

Burp Suite Extensions 作成のススメ

2024/12/06 @salty_byte

いきなりですが...

Burp Suite Extentions使ってますか？

Burp extensions(Burp Suite Extensions)って何？

Burp Suiteを使いやすくするための拡張機能のこと。

The screenshot shows the Burp Suite interface with the 'Extensions' tab selected. The 'BApp Store' is open, displaying a list of extensions. The 'Retire.js' extension is highlighted. The right panel shows details for 'Retire.js', including its description, estimated system impact, and rating.

Name	Installed	Rating	Popularity	Last updated	System impact	Detail
Param Miner		☆☆☆☆☆		19 Sep 2024	Low	
JSON Web Tokens	✓	☆☆☆☆☆		29 Aug 2024	Low	
Active Scan++		☆☆☆☆☆		23 Nov 2023	Low	Requires Burp Sui...
Retire.js		☆☆☆☆☆		15 Dec 2021	Low	Requires Burp Sui...
Authorize		☆☆☆☆☆		19 Sep 2024	Low	
Turbo Intruder		☆☆☆☆☆		08 Aug 2024	Medium	
JS Miner		☆☆☆☆☆		20 Jul 2023	Low	Requires Burp Sui...
J2EEScan		☆☆☆☆☆		26 Aug 2021	High	Requires Burp Sui...
Collaborator Everywhere		☆☆☆☆☆		09 Jan 2023	Low	Requires Burp Sui...
Content Type Converter		☆☆☆☆☆		24 Jan 2017	Low	
403 Bypass		☆☆☆☆☆		27 Sep 2022	Low	Requires Burp Sui...
JS Link Finder		☆☆☆☆☆		05 Sep 2019	Low	Requires Burp Sui...
Software Vulnerability Sca...		☆☆☆☆☆		09 Apr 2019	Low	Requires Burp Sui...
HTTP Request Smuggler		☆☆☆☆☆		17 Nov 2023	Low	
Hackvertor		☆☆☆☆☆		24 Jan 2024	Low	
Logger++		☆☆☆☆☆		06 Jul 2023	High	
JSON Web Token Attacker		☆☆☆☆☆		04 Feb 2022	Medium	
.NET Beautifier		☆☆☆☆☆		23 Jan 2017	Low	
Additional Scanner Checks		☆☆☆☆☆		22 Dec 2018	Low	Requires Burp Sui...
JWT Editor		☆☆☆☆☆		11 Sep 2024	Low	
XSS Validator		☆☆☆☆☆		10 Feb 2022	High	Requires Burp Sui...

Retire.js

This extension integrates Burp with the Retire.js repository to find vulnerable JavaScript libraries.

It passively looks at JavaScript files loaded and identifies those which are vulnerable based on various signature types (URL, filename, file content specific hash).

Estimated system impact

Overall: **Low**

Memory	CPU	Time	Scanner
Low	Low	Low	Low

Author: Philippe Arteau
Version: 3.0.4
Source: <https://github.com/portswigger/retire-js>
Updated: 15 Dec 2021
Rating: ☆☆☆☆☆ [Submit rating](#)

話すこと

- Burp extensionsの作り方 (Kotlin版)
- 具体例

動機

- なんで今さらExtensions作成の発表？
 - Extensions作成用のAPIが更新されているが古い記事が結構ヒットする。
 - Pythonで作られている記事が多く、Java/KotlinやRubyの記事が少ない。
 - 意外と使ったことはあるけど、作ったことがない人がそれなりにいる。

Burp extensionsで何ができる？

- HTTP リクエスト/レスポンスの書き換えや送信
- Burp Scanner のシグネチャ追加
- Burp Suite で取得したデータの操作
- UI変更(メニュー / タブ / コンテキストメニュー)
- 各種設定
- BCheck definitions
- etc...

<https://github.com/PortSwigger/BChecks/tree/main>

作り方

1. 環境構築
2. コード作成
3. ビルド
4. Burp Suiteにインポート / 使う
5. 公開

1. 環境構築

利用可能言語

- Java / Kotlin
- Python
- Ruby

Load Burp extension

Please enter the details of the extension, and how you would like to handle standard output and error.

Extension details

Extension type: Java

Extension file (.jar): Java Select file ...

Standard output

☐ Output to system console

☐ Save to file: Select file ...

☒ Show in UI

Standard error

☐ Output to system console

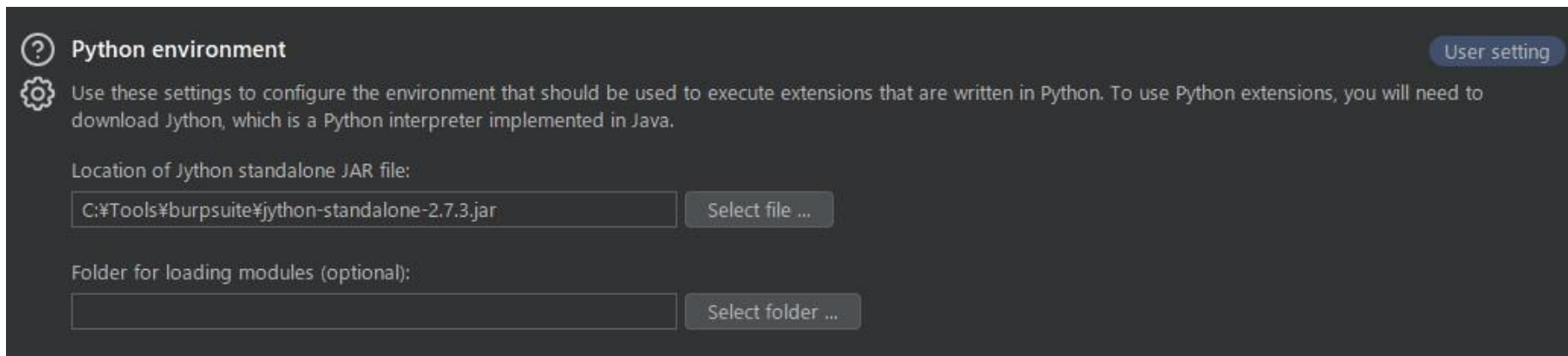
☐ Save to file: Select file ...

☒ Show in UI

Cancel Next

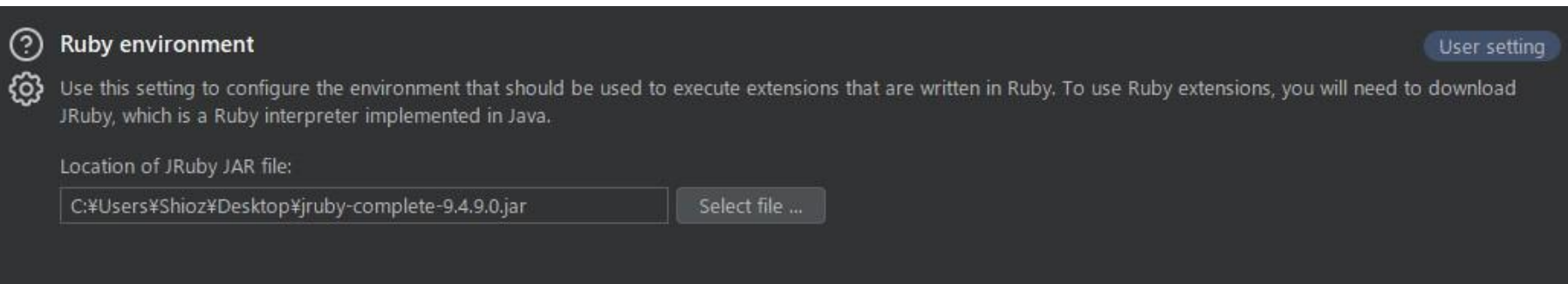
利用時の注意点: Python

- Jythonが必要
 - <https://www.jython.org/download>
- Python 2.x まで対応している



利用時の注意点: Ruby

- JRubyが必要
 - <https://www.jruby.org/download>
- Ruby 3.1.x まで対応している



ビルドツール(Java / Kotlin利用時)

