

```
1 import discord
2 import time
3 from selenium import webdriver
4 from discord.ext import commands, tasks
5
6 refresher = webdriver.Chrome('/Users/admin/Desktop/Gabriel Stefanecs
    Folder/chromedriver')
7
8 #refresher = webdriver.Chrome('/Users/stefa/Documents/chromedriver_win32/
    chromedriver')
9
10 refresher.get("https://webinterface.nitrado.net/8944851/wi/gameserver/")
11
12 client = commands.Bot(command_prefix="!")
13
14 @client.event
15 async def on_ready():
16     taskFunction.start()
17     print("Bot is ready")
18
19 @tasks.loop(seconds = 1)
20 async def taskFunction():
21     # For using Chrome
22
23     try:
24
25         try:
26             main_page = refresher.current_window_handle
27
28             cookieClicker = addButton =
                refresher.find_element_by_class_name("cb_button-dark")
29             cookieClicker.click()
30
31             loginMenu = addButton = refresher.find_element_by_class_name
                ("ni_user-dropdown")
32             loginMenu.click()
33
34             time.sleep(5)
35
36             # changing the handles to access login page
37             for handle in refresher.window_handles:
38                 if handle != main_page:
39                     login_page = handle
40
41             # change the control to signin page
42             refresher.switch_to.window(login_page)
43
44             username = refresher.find_element_by_id('username')
45             username.send_keys("REDACTED")
```

```
46
47     password = refresher.find_element_by_id('password')
48     password.send_keys("REDACTED")
49
50     time.sleep(5)
51
52     submitButton = refresher.find_element_by_class_name("mb-3")
53     submitButton.click()
54
55     time.sleep(6)
56
57     # change control to main page
58     refresher.switch_to.window(main_page)
59
60     accountMenu = addButton = refresher.find_element_by_class_name ↗
61     ("ni_user-dropdown")
62     accountMenu.click()
63
64     myServices = addButton = refresher.find_element_by_xpath("//* ↗
65     [@id=\"__layout\"]/div/nav/div[2]/div[4]/div[2]/div/div/ul/ ↗
66     li[4]/a")
67     myServices.click()
68
69     gameServer = addButton = ↗
70     refresher.find_element_by_css_selector('[href="/usa/ ↗
71     services/index#gameserver"]')
72     gameServer.click()
73
74     settings = addButton = refresher.find_element_by_class_name ↗
75     ("pull-right")
76     settings.click()
77
78     time.sleep(10)
79
80     refresher.switch_to.window(refresher.window_handles[1])
81
82     #refreshBool = False
83
84     #while not refreshBool:
85
86     # Red Circle \U0001F534
87     # Green Circle \U0001F7E2
88     # Arrows Counterclockwise \U0001F504
89
90     status = refresher.find_element_by_id('game-header')
91
92     online = "Server started"
93
94     offline = "Server stopped"
```

```
89
90     stopping = "Stopping server..."
91
92     restarting = "Restarting..."
93
94     if offline in status.text:
95
96         channel = client.get_channel(853748740224712754)
97         embed = discord.Embed(title="Server Status", description =
98             ":red_circle: Offline", color = 0xe74c3c)
99         message = await channel.send(embed = embed)
100         await message.add_reaction('\U0001F7E2')
101         await client.change_presence(status = discord.Status.dnd,
102             activity = discord.Activity(type =
103                 discord.ActivityType.watching, name = "Server:
104                 Offline"))
105
106     elif online in status.text:
107
108         channel = client.get_channel(853748740224712754)
109         embed = discord.Embed(title="Server Status", description =
110             ":green_circle: Online", color = 0x2ecc71)
111         message = await channel.send(embed = embed)
112         await message.add_reaction('\U0001F534')
113         await message.add_reaction('\U0001F504')
114         await client.change_presence(status =
115             discord.Status.online, activity = discord.Activity(type
116                 = discord.ActivityType.watching, name = "Server:
117                 Online"))
118
119     elif stopping in status.text:
120
121         channel = client.get_channel(853748740224712754)
122         embed = discord.Embed(title="Server Status", description =
123             ":octagonal_sign: Stopping...", color = 0xe67e22)
124         message = await channel.send(embed = embed)
125         await client.change_presence(status = discord.Status.idle,
126             activity = discord.Activity(type =
127                 discord.ActivityType.watching, name = "Server:
128                 Stopping"))
129
130     elif restarting in status.text:
131
132         channel = client.get_channel(853748740224712754)
133         embed = discord.Embed(title="Server Status", description =
134             ":arrows_counterclockwise: Restarting...", color =
135             0xf1c40f)
136         message = await channel.send(embed = embed)
137         await client.change_presence(status = discord.Status.idle,
```

```
        activity = discord.Activity(type =  
        discord.ActivityType.watching, name = "Server:  
        Restarting"))  
124  
125     else:  
126  
127         channel = client.get_channel(853748740224712754)  
128         embed = discord.Embed(title="Server Status", description =  
        ":grey_question: Unknown", color = 0x979c9f)  
129         message = await channel.send(embed = embed)  
130  
131     except:  
132  
133         time.sleep(10)  
134  
135         channel = client.get_channel(853748740224712754)  
136  
137         await channel.purge(limit = 1000)  
138  
139         status = refresher.find_element_by_id('game-header')  
140  
141         online = "Server started"  
142  
143         offline = "Server stopped"  
144  
145         stopping = "Stopping server..."  
