

UNLOCKING UNTAPPED SOFTWARE METRICS POTENTIAL WITH JIRA RESTFUL API

Blaze Smallwood, CCE/A, CSM, ICP-ACC, ICP-ATF
ICEAA Workshop 2024

MAY 2024

AGENDA

INTRODUCTION

Purpose

What is Jira?

GETTING DATA OUT OF JIRA

Export Via Jira UI

Using Jira's RESTful API

WHAT TO DO WITH DATA FROM THE API

What is JSON?

JSON Parsing Methods for Jira Data

Example Metrics

SUMMARY

PURPOSE

- Summarize Jira and show how valuable it can be as a data source for software metrics
- Review traditional ways of extracting data from Jira via its UI, then show how additional data can be extracted from its RESTful API
- Describe how one could process and utilize the data returned from the API for useful software metrics

WHAT IS JIRA?

- Jira is a web-based Application Lifecycle Management (ALM) tool, primarily used by software projects to manage efforts
 - Captures records called “issues” that represent pieces of work/effort a team can prioritize, assign to team members, manage across a workflow, capture knowledge about, and many other things
 - Leading software used by agile software projects

The screenshot displays a Jira Kanban board for the project 'APMT Web App Kanban (w Epics)'. The board is organized into columns representing different stages of the workflow: TO DO (5 items), BLOCKED (0 items), IN PROGRESS (5 items), READY FOR TEST (0 items), READY TO DEPLOY (0 items), READY TO VERIFY (0 items), and DONE (348 items). The 'TO DO' column contains two issues: APMT-2455, 'Update TLS version to address IT Service Hub vulnerability', and APMT-2456, 'Address sequence'. The 'IN PROGRESS' column contains two issues: APMT-1778, 'As a user, I want all views to reflect my sprint time boxes (1-40) Refactor, WebApp', and APMT-2469, 'Implement front-end and back-end changes to enable'. The 'DONE' column contains one issue: APMT-2070, 'Date drop-downs don't initialize after Jira Import'. The board also features a search bar, navigation tabs for 'Dashboards', 'Projects', 'Issues', 'Boards', 'Plans', and 'Tests', and a 'Create' button.

AGENDA

INTRODUCTION

Purpose

What is Jira?

GETTING DATA OUT OF JIRA

Export Via Jira UI

Using Jira's RESTful API

WHAT TO DO WITH DATA FROM THE API

What is JSON?

JSON Parsing Methods for Jira Data

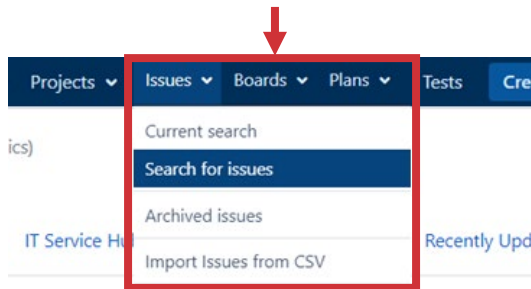
Example Metrics

SUMMARY

EXPORTING JIRA DATA VIA THE ISSUE SEARCH VIEW

- Primary and simplest way to get data out of Jira

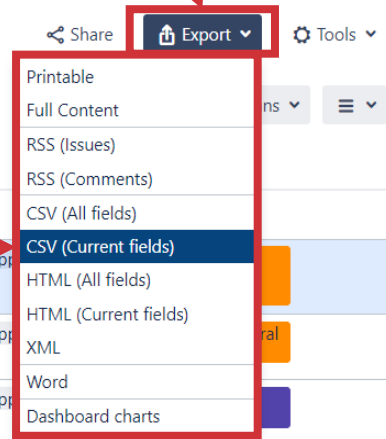
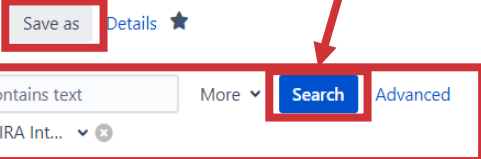
- Go to the **Issues Search** screen



- Create a filter and click **Search**, then **save** the filter for later

- Click the **Export** option

APMT Refactor Epics Valid Resolutions



T	Key	Summary	P	Status	Resolution	Created	Updated	Resolved	Story Points	Labels
↑	APMT-1905	Implement user input validation notifications on Configuration Dashboard	🚩	RESOLVED	Done	05/Sep/2018	21/Sep/2018	21/Sep/2018	2	WebApp
↑	APMT-1906	Implement project-level user input validations that show above any view	🚩	RESOLVED	Done	05/Sep/2018	20/Sep/2018	20/Sep/2018	2	WebApp
✓	APMT-1925	Merge code into 'refactor' branch and verify that 'refactor' builds/runs	🚩	RESOLVED	Done	01/Oct/2018	09/Oct/2018	09/Oct/2018	3	WebApp
↑	APMT-1835	Ensure data import process initializes the configuration	🚩	RESOLVED	Done	28/Feb/2018	15/Oct/2018	15/Oct/2018	3	Dataimport

