

```
1  import gi
2  gi.require_version('Gtk', '3.0')
3  from gi.repository import Gtk
4  import json
5  from os import path
6
7  class Setting(Gtk.Window):
8      def __init__(self):
9          self.importObject()
10         self.builder.connect_signals(self.signals())
11         self.setValues(self.settingJson(self.defaultSetting()))
12         self.lists = self.fetchSetting()["ddclientList"]
13         self.ddclientListInit(self.lists)
14
15
16     def importObject(self):
17         self.builder = Gtk.Builder()
18         self.builder.add_from_file("gui/setting.glade")
19         self.settingWindow = self.builder.get_object("settingWindow")
20
21         self.settingCancel = self.builder.get_object("settingCancel")
22         self.settingOk = self.builder.get_object("settingOk")
23
24         self.startWithGemp = self.builder.get_object("startWithGemp")
25         self.stopWithGemp = self.builder.get_object("stopWithGemp")
26         self.atStartup = self.builder.get_object("atStartup")
27
28         self.nginxPortEntry = self.builder.get_object("nginxPortEntry")
29         self.mysqlPortEntry = self.builder.get_object("mysqlPortEntry")
30         self.setDefaultPortButton =
31         self.builder.get_object("setDefaultPortButton")
32
33         self.ngIndexEntry = self.builder.get_object("ngIndexEntry")
34         self.ngTryFileEntry = self.builder.get_object("ngTryFileEntry")
35         self.ngCustomEntry = self.builder.get_object("ngCustomEntry")
36         self.gzipCompression = self.builder.get_object("gzipCompression")
37         self.wptweak = self.builder.get_object("wptweak")
38         self.nginxlogs = self.builder.get_object("nginxlogs")
39         self.ngFileSetDefault = self.builder.get_object("ngFileSetDefault")
40
41         self.mysqlPort = self.builder.get_object("mysqlPort")
42         self.mysqlChangePassButton =
43         self.builder.get_object("mysqlChangePassButton")
44         self.adminerButton = self.builder.get_object("adminerButton")
45         self.adminerButton.set_label("Adminer")
46         self.mysqlLogEntry = self.builder.get_object("mysqlLogEntry")
47         self.mysqlDefaultLogButton =
48         self.builder.get_object("mysqlDefaultLogButton")
49
50         self.phpMaxUploadLimit = self.builder.get_object("phpMaxUploadLimit")
51         self.phpMaxNo = self.builder.get_object("phpMaxNo")
52
53         self.ddProtocolEntry = self.builder.get_object("ddProtocolEntry")
54         self.ddServiceEntry = self.builder.get_object("ddServiceEntry")
55         self.ddlogin = self.builder.get_object("ddlogin")
56         self.ddpassword = self.builder.get_object("ddlogin")
57         self.ddclientClient = self.builder.get_object("ddclientClient")
58         self.ddclientAdd = self.builder.get_object("ddclientAdd")
59         self.ddclientRemove = self.builder.get_object("ddclientRemove")
60         self.ddclientTreeBox = self.builder.get_object("ddclientTreeBox")
61         self.ddclientList = []
```

```
60         self.treeView()
61
62     def onSelect(self, selection):
63         model, iter = selection.get_selected()
64         if iter is not None:
65             self.ddclientSeleted = model[iter][0]
66
67     def treeView(self):
68         self.listStore = Gtk.ListStore(str)
69         self.ddclientTreeView = Gtk.TreeView(model=self.listStore)
70         self.selection = self.ddclientTreeView.get_selection()
71         self.selection.connect("changed", self.onSelect)
72         self.ddclientTreeBox.add(self.ddclientTreeView)
73         render = Gtk.CellRendererText()
74         treeViewColumn = Gtk.TreeViewColumn('Client', render, text=0)
75         self.ddclientTreeView.append_column(treeViewColumn)
76
77     def ddclientAddClicked(self, widget):
78         if self.ddclientClient.get_text() != "" and not
79         (self.ddclientClient.get_text() in self.ddclientList):
80             self.ddclientList.append(self.ddclientClient.get_text())
81             self.listStore.append([self.ddclientClient.get_text()])
82
83     def ddclientListInit(self, List):
84         for list in List:
85             print(list)
86             self.listStore.append([list])
87
88     def signals(self):
89         signals = {
90             "delete-event": (self.quit, True),
91             "okButtonPressed": self.okButtonPressed,
92             "cancelButtonPressed": (self.quit),