146  
147         restarting = "Restarting..."  
148  
149         if offline in status.text:  
150  
151             embed = discord.Embed(title="Server Status", description =  
        ":red_circle: Offline", color = 0xe74c3c)  
152             message = await channel.send(embed = embed)  
153             await message.add_reaction('\U0001F7E2')  
154             await client.change_presence(status = discord.Status.dnd,  
        activity = discord.Activity(type =  
        discord.ActivityType.watching, name = "Server:  
        Offline"))  
155  
156         elif online in status.text:  
157  
158             embed = discord.Embed(title="Server Status", description =  
        ":green_circle: Online", color = 0x2ecc71)  
159             message = await channel.send(embed = embed)  
160             await message.add_reaction('\U0001F534')  
161             await message.add_reaction('\U0001F504')  
162             await client.change_presence(status =  
        discord.Status.online, activity = discord.Activity(type =
```

```
        = discord.ActivityType.watching, name = "Server:
        Online"))
163
164     elif stopping in status.text:
165
166         embed = discord.Embed(title="Server Status", description =
        ":octagonal_sign: Stopping...", color = 0xe67e22)
167         message = await channel.send(embed = embed)
168         await client.change_presence(status = discord.Status.idle,
        activity = discord.Activity(type =
        discord.ActivityType.watching, name = "Server:
        Stopping"))
169
170     elif restarting in status.text:
171
172         embed = discord.Embed(title="Server Status", description =
        ":arrows_counterclockwise: Restarting...", color =
        0xf1c40f)
173         message = await channel.send(embed = embed)
174         await client.change_presence(status = discord.Status.idle,
        activity = discord.Activity(type =
        discord.ActivityType.watching, name = "Server:
        Restarting"))
175
176     else:
177
178         embed = discord.Embed(title="Server Status", description =
        ":grey_question: Unknown", color = 0x979c9f)
179         message = await channel.send(embed = embed)
180         await client.change_presence(status = discord.Status.idle,
        activity = discord.Activity(type =
        discord.ActivityType.watching, name = "Server:
        Unknown"))
181
182 except:
183
184     refresher.get("https://webinterface.nitrado.net/8944851/wi/
        gameserver/")
185
186 # Command Code
187
188 @client.command()
189 async def on(ctx):
190     taskFunction.stop()
191
192     try:
193
194         status = refresher.find_element_by_id('game-header')
195
```

```
196         offline = "Server stopped"
197
198         if offline in status.text:
199
200             await ctx.send("Turning On Server.")
201
202             startServer = addButton = refresher.find_element_by_xpath('/  ↗
203                 html/body/div[4]/div[2]/div/button[3]')
204             startServer.click()
205
206             time.sleep(3)
207
208             startServerConfirm = addButton =
209                 refresher.find_element_by_xpath('/html/body/div[6]/div/div/  ↗
210                 div[2]/button[2]')
211             startServerConfirm.click()
212
213         else:
214
215             await ctx.send("You can only turn on the server when it is  ↗
216                 offline.")
217
218             time.sleep(1)
219
220             taskFunction.restart()
221
222     except:
223
224         await channel.send("An error has occurred, please try again.")
225
226         taskFunction.restart()
227
228 @client.command()
229 async def off(ctx):
230     taskFunction.stop()
231
232     try:
233
234         status = refresher.find_element_by_id('game-header')
235
236         online = "Server started"
237
238         if online in status.text:
239
240             await ctx.send("Turning Off Server.")
