaktualizowane od wczoraj
3. wyszukaj zadania, które należą do obecnie otwartego sprintu
4. wyszukaj zadania oflagowane
5. wyszukaj zadania, które należą do osób z grupy jira-administrators
6. wyszukaj zadania, które były przypisane do Ciebie, ale już nie są
7. Wyszukaj wszystkie zadania zaktualizowane przez Ciebie w okresie ostatniego tygodnia
 - Pokaż kolumny: Priority, Key, Summary, Original Time Estimate, fixVersion, Epic Name, Status

2.6 Filtry

- Tworzenie
- Subskrybcja
- Uprawnienia
 - Przydział do ról
 - Przydział do grup
 - Publiczny
- Współdzielenie

2.6.1 Assignments

- Stwórz filtr “Daily”
- Stwórz filtr “Przekroczony Deadline”, ustaw uprawnienia by był widoczny dla administratorów w projekcie
- Stwórz filtr “Praca mojego zespołu z ostatniego tygodnia”, ustaw by przychodził mail z zadaniami w poniedziałki o 6 rano

2.7 Dashboard

- Edit Layout
- Share
- Tworzenie
- Publikacja
- Dodawanie gadżetów
 - Filter Results
 - Issue Statistics
 - Average Age Chart
 - Resolution Time
- Wallboard plugin

- Tables
- Graphs
- Piecharts
- Jira Agile Reports
 - Sprint Health Report
 - Burndown
 - Days Remaining

2.7.1 Assignments

1. Stwórz dashboard z trzema kolumnami
2. Pierwsza kolumna:
 - Created vs. Resolved
 - Average Age Chart
3. Druga kolumna:
 - Version Report
 - Issue statistics (Priority)
4. Trzecia kolumna:
 - Sprint Health Gadget
 - Days remaining in Sprint Gadget
 - Sprint Burndown

2.8 Jira Software (Agile)

2.8.1 Project Management

- Prowadzenie projektów
- Kanban
- Scrum
- Portfolio
- Scrum + Kanban

2.8.2 Artifacts

- Backlog
- Sprintlog
- Task board
- Units:
 - Story Points
 - Business Value

2.8.3 Epic

- Brak worków (np. Poprawki błędów)
- Doważalne (określone w czasie, mają datę początku i końca)
- Dobre praktyki:
 - Due Date
 - Start Date
 - Assignee
- Doważalne
- optymalna długość
- kategoryzowanie
- timeline i roadmapa
- planowanie kwartalne
- przypisywanie epików do wersji
- board epików
- Business Value epików

2.8.4 Estimation

- Time Estimate
- Manday
- Story Point
- Business Value
- #NoEstimates and Monte Carlo simulation (<https://www.infoq.com/presentations/monte-carlo>)

2.8.5 Metrics

- Velocity
- Capacity
- Maturity

2.8.6 Planning and Refinement

- Estimation
- How big your tasks should be?
- Estimation support systems
- Sprint goal
- Acceptance Criteria
- Definition of Done
- Time Tracking

2.8.7 Dobre praktyki

- Kryteria akceptacyjne
- INFO
- BEFORE
- TODO
- AFTER
- używanie (/) i (x)

2.8.8 Board

- Scrum vs. Kanban
 - Scrum -> Rozwój (Story)
 - Kanban -> Utrzymanie (Task)
 - Praca w Scrum i Kanban jednocześnie
 - Konstytucja zespołu i dobre praktyki
- Board vs. Project
 - Board z wielu projektów
 - Board z części jednego projektu
 - Board dla Projektu
 - Wiele boardów do jednego projektu (różne estymaty)
 - Wiele projektów czy wiele boardów (np. po komponentach)?
- Sprints:
 - Wielkość (ilość zadań, capacity chart)
 - Długość (tydzień)
 - Konwencja nazewnicza (YYYY-MM week W) (2017-03 week 2, 2017-03 week 3)
- Uprawnienia
- Konfiguracja
- Kolumny
 - Column Constraint (max, min)
 - Dodawanie i usuwanie kolumn
 - Wiele statusów w jednej kolumnie
 - Statusy ciągnące pracę
- Swimlines
 - wg. priorytetów
 - wg. wersji
- Quick Filters
- Card Colors
- Card Layout
 - Backlog

- Active Sprint
- Days in Column
- Estimation
 - Original Estimate + Remaining Estimate and Time Spent
 - Story Points
 - Business Value
 - Issue Count
- Working Days
- Issue Detail View
- Portfolio na bazie Kanbana
- Scope Changes
- Otwieranie i zamykanie sprintów
- Auto assign
- Flagowanie zadań
- Quick Filters dla Daily

2.8.9 Charts

- Burn-down Chart
- Burn-up Chart
- Control Chart
- Cumulative Flow Diagram
- Epic Burndown
- Epic Report
- Release Burndown
- Sprint Report
- Velocity Chart
- Version Report
- Version Burndown
- Refine Reports

2.8.10 Kanban

- What's Kanban?
- Pull system
- JIT
- Context switching
- Kanban Board
- Improvement:
 - Muda

- Jidoka
- Kaizen
- Bottlenecks
- Metrics
- Lean
- Workflow:
 - Columns
 - Swimlanes
 - Expedite
 - Priority
 - SLA

2.8.11 Assignments

Board

1. Stwórz Board dla zadań rozwojowych (Story, Bug):
 - Dodaj kolumnę `In Test` oraz `In Review` wraz z odpowiadającymi im statusami
 - Dodaj status `Won't Do`, który będzie w kolumnie `Done` jednocześnie ze statusem `Done`
 - Stwórz Quick Filter `Daily`:
 - zadania są w trakcie wykonywania
 - zaktualizowane w ciągu ostatniego dnia
 - lub mają flagę
 - Stwórz wersję board z Estymacją `Time Estimate`
 - Stwórz wersję board z Estymacją w `Story Points`
2. Stwórz Board dla zadań utrzymaniowych (Task)
 - Kolumny: `To Do`, `In Progress`, `Blocked`, `Done`
 - Dodaj status `Won't Do`, który będzie w kolumnie `Done` jednocześnie ze statusem `Done`
3. Stwórz board Kanban z Epikami:
 - Stwórz swimline dla kwartałów
 - Określ aby w kolumnie "In Progress" mogły być maksymalnie 3 Epiku
4. Stwórz board zadań przypisanych do Ciebie:
 - zadania mogą być w dowolnym projekcie
 - board ma być publiczny

Backlog i Estymacja

- Stwórz epiki
 - Logowanie
 - Panel administracyjny
- oszacuj zadania używając Story Points i skali S,M,L (Small: 1, Medium: 2, Large: 3)

- Zadanie wyestymuj na 4h
- Zaloguj 1h 30m do zadania i ustaw remaining na 3h

Wersje

- Stwórz wersje
 - 2019-01 (rozpoczęcie: 1 styczeń 2019; zakończenie: 31 styczeń 2019)
 - 2019-02 (rozpoczęcie: 1 luty 2019; zakończenie: 28 luty 2019)
 - 2019-03 (rozpoczęcie: 1 marzec 2019; zakończenie: 31 marzec 2019)
- Zadania przydziel do wersji

Sprinty

- Stwórz Sprinty
 - 2019-01 week 1 (ma 4 Story Points)
 - 2019-01 week 2 (ma 10 Story Points)
 - 2019-01 week 3 (ma 8 Story Points)
 - 2019-01 week 4 (ma 10 Story Points)
 - 2019-02 week 5 (ma 8 Story Points)
- Wystartuj sprint 2019-01 week 1
 - Data rozpoczęcia 1 styczeń 2019, 9:00
 - Data zakończenia 7 styczeń 2017, 9:00
- Przenieś dwa zadania do “In progress”
- Przenieś jedno zadanie do “Done”
- Zamknij sprint
- Zadania które nie zostały zakończone w sprincie niech spadną do następnego tygodnia
 - Co się dzieje z otwartymi zadaniami?
 - Co się dzieje z zamkniętymi zadaniami?
 - Co się dzieje z zamkniętymi subtaskami, ale otwartym zadaniem?
 - Co się dzieje z otwartymi subtaskami ale zamkniętym zadaniem?
- Zobacz raporty

3.1 Configuration

3.1.1 Basic Configuration

- Skrót klawiszowy gg
- Estymacja różnych issuetype (nie tylko Story)
- Re-index
- Application Links
- Zmiana formatu daty

Key	Value	
jira.clone.prefix		<input type="button" value="Revert"/>
jira.date.picker.java.format <small>This part is only for the Java (server side) generated dates. Note that this should correspond to the javascript date picker format (jira.date.picker.javascript.format) setting.</small>	yyyy-MM-dd	<input type="button" value="Revert"/>
jira.date.picker.javascript.format <small>This part is only for the JavaScript (client side) generated dates. Note that this should correspond to the java date picker format (jira.date.picker.java.format) setting.</small>	%Y-%m-%d	<input type="button" value="Revert"/>
jira.date.time.picker.java.format <small>This part is only for the Java (server side) generated datetimes. Note that this should correspond to the javascript datetime picker format (jira.date.time.picker.javascript.format) setting.</small>	yyyy-MM-dd HH:mm	<input type="button" value="Revert"/>
jira.date.time.picker.javascript.format <small>This part is only for the JavaScript (client side) generated date times. Note that this should correspond to the java datetime picker format (jira.date.time.picker.java.format) setting.</small>	%Y-%m-%d %H:%M	<input type="button" value="Revert"/>

Figure 3.1.: Zmiana formatu daty w zaawansowanych opcjach konfiguracyjnych

3.1.2 General Configuration

System

- General configuration

Troubleshooting and support

- Audit Log

Security

- Project roles
- Global permissions
- Issue collectors

User Interface

- Default user preferences
- System dashboard
- Look and feel

Import and Export

- Backup manager
- External System Import
- Restore system

MAIL

- Global Mail Settings
- Outgoing Mail
- Incoming Mail
- Send email

Admin Helper

- Permission helper
- Notification helper

Shared Items

- Shared filters
- Shared dashboards

Advanced

- Attachments
- Events
- WebHooks
- Services
- LexoRank management

3.1.3 Assignments

Podstawowa Administracja

1. Wyłącz obsługę gravatar
2. Włącz obsługę Attachmentów
3. Ustaw maksymalny rozmiar attachmentów na 100 MB

3.2 License

3.2.1 Cloud vs. Server

- Dokumentacja [1]
- Jira Core vs. Software vs. Service Desk

3.2.2 Poziomy licencji

- Ilość użytkowników
- Evaluation license

3.2.3 Okres licencji

- Długość trwania licencji
- Co po zakończeniu okresu?
- Zarządzanie licencjami

3.3 Issue Types

3.3.1 Issue Types

3.3.2 Issue Type Schemes

3.3.3 Sub-tasks

- Czy stosować?
- Kiedy stosować?