93             "setDefaultPortButtonPressed": self.setDefaultPortButtonPressed,
94             "mysqlDefaultLogButtonClicked": self.mysqlDefaultLogButtonClicked,
95             "ngFileSetDefaultClicked": self.ngFileSetDefaultClicked,
96             "ddclientAddClicked": self.ddclientAddClicked,
97             "ddclientRemoveClicked": self.ddclientRemoveClicked
98         }
99         return signals
100
101     def setValue(self, setting):
102         self.startWithGemp.set_active(setting['startWithGemp'])
103         self.stopWithGemp.set_active(setting["stopWithGemp"])
104         self.atStartup.set_active(setting["atStartup"])
105
106         self.nginxPortEntry.set_text(str(setting["nginxPortEntry"]))
107         self.mysqlPortEntry.set_text(str(setting["mysqlPortEntry"]))
108
109         self.ngIndexEntry.set_text(setting["ngIndexEntry"])
110         self.ngTryFileEntry.set_text(setting["ngTryFileEntry"])
111         self.ngCustomEntry.set_text(setting["ngCustomEntry"])
112         self.wptweak.set_active(setting["wptweak"])
113         self.zipCompression.set_active(setting["gzipCompression"])
114         self.nginxlogs.set_text(setting["nginxlogs"])
115
116         self.mysqlPort.set_text(str(setting["mysqlPort"]))
117         self.mysqlLogEntry.set_text(setting["mysqlLogEntry"])
118
119         self.phpMaxUploadLimit.set_text(str(setting["phpMaxUploadLimit"]))
120         self.phpMaxNo.set_text(str(setting["phpMaxNo"]))
```

```
121
122     self.ddProtocolEntry.set_text(setting["ddProtocolEntry"])
123     self.ddServiceEntry.set_text(setting["ddServiceEntry"])
124     self.ddlogin.set_text(setting["ddlogin"])
125     self.ddPassword.set_text(setting["ddPassword"])
126     self.ddclientList = setting["ddclientList"]
127
128
129     def defaultSetting(self):
130         defaultSetting = {
131             "startWithGemp": True,
132             "stopWithGemp": False,
133             "atStartup": False,
134
135             "nginxPortEntry": 80,
136             "mysqlPortEntry": 3306,
137
138             "ngIndexEntry": "index.php index.htm index.html",
139             "ngTryFileEntry": "$uri $uri/ =404",
140             "ngCustomEntry": " ",
141             "gzipCompression": True,
142             "wptweak": True,
143             "nginxlogs": "/etc/nginx/error_logs.txt",
144
145             "mysqlPort": 3306,
146             "mysqlLogEntry": "/var/log/mysql/error.log",
147
148             "phpMaxUploadLimit": 120,
149             "phpMaxNo": 20,
150
151             "ddProtocolEntry": " ",
152             "ddServiceEntry": " ",
153             "ddlogin": " ",
154             "ddPassword": " ",
155             "ddclientList": []
156         }
157         return defaultSetting
158
159     def fetchSetting(self):
160         return {
161             "startWithGemp": self.startWithGemp.get_active(),
162             "stopWithGemp": self.stopWithGemp.get_active(),
163             "atStartup": self.atStartup.get_active(),
164
165             "nginxPortEntry": int(self.nginxPortEntry.get_text()),
166             "mysqlPortEntry": int(self.mysqlPortEntry.get_text()),
167
168             "ngIndexEntry": self.ngIndexEntry.get_text(),
169             "ngTryFileEntry": self.ngTryFileEntry.get_text(),
170             "ngCustomEntry": self.ngCustomEntry.get_text(),
171             "gzipCompression": self.gzipCompression.get_active(),
172             "wptweak": self.wptweak.get_active(),
173             "nginxlogs": self.nginxlogs.get_text(),
174
175             "mysqlPort": self.mysqlPort.get_text(),
176             "mysqlLogEntry": self.mysqlLogEntry.get_text(),
177
178             "phpMaxUploadLimit": int(self.phpMaxUploadLimit.get_text()),
179             "phpMaxNo": int(self.phpMaxNo.get_text()),
180
181             "ddProtocolEntry": self.ddProtocolEntry.get_text(),
182             "ddServiceEntry": self.ddServiceEntry.get_text(),
```

```
183         "ddlogin":self.ddlogin.get_text(),
184         "ddPassword":self.ddPassword.get_text(),
185         "ddclientList": self.ddclientList
186     }
187
188
189     def writeJson(self, string):
190         print(string)
191         with open("setting.json", "w") as file:
192             s = json.dumps(string)
193             file.write(s)
194             print("Writing Setting Done")
195
196
197     def settingJson(self, setting):
198         print("Checking File...!")
