

ZED ATTACK PROXY(ZAP)

Software Architecture and
Design Assignment!



Member



62010453 ນະກພັນຮູ ຮົງກາລ



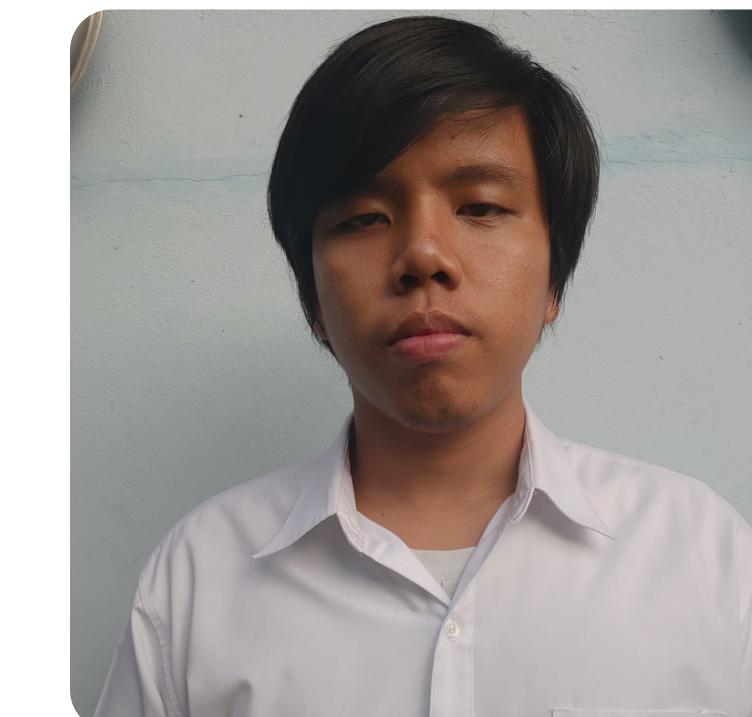
62010474 ນິວລ ກຽດພັນຮ



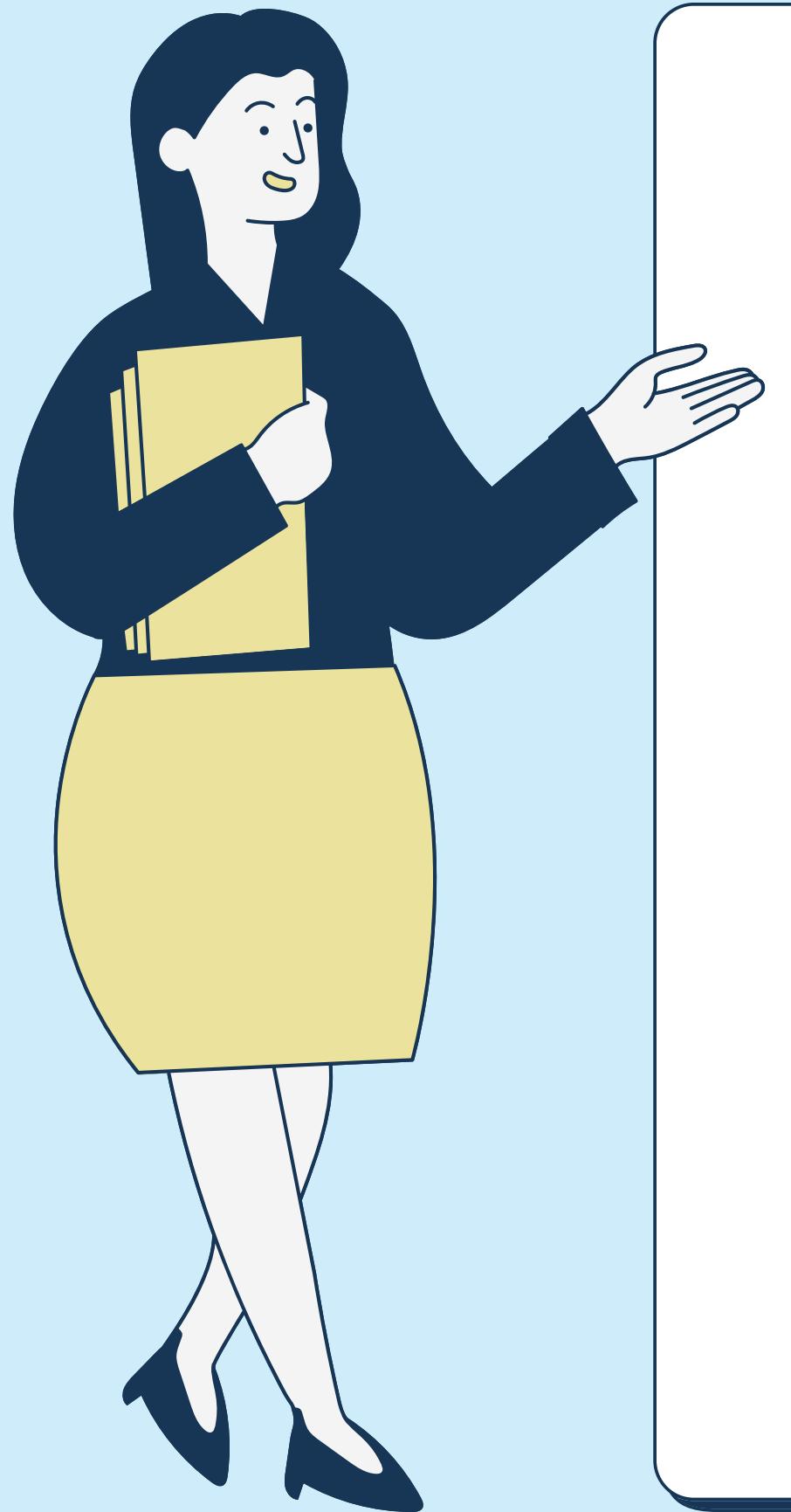
62010472 ນິວພຣະ ສົມບຸນເຮືອງ



62010494 ນິຕີພັນນ ບຸນເກຕຸ



62010496 ນິຕີກູນິ ຄລ້າຍເນີຍນ



About ZAP

เป็น Software Open Source สำหรับการทดสอบระบบ Security บน Web Application ด้วยการดักข้อมูลที่สื่อสารระหว่าง Browser กับ Web Application.

ทำงานได้หลายรูปแบบ: Stand-Alone, Daemon Process และ Service(กำลังพัฒนา)

CEO



Simon Bennetts
@psiinon

OWASP Zed Attack Proxy ([@zaproxy](#)) lead, [@Stackhawk](#) Distinguished Engineer,
Hac<3r

ข้อมูลเพิ่มเติมได้ที่

OWASP ZAP – Getting Started ([zaproxy.org](#))

ARCHITECTURAL PATTERN



MODEL VIEW CONTROL (MVC)

Model

```
/** This method should typically only be called from the Control class ...*/
public Session newSession() {...}

/** This method should typically only be called from the Control class */
public void openSession(String fileName)
    throws SQLException, SAXException, IOException, Exception {...}

public void openSession(String fileName, final SessionListener callback) { getSession().open(fileName, callback); }

/** This method should typically only be called from the Control class */
public void openSession(final File file, final SessionListener callback) { getSession().open(file, callback); }

/** This method should typically only be called from the Control class */
public void saveSession(final String fileName, final SessionListener callback) {...}

/** This method should typically only be called from the Control class */
public void saveSession(String fileName) throws Exception {...}

/** Persists the properties (e.g. name, description) of the current session. ...*/
public void persistSessionProperties() throws Exception {...}

/** This method should typically only be called from the Control class */
public void snapshotSession(final String fileName, final SessionListener callback) {...}

/** This method should typically only be called from the Control class */
public void discardSession() { getSession().discard(); }

/** This method should typically only be called from the Control class */