3.4 Workflow

3.4.1 Dobre praktyki

- Directed graph
- Complete graph
- Few vertices
- Lots of Edges
- Try simple and add statuses
- Keep transitions from all statuses
- Simplified Workflow

3.4.2 Workflow Functions

Triggers

Bitbucket

- Pull request created
- Pull request merged
- Pull request declined
- Pull request reopened
- Branch created
- Commit created

Bamboo

- Deployment successful
- Deployment failed

Crucible

- Review started
- Review abandoned
- Review submitted for approval
- Review closed
- Review rejected
- Review summarized

Condition

Table 3.1.: Condition

Name	Description
Always False Condition	This condition always fails
Block transition until approval	Condition to block issue transition if there is a pending approval
Compare Number Custom Field	Condition to allow transition if a comparison of specified Number Custom Field to a specified value is true
Hide From User Condition	Condition to hide a transition from the user. The transition can only be triggered from a workflow function or from REST
Only Assignee Condition	Condition to allow only the assignee to execute a transition
Only Bamboo Notifications Workflow Condition	Only makes this transition available to the Bamboo build notifications
Only Reporter Condition	Condition to allow only the reporter to execute a transition
Permission Condition	Condition to allow only users with a certain permission to execute a transition
Previous Status Condition	Condition to check if the issue has transitioned through a specified status or no
Separation of Duties condition	Condition preventing a user to perform the transition, if the user has already performed a transition on the issue
Sub-Task Blocking Condition	Condition to block parent issue transition depending on sub-task status
User Is In Any Group	Condition to allow only users in a given group to execute a transition
User Is In Any Project Role	Condition to allow only users in a given project role to execute a transition
User Is In	Custom field Allows only users in a given custom field to execute the transition
User Is In Group	Condition to allow only users in a given group to execute a transition
User Is In Group Custom Field	Condition to allow only users in a custom field-specified group to execute a transition
User Is In Project Role	Condition to allow only users in a given project role to execute a transition
Value Field	Allows to execute a transition if the given value of a field is equal to a constant value, or simply set

Validators

Table 3.2.: Validators

Name	Description
Date Compare Validator	Compare two dates during a workflow transition
Date Window Validator	Compares two date fields, by adding a time span in days to one of them
Field Required Validator	Field must not be empty during the transition
Field has been modified Validator	Field value must be changed during the transition
Field has single value Validator	Multi-select Field has not more than one value during transition
Parent Status Validator	Validates that the parent issue is in required state
Permission Validator	Validates that the user has a permission
Previous State Validator	Validates that the issue has previously transitioned through a specific state
Regular Expression Check	Validate field contents against a regular expression during a workflow transition
User Permission Validator	Validates that the user has a permission, where the OSWorkflow variable holding the username is configurable. Obsolete

Post Functions

Table 3.3.: Post Functions

Name	Description
Assign to Current User	Assigns the issue to the current user if the current user has the 'Assignable User' permission
Assign to Lead Developer	Assigns the issue to the project/component lead developer
Assign to Reporter	Assigns the issue to the reporter
Clear Field Value	Clear value of a given field
Copy Value From Other Field	Copies the value of one field to another, either within the same issue or from parent to sub-task
Create Crucible Review Workflow Function	Creates a Crucible review for all unreviewed code for this issue
Notify HipChat	Send a notification to one or more HipChat rooms
Set issue security level based on user's project role	Set the issue's Security Level to the specified level if the current user is in a specified Project Role
Trigger a Webhook	If this post-function is executed, Jira will post the issue content in JSON format to the URL specified
Update Issue Custom Field	Updates an issue custom field to a given value
Update Issue Field	Updates a simple issue field to a given value

3.4.3 Workflow Schemes

3.4.4 Assignments

Workflow

1. Dodaj do workflow status In Review, Blocked, In Test
2. Przy przenoszeniu do statusu Done ma wyświetlać się okienko z logownaiem czasu
3. Przy przenoszeniu do statusu Blocked ma wyświetlać się okienko z komentarzem (przyczyna zablokowania)

3.5 Screens

3.5.1 Screens

- Definiowanie
- Podłączanie do workflow
- Przypisywanie custom fields
- Jak dodać sreen do statusu w workflow?

3.5.2 Screen Schemes

- Screen dla edycji
- Screen dla view
- Rozróżnienie screenów jest deprecated

3.5.3 Issue Type Screen Scheme

3.6 Fields

3.6.1 Custom Fields

- Dobre praktyki
- Ile?
- Konsekwencje
- CF w bazie dancyh
- Javascript w opisie (nie używać)

3.6.2 Field Configuration

3.6.3 Field Configuration Schemes

3.6.4 Assignments

Custom Field

1. Stwórz Custom Field "People Assigned":
 - W polu mamy mieć możliwość przypisywania wielu użytkowników do zadania
 - Pole dodaj ekranu dla zadań w projekcie
 - Stwórz filtr który wyszuka zadania w których jesteś wymieniony w tym Custom Field
 - Na podstawie filtru stwórz tablicę Kanban, z zadaniami które są do Ciebie przypisane w tym Custom Fieldzie
 - Pole ma wyświetlać się w widoku Backlog w kolumnie po prawej stronie
 - Podpowiedź: typ `User Picker (Multiple Users)`
2. Stwórz Custom Field "Team Assigned":
 - Dodaj 4 zespoły: Team A, Team B, Team C, Team D
 - Można wybrać więcej niż jeden zespół
 - Pole dodaj ekranu dla zadań w projekcie
 - Pole ma być wymagane przy tworzeniu nowego zadania
 - Podpowiedź: typ `Checkbox`
3. Stwórz Custom Field "Manhours":
 - Pole dodaj ekranu dla zadań w projekcie
 - Stwórz nowy board do projektu z estymacją w Manhours
 - Stwórz filtr, który wyciągnie wszystkie zadania z projektu
 - Na filtrze mają być kolumny: Key, Summary, Original Time Estimate, Manhours, Status
 - Podpowiedź: typ `Number`

3.7 Issue Features

3.7.1 Time Tracking

- Time tracking provider
- Working hours per day
- Working days per week
- Time display format
- Default unit for time tracking
- Copying of comments to work description

3.7.2 Issue Linking

3.8 Issue Attributes

3.8.1 Statuses

Good practises

- Ile statusów w Workflow?
- Jakie nazwy statusów?
- Statusy ciągnące
- Ile statusów w Jirze?
- Jak to jest w wersji Cloud?

Proponowane nazwy

- Code Review
- Manual Testing
- PO Accept?

Status Category

- To Do
- In Progress
- Done

Terminal Statuses

- Closed vs Resolved vs Done

3.8.2 Resolutions

3.8.3 Priorities

- Ile?

Jakie nazwy?

- Lower, Low, Medium, High, Highest
- Low, Medium, High, Highest, Blocker
- Urgent, Important, Standard
- MoSCoW (Must, Should, Could)
- Important, Normal
- Expedite, Standard
- Top, Normal, Bottom
- Important, Normal, Someday/Maybe
- DEFCON-1, DEFCON-2, DEFCON-3

3.8.4 Issue security schemes

- Konfigurowanie widoku per zadanie

3.8.5 Notification schemes

- Dodawanie powiadomień
- Konfiguracja

3.8.6 Permission schemes

- Role projektowe
- Grupy vs. Role
- Grupy w AD
- Global Permissions

3.8.7 Assignments

Priorytety

1. Zmień priorytety na MoSCoW, zmień ikony i kolory (czerwony, zielony, szary)

Role

1. Dodaj rolę "Scrum Master"
2. Dodaj do roli w projekcie użytkownika z JIRY
3. Zmień w Permission scheme, aby tylko Scrum Master mógł otwierać i zamykać sprinty

3.9 User Management

3.9.1 Konfiguracja

- Go to Jira User Server (g+g and type JIRA User Server)
- Add application
- Set application name, password and IP Addresses (paste addresses from instances which you want connect with Jira User Server)

3.9.2 Dobre praktyki

- Always use LDAP (OpenLDAP or Active Directory)
- name groups as `jira-users` or `jira-administrators`
- local administrator `jira-administrator` only for fixing bugs with LDAP
- use `jira@example.com` (for easy email filterling)
- use `jira.example.com` as domain name with Firewall blocking external access
- `/etc/resolv.conf search example.com -> ustawianie przez DHCP`
- Internal and external users in one LDAP server
- Read only access via LDAPs
- avoid nested groups

- all tools in OU=ecosystem
- use LDAP groups for project roles from OU=projects
- do not use user accounts in project roles (only LDAP groups)
- Confluence page with all *-administrators + mailto: links
- Confluence page with JIRA project administrators
- Do not use technical accounts (use SSH keys)
- Use SSH keys with proper comment

3.10 Apps

3.10.1 Atlassian Marketplace

- Find new apps
- Manage apps

Licencje pluginów

- Darmowe
- Płatne
- Support
- Atlassian Verified

Cloud vs. Server

- Atlassian Connect
- p2

3.10.2 Dobre praktyki

- Kiedy instalować?
- Czy zgadzać się na wszystkie pluginy?