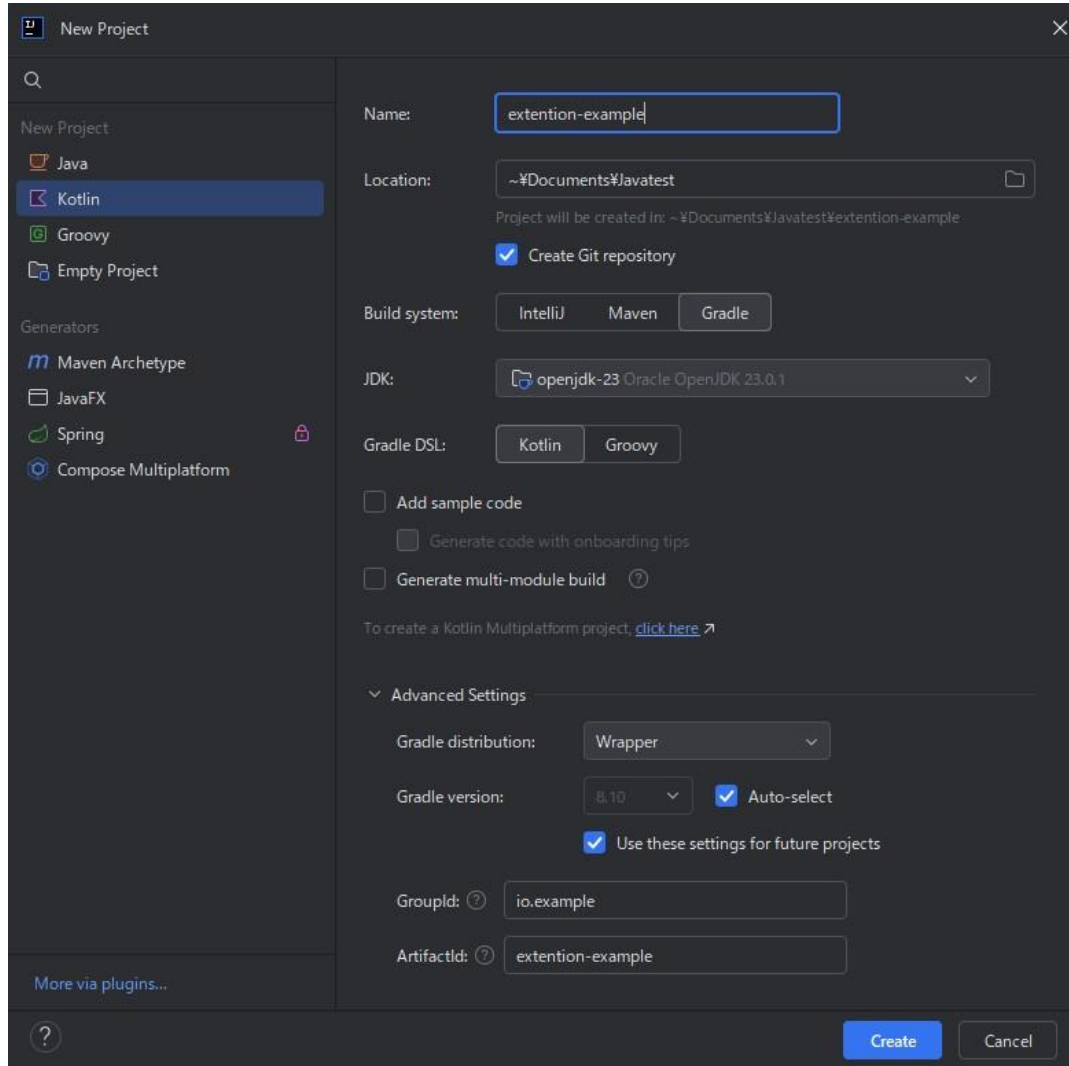
- Maven
- Gradle

今回の開発環境

- 対象
 - Burp Suite Community Edition v2024.9.5
- IntelliJ IDEA 2024.3 (Community Edition)
 - Kotlin 2.0.21
 - JDK: Oracle OpenJDK 23.0.1
 - Gradle 8.10

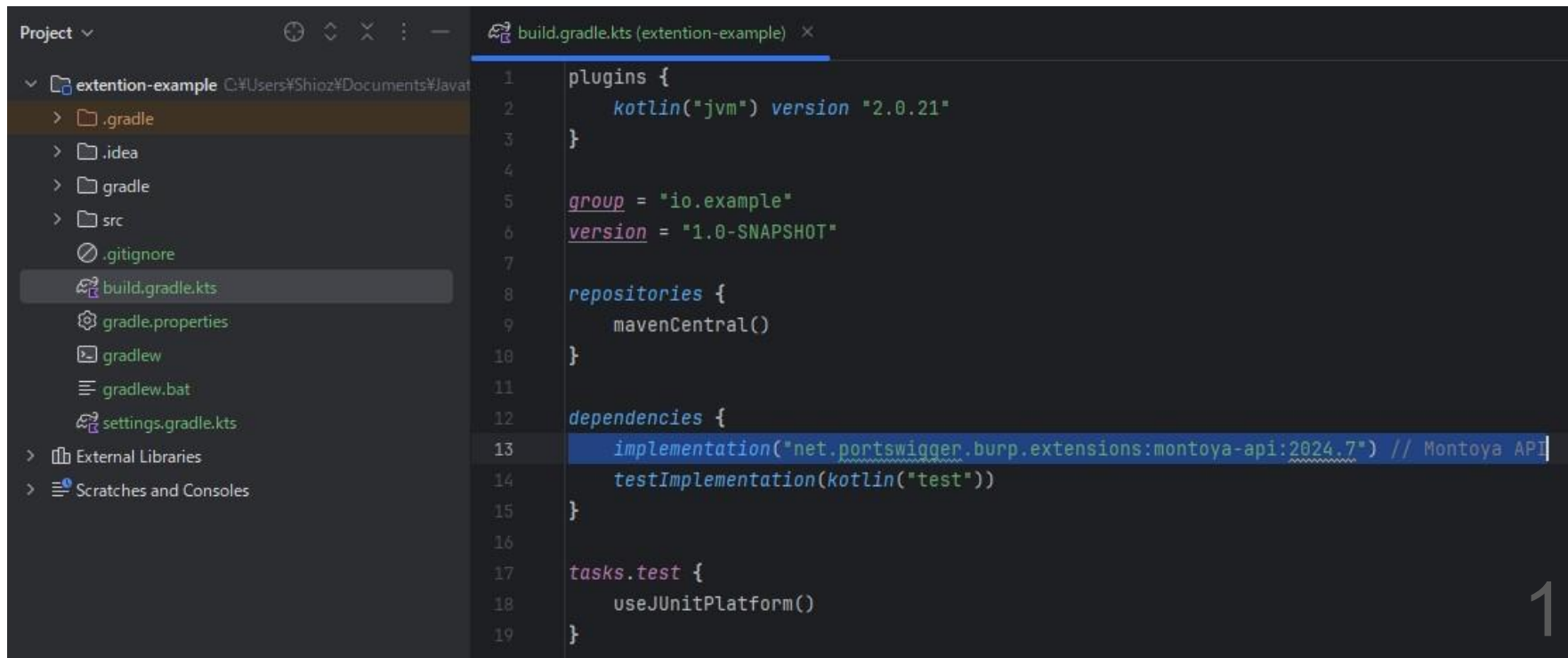
IntelliJ IDEA

- New > Project...
- Create



Montoya APIを依存関係に追加

- build.gradle.kts の dependencies
 - implementation("net.portswigger.burp.extensions:montoya-api:2024.7")