public void closeSession() { getSession().close(); }
```

ធ្វើការ Change State ដោយ Control

Link:

<https://github.com/zaproxy/zaproxy/blob/main/zap/src/main/java/org/parosproxy/paros/model/Model.java>

va

View

javax เป็น Library ที่ใช้ใน
การออกแบบ GUI

Instance ใน class มี GUI
ในส่วนต่าง ๆ ของโปรแกรม

Link:

<https://github.com/zaproxy/zaproxy/blob/main/zap/src/main/java/org/parosproxy/paros/view/View.java>

```
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JToggleButton;
import javax.swing.JToolBar;
import javax.swing.KeyStroke;
```

```
// private LogPanel logPanel = null;
private MainFrame mainFrame = null;
private HttpPanelRequest requestPanel = null;
private HttpPanelResponse responsePanel = null;
private SiteMapPanel siteMapPanel = null;
private OutputPanel outputPanel = null;
private Vector<JMenuItem> popupList = new Vector<>();

private JMenu menuShowTabs = null;

private JCheckBox rememberCheckbox = null;
private JCheckBox dontPromptCheckbox = null;

private List<AbstractContextPropertiesPanel> contextPanels = new ArrayList<>();
private List<ContextPanelFactory> contextPanelFactories = new ArrayList<>();
//
```

Control

```
protected Model model = null;
protected View view = null;

private boolean init(ControlOverrides overrides, boolean startProxy) {

    // Load extensions first as message bundles are loaded as a side effect
    loadExtension();

    // ZAP: Start proxy even if no view
    Proxy proxy = getProxy(overrides);
    proxy.setIgnoreList(model.getOptionsParam().getGlobalExcludeURLParam().getTokensNames());
   getExtensionLoader().hookProxyListener(proxy);
   getExtensionLoader().hookOverrideMessageProxyListener(proxy);
   getExtensionLoader().hookPersistentConnectionListener(proxy);
   getExtensionLoader().hookConnectRequestProxyListeners(proxy);

    if (hasView()) {
        // ZAP: Add site map listeners
        getExtensionLoader().hookSiteMapListener(view.getSiteTreePanel());
    }

    model.postInit();

    if (startProxy) {
        proxy.setShouldPrompt(true);
        boolean started = proxy.startServer();
        proxy.setShouldPrompt(false);
        return started;
    }
    return false;
}
```