EXPORTING JIRA DATA VIA THE ISSUE SEARCH VIEW – RESULTING DATA FORMAT

- Exported data typically in flat, CSV format

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Issue Type	Issue key	Issue id	Summary	Priority	Status	Resolution	Created	Updated	Resolved	Custom fie	Labels	Labels	Labels	Labels
2	Improvement	APMT-1905	478993	Implement user input validation notifications on	Major	Resolved	Done	9/5/2018 17:56	9/21/2018 16:57	9/21/2018 16:57	2	WebApp			
3	Improvement	APMT-1906	478994	Implement project-level user input validations th	Major	Resolved	Done	9/5/2018 17:57	9/20/2018 17:44	9/20/2018 17:44	2	WebApp			
4	Task	APMT-1925	489748	Merge code into 'refactor' branch and verify that	Critical	Resolved	Done	10/1/2018 12:08	10/9/2018 11:44	10/9/2018 11:44	3	WebApp			
5	Improvement	APMT-1835	340556	Ensure data import process initializes the configu	Major	Resolved	Done	2/28/2018 11:09	10/15/2018 11:38	10/15/2018 11:38	3	DataImpoi	WebApp		
6	Improvement	APMT-1912	489726	Enable user to delete previously imported data se	Major	Resolved	Done	10/1/2018 11:26	10/23/2018 11:31	10/23/2018 11:31	5	WebApp			
7	Improvement	APMT-1919	489737	Refine buffering input to be a single input vice by	Major	Resolved	Done	10/1/2018 11:35	10/17/2018 13:12	10/17/2018 13:12	8	WebApp			
8	Improvement	APMT-1922	489740	Improve UI on budget/schedule deltas bars	Major	Resolved	Done	10/1/2018 11:39	10/23/2018 11:31	10/23/2018 11:31	2	WebApp			
9	Improvement	APMT-1928	494973	Initialize all variables (user inputs and config dash	Major	Resolved	Done	10/15/2018 9:55	10/23/2018 15:43	10/23/2018 15:43	2	WebApp			
10	Improvement	APMT-1929	494975	Ensure all errors show up on top error drop-down	Major	Resolved	Done	10/15/2018 9:56	10/19/2018 19:42	10/19/2018 19:42	2	WebApp			
11	Improvement	APMT-1927	494972	Persist user inputs with a "Save" button that repl	Major	Resolved	Done	10/15/2018 9:53	10/22/2018 13:20	10/22/2018 13:20	2	WebApp			

- CSV data is very useful, but has limitations
 - Completion date data is often incomplete
 - Minimal insight into specific actions on issues over time
 - How it outputs data in overloaded fields is cumbersome

Jira's RESTful API can help mitigate these limitations

AGENDA

INTRODUCTION

Purpose

What is Jira?

GETTING DATA OUT OF JIRA

Export Via Jira UI

Using Jira's RESTful API

WHAT TO DO WITH DATA FROM THE API

What is JSON?

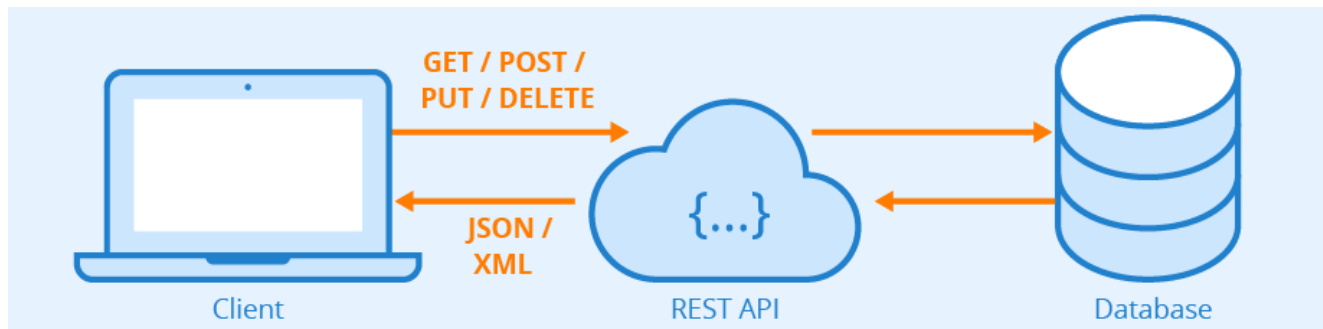
JSON Parsing Methods for Jira Data

Example Metrics

SUMMARY

WHAT IS A RESTFUL API?

- Representational State Transfer (REST)ful Application Program Interface (API) Defined:
 - *"A RESTful API is an interface that two computer systems use to exchange information securely over the internet."

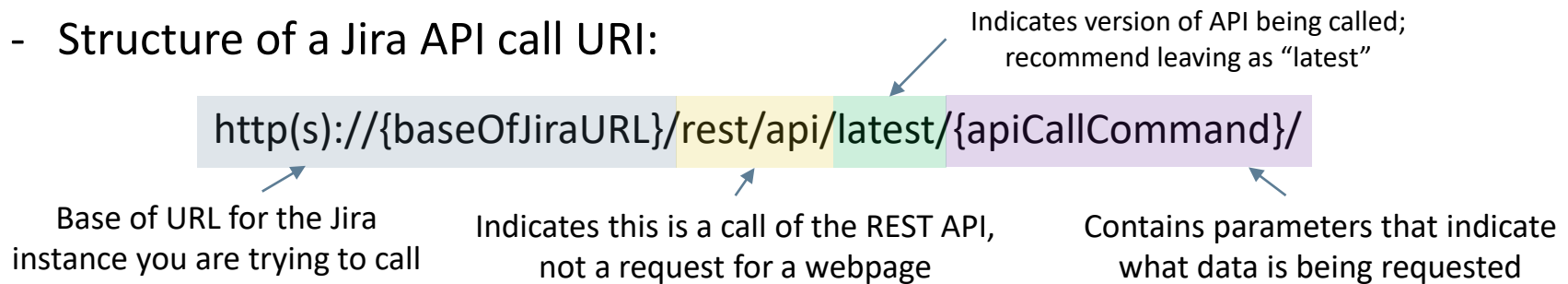


- Many RESTful APIs are implemented by websites as an additional capability that enables users to access data from the site's database directly with an API call versus via the site's UI (i.e. web pages)
- Most RESTful APIs are accessed with a specifically constructed Uniform Resource Identifier (URI) that executes the call using HTTP/HTTPS

HOW DO I ACCESS JIRA'S RESTFUL API?