The screenshot shows an IDE window with the file `build.gradle.kts (extention-example)` open. The left sidebar shows the project structure with `build.gradle.kts` selected. The main editor displays the following Kotlin Gradle build script:

```
1 plugins {  
2     kotlin("jvm") version "2.0.21"  
3 }  
4  
5 group = "io.example"  
6 version = "1.0-SNAPSHOT"  
7  
8 repositories {  
9     mavenCentral()  
10 }  
11  
12 dependencies {  
13     implementation("net.portswigger.burp.extensions:montoya-api:2024.7") // Montoya API  
14     testImplementation(kotlin("test"))  
15 }  
16  
17 tasks.test {  
18     useJUnitPlatform()  
19 }
```

jarビルド用のタスクを作成

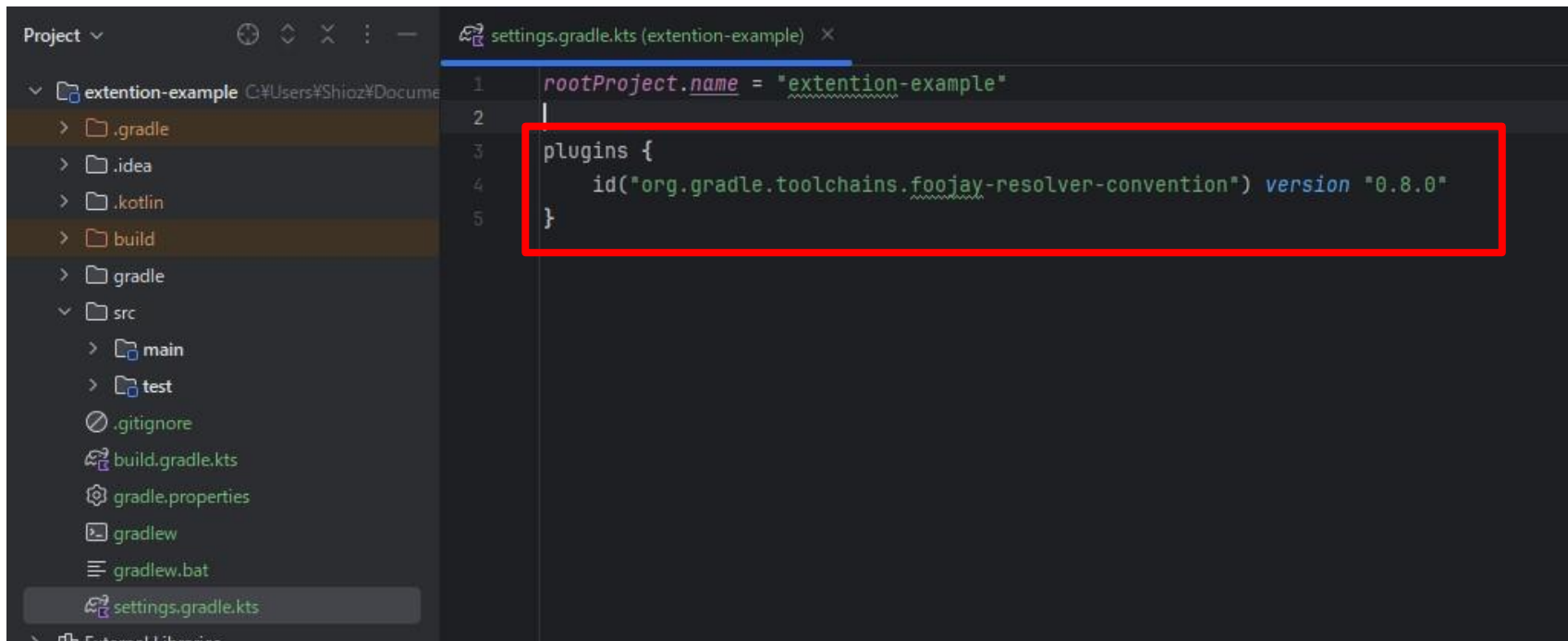
- build.gradle.kts
 - 新規タスク「fatJar」を追記

```
17 tasks.test {  
18     useJUnitPlatform()  
19 }  
20  
21 tasks {  
22     register(name: "fatJar", Jar::class.java) {  
23         archiveClassifier.set("all")  
24         duplicatesStrategy = DuplicatesStrategy.EXCLUDE  
25         from(configurations.runtimeClasspath.get().map { if (it.isDirectory) it else zipTree(it) })  
26         val sourcesMain = sourceSets.main.get()  
27         from(sourcesMain.output)  
28     }  
29 }
```


Compileターゲットバージョンの指定①

- settings.gradle.kts
 - ToolChainプラグインの追記

※Java CompileとKotlin Compileのバージョン違いのエラーが出る必要な人向け



Compileターゲットバージョンの指定②

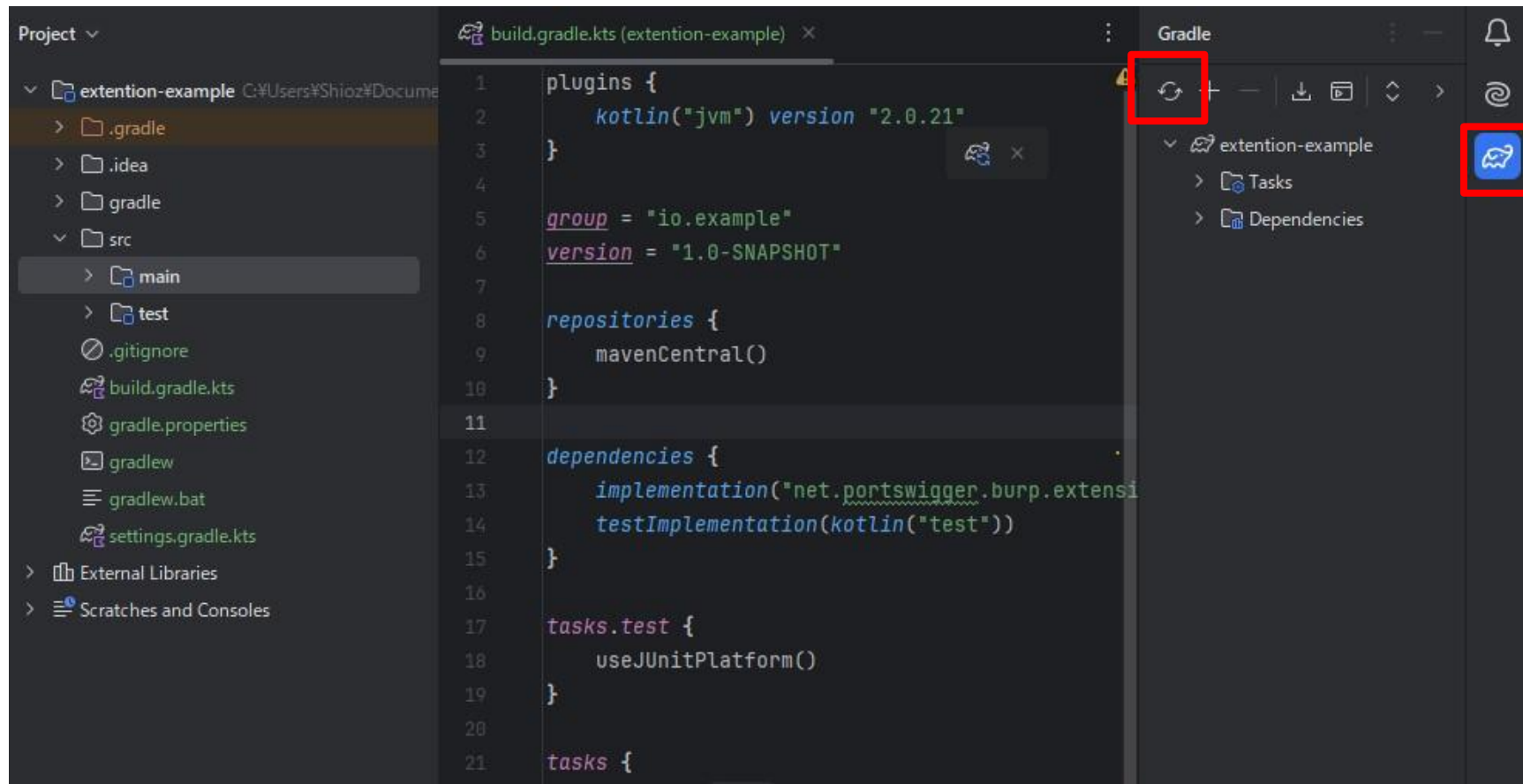
- build.gradle.kts

- 動作させるJDKバージョンを追記する。
- バージョンを上げすぎると、最新の Burp Suite以外で動かない可能性あり。

※Java CompileとKotlin Compileのバージョン違いのエラーが出る必要な人向け

```
11
12 dependencies {
13     implementation("net.portswigger.burp.extensions:montoya-api:2024.7") // Montoya API
14     testImplementation(kotlin("test"))
15 }
16
17 kotlin {
18     jvmToolchain(jdkVersion: 22)
19 }
20
21 tasks.test {
22     useJUnitPlatform()
23 }
```