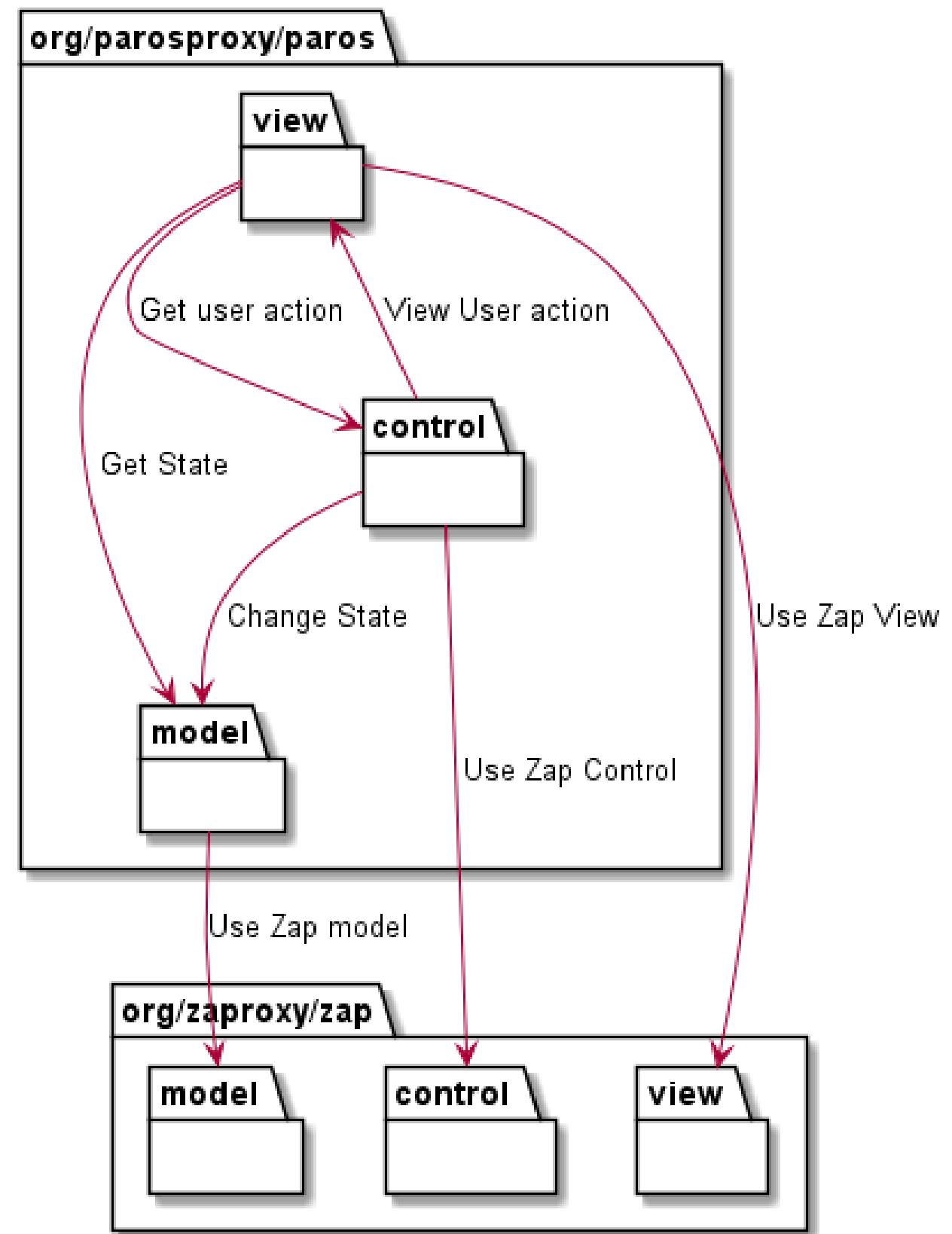
มี Instance Model และ View

โดยทำงานร่วมกับ View และ
Model ผ่าน Control

Link:

<https://github.com/zaproxy/zaproxy/blob/main/zap/src/main/java/org/parosproxy/paros/control/Control.java>