Strategia update'ów

- pluginy darmowe
- pluginy komercyjne

3.10.3 Pluginy i dane

Wykorzystywane zasoby

- Pamięć RAM
- Baza danych
- System operacyjny

- Zasoby sieciowe

Baza danych

- Tabelki
- Architektura
- Dirty hacki

3.10.4 Ciekawe Addony

- Jira Agile Cards
- Instalacja dodatkowych języków

Portfolio for Jira

- Portfolio Permissions
- Portfolio License
- Portfolio Hierarchy Configuration
- Portfolio Dependencies
- Portfolio vs. Roadmap in Jira Cloud

3.11 Dirty hacks

3.11.1 Manipulacje na bazie

3.11.2 Django ORM

Django Inspect DB + Jira = Django ORM

3.11.3 Skryptowanie

3.11.4 Time tracking

3.11.5 Atlassian Python API

- <https://github.com/atlassian-api/atlassian-python-api>
- `pip install atlassian-python-api`

Reindex

Listing 3.1: Jira reindex

```
from pprint import pprint
from atlassian import Jira

jira = Jira(
```

(continues on next page)

(continued from previous page)

```

url="http://localhost:8000/",
username="admin",
password="admin")

status = jira.reindex().json()
pprint(status)

```

Project Administrators

Listing 3.2: Jira Project Administrators

```

import logging
from atlassian import Confluence
from atlassian import Jira

logging.basicConfig(level=logging.DEBUG, format="[% (asctime)s] [% (levelname)s]
↳ %(message)s")
logging.getLogger("requests").setLevel(logging.WARNING)
log = logging.getLogger("jira-projects-administrators")

jira = Jira(
    url="http://localhost:8000/",
    username="admin",
    password="admin")

html = "<table><tr><th>Project Key</th><th>Project Name</th><th>Leader</th><th>
↳ Email</th></tr>"

for data in jira.project_leaders():
    log.info("{project_key} leader is {lead_name} <{lead_email}>".format(**data))
    html += "<tr><td>{project_key}</td><td>{project_name}<td></td>{lead_name}<td></
↳ td><a href='mailto:{lead_email}'>{lead_email}</a></td></tr>".format(**data)

html += "</table><p></p><p></p>"
html += "<p>Autogenerated with <a href='http://localhost:7999/projects/AGILE/repos/
↳ devops-utils-jira/browse/bin/jira-projects-administrators.py'>this script</a></p>
↳ "

confluence = Confluence(
    url="http://localhost:8090/",
    username="admin",
    password="admin")

confluence.update_page(
    page_id=13207798,
    parent_id=7471197,
    title="Administratorzy JIRA",
    body=html)

log.info("Confluence Page Created with JRIA Administrators at: http://
↳ localhost:8095/pages/viewpage.action?pageId=13207798")

```

3.11.6 Assignments

Atlassian Python API - Instalacja

1. Zainstaluj bibliotekę Atlassian Python API `atlassian-python-api`

Note: Kod biblioteki dostępny jest na GitHub <https://github.com/atlassian-api/atlassian-python-api>

Warning: Wymagany Python 3.4 lub nowszy

Atlassian Python API - Reindeksacja

1. Stwórz skrypt `jira-reindex.py`
2. Skrypt wykorzystując bibliotekę `atlassian-python-api` ma reindeksować JIRE
3. Skrypt `jira-reindex.py` dodaj Crontab by był uruchamiany o 4 w nocy
4. Pamiętaj, że cron ma inne zmienne środowiskowe

4.1 Instalacja

4.1.1 Instalacja

- skąd pobrać (<https://www.atlassian.com/software/jira/downloads/binary/atlassian-jira-software-X.X.X-x64.bin>)
- jakie polecenia
- forwarding portów
- SSL proxy

4.1.2 Database

Jaka baza danych?

- Wspierane: PostgreSQL, MySQL, Oracle, MSSQL
- Preferowana: PostgreSQL

Instalacja PostgreSQL na CentOS

```
yum install postgresql-server
postgresql-setup initdb
systemctl start postgresql
```

Instalacja PostgreSQL na Ubuntu

```
# Ubuntu / Debian
$ apt-get install postgresql
```

Tworzenie bazy danych

- Będąc zalogowanym jako root wykonaj:

```
su postgres
psql
```

```
CREATE USER jira WITH PASSWORD 'jira';
CREATE DATABASE jira;
GRANT ALL PRIVILEGES ON DATABASE jira TO jira;
```

Konfiguracja połączenia z bazą danych

- Konieczna jest modyfikacja pliku `pg_hba.conf` aby można było łączyć się z localhost (po IPv4 i IPv6) za pomocą hasła (md5)
- Ubuntu: `/etc/postgresql/9.5/main/pg_hba.conf`.
- CentOS: `/var/lib/pgsql/data/pg_hba.conf`

```
# "local" is for Unix domain socket connections only
local all all peer
# IPv4 local connections:
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all ::1/128 md5
```

Restart bazy danych

```
$ systemctl restart postgresql
```

4.1.3 Jira install

Listing 4.1: Jira install

```
VERSION='7.13.1'
wget https://www.atlassian.com/software/jira/downloads/binary/atlassian-jira-
↪software-${VERSION}-x64.bin
chmod +x atlassian-jira-software-${VERSION}-x64.bin
./atlassian-jira-software-${VERSION}-x64.bin
rm -fr atlassian-jira-software-${VERSION}-x64.bin
```

Environment

1. Poniższych edycji dokonujemy w pliku `atlassian-jira-XXX/bin/setenv.sh` gdzie XXX to numer wersji (nowej)

```
JIRA_HOME="/opt/jira/home"
JVM_SUPPORT_RECOMMENDED_ARGS="-server -XX:MaxPermSize=512m -XX:+UseG1GC -
↪XX:MaxGCHeapFreePercentage=200 -XX:+PrintGC -XX:+PrintGCDateStamps -
↪XX:+OptimizeStringConcat -XX:+PrintGCDetails -XX:+DisableExplicitGC -Xloggc:/opt/
↪jira/logs/gc-jira-$(hostname)-$(date +%Y.%m.%d).log -XX:+UseGCLogFileRotation -
↪XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=10M"
JVM_MINIMUM_MEMORY="512m"
JVM_MAXIMUM_MEMORY="2048m"
```

Firewall

```
# CentOS
$ firewall-cmd --zone=public --add-port=8080/tcp --permanet
$ firewall-cmd --zone=public --add-port=5432/tcp --permanet
$ firewall-cmd --reload

# Other Linux
$ iptables -I INPUT 1 -i eth0 -p tcp --dport 8080 -j ACCEPT
$ iptables -I INPUT 1 -i eth0 -p tcp --dport 5432 -j ACCEPT
```

4.1.4 Configuration

Websudo

- automatic admin logout
- admin rights notification

```
service jira stop
echo "jira.websudo.is.disabled = true" >> /var/atlassian/application-data/jira/
↪jira-config.properties
service jira start
```

4.1.5 Assignments

Install Jira

1. Zainstaluj Jirę z licencją evaluation (wykorzystaj 10 minute email * drugi wynik w Google)
2. Utwórz projekt Moon Village (klucz: MOON) z przykładowymi danymi

4.2 Upgrade

4.2.1 Instalacja nowej wersji

1. Wejdź na stronę <https://www.atlassian.com/software/jira/download>
2. Kliknij prawym na przycisk Download obok wydania Jira TAR.GZ i “Copy Link Location”
3. Uruchom polecenia poniżej:

```
# Tu wklej zawartość linku
$ URL=""

$ ssh root@localhost
$ cd /opt/jira
$ wget "$URL" -O jira.tgz
$ tar xzf jira.tgz
$ rm -fr jira.tgz
```

4.2.2 Ustawienia środowiskowe

1. Poniższych edycji dokonujemy w pliku `atlassian-jira-XXX/bin/setenv.sh` gdzie XXX to numer wersji (nowej)

```
JIRA_HOME="/opt/jira/home"
JVM_SUPPORT_RECOMMENDED_ARGS="-server -XX:MaxPermSize=512m -XX:+UseG1GC -
↪XX:MaxGCPauseMillis=200 -XX:+PrintGC -XX:+PrintGCDateStamps -
↪XX:+OptimizeStringConcat -XX:+PrintGCDetails -XX:+DisableExplicitGC -Xloggc:/opt/
↪jira/logs/gc-jira-$(hostname)-$(date +%Y.%m.%d).log -XX:+UseGCLogFileRotation -
↪XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=10M"
JVM_MINIMUM_MEMORY="512m"
JVM_MAXIMUM_MEMORY="2048m"
```

4.2.3 Zmiana portu działania Jiry

1. Edytuj linijkę `/opt/jira/install/conf/server.xml` i znajdź

```
Connector port="8080"
```

2. Zamień na:

```
Connector port="8000"
```

4.2.4 Sprawdzenie wersji Javy dla Jiry

1. Odpal poniższe polecenie

```
/opt/java/default/bin/java -version
```

2. Zobacz aktualną wersję na <http://www.oracle.com/technetwork/java/javase/downloads/1880261>
3. Jeżeli wersja się różni ściągnij za pomocą `wget` nową Javę do `/opt/java/`
4. rozpakuj i usuń symlink `/opt/java/default`
5. stwórz symlink `/opt/java/default` wskazujący na nową Javę

4.2.5 Backup bazy oraz home'a

1. Odpal skrypt `/opt/jira/backup-jira.sh`

4.2.6 Upgrade Jiry

```
$ service jira stop
$ rm -fr /opt/jira/install
$ ln -s /opt/jira/atlassian-jira-XXX
$ /opt/jira/install
$ service jira start
```

4.2.7 Sprzątanie

1. Możesz usunąć stary katalog instalacyjny Jiry.
2. Proponuję jednak zostawić jedną, poprzednią wersję tak na wszelki wypadek, gdyby jakieś zmiany nie zostały przeniesione.