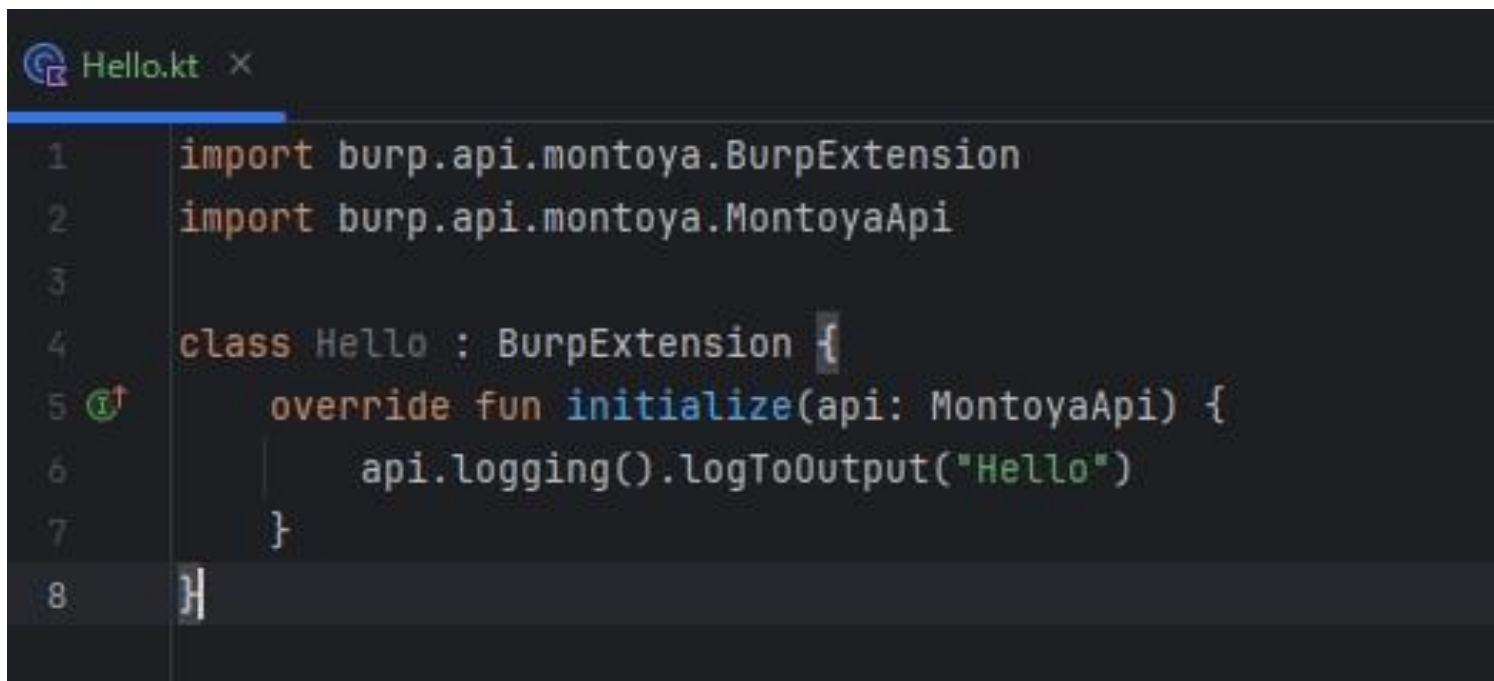
Gradleの設定を更新する



2. コード作成

コード作成

- src/main/kotlin/Hello.kt
 - コンソールに「Hello」を出力するだけのコード。
 - Montoya APIを継承することで、Burp Suiteの各機能にアクセスすることができる。

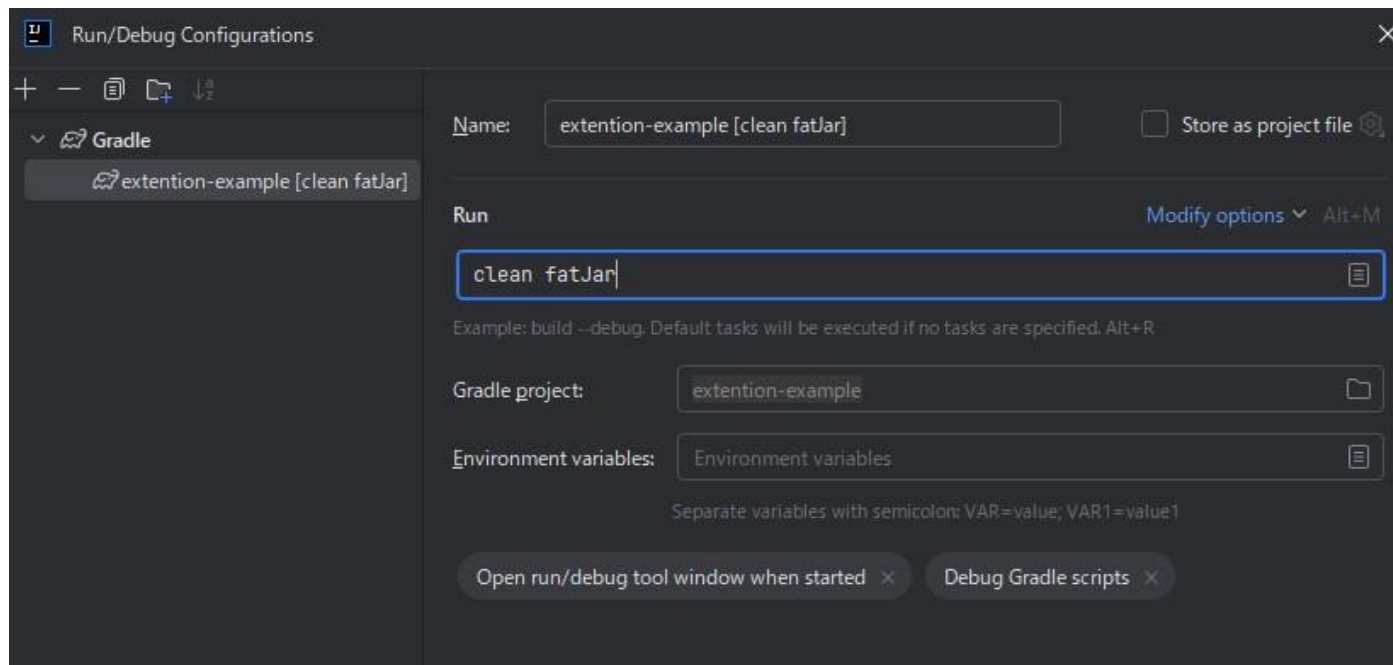


```
1 import burp.api.montoya.BurpExtension
2 import burp.api.montoya.MontoyaApi
3
4 class Hello : BurpExtension {
5     override fun initialize(api: MontoyaApi) {
6         api.logging().logToOutput("Hello")
7     }
8 }
```

3. ビルド

Gradleのタスク

- Run/Debug ConfigurationsでGradleのタスクを追加する。
 - + > Gradle
 - Runコマンドに「clean fatJar」を設定