- Jira's RESTful API allows you call it with an appropriate URI, then it sends back the requested data in JSON format

- Structure of a Jira API call URI:



- Example Jira RESTful API URI that requests all fields for all issues, including change log data, from a specific saved filter in Booz Allen's Jira instance:

`https://jira.boozallencsn.com/rest/api/latest/search?jql=filter=XXXXX&maxResults=XXX&expand=changelog`

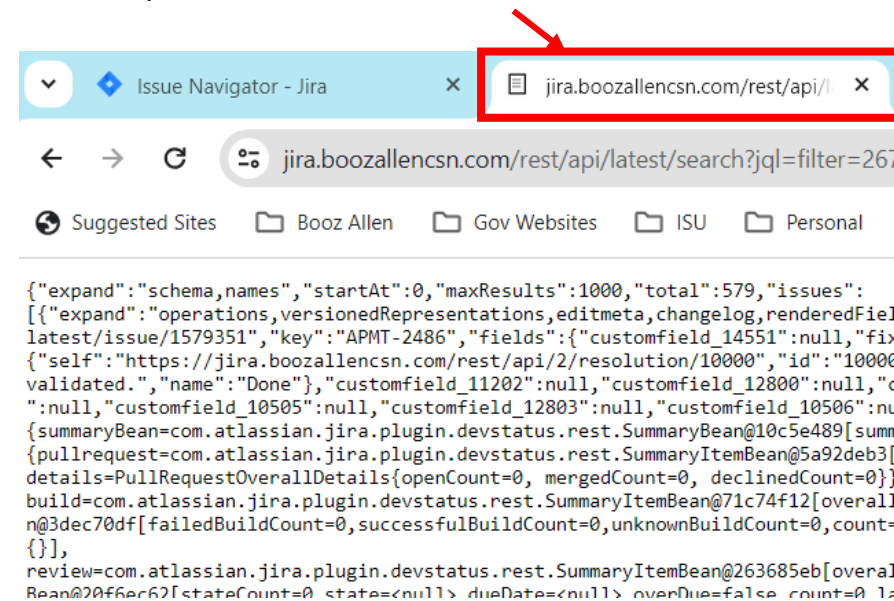
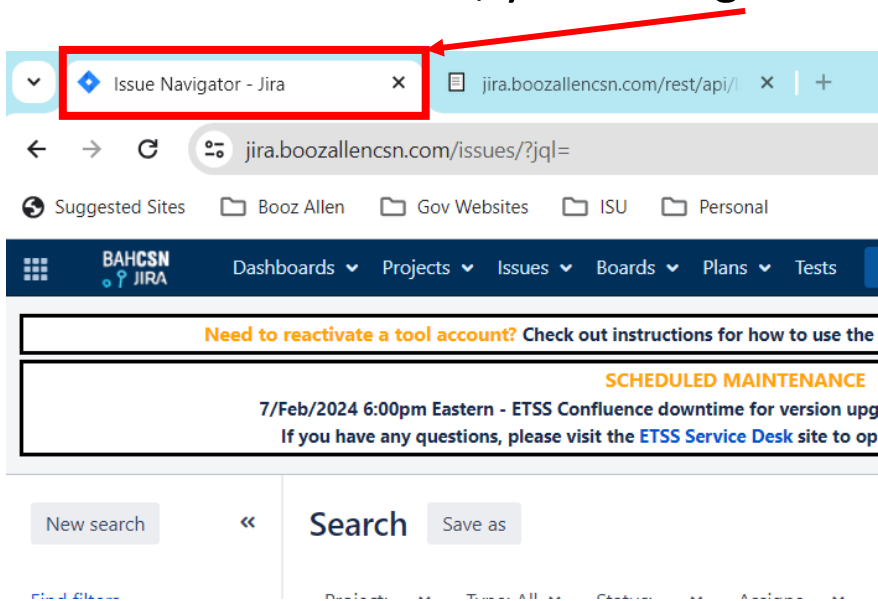
- Jira has thorough online documentation on their RESTful API, as well as a helpful online support community

- *Documentation:* <https://developer.atlassian.com/server/jira/platform/rest-apis/>

- *Support community:* <https://community.developer.atlassian.com/>

HOW DO I ACCESS JIRA'S RESTFUL API? (CONT'D)

- Calling Jira's API requires you to be authenticated (i.e. signed into your user account)
 - In the browser, you can sign in on one tab, then enter URI on a new tab



- In a script/program, you need to code the authentication procedure before/as a part of your API call procedure
 - Jira instances requiring smart card-based or multi-factor authentication increases complexity of writing an API-calling script

WHY SHOULD I USE JIRA'S RESTFUL API?

- Jira's RESTful API gives you access to important data that you cannot access from its web UI
 - **Change Log**: log of all changes that have happened on every issue record in the database
 - **Work Log**: log of all work, including time spent, that users have logged against any given issue record in the database
 - **Field Metadata**: information about each field, including key-name pairs that are important in parsing API data
- Enables use of parsing scripts/programs to create a data structure that best fits your analytical tools
 - e.g. better formatting of Jira's overloaded fields

AGENDA

INTRODUCTION

Purpose

What is Jira?