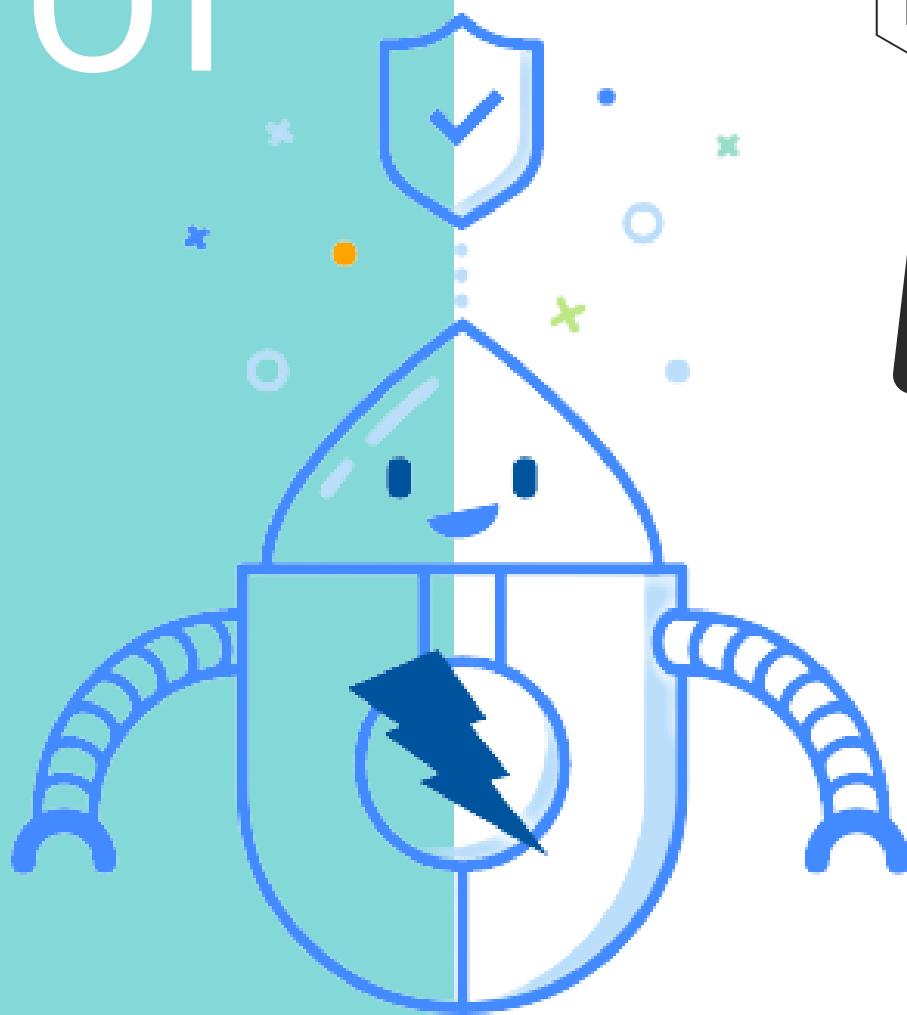
Implementation





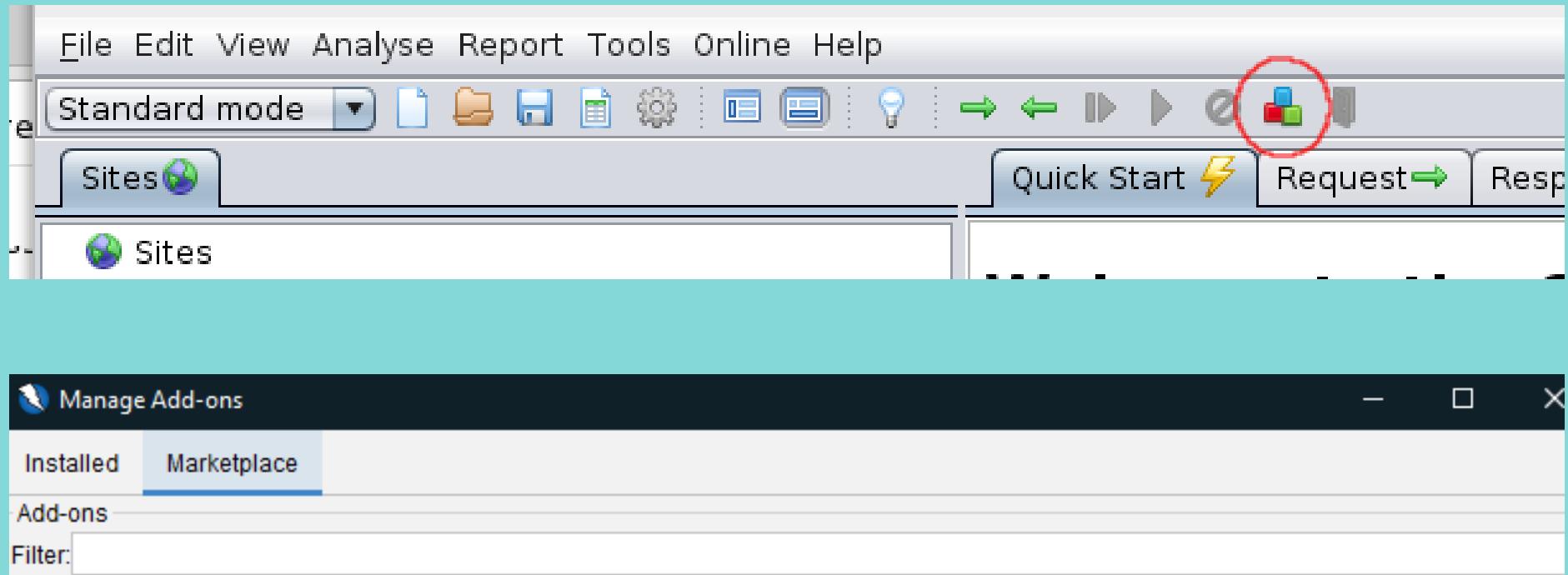
Quality Attributes of Zaproxy

Open Source project



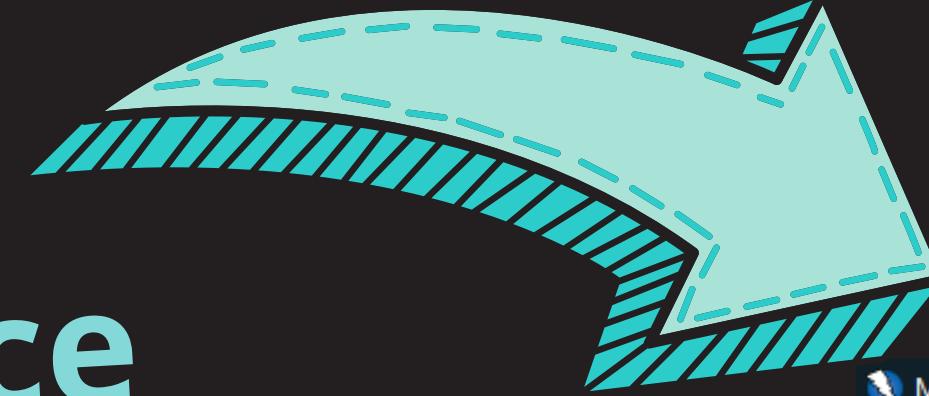
Shop

Marketplace Add-on



Marketplace Add-on

Zaproxy สามารถเพิ่ม add-ons ซึ่งมีส่วนช่วยในการใช้งาน Zaproxy ของผู้ใช้ ให้ตรงตามจุดประสงค์การใช้งานของผู้ใช้มากยิ่งขึ้น โดยที่ตัว Zaproxy จะมี Marketplace ที่รวม Add-ons ไว้ให้สำหรับ download โดยที่ไม่จำเป็นต้องไปโหลด หรือ import จาก ที่อื่นเข้ามา ซึ่งมีความยุ่งยากและเสียเวลามากกว่า



USABILITY

Manage Add-ons

Installed Marketplace

Add-ons

Filter:

Status	Name	Description	Update	
Release	Core Language Files	Translations of the core language files	14.0.0	<input type="checkbox"/>
Release	Directory List v2.3	Lists of directory names to be used with Forced Browse or...	4.0.0	<input type="checkbox"/>
Release	Directory List v2.3 LC	Lists of lower case directory names to be used with Force...	4.0.0	<input type="checkbox"/>
Release	FuzzDB Files	FuzzDB files which can be used with the ZAP fuzzer	8.0.0	<input type="checkbox"/>
Release	FuzzDB Offensive	FuzzDB web backdoors and attack files which can be use...	4.0.0	<input type="checkbox"/>
Release	Help - Portuguese, Brazilian	Portuguese, Brazilian version of the ZAP help file.	10.0.0	<input type="checkbox"/>
Release	Help - Spanish	Spanish version of the ZAP help file.	9.0.0	<input type="checkbox"/>
Release	Help - Turkish	Turkish version of the ZAP help file.	1.0.0	<input type="checkbox"/>