Gradleのタスク実行

- Run/Debug ConfigurationsでGradleのタスクを実行する。
 - Run

✓ extention-example [clean fatJar]: successful At 2024/11/ 959 ms

0:39:11: Executing 'clean fatJar'...

```
> Task :clean
> Task :checkKotlinGradlePluginConfigurationErrors SKIPPED
> Task :processResources NO-SOURCE
> Task :compileKotlin
> Task :compileJava NO-SOURCE
> Task :classes UP-TO-DATE
> Task :fatJar
```

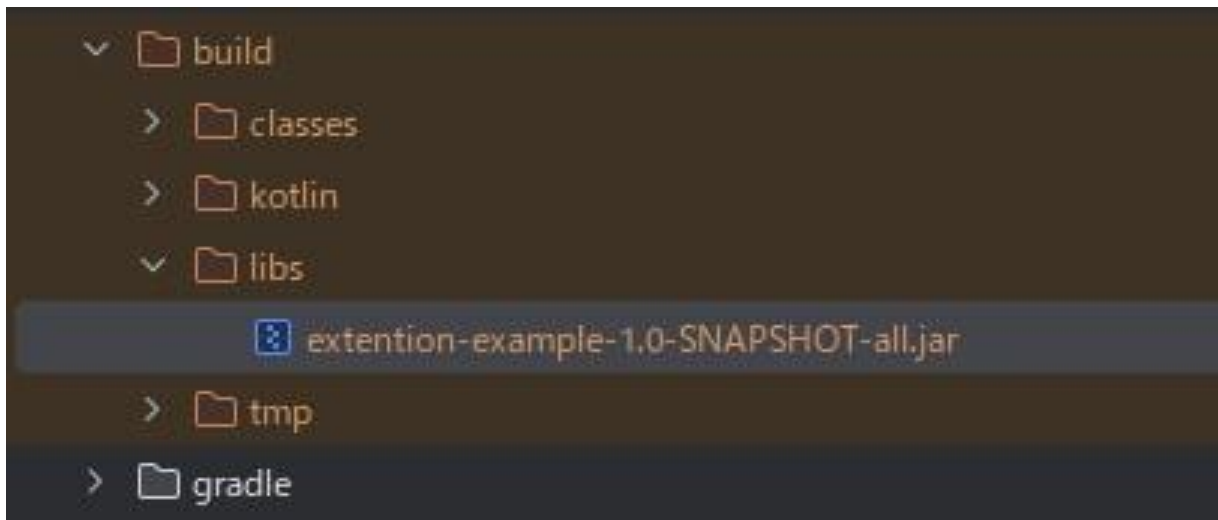
BUILD SUCCESSFUL in 890ms

3 actionable tasks: 3 executed

0:39:12: Execution finished 'clean fatJar'.

ビルドする

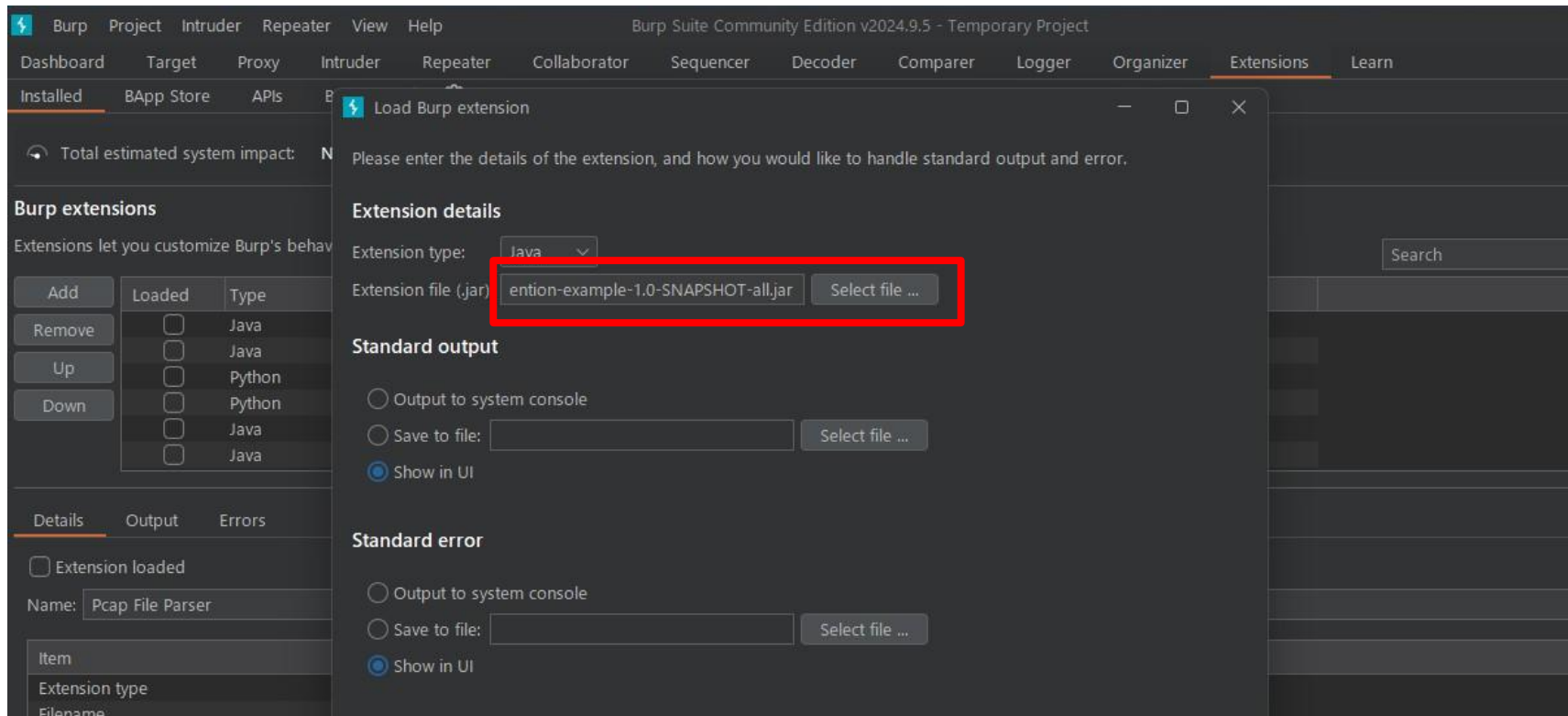
- Run/Debug ConfigurationsでGradleのタスクを実行する。
 - ビルドに成功すると、build/libs配下にjarファイルが生成される。



