



```
"WFQuickActionSurfaces": [],
"WfWorkflowClientVersion": "1206.4",
"WfWorkflowHasOutputFallback": false,
"WfWorkflowHasShortcutInputVariables": true,
"WfWorkflowIcon": {
  "WfWorkflowIconStartColor": 4251333119,
  "WfWorkflowIconGlyphNumber": 59727
},
"WfWorkflowImportQuestions": [
  {
    "Category": "Parameter",
    "ParameterKey": "WFTextActionText",
    "ActionIndex": 2,
    "Text": "Provide a GitHub Personal Access Token. (Required)",
    "DefaultValue": ""
  },
  {
    "Category": "Parameter",
    "ParameterKey": "WFItems",
    "ActionIndex": 53,
    "Text": "Set any additional Drafts tags you'd like included beyond the issue's current labels. (Optional.)",
    "DefaultValue": [
      "github",
      "automation"
    ]
  }
],
"WfWorkflowInputContentItemClasses": [
  "WFURLContentItem"
],
"WfWorkflowMinimumClientVersion": 1113,
"WfWorkflowMinimumClientVersionString": "1113",
"WfWorkflowNoInputBehavior": {
  "Name": "WfWorkflowNoInputBehaviorGetClipboard",
  "Parameters": {}
},
"WfWorkflowOutputContentItemClasses": [],
"WfWorkflowTypes": [
  "ActionExtension"
]
```

0 Comment

Retrieve and concatenate GitHub Issue contents and comments into a single draft.

Full documentation: <https://davidblue.wtf/issuedraft>

My contact information:

Email: davidblue@extratone.com  
Contact card: <https://davidblue.wtf/db.vcf>  
RoutineHub Profile: <https://routinehub.co/user/blue>  
GitHub: <https://github.com/extratone>  
iOS-specific Repository: <https://github.com/extratone/i>  
Telegram: <https://t.me/DavidBlue>  
Telegram Channel: <https://t.me/extratone>  
Discord: <https://davidblue.wtf/discord>  
Twitter: <https://twitter.com/NeoYokel>  
Reddit: <https://reddit.com/u/asphaltapostle>  
Mastodon: <https://mastodon.social/@DavidBlue>  
EVERYWHERE: <https://davidblue.wtf/socialdrop>

1 Get Path of URL [Shortcut Input] » 2 9

2 Split Text [1 Component of URL] by Custom » 3 5

TextCustomSeparator || /

3 Get List Item At Index 2 from [2 Split Text] »

4 Set Variable owner to [3 Item from List]

5 Get List Item At Index 3 from [2 Split Text] »

6 Set Variable repo to [5 Item from List]

7 Text »

ghp\_00000000000000000000000000000000

8 Set Variable auth to [7 Text]

9 URL <https://api.github.com/repos>[1 Component of URL] » 10 34

10 Get Contents of URL [9 URL] » 11 13 15 18 21 23 28 32

HTTPHeaders { "Accept": "application/vnd.github+json",  
"Authorization": "Bearer [auth]"  
}  
ShowHeaders true

11 Get Dictionary Value for html\_url in [10 Contents of URL] »

12 Set Variable link to [11 Dictionary Value]

13 Get Dictionary Value for user.login in [10 Contents of URL] »

14 Set Variable username to [13 Dictionary Value]

15 Get Dictionary Value for created\_at in [10 Contents of URL] »

16 Get Dates from [15 Dictionary Value] »

17 Set Variable created to [16 Dates{"Date Style":"Custom","WFDateFormat":"MMddYYYY-HH:mm:ss","WFISO8601IncludeTime":false}]

18 Get Dictionary Value for updated\_at in [10 Contents of URL] »

19 Get Dates from [18 Dictionary Value] »

20 Set Variable updated to [19 Dates{"Date Style":"Custom","WFDateFormat":"MMddYYYY-HH:mm:ss","WFISO8601IncludeTime":false}]

21 Get Dictionary Value for title in [10 Contents of URL] »

22 Set Variable title to [21 Dictionary Value]

23 Get Dictionary Value for labels in [10 Contents of URL] »

24 Repeat with Each in [23 Dictionary Value]

25 Get Dictionary Value for name in [Repeat Item]

26 End Repeat with Each »

27 Set Variable tags to [26 Repeat Results]

28 Get Dictionary Value for body in [10 Contents of URL] »

29 Set Variable body to [28 Dictionary Value]

30 Text »

# "[title]" - [owner]/[repo]  
Updated `[updated]`  
Created `[created]`  
  
- [\*\*Issue\*\*]([link])  
- @[username](https://github.com/[username])  
- [Repository](https://github.com/[owner's File Extension]/[repo])  
- [Maintainer/Org](https://github.com/[owner])  
  
---  
  
[body]

31 Set Variable md to [30 Text]

32 Get Dictionary Value for comments in [10 Contents of URL] »

33 If [32 Dictionary Value as Number] > 0

34 URL [9 URL]/comments »

35 Get Contents of URL [34 URL] »

HTTPHeaders { "Accept": "application/vnd.github+json",  
"Authorization": "Bearer [auth]"  
}  
ShowHeaders true

36 Repeat with Each in [35 Contents of URL]

37 Get Dictionary Value for user.login in [Repeat Item] → association » 43

38 Get Dictionary Value for user.html\_url in [Repeat Item] → authorURL » 43

39 Get Dictionary Value for updated\_at in [Repeat Item] »

40 Get Dates from [39 Dictionary Value]

41 Get Dictionary Value for body in [Repeat Item] → commentContent » 45

42 Get Dictionary Value for html\_url in [Repeat Item] → commentURL »

43 Text »

[\*]([42 commentURL])-[37 association]([38 authorURL])

44 Set Variable ref to [43 Text]

45 Text »

[41 commentContent].  
\*[ref]\*  
  
---

46 Add [45 Text] to Variable comments

47 End Repeat with Each Δ36

48 Combine Text [comments] with New Lines »

TextCustomSeparator || ---

49 Replace Text # with ## in [48 Combined Text] »

ReplaceTextCaseSensitive || false

50 Add [49 Updated Text] to Variable md

51 End If Δ33

52 Combine Text [md] with New Lines

53 List → additionalTags »

[github,  
capture]

54 com.agile tortoise.Drafts5.CaptureIntent » 55 56

ShowWhenRun || false  
tags || [repo],  
[tags],  
[53 additionalTags]  
content | [md]  
Show-tags || true

55 Copy [54 Draft's link] to Clipboard

56 com.agile tortoise.Drafts5.OpenDraftIntent

draft || [54 Draft]  
ShowWhenRun || false