

24.ZabbixJavaScriptWebhook

Zabbix4.4JavaScriptWebhookZabbixWebhook

ZabbixwebhookwebhookJavaScriptJavaScriptJavaScriptJavaScripthttp

ZabbixJavaScript

ZabbixJavaScriptWebhook

zabbix

Log(loglevel, message)	Zabbix

```
Zabbix.Log(3, "this is a log entry written with 'Warning' log level")
```

http

httphttp

AddHeader(name, value)	HTTPClearHeader()
ClearHeader()	HTTP json Content-Type application/json text/plain
GetHeaders()	HTTP
Get(url, data)	HTTP GET_data_URL
Put(url, data)	HTTP PUTdataURL
Post(url, data)	HTTP POST_data_URL
Delete(url, data)	HTTP DELETEdataURL
Status()	HTTP
SetProxy(proxy)	HTTP"proxy"

```

try {
    Zabbix.Log(4, 'jira webhook script value='+value);

    var result = {
        'tags': {
            'endpoint': 'jira'
        }
    },
    params = JSON.parse(value),
    req = new CurlHttpRequest(),
    fields = {},
    resp;

    req.AddHeader('Content-Type: application/json');
    req.AddHeader('Authorization: Basic '+params.
authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {"key": params.project_key};
    fields.issuetype = {"id": params.issue_id};
    resp = req.Post('https://tsupport.zabbix.lan/rest/api/2
/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.Status() != 201) {
        throw 'Response code: '+req.Status();
    }

    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;
} catch (error) {
    Zabbix.Log(4, 'jira issue creation failed json : '+JSON.
stringify({"fields": fields}));
    Zabbix.Log(4, 'jira issue creation failed : '+error);

    result = {};
}

return JSON.stringify(result);

```

Webhook

webhookgitxmlwebhookwebhook webhookhttpwebhookURL :, HTTPProxy:, To:{ALERT. SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}httpProcess tags__ webhookjson

Webhook

webhook

- Webhook Webhook
- webhook
-

- HTTP
- /JSON/XML/
-
-

webhook webhook OK webhook Process tagswebhook JSON {tags: {}}

Webhooks Zabbix

- webhook
- Zabbix ""3 webhooks
- WEBHOOK webhook Zabbix Server
- WEBHOOK

webhook

webhookwebhook



webhookhttps://oapi.dingtalk.com/robot/send?access_token=e5906a9b879fe615323bd9489f46334f5539d6a46f8e29c5740ba02c7ee90e84



webhookwebhook

```
var Dingding = {
  params: {},
  proxy: null,
  setParams: function (params) {
    if (typeof params !== 'object') {
      return;
    }
    Dingding.params = params;
  },
  request: function () {
    var data = {
      msgtype: "markdown",
      markdown: {
        title: "",
        text: "## :\n " + Dingding.params.Message,
      },
      at: {
        atUserIds: [Dingding.params.To],
        isAtAll: false,
      },
    },
    response,
    url = Dingding.params.URL,
    request = new HttpRequest();

    request.addHeader('Content-Type: application/json');
```

```

        if (typeof Dingding.HTTPProxy !== 'undefined' && Dingding.
HTTPProxy !== '') {
            request.setProxy(Dingding.HTTPProxy);
        }

        if (typeof data !== 'undefined') {
            data = JSON.stringify(data);
        }
        Zabbix.Log(4, "[Dingding Webhook] message is: " + data);
        response = request.post(url, data);

        Zabbix.log(4,
' [ Dingding Webhook ] Received response with status code ' +
            request.getStatus() + '\n' + response);

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4,
' [ Dingding Webhook ] Failed to parse response received from Dingding'
);
                response = null;
            }
        }

        if (request.getStatus() !== 200 || response.errcode !== 0) {
            var message = 'Request failed with status code '+request.
getStatus();

            if (response !== null && typeof response.errmsg !==
'undefined') {
                message += ': ' + JSON.stringify(response.errmsg) ;
            }

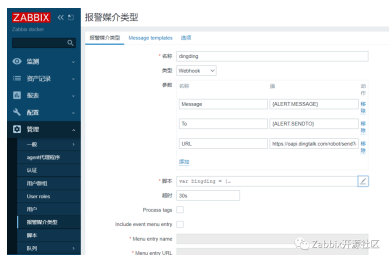
            throw message + '. Check debug log for more information.';
        }

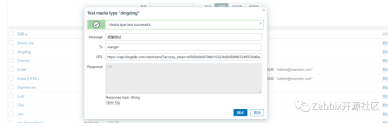
        return response;
    },
};

try {
    var params = JSON.parse(value);

    if (typeof params.URL !== 'undefined'
&& typeof params.To !== 'undefined'
&& typeof params.Message !== 'undefined') {
        Zabbix.log(4, '[ Dingding Webhook ] webhookURL "' + params.
URL +
            '" sendto "' + params.To )+'";
    }
    else {
        throw
'Missing parameter. URL, message, to parameter is required'
    }
    if (params.HTTPProxy) {
        Dingding.proxy = params.HTTPProxy;
    }
    Dingding.setParams(params);
    Dingding.request();
    return 'OK';
} catch (error) {
    Zabbix.log(3, '[ Dingding Webhook ] ERROR: ' + error);
    throw 'Sending failed: ' + error;
}
}

```





markdownMarkdown



#Zabbix#ZabbixJavaScriptWebhook #Zabbix

Wanger | Zabbix