4. Burp Suiteにインポート / 使う

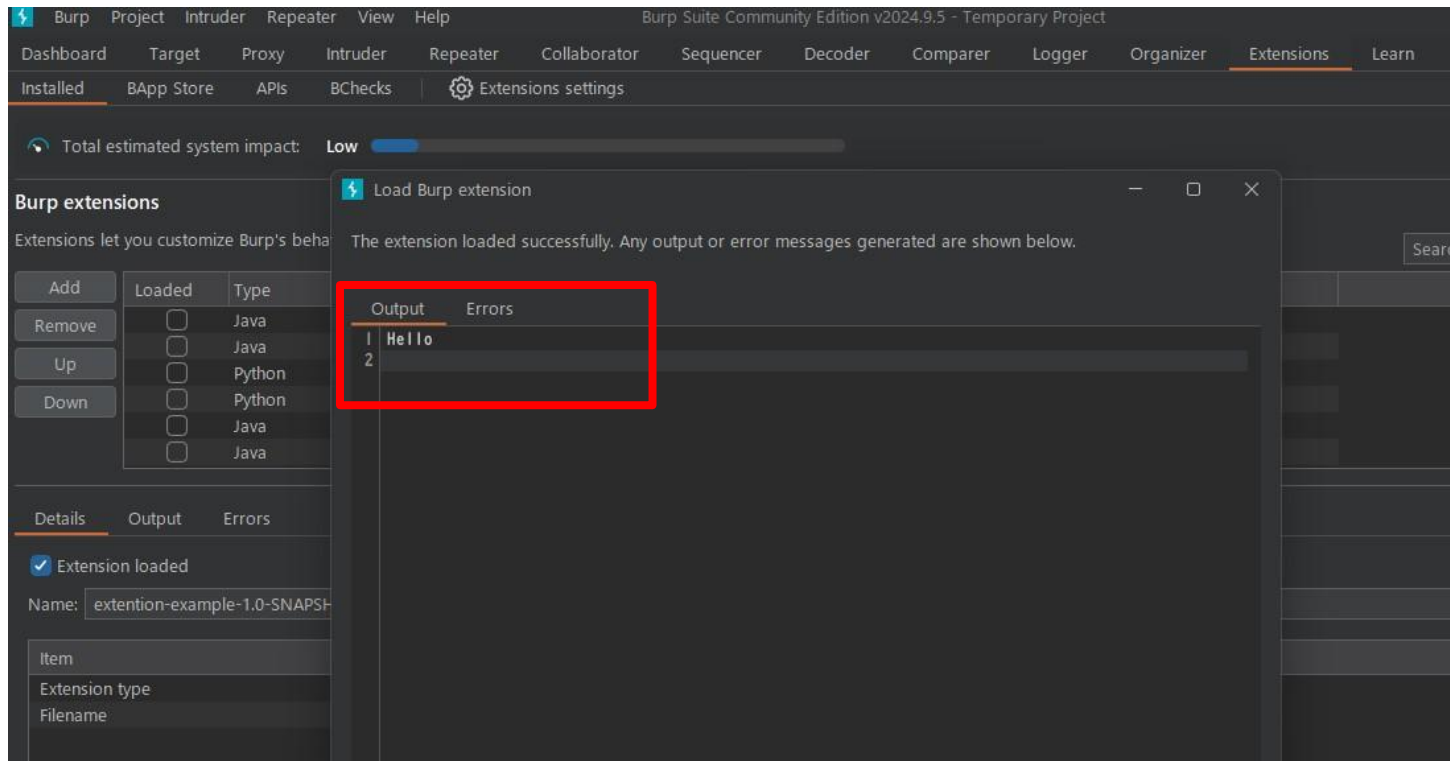
Burp Suiteにインポート

- Burp SuiteのExtensionsタブから追加する。



使う

- 今回はコンソールログに「Hello」が出力されるだけ。



5. 公開する

公開方法

1. 一般的なサービスで公開
2. BAppsに登録

公開方法1: 一般的なサービスで公開

- GitHub
- GitLab
- Bitbucket
- etc...

公開方法2: BAppsに登録

- 以下を含めてメールで連絡する。
 - ソースコードが保存されている GitHub リポジトリのリンク
 - 拡張機能の名前と機能の適切な説明
 - 拡張機能の動作、使用方法、および拡張機能の使用に必要な設定情報の説明
- 注意: 承認基準がある
 - <https://portswigger.net/burp/documentation/desktop/extensions/creating/bapp-store-acceptance-criteria>

<https://portswigger.net/burp/documentation/desktop/extensions/creating/bapp-store-submitting-extensions>

具体例

Montoya API

- 使える機能がたくさんある。

<https://portswigger.github.io/burp-extensions-montoya-api/javadoc/burp/api/montoya/MontoyaApi.html>

公式サンプル

- <https://github.com/PortSwigger/burp-extensions-montoya-api-examples>

Example Extensions

Extension	Description
Hello World	Prints output to various locations in Burp
HTTP Handlers	Demonstrates performing various actions on requests passing through any tool in Burp
Proxy Handlers	Demonstrates performing various actions on requests passing through the Proxy
Event Listeners	Registers handlers for various runtime events, and prints a message when each event occurs
Traffic Redirector	Redirects all outbound requests from one host to another
Custom Logger	Adds a new tab to Burp's UI and displays a log of HTTP traffic for all Burp tools
Custom Request Editor Tab	Adds a new tab to Burp's HTTP message editor, in order to handle a data serialization format
Custom Scan Insertion Points	Provides custom attack insertion points for active scanning
Custom Scan Checks	Implements custom checks to extend the capabilities of Burp's passive and active scan checks

更新履歴

- 最近も機能追加がある。

Changelog

v2024.11

- Added `requestResponse` and `issues` methods to *SiteMapNode*.

v2024.7

- Added JSON parsing / manipulation support with *JsonUtils* and *JsonNode*.
- Added ability to control redirection behavior when issuing HTTP requests using *RedirectionMode* and *RequestOptions*.
- Added utility methods to *HttpRequest* and *HttpResponse* which add, update or remove multiple headers.
- Added *EditorOptions.SHOW_NON_PRINTABLE_CHARACTERS* and *EditorOptions.WRAP_LINES* which can be applied when creating *RawEditors*.
- Added method to *Intruder* which enables sending of HTTP requests with an associated tab name.
- Added *Project* which allows retrieval of the current project name.
- Added method to *Proxy* to determine the current interception state.

例1: Active Scanの危険なペイロードを置き換える

- 「 or 」、「%20or%20」等
- Http#registerHttpRequestHandlerを使って実装が可能。
 - Scanner等で生成されたリクエストの書き換え
 - handleHttpRequestToBeSent
 - レスポンスの書き換え
 - handleHttpResponseReceived

例2: WebSocket

- 通信生成
 - `WebSockets#createWebSocket`で作成。
- 内容確認 / 改変
 - Halder
 - `WebSocketCreatedHandler`
 - `MessageHandler`
 - `registerWebSocketCreatedHandler`

例3: データ保存

- プロジェクトファイルに拡張機能のデータを保存できる。

※Community版ではメモリに保存されるので、使う機会はあまりないかも。

```
override fun initialize(api: MontoyaApi) {  
    val storage = api.persistence().extensionData()  
    storage.setString("test", "保存したいデータ")  
    val data = storage.getString(key: "test")  
    api.logging().logToOutput(data) // -> 保存したいデータ  
}
```

作成例

こんなことができるよ！①

- サイト巡回(クローリング)を楽にするためのツール。

The screenshot displays the Burp Suite interface, specifically the Crawling tab. The top section shows a list of requests with columns for No, リクエスト名, メソッド, URL, パラメータ, ステータス, MIME, 拡張子, 重複, 重複箇所, 自動..., 手動..., and 備考. The bottom section shows the raw HTTP request and response data.