GETTING DATA OUT OF JIRA

Export Via Jira UI

Using Jira's RESTful API

WHAT TO DO WITH DATA FROM THE API

What is JSON?

JSON Parsing Methods for Jira Data

Example Metrics

SUMMARY

WHAT IS JSON?

- Jira's RESTful API returns data in XML or JSON (default) formats
- JSON (JavaScript Object Notation) is a lightweight data-interchange format*, heavily used in web-based systems
 - Coding language-independent format, primarily composed of:
 - Objects – surrounded by braces “{}”
 - Collection of name/value pairs
 - Value in a pair can be a primitive or another object or array
 - Names and values in a pair separated by a colon (“:”)
 - Name/value pairs separated by commas (“,”)
 - Arrays – surrounded by brackets “[]”
 - Ordered list of values, separated by commas (“,”)
 - Values are often objects

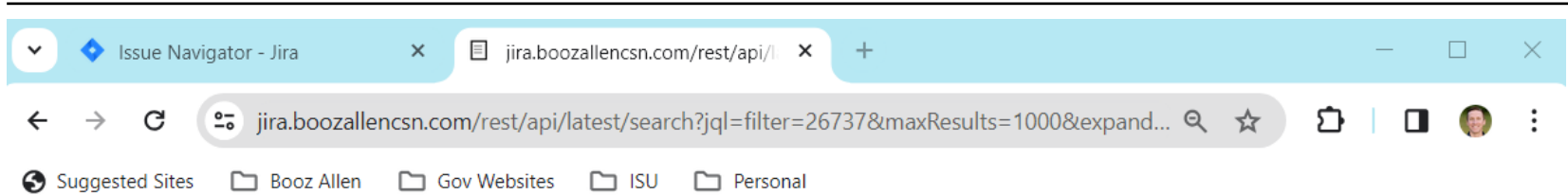
SIMPLE JSON EXAMPLE*

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "selection": null,
    "popup": {
      "size": 300,
      "enabled": true,
      "menuitem": [
        {"value": "New", "onclick": "CreateNewDoc()"},
        {"value": "Open", "onclick": "OpenDoc()"},
        {"value": "Close", "onclick": "CloseDoc()"}
      ]
    }
  }
}
```

A *value* can be a:

- String
- null
- object
- number
- Boolean
- array

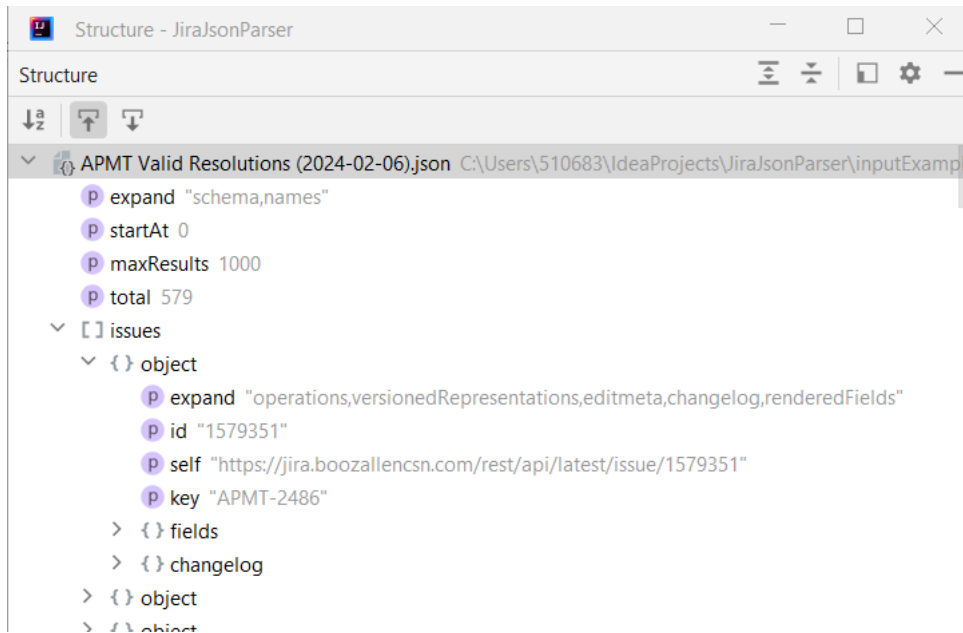
VIEWING JIRA JSON DATA



```
{
  "expand": "schema,names",
  "startAt": 0,
  "maxResults": 1000,
  "total": 579,
  "issues": [
    {
      "expand": "operations,versionedRepresentations,editmeta,changelog,renderedFields",
      "id": "1579351",
      "self": "https://jira.boozallencsn.com/rest/api/latest/issue/1579351",
      "key": "APMT-2486",
      "fields": {
        "customfield_14551": null,
        "fixVersions": [],
        "customfield_12257": null,
        "resolution": {
          "self": "https://jira.boozallencsn.com/rest/api/2/resolution/10000",
          "id": "10000",
          "description": "The request has been completed, verified and/or validated.",
          "name": "Done"
        },
        "customfield_11202": null,
        "customfield_12800": null,
        "customfield_10501": null,
        "customfield_12801": null,
        "customfield_12804": null,
        "customfield_10505": null,
        "customfield_12803": null,
        "customfield_10506": null,
        "customfield_10509": null,
        "customfield_18100": null
      },
      "summaryBean": com.atlassian.jira.plugin.devstatus.rest.SummaryBean@10c5e489[summary=
      {pullrequest=com.atlassian.jira.plugin.devstatus.rest.SummaryItemBean@5a92deb3[overall=PullRequestOverallBean{stateCount=0, state='OPEN', details=PullRequestOverallDetails{openCount=0, mergedCount=0, declinedCount=0}},byInstanceType={}],
    }
  ]
}
```

Data comes unformatted from Jira's API

Better to view data using an IDE or JSON formatting tool to see it in a tree structure



JIRA JSON DATA STRUCTURE