Release	Linux WebDrivers	Linux WebDrivers for Firefox and Chrome.	33.0.0	<input type="checkbox"/>
Release	MacOS WebDrivers	MacOS WebDrivers for Firefox and Chrome.	33.0.0	<input type="checkbox"/>
Release	SVN Digger Files	SVN Digger files which can be used with ZAP forced brow...	4.0.0	<input type="checkbox"/>
Release	Wappalyzer - Technology Dete...	Technology detection using Wappalyzer: wappalyzer.com	21.5.0	<input type="checkbox"/>
Beta	Active scanner rules (beta)	The beta quality Active Scanner rules	37.0.0	<input type="checkbox"/>
Beta	Advanced SQLInjection Scanner	An advanced active injection bundle for SQLi (derived by S...	15.0.0	<input type="checkbox"/>
Beta	BeanShell Console	Provides a BeanShell Console	7.0.0	<input type="checkbox"/>
Beta	Groovy Support	Adds Groovy support to ZAP	3.1.0	<input type="checkbox"/>
Beta	Help Chinese Simplified	Chinese Simplified version of the ZAP help file.	2.0.0	<input type="checkbox"/>
Beta	Help Indonesian	Indonesian version of the ZAP help file.	2.0.0	<input type="checkbox"/>
Beta	Help - Japanese	Japanese version of the ZAP help file.	9.0.0	<input type="checkbox"/>
Beta	Image Location and Privacy Sc...	Image Location and Privacy Passive Scanner	3.0.0	<input type="checkbox"/>
Beta	Passive scanner rules (beta)	The beta quality Passive Scanner rules	27.0.0	<input type="checkbox"/>
Beta	Plug-n-Hack Configuration	Supports the Mozilla Plug-n-Hack standard: https://develo...	12.0.0	<input type="checkbox"/>