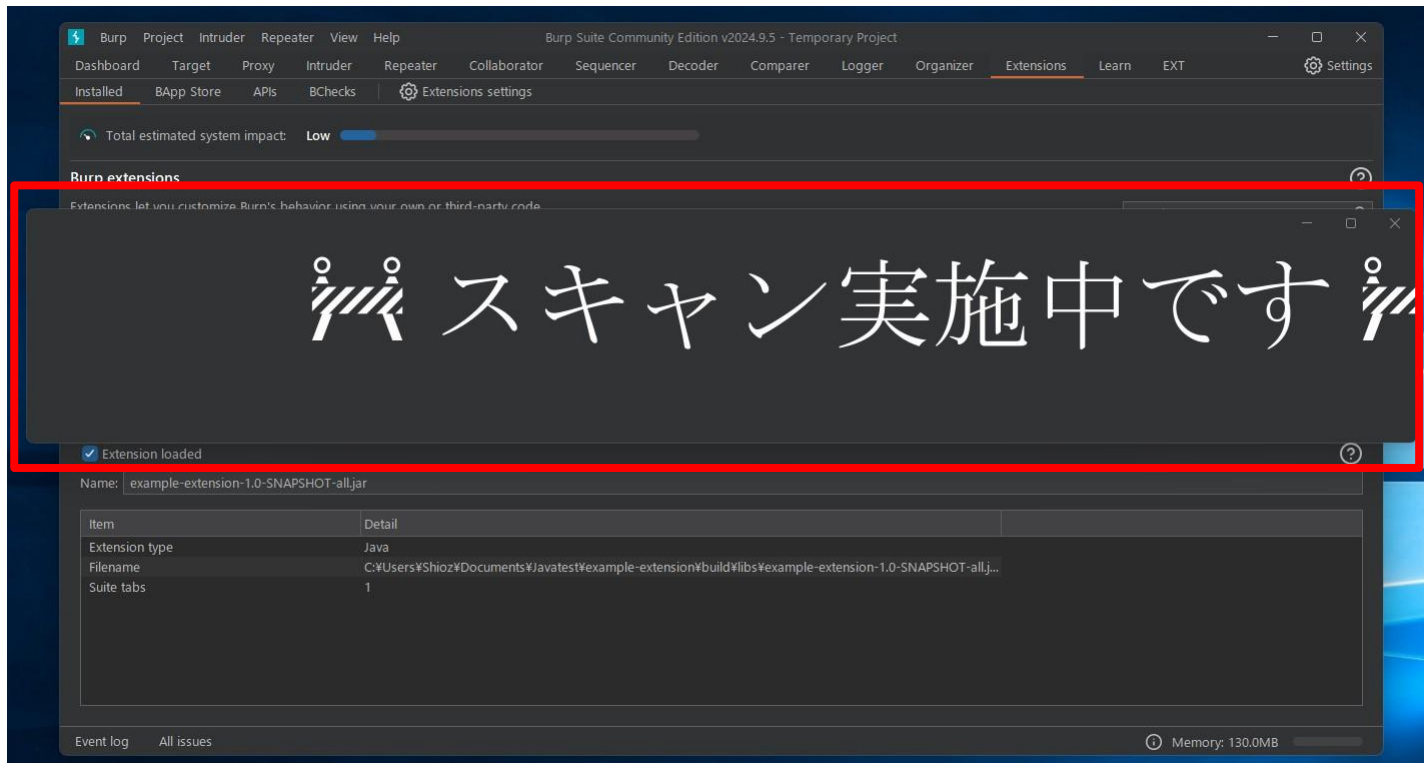
No	リクエスト名	メソッド	URL	パラメータ	ステータス	MIME	拡張子	重複	重複箇所	自動...	手動...	備考
1	TOP	GET	https://www.google.com/search?q=example...		200	HTML						
2	TOP#2	GET	https://www.google.com/images/branding/g...		200	PNG	png					
3	TOP#3	GET	https://www.google.com/images/searchbox/...		200	image	webp					
4	TOP#4	GET	https://www.google.com/xjs/_fs/k=xjs.sja_O...		200	script						
5	TOP#5	GET	https://www.google.com/images/nav_logo32...		200	image	webp					
6	TOP#6	POST	https://www.google.com/gen_204?s=web&t=...		204	HTML						
43	TOP#7	OPTIONS	https://play.google.com/log?format=json&h...		200	text						
44	TOP#8	GET	https://www.gstatic.com/_mss/boq-search/_/...		200	script						
11	TOP>検索>Example	GET	https://example.com/		200	HTML						
10	TOP>検索>Example>More information...	GET	https://www.iana.org/domains/example		301	HTML						
11	TOP>検索>Example>More information...#2	GET	https://www.iana.org/js/iana.js		200	script	js					
12	TOP>検索>Example>More information...#3	GET	https://www.iana.org/css/2022/iana_website...		200	CSS	css					
13	TOP>検索>Example>More information...#4	GET	https://www.iana.org/js/query.js		200	script	js					
14	TOP>検索>Example>More information...#5	GET	https://www.iana.org/img/2022/iana-logo-h...		200	SVG	svg					
15	TOP>検索>Example>More information...#6	GET	https://www.iana.org/img/2022/fonts/Source...		200	woff						
16	TOP>検索>Example>More information...#7	GET	https://www.iana.org/img/2022/fonts/NotoS...		200	woff						
17	TOP>検索>Example>More information...#8	GET	https://www.iana.org/img/2022/fonts/NotoS...		200	woff						

The bottom section shows the raw HTTP request and response data. The request is a GET request to https://example.com/. The response is a 200 OK status with a Content-Type of text/html; charset=UTF-8. The response body contains HTML code, including a <doctype html> declaration and a <html> tag.

