Http-Request

Install the plugin

```
HTTP Request Plugin 1.18

This plugin sends a http request to an url with some parameters Report an issue with this plugin

This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
```

In a Pipeline job, you have total control over how the url is formed. Suppose you have a build parameter called "param1", you can pass it to the HTTP request programmatically like so:

```
httpRequest "http://httpbin.org/response-headers?param1=${param1}"
```

If you wish to save the response to a file, you need to grab a workspace. You can do this with a node Pipeline step. For example:

```
def response = httpRequest "http://httpbin.org/response-headers?param1=${param1}"
node() {
    writeFile file: 'response.txt', text: response.content
}
```

You can access the response status code, content and headers programmatically:

```
def response = httpRequest "http://httpbin.org/response-headers?param1=${param1}"
println("Status: ${response.status}")
println("Response: ${response.content}")
println("Headers: ${response.headers}")
```

```
pipeline {
   agent any

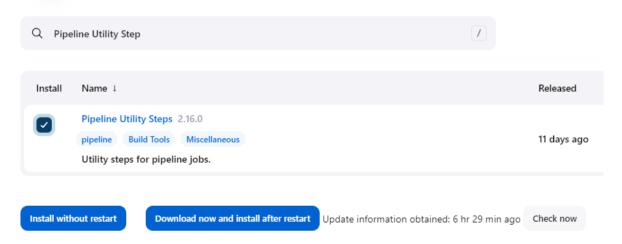
parameters {
    string(name: 'param1', defaultValue: 'example-value', description: 'Parameter 1')
}

stages {
```

```
stage('HTTP Request') {
    steps {
        script {
            def response = httpRequest "http://httpbin.org/response-headers?param1=${param1}"
            writeFile file: 'response.txt', text: response.content

            println("Status: ${response.status}")
            println("Response: ${response.content}")
            println("Headers: ${response.headers}")
            }
        }
    }
}
```

Plugins



```
pipeline {
  agent any
  stages {
    stage('Get Organizations') {
       steps {
         script {
           def username = "amitopenwriteup"
           // Send GET request to retrieve organizations
           def response = httpRequest(
             url: "https://api.github.com/users/${username}/orgs"
           )
           // Process the response
           if (response.status == 200) {
             def orgs = readJSON(text: response.content)
             for (org in orgs) {
               println("Organization Name: ${org.login}")
             }
           } else {
             println("Failed to retrieve organizations. Status: ${response.status}")
           }
         }
      }
    }
  }
}
```