MODIFIABILITY



The Z Marketplace
with Simon Bennetts

ADDO[®]
Add-on DevOps

Name	Version	Description	Update
Active scanner rules	35.0.0	The release quality Active Scanner rules.	<input type="checkbox"/>
Active scanner rules (alpha)	28.0.0	The alpha quality Active Scanner rules.	<input type="checkbox"/>
Active scanner rules (beta)	30.0.0	The beta quality Active Scanner rules.	<input type="checkbox"/>
Advanced SQLInjection Scanner	13.0.0	An advanced active injection bundle for SQLi (derived by SQLMap)	<input type="checkbox"/>

Check for Updates on startup

ZAP Releases

Automatically download new ZAP releases (you will be prompted to install them)

Add-on updates

Check for updates to the add-ons you have installed

Automatically install updates to the add-ons you have installed

Automatically install updates to the scanner rules you have installed

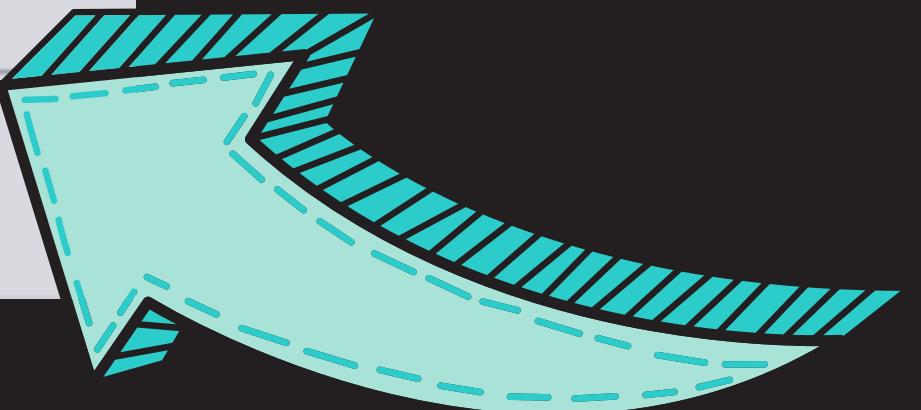
New Add-ons

Report new release quality add-ons

Report new beta quality add-ons

Report new alpha quality add-ons

อีกทั้งการผู้พัฒนาไม่ได้เพียงใส่ Add-ons ไปเฉย ๆ แล้วหยุดการพัฒนา แต่ยังมีการพัฒนา ปรับปรุง และแก้ไข Add-ons ที่ออกมาเพื่อ คุณภาพ และความ ปลอดภัย ดังในภาพที่มีการปล่อยเวอร์ชัน ที่ทำให้ active scanner rules นั้นมีประสิทธิภาพมากยิ่งขึ้น



COUPLING COHESION

The image shows a screenshot of a GitHub interface. At the top, there are two repository cards: 'zaproxy' (The OWASP ZAP core project) and 'zap-extensions' (OWASP ZAP Add-ons). Both are marked as 'Public'. Below these are three detailed repository views:

- zap-extensions / addOns /**: Shows a list of modules: accessControl, alertFilters, allinonenotes, amf, ascanrules, and ascanrulesAlpha. Each module has a small icon, a name, and a brief description.
- zap-extensions / addOns / directorylistv2_3_lc /**: Shows files: zapbot, src/main, CHANGELOG.md, directorylistv2_3_lc.gradle.kts, and gradle.properties. Each file has a small icon, a name, and a brief description.
- zap-extensions / addOns / allinonenotes /**: Shows files: zapbot, src/main, CHANGELOG.md, allinonenotes.gradle.kts, and gradle.properties. Each file has a small icon, a name, and a brief description.

At the bottom, there is a table titled 'Name' with columns 'Name', 'Version', and 'Description'. It lists several addOns:

Name	Version	Description
Active scanner rules	35.0.0	The release quality Active Scanner rules
Active scanner rules (alpha)	28.0.0	The alpha quality Active Scanner rules
Active scanner rules (beta)	30.0.0	The beta quality Active Scanner rules
Advanced SQLInjection Scanner	13.0.0	An advanced active injection bundle for SQLi (derived by SQLMap)

SPLIT MODULE

มีการแยก Repository ที่เก็บ addOns กับ Zap proxy
ออกจากกัน

ใน Repository ของ addOns มีการแยก Module
ของแต่ละ addOn ออกจากกัน
แต่ละ addOn จะมีการทำงานของตัวเองไม่เกี่ยวข้องกับ
Modules อื่นๆ

สามารถพัฒนา, แก้ไข และลบ addOn ได้โดยไม่ส่งผล
กระทบต่อ Modules อื่นๆ

Path: zaproxy/zap-extensions/tree/main/addOns
zaproxy/zap-extensions/tree/main/addOns/allinonenotes
zaproxy/zap-extensions/tree/main/addOns/directorylistv2_3_lc