<https://github.com/salty-byte/burp-crawling>

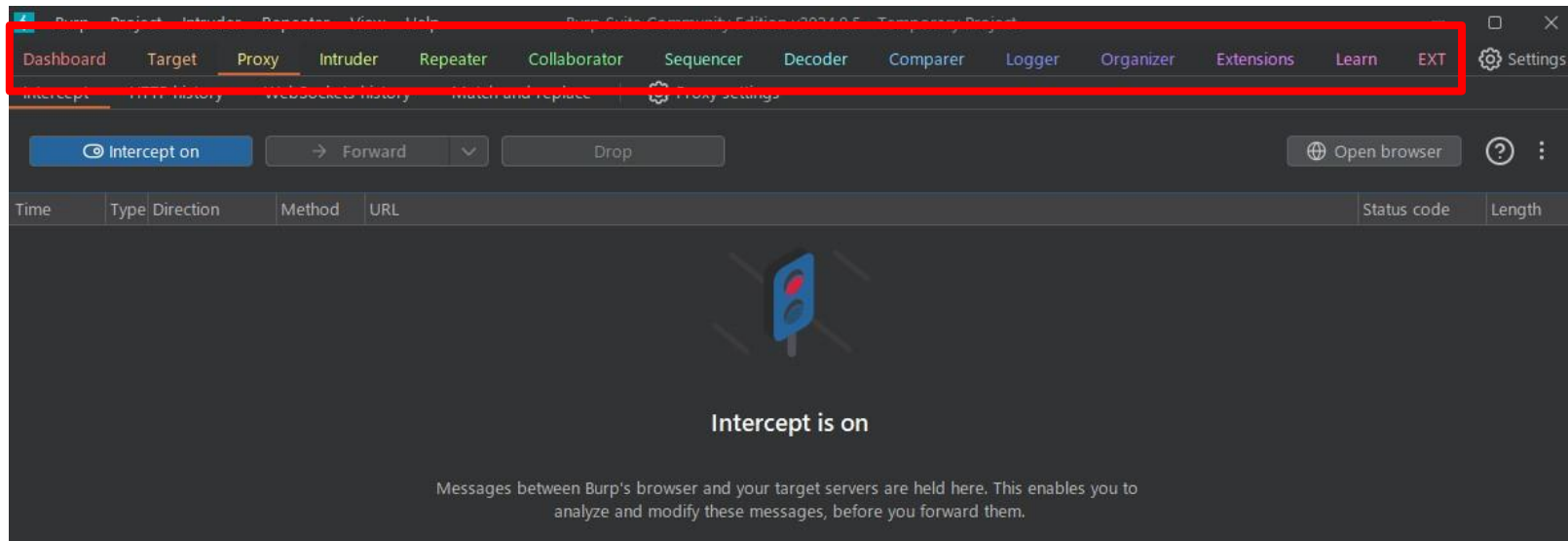
こんなことができるよ！②

- スキャン実施中を知らせるポップアップメッセージ。



こんなことができるよ！③

- Burpのタブの色を変える(APIの機能外)。

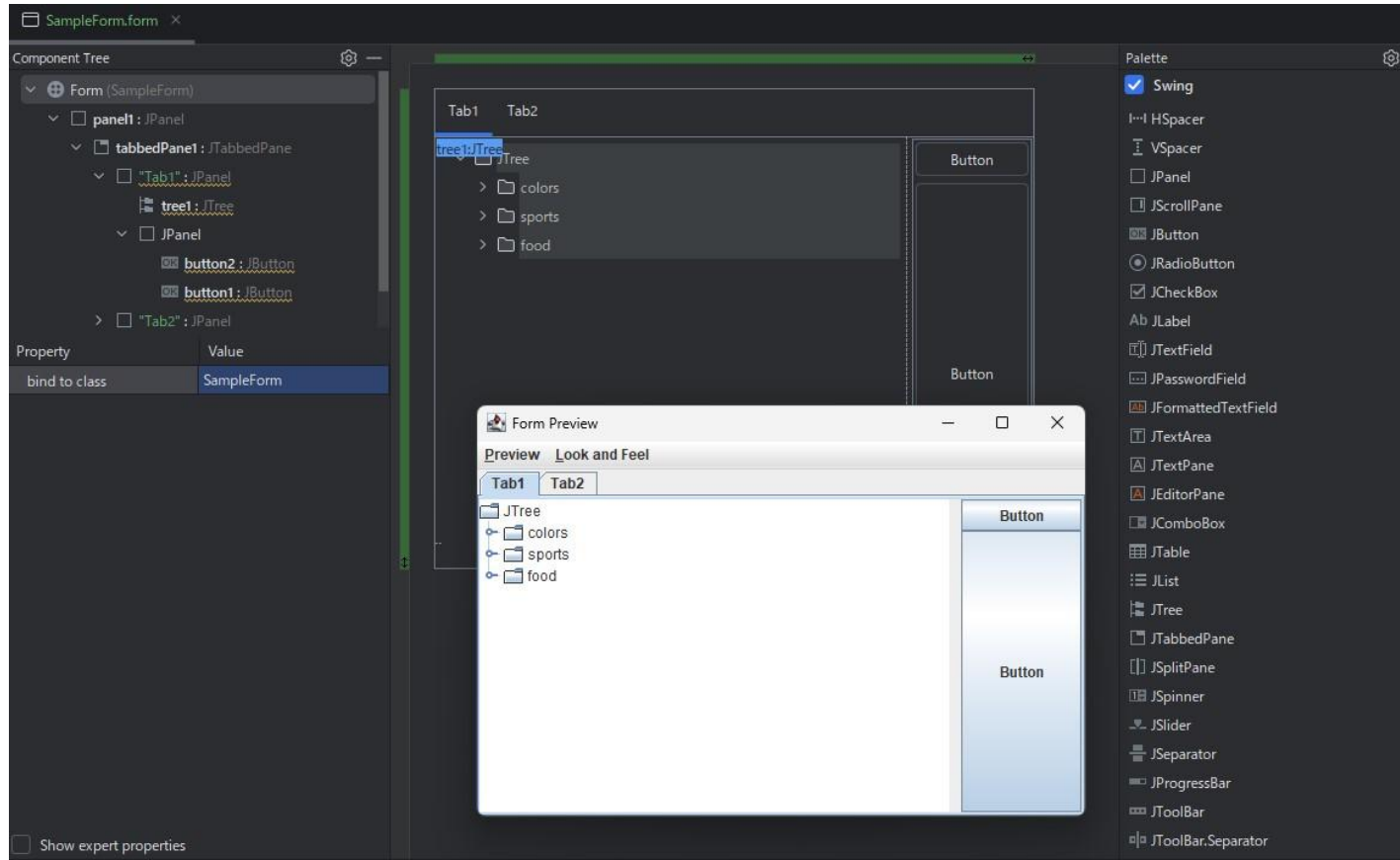


GUI作成

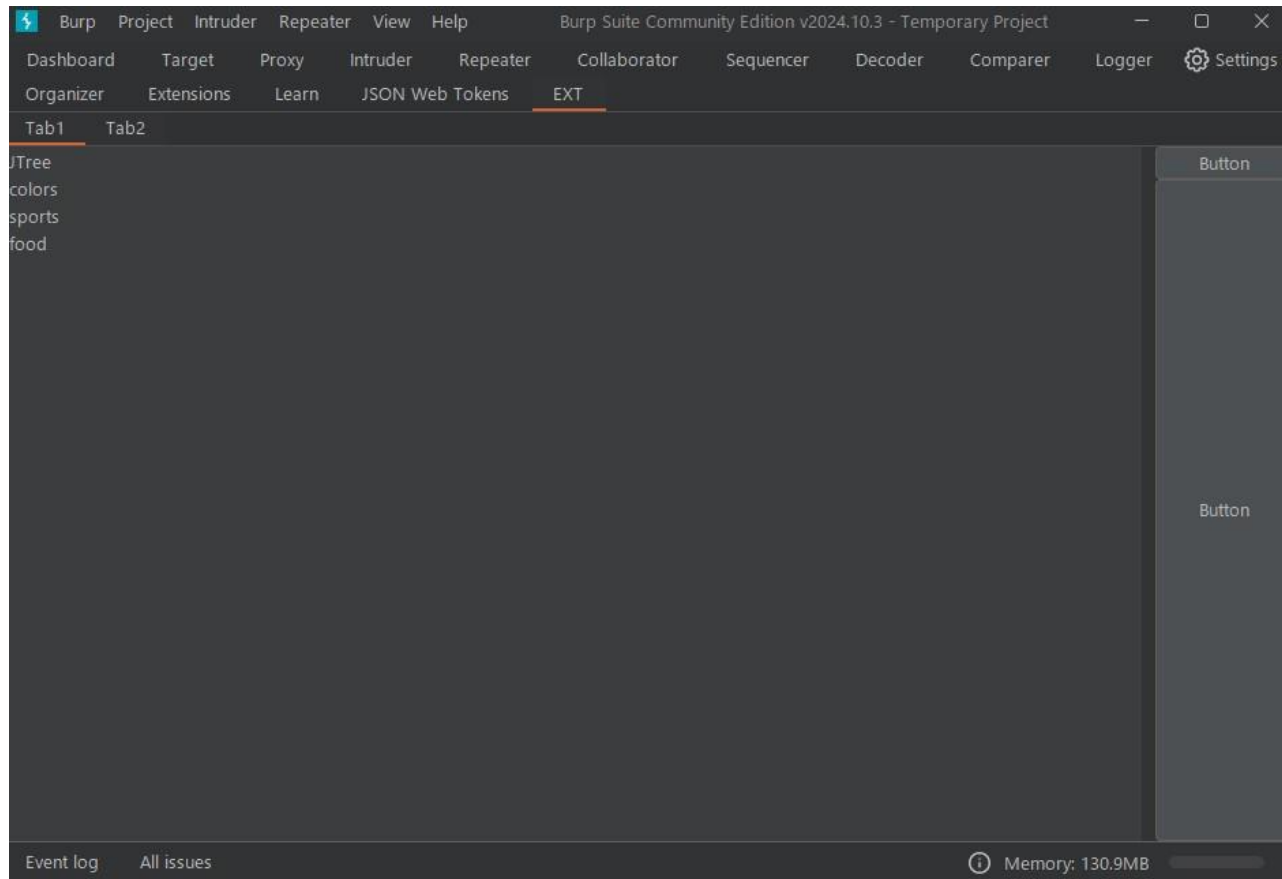
GUI作成どうしてる？

- 方法は複数ある
 - JavaFXを使う。
 - Swing UI Designer(IntelliJ IDEA)を使う。
 - WindowBuilder(Eclipse)を使う。
 - JFormDesignerを使う。
 - 自力でSwing。

Swing UI Designer (IntelliJ IDEA)



Swing UI Designer(IntelliJ IDEA)



SwingとJavaFXどっちがいい？

- 公式の見解ではSwing推奨らしい。

<https://forum.portswigger.net/thread/javafx-vs-swing-1c6356388b9a55bd5849>

- 個人的にJavaFXの方がXMLで書けるのでUIが作りやすいと思われる。
 - ただし、ビルド時の依存関係をちゃんとしないといけない。

まとめ

まとめ

- Burp extensionsを作るのは簡単！
- 便利な機能は誰かが作っていることが多いけど...
 - かゆいところに手が届かない ...とかあれば自分で作ればいい！
- 作ったら公開してみよう！
 - BApp Store
 - GitHub

参考

- <https://portswigger.net/burp/documentation/desktop/extensions/creating>
- <https://github.com/PortSwigger/burp-extensions-montoya-api>