4.3 Backup

4.3.1 Procedure

- XML (<http://localhost:8080/secure/admin/XmlBackup!default.jspa>)
- rsync:
 - JIRA_HOME="/var/atlassian/application-data/jira"
 - JIRA_INSTALL="/opt/atlassian/jira/"
 - database replication
 - pg_dump i pg_restore
- database replication consistency and rsync while upgrading
- /var/atlassian/application-data/jira/.jira-home.lock
- Cold standby in alternative datacenter
- database replication between datacenter
- cold standby and licensing (same SEN number)

4.3.2 Script

Listing 4.2: Jira backup

```
#!/bin/bash

echo "[$(date)] Reading database configuration..."
USER=$(cat /opt/jira/home/dbconfig.xml |xmllint --xpath '//username/text()' -)
export PGPASSWORD=$(cat /opt/jira/home/dbconfig.xml |xmllint --xpath '//password/
↪text()' -)
HOST=$(cat /opt/jira/home/dbconfig.xml |xmllint --xpath '//url/text()' - |awk -F'/
↪' '{print $3}' |awk -F':' '{print $1}')
PORT=$(cat /opt/jira/home/dbconfig.xml |xmllint --xpath '//url/text()' - |awk -F'/
↪' '{print $3}' |awk -F':' '{print $2}')
DATABASE=$(cat /opt/jira/home/dbconfig.xml |xmllint --xpath '//url/text()' - |awk -
↪F'/' '{print $4}')

echo "[$(date)] Database connection settings..."
echo "User: $USER"
echo "Pass: *****" #PGPASSWORD"
echo "Host: $HOST"
echo "Port: $PORT"
echo "Database: $DATABASE"

echo "[$(date)] Cleanup previous backups..."
rm -fr /opt/jira/backup/*

echo "[$(date)] Backup database..."
pg_dump -h $HOST -p $PORT -U $USER $DATABASE |gzip > /opt/jira/backup/jira-
↪database_$(date +%F).tar.gz

echo "[$(date)] Backup home directory..."
tar -jcf /opt/jira/backup/jira-home_$(date +%F).tar.bz2 --exclude-caches-all /opt/
↪jira/home

echo "[$(date)] All done."
```

4.3.3 Assignments

Administracja - backup

1. Zrób backup `$JIRA_HOME` i `$JIRA_INSTALL` wykorzystując `tar.gz` (musi być data w nazwie pliku)
2. Wylistuj pliki w archiwum (możesz przeglądać za pomocą `midnight commander`)
3. Usuń katalogi `$JIRA_HOME` i `$JIRA_INSTALL`
4. Przywróć oba katalogi do:
 - `/opt/jira/home`
 - `/opt/jira/install`
5. Podmienić skrypty startowe
6. Uruchom Jirę z nowej lokalizacji
7. Dodaj polecenie backupu `$JIRA_HOME` i `$JIRA_INSTALL` do `crontab` z `@midnight`

4.4 Migrating

4.4.1 Migracja danych

Listing 4.3: Jira Migrate

```
#!/usr/bin/env python2

import httpplib
import ldap
import urllib
import logging
import re

"""
Basic configuration
"""

class Config(object):
    host = 'localhost:8000'
    token = '...'
    jsessionid = '...'
    entities_file = '/Users/matt/Projects/clitools/entities.xml'
    entities_attrs = ['author', 'caller', 'lead', 'name', 'newstring', 'reporter',
↪ 'roletypeparameter', 'updateauthor', 'user', 'username', 'userName',
↪ 'lowerUserName', 'childName', 'lowerChildName']
    ldap_addr = 'ldap://localhost:3268'
    ldap_user = '...'
    ldap_pass = '...'
    ldap_dc = '...'
    fallback_domains = []

class DoNotDelete(object):
    groups = ["jira-administrators", "jira-users"]
    custom_fields = [11152, 10507, 11210, 10068, 11240, 10773, 11190, 10086, 10105,
↪ 10797, 11063, 11064, 10356]
    projects = [10605, 11600, 10607, 10800, 10600]
    permission_schemes = [0]
```

(continues on next page)

(continued from previous page)

```

workflow_schemes = []
workflows = []
issue_types = [1, 2, 3, 4, 5, 9, 107, 146, 126, 127, 128, 157, 237]
issue_type_schemes = []
statuses = []
notification_schemes = []
project_roles = [10002, 10001, 10000]

"""
You shouldn't change anything below this point,
unless you know what are you doing.
"""

logging.basicConfig(
    level=logging.INFO,
    format='[%asctime].19s] %(levelname)s: %(message)s'
)

class Http(object):
    """ Http request class """

    @staticmethod
    def GET(url, params={}):
        params["atl_token"] = Config.token
        params = urllib.urlencode(params)
        return Http._request("GET", "%s?%s" % (url, params))

    @staticmethod
    def POST(url, params={}):
        params["atl_token"] = Config.token
        params["Delete"] = "Delete"
        params["confirmedDelete"] = "true"
        params["workflowMode"] = "live"
        params["confirm"] = "true"
        params["confirmed"] = "true"
        return Http._request("POST", url, params)

    @staticmethod
    def _request(method, url, params={}):
        params = urllib.urlencode(params)
        headers = {
            "Cookie": "atlassian.xsrf.token=%s; JSESSIONID=%s" % (Config.token,
↪Config.jsessionId),
            "Content-Type": "application/x-www-form-urlencoded",
        }
        conn = httplib.HTTPConnection(Config.host)
        logging.debug("curl -X %(method)s -d '%(params)s' --cookie '%(cookie)s' ↪
↪http://%(host)s%(path)s" % {
            'method': method,
            'params': params,
            'cookie': headers['Cookie'],
            'host': Config.host,
            'path': url,
        })
        conn.request(method, url, params, headers)
        response = conn.getresponse()
        logging.debug("%s %s" % (response.status, response.reason))
        ret = response.read()
        response.close()
        return ret

```

(continues on next page)

(continued from previous page)

```

class Delete(object):
    """ Delete Abstract Class """

    pretty_name = None
    list_url = None
    list_re = None
    safe_data = []
    delete_url = None
    delete_param = "id"

    def __init__(self):
        if not self.pretty_name:
            self.pretty_name = self.__class__.__name__

        if self.__class__.__name__ == "Delete":
            raise NotImplementedError

        logging.warning("Deleting %ss" % self.pretty_name)

    def get_delete_data(self):
        html = Http.GET(self.list_url, {"start": 0, "max": 10000})
        matches = re.findall(self.list_re, html)

        try:
            if isinstance(matches[0], tuple):
                matches = [id for string, id in matches]
        except IndexError:
            print("Not authorized or no entries.")

        def clean(matches):
            matches = [urllib.unquote(name).decode('utf8') for name in matches]
            matches = [name.replace('+', ' ') for name in matches]
            return matches

        return clean(matches)

    def run(self):
        for id in self.get_delete_data():
            if str(id) not in [str(x) for x in self.safe_data]:
                logging.info("Deleting %s: %s" % (self.pretty_name, id))
                Http.POST(self.delete_url, {self.delete_param: id})

"""
Migration procedures
"""

class DeleteGroups(Delete):
    pretty_name = "Group"
    list_url = "/GroupBrowser.jspa"
    list_re = r'DeleteGroup!default.jspa(.*);name=(.*)"'
    safe_data = DoNotDelete.groups
    delete_url = "/DeleteGroup.jspa"
    delete_param = "name"

class DeleteCustomFields(Delete):
    pretty_name = "Custom Field"
    list_url = "/ViewCustomFields.jspa"
    list_re = r'DeleteCustomField!default.jspa\?(.*)id=([0-9]*)"'
    safe_data = DoNotDelete.custom_fields

```

(continues on next page)