COUPLING COHESION

SPLIT MODULE

```
public AddOnCollection(File[] dirs) {  
    if (dirs != null) {  
        for (File dir : dirs) {  
            try {  
                this.addDirectory(dir);  
            } catch (Exception e) {  
                logger.error(e.getMessage(), e);  
            }  
        }  
    }  
}
```

```
// Load the addons  
try (DirectoryStream<Path> addOnFiles =  
    Files.newDirectoryStream(dir.toPath(), "*" + AddOn.FILE_EXTENSION)) {  
    for (Path addOnFile : addOnFiles) {  
        AddOn.createAddOn(addOnFile)
```

```
    if (add) {  
        logger.debug(  
            "Found addon "  
            + ao.getId()  
            + " version "  
            + ao.getVersion());  
        this.addOns.add(ao);  
    }  
}
```

Zaproxy จะหา Directory ของ addOns เพื่อใช้งาน addOns ไม่เกี่ยวข้องกับ Core ของ Zaproxy ทำให้ไม่เกิดปัญหา กับ Product โดยตรง

Zaproxy เป็น Open-Source Project การทำแบบนี้สามารถช่วยให้ผู้พัฒนาสามารถพัฒนา addOn และจากตัวระบบได้โดยไม่ส่งผลต่อตัว Modules อื่นๆ ของระบบ

TESTABILITY

Tactic : Specialized Interfaces

เป็น setter() ที่ Validate ค่าไม่ให้เป็น Null เพื่อป้องการความผิดพลาดในการทำงานของโปรแกรม

และก็มีการ throwException() เพื่อกำหนดให้ Developer ที่จะมาใช้งานสามารถหาจุดบกพร่องได้ง่าย มีความสะดวกรวดเร็ว ในการทำงานและทำความเข้าใจ

```
private void addDirectory(File dir) throws Exception {
    if (dir == null) {
        logger.error("Null directory supplied");
        return;
    }
    if (!dir.exists()) {
        logger.warn(
            "Skipping enumeration of add-ons, the directory does not exist: "
            + dir.getAbsolutePath());
        return;
    }
    if (!dir.isDirectory()) {
        logger.warn("Not a directory: " + dir.getAbsolutePath());
        return;
    }
}
```

```
public Builder setHeaders(List<HttpHeaderField> headers) {
    if (headers == null) {
        throw new IllegalArgumentException("Parameter headers must not be null.");
    }
    if (!headers.isEmpty()) {
        List<HttpHeaderField> validHeaders = new ArrayList<>();
        for (HttpHeaderField headerField : headers) {
            if (headerField == null) {
                throw new IllegalArgumentException("Element of headers must not be null.");
            }
            if (headerField.getName() != null
                && !headerField.getName().trim().isEmpty()
                && headerField.getValue() != null) {
                validHeaders.add(headerField);
            }
        }
        this.headers = Collections.unmodifiableList(validHeaders);
    } else {
        this.headers = Collections.emptyList();
    }
    return this;
}
```

```
private static final String USER_NAME = "username";
private static int CONTEXT_ID = 23;
```

```
@Test
public void shouldEncodeAndDecodeProperly() {
    // Given
    User user = spy(new User(CONTEXT_ID, USER_NAME));
    user.setAuthenticationCredentials(mockedCredentials);
    doReturn(mockedContext).when(user).getContext();
    // When
    String encoded = User.encode(user);
    User result = User.decode(CONTEXT_ID, encoded, mockedExtension);
    // Then
    assertEquals(user.getName(), result.getName());
    assertEquals(user.isEnabled(), result.isEnabled());
    assertEquals(user.getId(), result.getId());
    assertEquals(user.getContextId(), result.getContextId());
}
```

```
@Test
public void shouldGenerateUniqueIds() {
    // Given
    User u1 = new User(CONTEXT_ID, USER_NAME);
    User u2 = new User(CONTEXT_ID, USER_NAME);
    User u3 = new User(CONTEXT_ID, USER_NAME);
    User u4 = new User(CONTEXT_ID, USER_NAME);
    // When/Then
    assertThat(u1.getId(), not(anyOf(is(u2.getId()), is(u3.getId()), is(u4.getId()))));
    assertThat(u2.getId(), not(anyOf(is(u3.getId()), is(u4.getId()))));
    assertThat(u3.getId(), not(is(u4.getId())));
    User u5 = new User(CONTEXT_ID, USER_NAME, u4.getId() + 5);
    User u6 = new User(CONTEXT_ID, USER_NAME);
    assertThat(u6.getId(), greaterThan(u5.getId()));
}
```

