ChatGPT

📦 DOXBROWSER Plugin API Documentation

Welcome to the DOXBROWSER Plugin API! This document provides a guide for developing Python-based plugins for DOXBROWSER.

Requirements

- Python 3.7+
- Plugins are written in Python (*.py)
- DOXBROWSER loads plugins dynamically via a GUI

Plugin Structure

A plugin must define a `run_plugin()` function. Example:

```
# myplugin.py
def run_plugin():
    print("Plugin launched!")
```

Inside `run_plugin`, you have access to the following global variables:

Variable	Type	Description
`browser`	`QWebEngineView`	Main browser view
`page`	`QWebEnginePage`	Current web page object
`url_bar`	`QLineEdit`	URL input field

You can use these to interact with the browser:

```
def run_plugin():
    url_bar.setText("https://example.com")
    browser.load(QUrl("https://example.com"))
```

Auto-run Plugins

To make your plugin run automatically at startup, name it `autorun_*.py` and place it in the `plugins/` folder (will be auto-loaded in future versions).

🛠 Available Extensions

You can extend the browser with:

- Buttons in plugin tab
- JavaScript injection
- Custom popups or messages
- Simulated CRX extension via JS loading

Emulating `.crx` Extensions

You can write JavaScript extensions and load them like this:

```
def run_plugin():
    js_code = '''
        alert('Hello from a simulated CRX plugin!');
    '''
    page.runJavaScript(js_code)
```

This allows adding JS-based functionality similar to Chrome extensions.

Limitations

- Plugins run in Python context, not in sandbox — **they have full system access**.
- Browser does not yet isolate plugins per tab.

Use this API responsibly.

Example Plugin

```
from PyQt5.QtCore import QUrl

def run_plugin():
    url_bar.setText("https://wikipedia.org")
    browser.load(QUrl("https://wikipedia.org"))
```

🛠 Create EXE (Windows)

You can bundle your browser into a standalone Windows executable:

1. Install PyInstaller

```
pip install pyinstaller
```

2. Generate .exe

From the directory with your main browser script:

```
pyinstaller --onefile --windowed your_browser_script.py
```

This will create a standalone `.exe` in the `dist/` folder.

If your browser uses images/assets, use a `.spec` file to include them.

Contributing

Want to submit your plugin to the public plugin list? Open a pull request at:

https://github.com/YOUR-GITHUB/DOXBROWSER

Happy hacking!