```
Structure
APMT Valid Resolutions (2024-02-06).json
  expand "schema, names"
  startAt 0
  maxResults 1000
  total 579
  issues
    object
      expand "operations, versionedRepresentations, editmeta, changelog, renderedFields"
      id "1579351"
      self "https://jira.boozallencsn.com/rest/api/latest/issue/1579351"
      key "APMT-2486"
      fields
      changelog
      object
      object
      object
```

Most of the data is in the "issues" array

The "key" field is the issue object's unique identifier

For each issue, the main useful data is in the "fields" and "changelog" objects

JIRA JSON DATA STRUCTURE: FIELDS OBJECT

All of the data you can export from the UI (plus more) is in the "fields" object

Some fields are objects or arrays that have additional data that you can't access by downloading data from the web UI

```
Structure - JiraJsonParser
Structure
  { } object
    expand "operations,versionedRepresentations,editmeta,changelog,renderedFields"
    id "1579351"
    self "https://jira.boozallencsn.com/rest/api/latest/issue/1579351"
    key "APMT-2486"
    { } fields
      summary "Update Spring framework to resolve Spring Security cyber vulnerabilities"
      { } issuetype
        self "https://jira.boozallencsn.com/rest/api/2/issuetype/3"
        id "3"
        description "A task that needs to be done."
        iconUrl "https://jira.boozallencsn.com/secure/viewavatar?size=xsmall&avatarId=10318"
        name "Task"
        subtask false
        avatarId 10318
      [ ] components
      created "2023-06-29T11:23:46.000-0400"
      > { } reporter
      [ ] fixVersions
      > { } priority
      > { } resolution
      [ ] labels
      resolutiondate "2023-09-07T12:08:23.000-0400"
      > [ ] issuelinks
      > { } assignee
      updated "2023-09-07T12:08:23.000-0400"
      { } status
        self "https://jira.boozallencsn.com/rest/api/2/status/5"
        description "A resolution has been taken, and it is awaiting verification by reporter. f"
        iconUrl "https://jira.boozallencsn.com/images/icons/statuses/resolved.png"
        name "Resolved"
        id "5"
      > { } statusCategory
```

JIRA JSON DATA STRUCTURE: CHANGELOG OBJECT

The “changelog” object within each “issue” object contains data regarding every change that happened to that issue record

The “histories” array contains a collection of objects that represent each change

The “author” object shows who made the change

The “created” item shows when the change was made

The “items” array has a collection of objects that shows what changed and how