241
242             stopServer = addButton = refresher.find_element_by_xpath('/  ↗
243                 html/body/div[4]/div[2]/div/button[1]')
244             stopServer.click()
```

```
240
241     time.sleep(3)
242
243     stopServerConfirm = addButton =
244         refresher.find_element_by_xpath('/html/body/div[6]/div/div/
245             div[2]/button[2]')
246     stopServerConfirm.click()
247
248     taskFunction.restart()
249
250     else:
251         await ctx.send("You can only turn off the server when it is
252             online.")
253
254         time.sleep(1)
255
256         taskFunction.restart()
257
258     except:
259         await channel.send("An error has occurred, please try again.")
260
261         taskFunction.restart()
262
263 @client.command()
264 async def restart(ctx):
265     taskFunction.stop()
266
267     try:
268         status = refresher.find_element_by_id('game-header')
269
270         online = "Server started"
271
272         if online in status.text:
273             await ctx.send("Restarting Server.")
274
275             restartServer = addButton = refresher.find_element_by_xpath('/
276                 html/body/div[4]/div[2]/div/button[2]')
277             restartServer.click()
278
279             time.sleep(3)
280
281             restartServerConfirm = addButton =
282                 refresher.find_element_by_xpath('/html/body/div[6]/div/div/
283                     div[2]/button[2]')
```

```
283         restartServerConfirm.click()
284
285         taskFunction.restart()
286
287     else:
288
289         await ctx.send("You can only restart the server when it is online.")
290
291         time.sleep(1)
292
293         taskFunction.restart()
294
295     except:
296
297         await channel.send("An error has occurred, please try again.")
298
299         taskFunction.restart()
300
301     # Emoji Code
302
303     @client.event
304     async def on_reaction_add(reaction, user):
305
306         try:
307
308             if user == client.user:
309                 return
310
311             taskFunction.stop()
312
313             status = refresher.find_element_by_id('game-header')
314
315             offline = "Server stopped"
316
317             online = "Server started"
318
319             channel = client.get_channel(853748740224712754)
320
321             if reaction.emoji == '\U0001F7E2':
322
323                 if offline in status.text:
324
325                     await channel.send("Turning On Server.")
326
327                     startServer = addButton = refresher.find_element_by_xpath
328                     ('/html/body/div[4]/div[2]/div/button[3]')
329                     startServer.click()
```



```
330         time.sleep(3)
331
332         startServerConfirm = addButton =
333             refresher.find_element_by_xpath('/html/body/div[6]/div/
334             div/div[2]/button[2]')
335         startServerConfirm.click()
336
337         time.sleep(1)
338
339     else:
340
341         await channel.send("Don't do that.:angry:")
342
343         time.sleep(1)
344
345     elif reaction.emoji == '\U0001F534':
346
347         if online in status.text:
348
349             await channel.send("Turning Off Server.")
350
351             startServer = addButton = refresher.find_element_by_xpath
352             ('/html/body/div[4]/div[2]/div/button[1]')
353             startServer.click()
354
355             time.sleep(3)
356
357             startServerConfirm = addButton =
358                 refresher.find_element_by_xpath('/html/body/div[6]/div/
359                 div/div[2]/button[2]')
360             startServerConfirm.click()
361
362             time.sleep(1)
363
364         else:
365
366             await channel.send("Don't do that.:angry:")
367
368             time.sleep(1)
369
370     elif reaction.emoji == '\U0001F504':
371
372         if online in status.text:
373
374             await channel.send("Restarting Server.")
375
376             restartServer = addButton =
377                 refresher.find_element_by_xpath('/html/body/div[4]/div
378                 [2]/div/button[2]')
```

```
372         restartServer.click()
373
374         time.sleep(3)
375
376         restartServerConfirm = addButton =
            refresher.find_element_by_xpath('/html/body/div[6]/div/
            div/div[2]/button[2]')
377         restartServerConfirm.click()
378
379         time.sleep(1)
380
381     else:
382
383         await channel.send("Don't do the that.:angry:")
384
385         time.sleep(1)
386
387     else:
388
389         await channel.send("Don't do the that.:angry:")
390
391         time.sleep(1)
392
393     except:
394
395         await channel.send("An error has occurred, please try again.")
396
397     taskFunction.restart()
398
399 client.run("REDACTED")
400
401
402
403
```