

# Website Builder Addon Engine Documentation

## 1. Addon Folder

All addons go inside:

```
data/addons/
```

Example:

```
data/addons/example.py
data/addons/green_theme.py
data/addons/test_gui.py
```

## 2. Required Addon Header

Every addon file must start with this exact first line:

```
#WEBSITEBUILDERADDONENGINE
```

If this line is missing, the app will skip the addon.

## 3. Required INIT Function

Every addon must have an `INIT(api)` function:

```
#WEBSITEBUILDERADDONENGINE

def INIT(api):
    print("Addon loaded")
```

The app calls `INIT(api)` when it loads the addon.

## 4. Basic Button Addon

```
#WEBSITEBUILDERADDONENGINE
from PyQt5.QtWidgets import QMessageBox

def INIT(api):
    def clicked():
        QMessageBox.information(api.window, "Addon", "Hello from addon!")
```

```
api.add_addon_button("Hello Addon", clicked)
```

This adds a button to the addon sidebar.

## 5. Insert Text Into Current Editor

```
#WEBSITEBUILDERADDONENGINE

def INIT(api):
    def insert_html():
        editor = api.current_editor()
        if editor:
            editor.insertPlainText("<h1>Hello from addon</h1>")

    api.add_addon_button("Insert HTML", insert_html)
```

## 6. Create a New Tab

```
#WEBSITEBUILDERADDONENGINE

def INIT(api):
    def new_page():
        api.add_tab("addon.html", "<h1>New addon page</h1>")

    api.add_addon_button("New Addon Page", new_page)
```

## 7. Change App Theme

```
#WEBSITEBUILDERADDONENGINE

def INIT(api):
    def green_theme():
        api.apply_full_style("""
        QMainWindow, QWidget, QDialog {
            background-color: #061a10;
            color: #d6ffe6;
        }

        QPushButton {
            background-color: #0f3d25;
            color: white;
            border: 1px solid #22aa55;
            border-radius: 6px;
            padding: 5px;
        }
        """)
```

```

    QPlainTextEdit, QTextEdit, QLineEdit {
        background-color: #03100a;
        color: #b6ffcc;
        border: 1px solid #22aa55;
    }
    """

api.add_addon_button("Green Theme", green_theme)

```

## 8. Create a Custom Addon Window

Do not create another `QApplication`. The main app already has one.

Correct:

```

#WEBSITEBUILDERADDONENGINE
from PyQt5.QtWidgets import QWidget, QPushButton, QVBoxLayout

def INIT(api):
    windows = []

    def show_window():
        window = QWidget()
        window.setWindowTitle("Addon Window")

        layout = QVBoxLayout()
        layout.addWidget(QPushButton("Top"))
        layout.addWidget(QPushButton("Bottom"))

        window.setLayout(layout)
        window.resize(250, 150)
        window.show()

        windows.append(window)

    api.add_addon_button("Show Window", show_window)

```

Wrong:

```
app = QApplication([])
```

Only one `QApplication` should exist.

## 9. Disable Addons

Create this file:

```
data/addons/.disable
```

Write addon filenames line by line:

```
example.py  
bad_theme.py  
test_gui.py
```

The app will not load addons listed there.

## 10. Comments in `.disable`

You can use comments:

```
# disabled because it crashes  
bad_theme.py  
  
# testing later  
old_addon.py
```

Empty lines are ignored.

## 11. Available API Functions

The addon receives `api`.

### Add button to addon sidebar

```
api.add_addon_button("Button Text", callback)
```

### Add button to normal sidebar

```
api.add_sidebar_button("Button Text", callback)
```

### Get current editor

```
editor = api.current_editor()
```

### Add a new editor tab

```
api.add_tab("name.html", "<h1>Hello</h1>")
```

### Change status label

```
api.log("Addon loaded")
```

### Change whole app style

```
api.apply_full_style(css)
```

### Add extra CSS to app

```
api.add_app_style(css)
```

### Set editor style

```
api.set_editor_style(css)
```

### Change font

```
api.set_font("Consolas", 14)
```

## 12. Direct App Access

Addons can access important app parts:

```
api.window  
api.app  
api.tabs  
api.toolbar  
api.sidebar_layout  
api.addon_layout  
api.commandentry  
api.infolabel  
api.tree  
api.model
```

Example:

```
#WEBSITEBUILDERADDONENGINE  
  
def INIT(api):  
    api.window.setWindowTitle("Website Builder - Modified by Addon")
```

## 13. Safety Warning

Addons are normal Python files. They can:

```
read files
delete files
change the GUI
run programs
access variables
break the app
```

Only install addons you trust.

## 14. Recommended Addon Template

```
#WEBSITEBUILDERADDONENGINE

ADDON_NAME = "My Addon"
ADDON_VERSION = "1.0"
ADDON_AUTHOR = "Your Name"

def INIT(api):
    api.log(f"{ADDON_NAME} loaded")

    def run():
        editor = api.current_editor()
        if editor:
            editor.insertPlainText("<!-- Added by addon -->")

    api.add_addon_button("Run My Addon", run)
```

## 15. Example Full Addon

```
#WEBSITEBUILDERADDONENGINE
from PyQt5.QtWidgets import QMessageBox

ADDON_NAME = "HTML Helper"
ADDON_VERSION = "1.0"

def INIT(api):
    def insert_basic_page():
        editor = api.current_editor()
        if not editor:
            return

        editor.setPlainText("""<!DOCTYPE html>
<html>
```

```
<head>
  <title>New Page</title>
</head>
<body>
  <h1>Hello Website Builder</h1>
</body>
</html>""")

def about():
    QMessageBox.information(
        api.window,
        "HTML Helper",
        f"{ADDON_NAME} version {ADDON_VERSION}"
    )

api.add_addon_button("Insert Basic HTML", insert_basic_page)
api.add_addon_button("About HTML Helper", about)
```