```
Structure
├── changelog
│   ├── startAt 0
│   ├── maxResults 8
│   └── total 8
├── histories
│   ├── object
│   ├── object
│   ├── object
│   ├── object
│   ├── object
│   └── object
│       ├── id "14057821"
│       ├── author
│       │   ├── self "https://jira.boozallencsn.com/rest/api/2/user?u..."
│       │   ├── name "smallwood_blaze@bah.com"
│       │   ├── key "smallwood_blaze@bah.com"
│       │   └── emailAddress "smallwood_blaze@bah.com"
│       ├── avatarUrls
│       │   ├── displayName "Smallwood, Blaze [USA]"
│       │   ├── active true
│       │   └── timeZone "America/New_York"
│       ├── created "2023-06-29T11:26:49.566-0400"
│       └── items
│           └── object
│               ├── field "status"
│               ├── fieldtype "jira"
│               ├── from "1"
│               ├── fromString "Open"
│               ├── to "3"
│               └── toString "In Progress"
├── object
└── object
```

As an example, using the changelog, you can see that Blaze changed this issue's status from Open to In Progress on 6/29/2023

PARSING JIRA JSON DATA

- Since JSON is language independent, various tools and programming languages can be used to parse it into a usable format for analysis. Some examples:

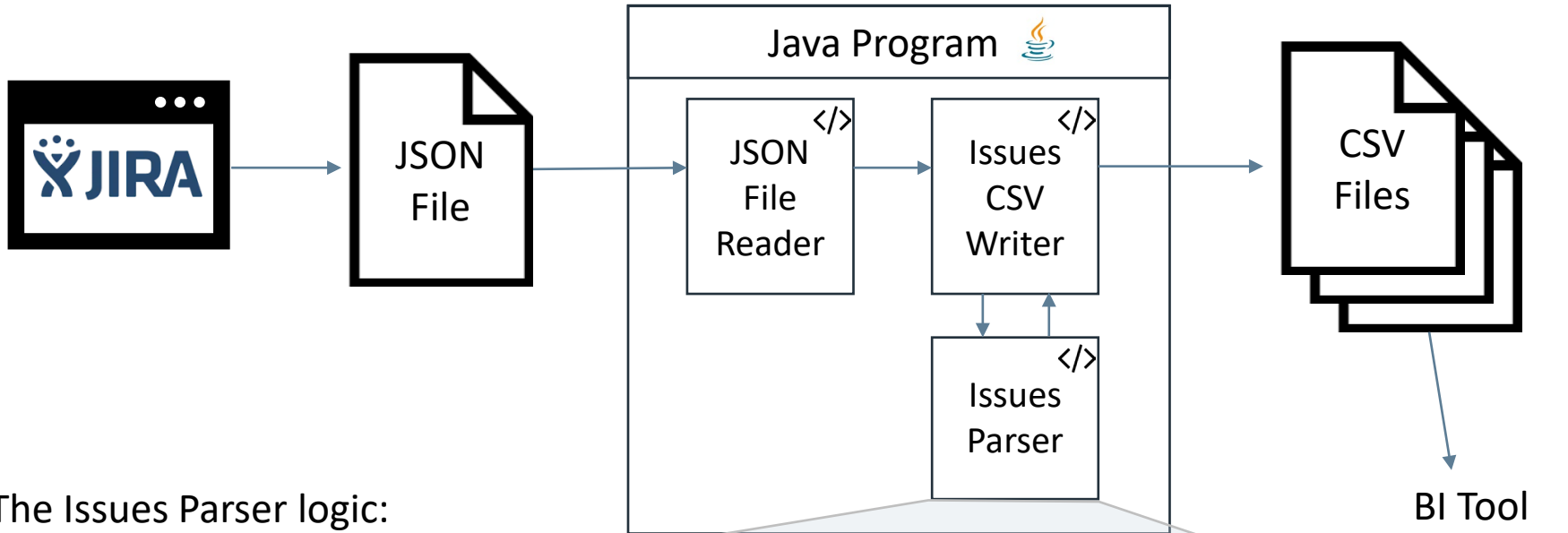
- Python
- Java
- JavaScript
- R
- Power Query

The image shows two screenshots. The top screenshot is an IDE window titled 'JiraJsonParser' showing Java code for parsing JIRA JSON data. The code includes comments and logic to check for the presence of an 'issues' array and 'fields' object, and to iterate over the issues array to extract data. The bottom screenshot is a Power Query interface showing a table with columns 'Issues.Key', 'Issues.Fields.Summary', and 'Issues.Fields.IssueType.Self'. The table contains 18 rows of data, including issue keys like 'APMT-2486' and their corresponding summaries and issue types.

Issues.Key	Issues.Fields.Summary	Issues.Fields.IssueType.Self
APMT-2486	Update Spring framework to resolve Spring Security cyber vulnerability	https://jira.boozallen.com/rest/a
APMT-2485	Update Spring framework to resolve Log4j cyber vulnerabilities	https://jira.boozallen.com/rest/a
APMT-2484	Update Spring framework to resolve Spring Framework cyber vulnerab...	https://jira.boozallen.com/rest/a
APMT-2483	Update OpenJDK to address related critical cyber vulnerability	https://jira.boozallen.com/rest/a
APMT-2482	Make appropriate environment updates to close critical RHEL-related ...	https://jira.boozallen.com/rest/a
APMT-2481	Update Spring framework to resolve Apache Struts cyber vulnerabilities	https://jira.boozallen.com/rest/a
APMT-2480	Disable nginx server tokens to address IT Service Hub vulnerability	https://jira.boozallen.com/rest/a
APMT-2479	Update OpenSSH version to address IT Service Hub vulnerability	https://jira.boozallen.com/rest/a
APMT-2478	Update nginx version to address IT Service Hub vulnerability	https://jira.boozallen.com/rest/a
APMT-2477	Plan how to allow non-Booz Allen users to access/use APMT	https://jira.boozallen.com/rest/a
APMT-2476	Enable non-Booz Allen users to access APMT	https://jira.boozallen.com/rest/a
APMT-2475	Add Clear Filters button to Feedback Admin view	https://jira.boozallen.com/rest/a
APMT-2471	As a user, I want all views to reflect my sprint time boxes with custom ...	https://jira.boozallen.com/rest/a
APMT-2467	Make filters on feedback admin table update asynchronously	https://jira.boozallen.com/rest/a
APMT-2463	Develop technical plan for enabling users to share projects with each o...	https://jira.boozallen.com/rest/a
APMT-2462	As an admin user, I would like to export data from user feedback view	https://jira.boozallen.com/rest/a
APMT-2461	Investigate appropriate system-generated e-mail solutions	https://jira.boozallen.com/rest/a
APMT-2460	Refine UI on RSP budget/schedule delta chart	https://jira.boozallen.com/rest/a

- Tool/parsing program traverses the JSON tree, extracts wanted data, and compiles it into desired data format for analysis

PARSING JIRA JSON DATA: JAVA EXAMPLE OVERVIEW



The Issues Parser logic:

- Iterates through the “issues” array
- Traverses each “issue” object to capture the desired data from the “fields” and “changelog” objects
- Stores the desired data in a structured format
- Passes structured data object back to CSV Writer

```
File Edit View Window Help JiraJsonParser - IssuesParser.java
IssuesCSVWriter.java IssuesParser.java ChangelogParser.java IssuesFromFilesConverterForm
79
80 //check if issues array exists; if it does, check for fields array before executing parsing logic
81 if(!JSONObject.has(memberName: "issues")){
82     errorStr = "JSON file has no issues array.";
83 } else {
84     //get issues array from JSONObject
85     JSONArray issuesArray = JSONObject.getAsJSONArray(memberName: "issues");
86
87     //check if first issue has fields object; if so, execute parsing logic, if not, log error message
88     JSONObject firstIssueElement = issuesArray.get(0).getAsJSONObject();
89     if(!firstIssueElement.has(memberName: "fields")){
90         errorStr = "JSON data has no fields data.";
91     } else {
92         //iterate over issues array to parse each issue element and feed the appropriate SB objects
93         int issueCount = 1;
94         for(JsonElement issue : issuesArray) {
95             //get issue object
96             JSONObject issueObject = issue.getAsJSONObject();
97
98             //get key and store in String variable
99             String issueKey = issueObject.get("key").getString();
100
```

ISSUE PARSER LOGIC LOOK-IN