199         if (path.isfile("setting.json")):
200             print("File Exists")
201             with open("setting.json", "r") as file:
202                 setting = json.load(file)
203         else:
204             print("File Not Found, Creating a new one.")
205             with open("setting.json", "w+") as file:
206                 setting = json.dumps(setting)
207                 file.write(setting)
208                 setting = json.loads(setting)
209             return setting
210
211     def nginxConfig(self, setting):
212         string = """
213         #GEMP Server Nginx Configuration File
214         server {{
215             listen {{;
216             server_name _;
217             root {{;
218             index {{;
219             location / {{
220                 try_files {{;
221             }}
222             access_log {{;
223         }}
224         """.format(int(setting['nginxPortEntry']), "/var/www/html/",
225         setting["ngIndexEntry"], "$uri $uri/ =404", setting["nginxlogs"])
226         with open("/etc/nginx/sites-enabled/GEMP.conf", "w") as file:
227             file.write(string)
228
229     def mysqlConfig(self, setting):
230         with open('mysqld.cnf.sample', 'r') as f:
231             s = f.read().replace('{port}', str(setting['mysqlPortEntry']))
232             s = s.replace('{log}', str(setting['mysqlLogEntry']))
233         with open('/etc/mysql/mysql.conf.d/mysqld.cnf', 'w') as f:
234             f.write(s)
235
236     def phpConfig(self, setting):
237         with open('php.ini.sample', 'r') as f:
238             s = f.read().replace("{maxFileSize}",
239         str(setting["phpMaxUploadLimit"]))
240             s = s.replace("{maxFileNo}", str(setting["phpMaxNo"]))
241         with open("/etc/php/7.1/fpm/php.ini", "w") as f:
242             f.write(s)
243
244     def ddclient(self, setting):
```

```
243         string = ""
244         # Configuration file for ddclient generated by GEMP
245         #
246         # /etc/ddclient.conf
247
248         daemon=3600
249         protocol={}
250         server={}
251         login={}
252         password={}
253         client={}
254         """.format(setting["ddProtocolEntry"], setting["ddServiceEntry"],
255         setting["ddlogin"], setting["ddPassword"], ','.join(setting["ddclientList"]))
256         with open("config/ddclient.conf", "w") as f:
257             f.write(string)
258
259     def ddclientRemoveClicked(self, widget):
260         self.ddclientList.remove(self.ddclientSeleted)
261         self.listStore.clear()
262         self.ddclientListInit(self.lists)
263
264     def okButtonPressed(self, widget):
265         self.writeJson(self.fetchSetting())
266
267         self.nginxConfig(self.fetchSetting())
268
269         self.mysqlConfig(self.fetchSetting())
270
271         self.phpConfig(self.fetchSetting())
272
273         self.ddclient(self.fetchSetting())
274
275         self.quit(True)
276         print("OK")
277
278     def setDefaultPortButtonPressed(self, widget):
279         self.nginxPortEntry.set_text("80")
280         self.mysqlPortEntry.set_text("3306")
281
282     def mysqlDefaultLogButtonClicked(self, widget):
283         self.mysqlLogEntry.set_text(self.defaultSetting()["mysqlLogEntry"])
284         print(self.defaultSetting()["mysqlLogEntry"])
285
286     def ngFileSetDefaultClicked(self, widget):
287         self.ngIndexEntry.set_text(self.defaultSetting()["ngIndexEntry"])
288         self.ngTryFileEntry.set_text(self.defaultSetting()["ngTryFileEntry"])
289         self.ngCustomEntry.set_text(self.defaultSetting()["ngCustomEntry"])
290         self.wptweak.set_active(self.defaultSetting()["wptweak"])
291         self.gzipCompression.set_active(self.defaultSetting()
292         ["gzipCompression"])
293         self.nginxlogs.set_text(self.defaultSetting()["nginxlogs"])
294
295     def main(self):
296         self.settingWindow.show_all()
297
298     def quit(self, widget):
299         self.settingWindow.close()
```