

100-使用Postman测试JIRA接口

场景

使用接口，查询一个问题的详情

接口

/api/2/issue/{issueIdOrKey}

方法

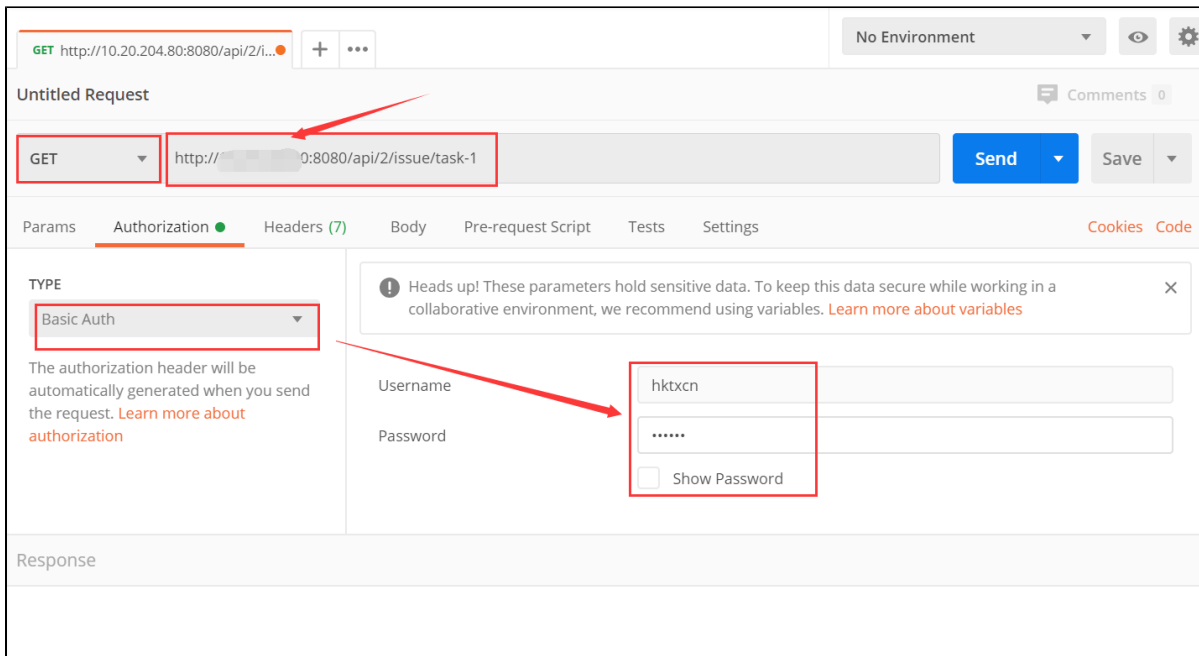
GET

参数

*issueIdOrKey		问题编号或者此问题的ID值
fields	文本	以, 号分割的字段list。默认会返回所有的字段信息
expand	文本	以, 号分割的，查询更多信息的指定信息。可包括以下项 - renderedFields - 以HTML显示的字段信息 - names - 显示每一个字段的名字 - schema - 显示每一个字段它对应字段类型的配置项 - transitions - 针对指定的问题显示所有可能的工作流动作项 - operations - 在此问题中显示此所有可能的操作项 - editmeta - 显示每一个字段是否可以被编辑的信息。它同时也包括在字段的配置信息项中. - changelog - 指定问题的所有变更历史 - versionedRepresentations -
properties	文本	当前问题属性的一个list。默认情况下不会返回此属性信息
updateHistory	boolean	是否返回更新历史信息

Postmain配置

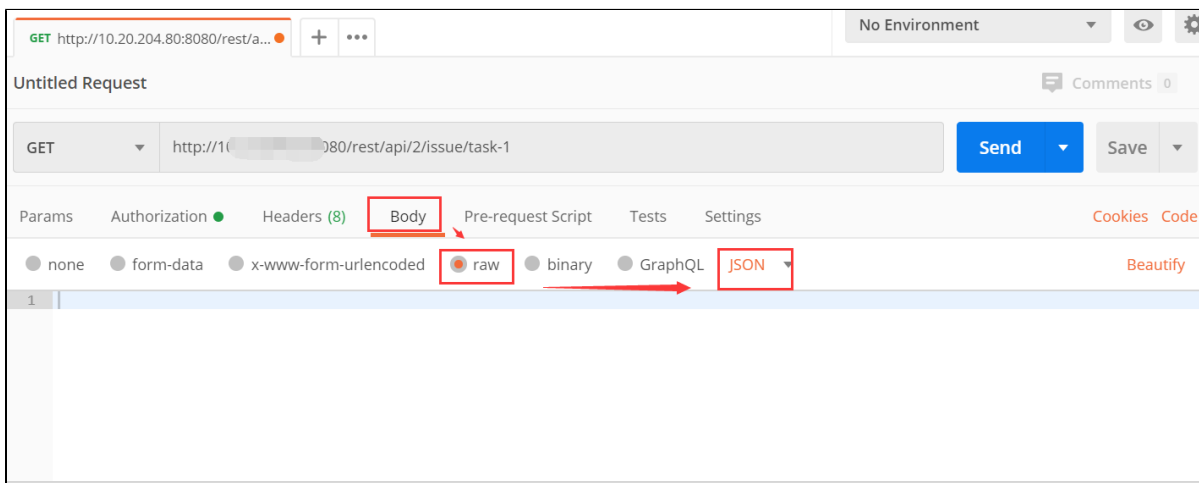
配置请求数据



- 选择调用方法，填写调用地址
- 在Authorization中，选择类型为：Base Auth
- 在展示出的参数中，填写登录jira的账户信息，包括账户和密码

配置接受数据格式

配置完成 信息后，配置获得的数据格式，我们选择JSON



查看调用结果

配置没有问题，我们可以点击“send”，来查看获得的结果信息

GET http://10.20.204.80:8080/rest/a...

No Environment

GET

http://10.20.204.80:8080/rest/api/2/issue/task-1

Send

Save

Body Cookies (2) Headers (15) Test Results

Status: 200 OK Time: 1928 ms Size: 15.71 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "expand": "renderedFields,names,schema,operations,editmeta,changelog,versionedRepresentations",
3   "id": "10029",
4   "self": "http://10.20.204.80:8080/rest/api/2/issue/10029",
5   "key": "TASK-1",
6   "fields": {
7     "issuetype": {
8       "self": "http://10.20.204.80:8080/rest/api/2/issuetype/10001",
9       "id": "10001",
10      "description": "Created by Jira Agile - do not edit or delete. Issue type for a user story.",
11      "iconUrl": "http://10.20.204.80/images/icons/issuetypes/story.svg",
12      "name": "Story",
```