

```
"WFQuickActionSurfaces": [],
"WFWorkflowClientVersion": "1206.4",
"WFWorkflowHasOutputFallback": false,
"WFWorkflowHasShortcutInputVariables": true,
"WFWorkflowIcon": {
  "WFWorkflowIconStartColor": 4251333119,
  "WFWorkflowIconGlyphNumber": 59727
},
"WFWorkflowImportQuestions": [
  {
    "Category": "Parameter",
    "ParameterKey": "WFTextActionText",
    "ActionIndex": 6,
    "Text": "Provide a GitHub Personal Access Token. (Required.)",
    "DefaultValue": "ghp_00000000000000000000000000000000"
  },
  {
    "Category": "Parameter",
    "ParameterKey": "WFItems",
    "ActionIndex": 52,
    "Text": "Specify any additional tags (apart from the Issue's labels) you'd like attached to the result draft. (Optional) ",
    "DefaultValue": [
      "github",
      "documentation"
    ]
  }
],
"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 32

10 Get Contents of URL **[9 URL]** » 11 13 16 19 21 26 30

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 created_at in **[10 Contents of URL]** »

14 Get Dates from **[13 Dictionary Value]** »

15 Set Variable created to **[14 Dates{"Date Style":"Custom","WFDateFormat":"MMddYYYY-HH:mm:ss","WFISO8601IncludeTime":false}]**

16 Get Dictionary Value for updated_at in **[10 Contents of URL]** »

17 Get Dates from **[16 Dictionary Value]** »

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

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

20 Set Variable title to **[19 Dictionary Value]**

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

22 Repeat with Each in **[21 Dictionary Value]**

23 Get Dictionary Value for name in **[Repeat Item]**

24 End Repeat with Each »

25 Set Variable tags to **[24 Repeat Results]**

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

27 Set Variable body to **[26 Dictionary Value]**

28 Text »

[title]
Updated **[updated]**

- **[**GitHub Issue**](**[link]**)**
- Created: **[created]**

[body]

29 Add **[28 Text]** to Variable md

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

31 If **[30 Dictionary Value as Number]** > 0

32 URL **[9 URL]/comments** »

33 Get Contents of URL **[32 URL]** »

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

34 Repeat with Each in **[33 Contents of URL]**

35 Get Dictionary Value for user.login in **[Repeat Item]** → association » 41 43

36 Get Dictionary Value for updated_at in **[Repeat Item]** »

37 Get Dates from **[36 Dictionary Value]** » 40

38 Get Dictionary Value for body in **[Repeat Item]** → commentContent » 40

39 Get Dictionary Value for html_url in **[Repeat Item]** → commentURL »

40 Text » 46

- **[37 Dates{"Date Style":"Custom","WFDateFormat":"MMddYYYY-HH:mm:ss","WFISO8601IncludeTime":false}]](39 commentURL)**

[38 commentContent]

41 If **[35 association as Text's Name]** is not **[owner]**

42 Get Dictionary Value for user.html_url in **[Repeat Item]** → authorURL »

43 Text »

@**[35 association]](42 authorURL)**

44 Add **[43 Text]** to Variable comments

45 End If

46 Add **[40 Text]** to Variable comments

47 End Repeat with Each Δ34

48 Combine Text **[comments]** with New Lines »

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

ReplaceTextCaseSensitive false

50 Add **[49 Updated Text]** to Variable md

51 End If Δ31

52 Combine Text **[md]** with New Lines

53 List → additionalTags »

[github,
capture]

54 com.agiletortoise.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.agiletortoise.Drafts5.OpenDraftIntent

draft **[54 Draft]**
ShowWhenRun false