```
APMT Valid Resolutions (2024-02-07).json CAU
├── expand "schema, names"
├── startAt 0
├── maxResults 1000
├── total 579
├── issues
│   ├── {} object
│   └── {} object
│       ├── expand "operations, versionedRepresentations"
│       ├── id "1579350"
│       ├── self "https://jira.boozallen.com/rest/api/2/issue/1579350"
│       ├── key "APMT-2485"
│       └── fields
│           ├── summary "Update Spring framework"
│           ├── issueType {} object
│           ├── components []
│           ├── created "2023-06-29T11:23:06.000Z"
│           ├── reporter {} object
│           ├── fixVersions []
│           ├── priority {} object
│           ├── resolution {} object
│           ├── labels []
│           ├── resolutiondate "2023-07-10T11:00:00.000Z"
│           ├── issuelinks []
│           ├── assignee {} object
│           ├── updated "2023-07-10T11:07:49.000Z"
│           ├── status {} object
│           ├── changelog []
│           ├── {} object
│           ├── {} object
│           └── {} object
```

```
for(JsonElement issue : issuesArray) {
    //get issue object
    JsonObject issueObject = issue.getAsJsonObject();

    //get key and store in String variable
    String issueKey = issueObject.get("key").getAsString();

    //get fields object from issue object
    JsonObject fieldsObject = issueObject.get("fields").getAsJsonObject();








    //loop through the fields map and parse the appropriate information from the issue object
    int fieldCount = 1;
    String issueValueStr = issueKey + ",";
    for(String[] field : jiraFields.values()) {
        //get fieldKey and initialize fieldValueStr variable
        String fieldKey = field[0];
        String fieldValueStr = "";

        //fields parsing logic for each issue object
        if(!(fieldsObject.get(fieldKey) == null) && !(fieldKey.equalsIgnoreCase("issuekey"))) {
            if(fieldsObject.get(fieldKey).isJsonPrimitive()) {
                //add primitive value to fieldValueStr
                fieldValueStr = fieldsObject.get(fieldKey).getAsString();

                //extract value from complex primitive string values that really represent Jira objects, like
                fieldValueStr = this.findDataInPrimitiveObjectString(fieldValueStr);