(continued from previous page)

```

delete_url = "/DeleteCustomField.jspa"

class DeletePermissionSchemes (Delete):
    pretty_name = "Permission Scheme"
    list_url = "/ViewPermissionSchemes.jspa"
    list_re = r'DeletePermissionScheme!default.jspa\?schemeId=([0-9]*)'
    safe_data = DoNotDelete.permission_schemes
    delete_url = "/DeletePermissionScheme.jspa"
    delete_param = "schemeId"

class DeleteNotificationSchemes (Delete):
    pretty_name = "Notification Scheme"
    list_url = "/ViewNotificationSchemes.jspa"
    list_re = r'DeleteNotificationScheme!default.jspa\?(.*)schemeId=([0-9]*)'
    safe_data = DoNotDelete.notification_schemes
    delete_url = "/DeleteNotificationScheme.jspa"
    delete_param = "schemeId"

class DeleteOutgoingMailServers (Delete):
    pretty_name = "Outgoing Mail Sever"
    list_url = "/OutgoingMailServers.jspa"
    list_re = r'DeleteMailServer!default.jspa\?id=([0-9]*)'
    delete_url = "/DeleteMailServer.jspa"

class DeleteProjectRoles (Delete):
    pretty_name = "Project Role"
    list_url = "/ViewProjectRoles.jspa"
    list_re = r'DeleteProjectRole!default.jspa\?id=([0-9]*)'
    safe_data = DoNotDelete.project_roles
    delete_url = "/DeleteProjectRole.jspa"

class DeleteProjects (Delete):
    pretty_name = "Projects"
    list_url = "/ViewProjects.jspa"
    list_re = r'DeleteProject!default.jspa\?pid=([0-9]*)'
    safe_data = DoNotDelete.projects
    delete_url = "/DeleteProject.jspa"
    delete_param = "pid"

class DeleteIssueTypes (Delete):
    pretty_name = "Issue Type"
    list_url = "/ViewIssueTypes.jspa"
    list_re = r'DeleteIssueType!default.jspa\?id=([0-9]*)'
    safe_data = DoNotDelete.issue_types
    delete_url = "/DeleteIssueType.jspa"

class DeleteIssueTypeSchemes (Delete):
    pretty_name = "Issue Type Scheme"
    list_url = "/ManageIssueTypeSchemes.jspa"
    list_re = r'DeleteOptionScheme!default.jspa\?(.*)schemeId=([0-9]*)'
    safe_data = DoNotDelete.issue_type_schemes
    delete_url = "/DeleteOptionScheme.jspa"
    delete_param = "schemeId"

class DeleteWorkflows (Delete):
    pretty_name = "Workflow"
    list_url = "/ListWorkflows.jspa"
    list_re = r'DeleteWorkflow.jspa\?(.*)workflowName=(.*)'
    safe_data = DoNotDelete.workflows
    delete_url = "/DeleteWorkflow.jspa"
    delete_param = "workflowName"

```

(continues on next page)

(continued from previous page)

```

class DeleteWorkflowSchemes(Delete):
    pretty_name = "Workflow Scheme"
    list_url = "/ViewWorkflowSchemes.jspa"
    list_re = r'DeleteWorkflowScheme!default.jspa\?(.*)schemeId=([0-9]*)'
    safe_data = DoNotDelete.workflow_schemes
    delete_url = "/DeleteWorkflowScheme.jspa"
    delete_param = "schemeId"

class DeleteStatuses(Delete):
    pretty_name = "Status"
    list_url = "/ViewStatuses.jspa"
    list_re = r'DeleteStatus!default.jspa\?id=([0-9]*)'
    safe_data = DoNotDelete.statuses
    delete_url = "/DeleteStatus.jspa"

def rename_users():
    logging.warning("Renaming Users")
    conn = ldap.initialize(Config.ldap_addr)
    conn.simple_bind_s(Config.ldap_user, Config.ldap_pass)

    with open(Config.entities_file) as f:
        logging.info("Looking up for usernames in %s" % Config.entities_file)
        content = f.read()
        userdata = re.compile(r'\n\s{4}<User\s.+?userName="([a-zA-Z0-9]+?)".*?
↪emailAddress="([a-zA-Z0-9.+_-@]+?)"')
        unames = userdata.findall(content)
        unames = sorted(unames)

    def get_ldap_username(email, oldname=None):
        oldname = oldname.strip(' ')
        email = email.strip(' ')
        result = conn.search_s(Config.ldap_dc, ldap.SCOPE_SUBTREE,
↪'(proxyAddresses=*%s*)' % email, ['sAMAccountName'])

        try:
            newname = result[0][1]['sAMAccountName'][0]
            logging.debug("User %s matched as %s" % (email, newname))
            return newname
        except Exception:
            logging.debug("User %s not found in LDAP directory" % email)
            try:
                loginname = re.match("(.*?)@(%s)" % "|".join(Config.fallback_
↪domains), email).group(1)
                logging.debug("Will use loginname %s from email address %s" %
↪(loginname, email))
                return loginname.lower()
            except Exception:
                logging.debug("Domain is not whitelisted, will use old username: %s
↪" % oldname)
                return oldname.lower()

    for oldname, email in unames:
        newname = get_ldap_username(email, oldname)
        logging.info("Substituting username %s with %s" % (oldname, newname))
        replace_from = re.compile(r'="%s"' % oldname)
        content = replace_from.sub(r'="%s"' % newname, content)

    with open(Config.entities_file, "w") as f:
        logging.info("Writing new content to file %s" % Config.entities_file)

```

(continues on next page)

(continued from previous page)

```

        f.write(content)
        logging.info("Content of %s updated." % Config.entities_file)

if __name__ == "__main__":
    #DeleteOutgoingMailServers().run()
    #DeleteGroups().run()
    #Do not run now new cf added#DeleteCustomFields().run()
    #DeleteProjectRoles().run()
    #DeleteProjects().run()
    #DeleteIssueTypes().run()
    #DeleteIssueTypeSchemes().run()
    #DeleteWorkflowSchemes().run()
    #DeleteWorkflows().run()
    #DeleteStatuses().run()
    #DeletePermissionSchemes().run()
    #DeleteNotificationSchemes().run()

    rename_users()

```

4.5 Testing Environment

4.5.1 Clone environment

Listing 4.4: Jira test environment

```

#!/usr/bin/env python2

from fabric.api import *
from fabric.colors import *
from fabric.contrib.console import confirm
from fabric.tasks import Task
from datetime import datetime

class CloneTask(Task):
    name = "clone"
    origin_user = None
    origin_host = None
    clone_user = None
    clone_host = None

    def run(self):
        timestamp_start = datetime.now().strftime("%Y-%m-%d %H:%M")
        print(yellow("[%s] Starting executing jobs..." % timestamp_start))
        execute(self.ask_user)
        execute(self.env_create)
        execute(self.env_rsync)
        execute(self.env_install)
        execute(self.deb_install)
        execute(self.jdk_rsync)
        execute(self.jdk_install)
        execute(self.clitools_rsync)
        execute(self.clitools_install)
        execute(self.preinstall)
        execute(self.install)
        execute(self.postinstall)
        timestamp_end = datetime.now().strftime("%Y-%m-%d %H:%M")

```

(continues on next page)

(continued from previous page)

```

    print(yellow("[%s] Everything done." % timestamp_end))

    def ask_user(self):
        self.origin_user = prompt('What is the origin user?', default=self.origin_
↪user, validate=r'^(\w+)$')
        self.origin_host = prompt('What is the origin host address?', default=self.
↪origin_host)
        self.clone_user = prompt('What is the clone user?', default=self.clone_
↪user, validate=r'^(\w+)$')
        self.clone_host = prompt('What is the clone host address?', default=self.
↪clone_host)
        env.roledefs['origin'] = ["%s@%s" % (self.origin_user, self.origin_host)]
        env.roledefs['clone'] = ["%s@%s" % (self.clone_user, self.clone_host)]
        print("Origin: %s\nClone: %s" % (env.roledefs['origin'], env.roledefs[
↪'clone']))
        if not confirm("Continue with this settings?", default=False):
            abort("Aborting at user request.")

    @roles('origin')
    def get_ssh_pubkey(self):
        print(green("Getting ssh pubkey from origin host..."))
        return sudo('cat /root/.ssh/id_rsa.pub')

    @roles('clone')
    def env_create(self):
        ssh_pubkey = execute(self.get_ssh_pubkey).popitem()[1]
        print(green("Creating Environment..."))
        sudo('mkdir -p /home/%s/.ssh' % self.clone_user)
        sudo('mkdir -p /opt/java')
        sudo('mkdir -p /opt/clitools')
        sudo('echo "%s" >> /home/%s/.ssh/authorized_keys' % (ssh_pubkey, self.
↪clone_user))
        sudo('echo \'export PS1="[u@\[$(tput bold)\]\[$(tput setaf 1)]\h\[$(tput_
↪sgr0)]:\w$] "\' >> /root/.bashrc')
        sudo('chown -R %s /home/%s' % (self.clone_user, self.clone_user))
        sudo('chown -R %s /opt/java' % self.clone_user)
        sudo('chown -R %s /opt/clitools' % self.clone_user)
        sudo('echo "nameserver 8.8.8.8" >> /etc/resolvconf/resolv.conf.d/head')
        sudo('echo "nameserver 8.8.6.6" >> /etc/resolvconf/resolv.conf.d/head')
        sudo('resolvconf -u')

    @roles('origin')
    def env_rsync(self):
        sudo('rsync -raz --delete /etc/environment %s:/tmp/environment' % env.
↪roledefs['clone'][0])

    @roles('clone')
    def env_install(self):
        sudo('mv /tmp/environment /etc/environment')
        sudo('chown root:root /etc/environment')

    @roles('clone')
    def deb_install(self):
        print(green("Installing deb..."))
        sudo('curl http://repo.varnish-cache.org/debian/GPG-key.txt | sudo apt-key_
↪add -')
        sudo('echo "deb http://repo.varnish-cache.org/ubuntu/ precise varnish-3.0"_
↪| sudo tee -a /etc/apt/sources.list')
        sudo('apt-get --quiet update')
        sudo('DEBIAN_FRONTEND=noninteractive apt-get --quiet --yes install mysql-
↪server')

```

(continues on next page)

(continued from previous page)

```

sudo('''sed -i -r -b "N;s/[mysqld]\\n#[/mysqld]\\ninnodb_file_per_
↪table\\nmax_allowed_packet=1024M/g" /etc/mysql/my.cnf''')
sudo('service mysql restart')
sudo('apt-get install --quiet --yes varnish')
sudo('apt-get install --quiet --yes htop')
sudo('apt-get install --quiet --yes memcached')
sudo('apt-get install --quiet --yes libmemcached-dev')
sudo('apt-get install --quiet --yes wget')
sudo('apt-get install --quiet --yes libxml2-utils')
sudo('apt-get install --quiet --yes curl')
sudo('apt-get install --quiet --yes git')
sudo('apt-get install --quiet --yes nmap')
sudo('apt-get install --quiet --yes gcc')
sudo('apt-get install --quiet --yes python-pip')
sudo('apt-get install --quiet --yes python-virtualenv')
sudo('apt-get install --quiet --yes libsasl2-dev')
sudo('apt-get install --quiet --yes python-dev')
sudo('apt-get install --quiet --yes libldap2-dev')
sudo('apt-get install --quiet --yes libmysqld-dev')
sudo('apt-get install --quiet --yes mc')

@roles('origin')
def jdk_rsync(self):
    print(green("Rsyncing jdk..."))
    sudo('rsync -raz --delete /opt/java/ %s:/opt/java' % env.roledefs['clone
↪'] [0])

@roles('clone')
def jdk_install(self):
    print(green("Installing jdk..."))
    sudo('update-alternatives --install /usr/bin/java java /opt/java/default/
↪bin/java 1')
    sudo('update-alternatives --set java /opt/java/default/bin/java')
    sudo('chown -R root:root /opt/java')

@roles('origin')
def clitools_rsync(self):
    print(green("Installing clitools..."))
    sudo('rsync -raz --delete /opt/clitools/ %s:/opt/clitools' % env.roledefs[
↪'clone'] [0])

@roles('clone')
def clitools_install(self):
    sudo('chown -R root:root /opt/clitools')

def preinstall(self):
    raise NotImplementedError

def install(self):
    raise NotImplementedError

def postinstall(self):
    raise NotImplementedError

class UpdateTask(Task):
    name = "update"
    origin_user = None
    origin_host = None
    clone_user = None
    clone_host = None

```

(continues on next page)

(continued from previous page)

```

def run(self):
    timestamp_start = datetime.now().strftime("%Y-%m-%d %H:%M")
    print(yellow("[s] Starting executing jobs..." % timestamp_start))
    execute(self.ask_user)
    execute(self.jdk_rsync)
    execute(self.preinstall)
    execute(self.install)
    execute(self.postinstall)
    timestamp_end = datetime.now().strftime("%Y-%m-%d %H:%M")
    print(yellow("[s] Everything done." % timestamp_end))

def ask_user(self):
    self.origin_user = prompt('What is the origin user?', default=self.origin_
↪user, validate=r'^(\w+)$')
    self.origin_host = prompt('What is the origin host address?', default=self.
↪origin_host)
    self.clone_user = prompt('What is the clone user?', default=self.clone_
↪user, validate=r'^(\w+)$')
    self.clone_host = prompt('What is the clone host address?', default=self.
↪clone_host)
    env.roledefs['origin'] = ["%s@%s" % (self.origin_user, self.origin_host)]
    env.roledefs['clone'] = ["%s@%s" % (self.clone_user, self.clone_host)]
    print("Origin: %s\nClone: %s" % (env.roledefs['origin'], env.roledefs[
↪'clone']))
    if not confirm("Continue with this settings?", default=False):
        abort("Aborting at user request.")

@roles('origin')
def jdk_rsync(self):
    print(green("Rsyncing jdk..."))
    sudo('rsync -raz --delete /opt/java/ %s:/opt/java' % env.roledefs['clone
↪'] [0])

def preinstall(self):
    raise NotImplementedError

def install(self):
    raise NotImplementedError

def postinstall(self):
    raise NotImplementedError

"""
.. todo:
* Add consolidate DeleteProjects.* to this module
* Add dry-run option
"""

class Clone(CloneTask):
    name = "clone"
    origin_user = None
    origin_host = "localhost"
    clone_user = "ubuntu"
    clone_host = None

    @roles('clone')
    def preinstall(self):
        print(green("Configuring preinstall actions..."))

```

(continues on next page)

(continued from previous page)

```

with settings(warn_only=True):
    sudo('useradd --system jira')
    sudo('mkdir -p /opt/jira')
    sudo('chown -R %s /opt/jira' % self.clone_user)

@roles('origin')
def install(self):
    print(green("Rsynchroning..."))
    clone = env.roledefs['clone'][0]
    exclude = [
        "home/caches/*",
        "home/data/attachments/*",
        "home/export/*",
        "home/import/*",
        "home/log/*",
        "home/tmp/*",
        "*/logs/*",
        "*/jre/*",
        "home/plugins/.bundled-plugins/*",
        "*/temp/*",
        "home/plugins/.osgi-plugins"]
    sudo('rsync -raz --delete --exclude=%(exclude)s /opt/jira/ %(clone)s:/opt/
↪jira' % {
        "clone": clone,
        "exclude": " --exclude=".join(exclude)})
    sudo('rsync -raz --delete /etc/init.d/jira %s:/opt/jira/initd.sh' % clone)

@roles('clone')
def postinstall(self):
    print(green("Configuring postinstall actions..."))
    sudo('sed -i "s/localhost/127.0.0.1/g" /opt/jira/home/dbconfig.xml')
    sudo('mysql -e "drop database if exists jira;"')
    sudo('mysql -e "create database jira /*!40100 DEFAULT CHARACTER SET utf8
↪COLLATE utf8_bin */;"')
    sudo('mysql -e "create user jira@localhost identified by \'jira\';"')
    sudo('mysql -e "grant all privileges on jira.* to jira@localhost
↪identified by \'jira\';"')
    sudo('mysqldump -hlocalhost -ujira -pjira --lock-all-tables jira |mysql
↪jira')
    sudo('mysql -e "delete from jira.filtersubscription;"')
    sudo('mysql -e "delete from jira.mailserver;"')
    sudo('mysql -e "update jira.propertystring set propertyvalue=\'http://
↪%s:8080\' where id in (select id from jira.propertyentry where property_key like
↪\'%%baseurl%%\');" % env.roledefs['clone'][0].split('@')[1])
    sudo('mysql -e "update jira.cwd_user set credential=
↪\'x61Ey612Kl2gpFL56FT9weDnpSo4AV8j8+qx2AuTHdRyY036xxzTTrw10Wq3+4qQyB+XURPWx1ONxp3Y3pB37A==\'
↪\' where user_name=\'admin\';"')
    sudo('mysql -e "update jira.propertytext set propertyvalue=\'<h3>This is a
↪JIRA test instance</h3>\' where ID=\'11216\';"')
    sudo('date > /opt/jira/database_lastupdate')
    sudo(''sed -i 's/JVM_MINIMUM_MEMORY=".*"/JVM_MINIMUM_MEMORY="256M"/g' /
↪opt/jira/install/bin/setenv.sh'')
    sudo(''sed -i 's/JVM_MAXIMUM_MEMORY=".*"/JVM_MAXIMUM_MEMORY="768M"/g' /
↪opt/jira/install/bin/setenv.sh'')
    sudo(''sed -i 's/scheme="https"/g' /opt/jira/install/conf/server.xml'')
    sudo(''sed -i 's/proxyName="localhost"/g' /opt/jira/install/conf/server.
↪xml'')
    sudo(''sed -i 's/proxyPort="443"/g' /opt/jira/install/conf/server.xml'')
    sudo('echo "jira.autoexport=false" >> /opt/jira/home/jira-config.properties
↪')
    sudo('mv /opt/jira/initd.sh /etc/init.d/jira')

```

(continues on next page)

(continued from previous page)

```

sudo('chown root:root /etc/init.d/jira')
sudo('chmod +x /etc/init.d/jira')
sudo('chown -R jira:jira /opt/jira')
print(red('/etc/init.d/jira start'))
print(red('sleep 240 && /opt/jira/home/reindex.sh'))

class Update(UpdateTask):
    name = "update"

    @roles('clone')
    def preinstall(self):
        print(green("Configuring preinstall actions..."))
        sudo('/etc/init.d/jira stop')
        sudo('chown -R %s /opt/jira' % self.clone_user)

    @roles('origin')
    def install(self):
        print(green("Rsyncing..."))
        clone = env.roledefs['clone'][0]
        exclude = [
            "home/caches/*",
            "home/data/attachments/*",
            "home/export/*",
            "home/import/*",
            "home/log/*",
            "home/tmp/*",
            "*/logs/*",
            "*/jre/*",
            "home/plugins/bundled-plugins/*",
            "*/temp/*",
            "home/plugins/osgi-plugins"
        ]
        sudo('rsync -raz --delete --exclude=%(exclude)s /opt/jira/ %(clone)s:/opt/
↪jira' % {
            "clone": clone,
            "exclude": " --exclude=" + join(exclude)})
        sudo('rsync -raz --delete /etc/init.d/jira %s:/opt/jira/initd.sh' % clone)

    @roles('clone')
    def postinstall(self):
        print(green("Configuring postinstall actions..."))
        sudo('sed -i "s/localhost/127.0.0.1/g" /opt/jira/home/dbconfig.xml')
        print(red('mysql -e "drop database if exists jira;"'))
        print(red('mysql -e "create database jira /*!40100 DEFAULT CHARACTER SET
↪utf8 COLLATE utf8_bin */;"'))
        print(red('mysql -e "grant all privileges on jira.* to jira@localhost
↪identified by \'localhost\';"'))
        print(red('mysqldump -hlocalhost -ujira -p"jira" --lock-all-tables jira
↪|mysql jira'))
        print(red('mysql -e "delete from jira.filtersubscription;"'))
        print(red('mysql -e "delete from jira.mailserver;"'))
        print(red('mysql -e "update jira.propertystring set propertyvalue=\'http://
↪%s:8080\' where id in (select id from jira.propertyentry where property_key like
↪\'%%baseurl%\';" % env.roledefs['clone'][0].split('@')[1]))
        print(red('mysql -e "update jira.cwd_user set credential=
↪\'x61Ey612Kl2gpFL56FT9weDnpSo4AV8j8+qx2AuTHdRyY036xxzTTrw10Wq3+4qQyB+XURPwx1ONxp3Y3pB37A==\'
↪ where user_name=\'admin\';"'))
        print(red('mysql -e "update jira.propertytext set propertyvalue=\'<h3>This
↪is a JIRA test instance</h3>\' where ID=\'11216\';"'))
        sudo('date > /opt/jira/database_lastupdate')
        sudo('sed -i "s/JVM_MINIMUM_MEMORY=.*"/JVM_MINIMUM_MEMORY="256M"/g' /
↪opt/jira/install/bin/setenv.sh')