                //clean fieldValueStr to preserve CSV structure
                fieldValueStr = this.cleanFieldValueStr(fieldValueStr);
            } else if(fieldsObject.get(fieldKey).isJsonObject()) {
                //get the field value as object, then parse into fieldValueStr
                JsonObject fieldsValueObject = fieldsObject.get(fieldKey).getAsJsonObject();
                fieldValueStr = this.parseJsonFieldObject(fieldsValueObject);
            }
        }
    }
}
```

EXAMPLE PARSED DATA SETS

-  APMT Data (FixVersions).csv
-  APMT Data (issues).csv
-  APMT Data (Labels).csv
-  APMT Data (LinkedIssues).csv
-  APMT Data (Sprint).csv
-  APMT Data (sprintChangelog).csv
-  APMT Data (statusChangelog).csv

Key	Summary	Issue Type	Status	Priority	Labels	Reporter	Assignee	Created	Updated	Resolved	Resolution	Story Points	Sprint	Epic Link
APMT-2486	Update Spring frame Task	Resolved	Major	GovCloud; ITServiceHub	Smallwooc Windes; D;	2023-06-29	2023-09-0	2023-09-0	Done			5		APMT-2270
APMT-2485	Update Spring frame Task	Resolved	Major	GovCloud; ITServiceHub	Smallwooc Windes; D;	2023-06-29	2023-07-1	2023-07-1	Done			5		APMT-2270
APMT-2484	Update Spring frame Task	Resolved	Major	GovCloud; ITServiceHub	Smallwooc Windes; D;	2023-06-29	2023-06-2	2023-06-2	Done			5		APMT-2270
APMT-2483	Update OpenJDK to 8 Task	Resolved	Major	GovCloud; ITServiceHub	Smallwooc Windes; D;	2023-05-04	2023-09-0	2023-09-0	Done			8		APMT-2270
APMT-2475	Add Clear Filters butt Improvem	Backlog	Minor	Refactor; WebApp	Smallwooc; Blaze [US	2021-05-21	2022-12-13	T10:53:13.000-0500				3		APMT-1999
APMT-2471	As a user, I want all v Story	Backlog	Major	Refactor; WebApp	Smallwooc Windes; D;	2021-05-06	2021-05-06	T12:39:10.000-0400				40		APMT-1888
APMT-2467	Make filters on feed Improvem	Resolved	Major	Refactor; WebApp	Smallwooc Santi Pierri	2021-05-06	2021-05-2	2021-05-2	Done			5		APMT-1999
APMT-2463	Develop technical pl; Task	Resolved	Major	Refactor; WebApp	Smallwooc Nuzzolo; K	2021-04-30	2021-05-0	2021-05-0	Done			5		APMT-2159

Main data set similar to Jira UI download, but in a cleaner format better suited for analysis and BI tools

Key	Type Name	Type Description	Linked Issue Key
APMT-2486	Clone		
APMT-2485	Clone		
APMT-2484	Clone		
APMT-2481	Clone		
APMT-2481	Clone		
APMT-2481	Clone		
APMT-2480	Clone		
APMT-2479	Super		
APMT-2477	Blocks	blocks	APMT-2476

Provides a cleaner, more descriptive, and pivoted table of **issue link** data than Jira UI download

Key	Date	From Status	To Status
APMT-2486	6/29/2023	Backlog	Open
APMT-2486	6/29/2023	Open	In Progress
APMT-2484	6/29/2023	Open	In Progress
APMT-2484	6/29/2023	In Progress	Resolved
APMT-2482	5/4/2023	Backlog	Open

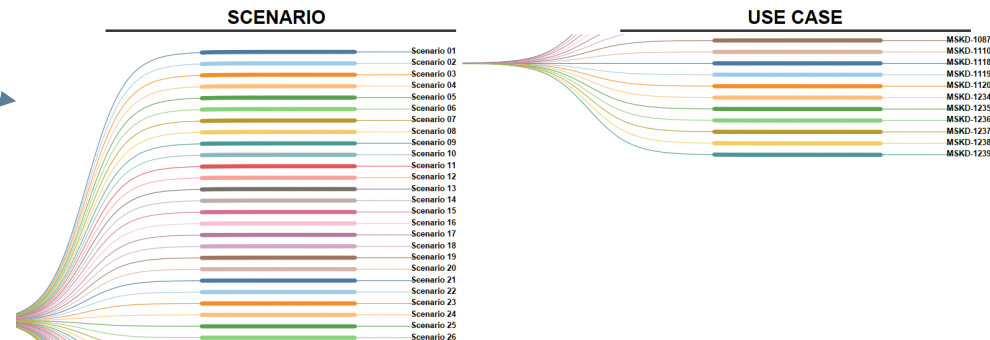
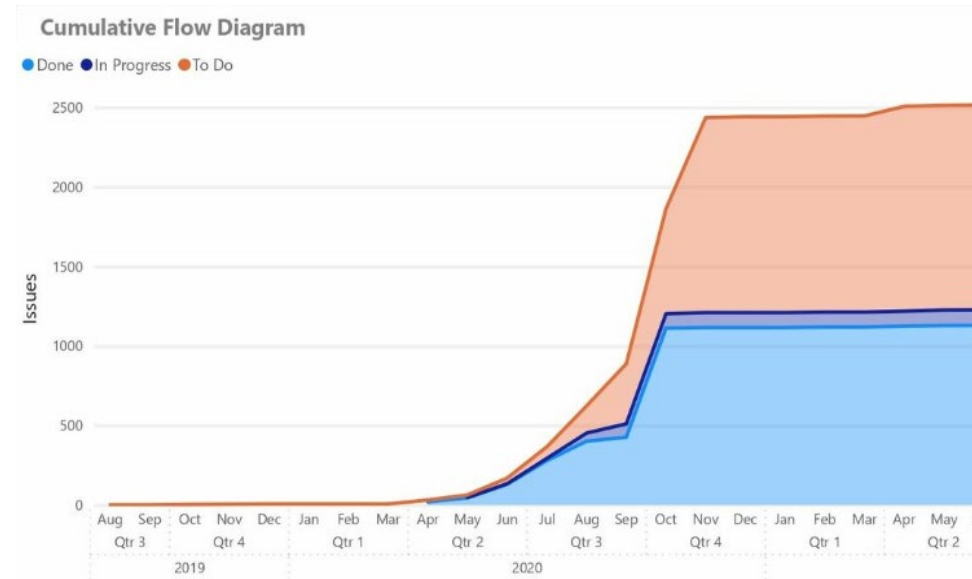
Captures data that shows when an issue moves from one **status** to another; Can't get that from Jira UI

Key	Date	From Sprint	To Sprint
APMT-2111	3/25/2019	(none)	APMT Web App FY19 Sprint 15
APMT-2111	3/27/2019	APMT Web App FY19 Sprint 15	(none)
APMT-2109			int 15
APMT-2108			int 15
APMT-2106			int 15
APMT-2099			int 15
APMT-2096			int 15
APMT-2095			int 15
APMT-2093			int 15
APMT-2092			int 15
APMT-2092	3/15/2019	APMT Web App FY19 Sprint 15	APMT Web App FY19 Sprint 14
APMT-2092	3/18/2019	APMT Web App FY19 Sprint 14	APMT Web App FY19 Sprint 15
APMT-2091	3/11/2019	(none)	APMT Web App FY19 Sprint 15

Captures when an issue moves from one **sprint** to another or from **backlog** to a **sprint**; Can't get that from Jira UI

BENEFITS AND EXAMPLE METRICS

- Changelog data enables more time-based visuals
 - Cumulative Flow Diagrams →
 - Average time in status
- Better formatted issue link data enables more hierarchy-based analysis
 - Dendrograms →
- More accurate velocity with changelog dates
- Cleaner parsed format enables quicker data refresh



AGENDA

INTRODUCTION

Purpose

What is Jira?

GETTING DATA OUT OF JIRA

Export Via Jira UI

Using Jira's RESTful API

WHAT TO DO WITH DATA FROM THE API

What is JSON?

JSON Parsing Methods for Jira Data

Example Metrics

SUMMARY

SUMMARY

- Jira provides a wealth of information/data to analyze for cost and project management analyses
- Jira's UI is useful for data export, but has limitations
- Jira's RESTful API provides more options for data export
- JSON data returned by Jira's RESTful API can be parsed by numerous tools and custom programs in various languages
- Resulting parsed data from Jira's RESTful API can provide valuable enhancements for analysis

THANK YOU

For more information, contact . . .

Blaze Smallwood

Lead Associate

Booz | Allen | Hamilton

Booz Allen Hamilton Inc.
Mobile 619.850.6123
smallwood_blaze@bah.com

BOOZALLEN.COM