```

(continues on next page)

(continued from previous page)

```

sudo(''sed -i 's/JVM_MAXIMUM_MEMORY=".*/JVM_MAXIMUM_MEMORY="768M"/g' /
↔opt/jira/install/bin/setenv.sh'')
sudo(''sed -i 's/scheme="https"//g' /opt/jira/install/conf/server.xml'')
sudo(''sed -i 's/proxyName="localhost"//g' /opt/jira/install/conf/server.
↔xml'')
sudo(''sed -i 's/proxyPort="443"//g' /opt/jira/install/conf/server.xml'')
sudo('mv /opt/jira/initd.sh /etc/init.d/jira')
sudo('chown root:root /etc/init.d/jira')
sudo('chown -R jira:jira /opt/jira')
sudo('chmod +x /etc/init.d/jira')
print(red('/etc/init.d/jira start'))
print(red('sleep 240'))
print(red('/opt/jira/home/reindex.sh'))

clone = Clone()
update = Update()

```

4.5.2 Clean data

Listing 4.5: Jira delete projects

```

import httplib
import urllib
import logging
import re

"""
.. todo::
* Add autodiscovery of token and jsessionid
* Do not delete projects by porojectkey not id
* Simplify
* Remove not used headers and params
"""

class Config(object):
    host = 'localhost:8080'
    do_not_delete_project = [10300] #EKO

    # You can get this from inspecting HTTP request with WebInspector in your
    ↔Browser
    token = '...'
    jsessionid = '...'

"""
You shouldn't change anything below this point,
unless you know what are you doing.
"""
logging.basicConfig(
    level=logging.INFO,
    format='[% (asctime)s] %(levelname)s: %(message)s'
)

class Http(object):

    @staticmethod

```

(continues on next page)

(continued from previous page)

```

def GET(url, params={}):
    params["atl_token"] = Config.token
    params = urllib.urlencode(params)
    return Http._request("GET", "%s?%s" % (url, params))

@staticmethod
def POST(url, params={}):
    params["atl_token"] = Config.token
    params["Delete"] = "Delete"
    params["confirmedDelete"] = "true"
    params["workflowMode"] = "live"
    params["confirm"] = "true"
    params["confirmed"] = "true"
    return Http._request("POST", url, params)

@staticmethod
def _request(method, url, params={}):
    params = urllib.urlencode(params)
    headers = {
        "Cookie": "atlassian.xsrf.token=%s; JSESSIONID=%s" % (Config.token,
↪Config.jsessionId),
        "Content-Type": "application/x-www-form-urlencoded",
    }
    conn = httplib.HTTPConnection(Config.host)
    logging.debug("curl -X %(method)s -d '%(params)s' --cookie '%(cookie)s' ↪
↪http://%(host)s%(path)s" % {
        'method': method,
        'params': params,
        'cookie': headers['Cookie'],
        'host': Config.host,
        'path': url,
    })
    conn.request(method, url, params, headers)
    response = conn.getresponse()
    logging.debug("%s %s" % (response.status, response.reason))
    ret = response.read()
    response.close()
    return ret

class DeleteAbstract(object):
    pretty_name = None
    list_url = None
    list_re = None
    safe_data = []
    delete_url = None
    delete_param = "id"

    def __init__(self):
        if not self.pretty_name:
            self.pretty_name = self.__class__.__name__
        if self.__class__.__name__ == "DeleteAbstract":
            raise NotImplementedError
        logging.warning("%s" % self.pretty_name)

    def get_delete_data(self):
        html = Http.GET(self.list_url, {"start":0, "max":10000})
        matches = re.findall(self.list_re, html)
        try:
            if isinstance(matches[0], tuple):
                matches = [id for string, id in matches]

```

(continues on next page)

(continued from previous page)

```

except IndexError:
    print("Not authorized or no entries.")

def clean(matches):
    matches = [urllib.unquote(name).decode('utf8') for name in matches]
    matches = [name.replace('+', ' ') for name in matches]
    return matches
return clean(matches)

def run(self):
    for id in self.get_delete_data():
        if str(id) not in [str(x) for x in self.safe_data]:
            logging.info("Deleting %s: %s" % (self.pretty_name, id))
            Http.POST(self.delete_url, {self.delete_param: id})

class DeleteProjects(DeleteAbstract):
    pretty_name = "Deleting Projects"
    list_url = "/ViewProjects.jspa"
    list_re = r'DeleteProject!default.jspa?pid=([0-9]*)'
    safe_data = Config.do_not_delete_project
    delete_url = "/DeleteProject.jspa"
    delete_param = "pid"

if __name__ == "__main__":
    DeleteProjects().run()

```

4.6 Performance Tuning

- JProfiler
- MAT (Memory Analyzer Tool) [heapdump and MAT from Eclipse]
- Performance SQL
- own database indexes
- *pgpool* and database cache
- *nginx* as a SSL terminator
- *Varnish* caching *REST* responses (JSON) and static files
- Java Melody
- New Relic

4.6.1 Optymalizacje

- Wyłączyć Activity Stream
- Update gadżetów na Dashboardzie (update na bazie dla wszystkich gadżetów)
- Edukacja użytkowników aby nie mieli odpalonych miliona zakładek z JIRĄ
- Czy wszystkie monitory z Wallboardami są potrzebne?

4.6.2 Database

- /var/atlassian/application-data/jira/dbconfig.xml

```

<pool-min-size>20</pool-min-size>
<pool-max-size>20</pool-max-size>
<pool-max-wait>30000</pool-max-wait>
<validation-query>select 1</validation-query>
<min-evictable-idle-time-millis>60000</min-evictable-idle-time-millis>
<time-between-eviction-runs-millis>300000</time-between-eviction-runs-millis>
<pool-max-idle>20</pool-max-idle>
<pool-remove-abandoned>true</pool-remove-abandoned>
<pool-remove-abandoned-timeout>300</pool-remove-abandoned-timeout>
<pool-test-on-borrow>false</pool-test-on-borrow>
<pool-test-while-idle>true</pool-test-while-idle>

```

4.6.3 Garbage Collector

- Jakub Kubryński on Garbage Collector <https://www.youtube.com/watch?v=LCr3XyHdaZk>
- G1 GC -XX:+UseG1GC
- Xmx
- /opt/atlassian/jira/bin/setenv.sh

Listing 4.6: Jira Garbage Collector

```

JIRA_HOME="/opt/jira/home"
JVM_SUPPORT_RECOMMENDED_ARGS="-server -XX:MaxPermSize=512m -XX:+UseG1GC -
↪XX:MaxGCPauseMillis=200 -XX:+PrintGC -XX:+PrintGCDateStamps -
↪XX:+OptimizeStringConcat -XX:+PrintGCDetails -XX:+DisableExplicitGC -Xloggc:/opt/
↪jira/logs/gc-jira-$(hostname)-$(date +%Y.%m.%d).log -XX:+UseGCLogFileRotation -
↪XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=10M"
JVM_MINIMUM_MEMORY="512m"
JVM_MAXIMUM_MEMORY="2048m"

# -server
# -XX:MaxPermSize=512m
# -XX:+UseG1GC
# -XX:+PrintGC
# -XX:MaxGCPauseMillis=200
# -XX:+PrintGCDateStamps
# -XX:+PrintGCDetails
# -XX:+UseGCLogFileRotation
# -XX:GCLogFileSize=10M
# -Xloggc:/opt/jira/logs/gc-jira-$(hostname)-$(date +%F).log
# -XX:NumberOfGCLogFiles=10
# -XX:+OptimizeStringConcat
# -XX:+DisableExplicitGC

# -Xms --> Minimum Memory
# -Xmx --> Maximum Memory
# -Xmn --> Heap of Younger Generation
# -Xss --> Thread Stack Size
# -XX:MaxMetaspaceSize --> Maximum Memory for Non-Heap Metaspace.
# -XX:NewRatio --> Ratio between Younger and Older Generation Memory sizes.
# -XX:ParallelGCThreads --> No of Parallel GC threads. By default, the GC threads
↪will be equal to the number of CPUs of the Node / VM. Used when Parallel Garbage
↪collectors are configured.

```

(continues on next page)

(continued from previous page)

```

GC_JVM_PARAMETERS=""
GC_JVM_PARAMETERS="-XX:+PrintGCDetails -XX:+PrintGCDateStamps -
↪XX:+PrintGCTimeStamps -XX:+PrintGCCause ${GC_JVM_PARAMETERS}"
GC_JVM_PARAMETERS="-Xloggc:$LOGBASEABS/logs/atlassian-jira-gc-%t.log -
↪XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=20M ${GC_JVM_
↪PARAMETERS}"

## Defaultowe ustawienia Jiry po instalacji:
/opt/atlassian/jira/jre/bin/java
-Djava.util.logging.config.file=/opt/atlassian/jira/conf/logging.properties
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
-Xms384m
-Xmx768m
-Djava.awt.headless=true
-Datlassian.standalone=JIRA
-Dorg.apache.jasper.runtime.BodyContentImpl.LIMIT_BUFFER=true
-Dmail.mime.decodeparameters=true
-Dorg.dom4j.factory=com.atlassian.core.xml.InterningDocumentFactory
-XX:-OmitStackTraceInFastThrow
-Datlassian.plugins.startup.options=
-Djdk.tls.ephemeralDHKeySize=2048
-Djava.protocol.handler.pkgs=org.apache.catalina.webresources
-Xloggc:/opt/atlassian/jira/logs/atlassian-jira-gc-%t.log
-XX:+UseGCLogFileRotation
-XX:NumberOfGCLogFiles=5
-XX:GCLogFileSize=20M
-XX:+PrintGCDetails
-XX:+PrintGCDateStamps
-XX:+PrintGCTimeStamps
-XX:+PrintGCCause
-classpath /opt/atlassian/jira/bin/bootstrap.jar:/opt/atlassian/jira/bin/
↪tomcat-juli.jar
-Dcatalina.base=/opt/atlassian/jira
-Dcatalina.home=/opt/atlassian/jira
-Djava.io.tmpdir=/opt/atlassian/jira/temp
org.apache.catalina.startup.Bootstrap start

```

4.6.4 Monitorowanie

- <http://www.stagemonitor.org/>
- New Relic
- JavaMelody
- JIRA embedded tools (in settings):
 - JMX monitoring
 - SQL profiling

4.6.5 Rozwiązywanie problemów

```
grep '/rest' /opt/atlassian/jira/logs/access_log.* |awk '{print $7}' |sort |uniq -  
↵c |sort -n
```

- Dużo zapytań API (varnish requestów, np. dashboardów)
- Inne usługi wysycające pamięć na maszynie, aż do limitów JAVY
- Przy port forwardnig `ssh -L 5432:localhost:5432 root@adresIP w /var/lib/pgsql/data/pg_hba.conf` musi być md5 przy IPv4 i IPv6
- Create issue by URL: <http://localhost:8080/secure/CreateIssueDetails!init.jsps?pid=10000&issuetype=10002>

4.6.6 Assignments

Administracja - Garbage Collector

1. Zmień Garbage Collector na G1
2. Zmień Xmx na 1GB
3. Wepnij Java Melody do monitorowania

Administracja - Zmiana Javy

1. Zainstaluj nową Javę na serwerze w katalogu `/opt/java/$VERSION`
2. Utwórz symlink `/opt/java/default/` wskazujący na `/opt/java/$VERSION` (dlaczego to dobra praktyka?)
3. Zrestartuj Jirę by wykorzystywała nową Javę

4.7 DB Performance

4.7.1 Dobre praktyki

- Terminal z połączeniem SSH do produkcji Background color RED

4.7.2 About

- AO = Add-On (plugins)
- `cwd_user` i `cwd_directories`
- `jiraissue`
- `mailserver`
- `filtersubscription`
- `worklog`
- `customfieldvalue` i `customfield`
- `project` i `project_key`
- `fileattachment`

```
ssh -L 5432:localhost:5432 root@adresIP
```

4.7.3 Backup data with pg_dump

```
$ service jira stop
$ pg_dump -i -h localhost -p 5432 -U jira -F c -b -v -f "/tmp/$(date +%F)_jira.
↪pgdump" jira
```

```
$ pg_dump -?
-p, -port=PORT database server port number
-i, -ignore-version proceed even when server version mismatches
-h, -host=HOSTNAME database server host or socket directory
-U, -username=NAME connect as specified database user
-W, -password force password prompt (should happen automatically)
-d, -dbname=NAME connect to database name
-v, -verbose verbose mode
-F, -format=c|t|p output file format (custom, tar, plain text)
-c, -clean clean (drop) schema prior to create
-b, -blobs include large objects in dump
-v, -verbose verbose mode
-f, -file=FILENAME output file name
```

4.7.4 Restore data with pg_restore

```
DROP DATABASE jira;
CREATE DATABASE jira_new;
GRANT ALL PRIVILEGES ON DATABASE jira_new TO jira;
```

```
$ pg_restore -i -h localhost -p 5432 -U jira -v "/tmp/$(date +%F)_jira.pgdump" -d_
↪jira_new
```

```
$ pg_restore -?
-p, -port=PORT database server port number
-i, -ignore-version proceed even when server version mismatches
-h, -host=HOSTNAME database server host or socket directory
-U, -username=NAME connect as specified database user
-W, -password force password prompt (should happen automatically)
-d, -dbname=NAME connect to database name
-v, -verbose verbose mode
```

4.7.5 Restore data with psql from plaintext SQL

```
$ psql -h localhost -p 5432 -U jira -d jira < "/tmp/$(date +%F)_jira.pgdump"
```

4.7.6 Change JIRA DB config

- Change /var/atlassian/application-data/jira/dbconfig.xml

```
$ service jira start
```

4.7.7 Assignments

Administracja - bazą danych

1. Zrób backup bazy danych (musi być data w nazwie pliku)
2. Zrób drop bazy
3. Zmień DB Pool connection
4. Przywróć backup do bazy jira_new
5. Dodaj polecenie backupu bazy danych do *crontab* z @midnight

4.8 API and Scripting

4.8.1 Accessing Jira

REST API

- <https://docs.atlassian.com/software/jira/docs/api/REST/server/>
- <https://developer.atlassian.com/jiradev/jira-apis/about-the-jira-rest-apis/jira-rest-api-tutorials>
- <https://docs.atlassian.com/jira/REST/latest/>
- <https://jira.atlassian.com/plugins/servlet/restbrowser#/>

Atlassian CLI

- <https://marketplace.atlassian.com/plugins/org.swift.atlassian.cli/cloud/overview>
- <https://bobswift.atlassian.net/wiki/spaces/ACLI/overview>

Atlassian Python API

- <https://github.com/atlassian-api/atlassian-python-api>
- `pip install atlassian-python-api`

4.8.2 Scripting

Reindex

Listing 4.7: Jira reindex

```
from pprint import pprint
from atlassian import Jira

jira = Jira(
    url="http://localhost:8000/",
    username="admin",
    password="admin")

status = jira.reindex().json()
pprint(status)
```


Project Administrators

Listing 4.8: Jira Project Administrators

```

import logging
from atlassian import Confluence
from atlassian import Jira

logging.basicConfig(level=logging.DEBUG, format="[% (asctime)s] [% (levelname)s]
↳ %(message)s")
logging.getLogger("requests").setLevel(logging.WARNING)
log = logging.getLogger("jira-projects-administrators")

jira = Jira(
    url="http://localhost:8000/",
    username="admin",
    password="admin")

html = "<table><tr><th>Project Key</th><th>Project Name</th><th>Leader</th><th>
↳ Email</th></tr>"

for data in jira.project_leaders():
    log.info("{project_key} leader is {lead_name} <{lead_email}>".format(**data))
    html += "<tr><td>{project_key}</td><td>{project_name}</td><td>{lead_name}</td></
↳ td><a href='mailto:{lead_email}'>{lead_email}</a></td></tr>".format(**data)

html += "</table><p></p><p></p>"
html += "<p>Autogenerated with <a href='http://localhost:7999/projects/AGILE/repos/
↳ devops-utils-jira/browse/bin/jira-projects-administrators.py'>this script</a></p>
↳ "

confluence = Confluence(
    url="http://localhost:8090/",
    username="admin",
    password="admin")

confluence.update_page(
    page_id=13207798,
    parent_id=7471197,
    title="Administratorzy JIRA",
    body=html)

log.info("Confluence Page Created with JIRA Administrators at: http://
↳ localhost:8095/pages/viewpage.action?pageId=13207798")

```

4.8.3 Assignments

Atlassian Python API - Instalacja

1. Zainstaluj bibliotekę Atlassian Python API `atlassian-python-api`

Note: Kod biblioteki dostępny jest na GitHub <https://github.com/atlassian-api/atlassian-python-api>

Warning: Wymagany Python 3.4 lub nowszy

Atlassian Python API - Reindeksacja

1. Stwórz skrypt `jira-reindex.py`
2. Skrypt wykorzystując bibliotekę `atlassian-python-api` ma reindeksować JIRE
3. Skrypt `jira-reindex.py` dodaj Crontab by był uruchamiany o 4 w nocy
4. Pamiętaj, że cron ma inne zmienne środowiskowe

Atlassian Python API - Project Administrators

1. Stwórz skrypt `jira-administrators.py`
2. Skrypt ma wylistować wszystkich administratorów projektów w JIRA w tabelce, wraz z ich emailem jako link "mailto"
 - Wynik zapisz w Confluence i dodaj się do watchers strony, by być powiadamianym o zmianach
 - Jeżeli nie masz zainstalowanego Confluence to zrzuć do pliku `/var/www/jira-admins.html` i skonfiguruj nginx aby wyświetlał tą stronę

Podpowiedź Aby uruchomić Confluence możesz wykorzystać Docker

```
$ apt-get update
$ apt-get install docker.io
$ docker run -v /var/atlassian/application-data/confluence:/var/
↪atlassian/application-data/confluence -d -p 8090:8090 atlassian/
↪confluence-server
```

Atlassian Python API - Changelog

1. Napisz skrypt `jira-changelog.py`
2. Wygeneruj Changelog, tj. listę zadań które zmieniły się pomiędzy dwoma wersjami (wykorzystaj JQL)
 - Wynik zapisz w Confluence na osobnej stronie dla każdej wersji
 - Jeżeli nie masz zainstalowanego Confluence to zrzuć do pliku `/var/www/changelog-XXX.html` i skonfiguruj nginx aby wyświetlał tą stronę, XXX to nazwa wersji

5.1 License

MIT License

Copyright (c) 2019 Matt Harasymczuk

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

5.2 Bibliography

5.3 Glossary

Bibliography

- [1] Atlassian. Jira official documentation. 2019. URL: <https://confluence.atlassian.com/jirasoftwareserver/>.