Tactic : Abstract data source

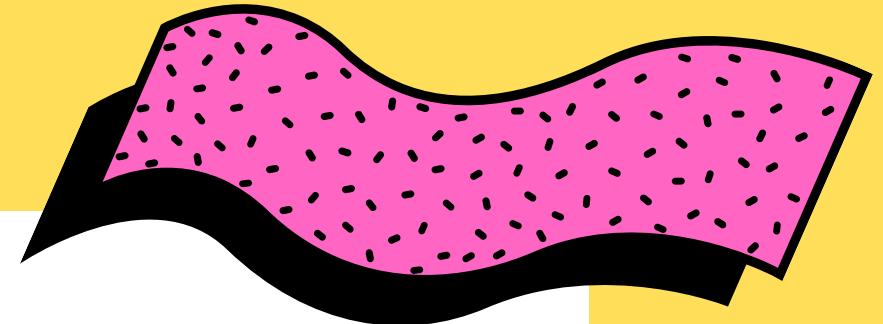
Tactic : Sandbox

มีการสร้าง dummy User เพื่อทำการนำมาใช้สำหรับการทำ testing ระบบ

มีการใช้ assertion เพื่อกำเป็นตัว Tester

ส่งผลให้สามารถจำลองข้อมูลเพื่อมาทดสอบระบบให้คล้ายกับ OnProduct ที่สุด ทำให้สามารถหาข้อผิดพลาดของระบบได้

AVAILABILITY



Tactic : Exception handling

มีการคาดการณ์ Error กี่จะเกิดขึ้น จึงใช่ Try Catch และ Throw exception เพื่อจัดการกับ Error กี่คาดว่าจะเกิดขึ้น

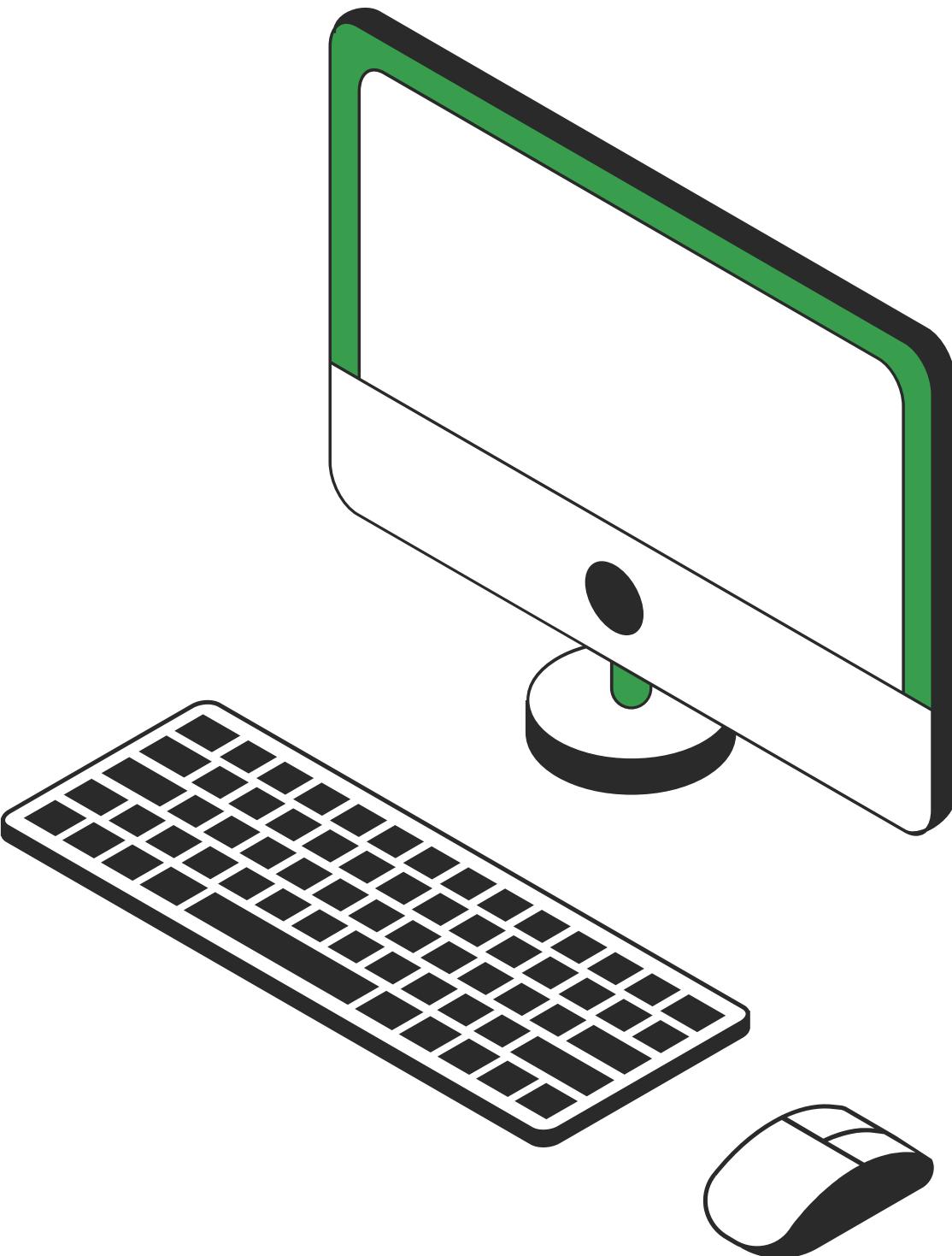
ทำให้ระบบสามารถรับรู้ปัญหาได้ และทำให้ระบบยังทำงานต่อไปได้

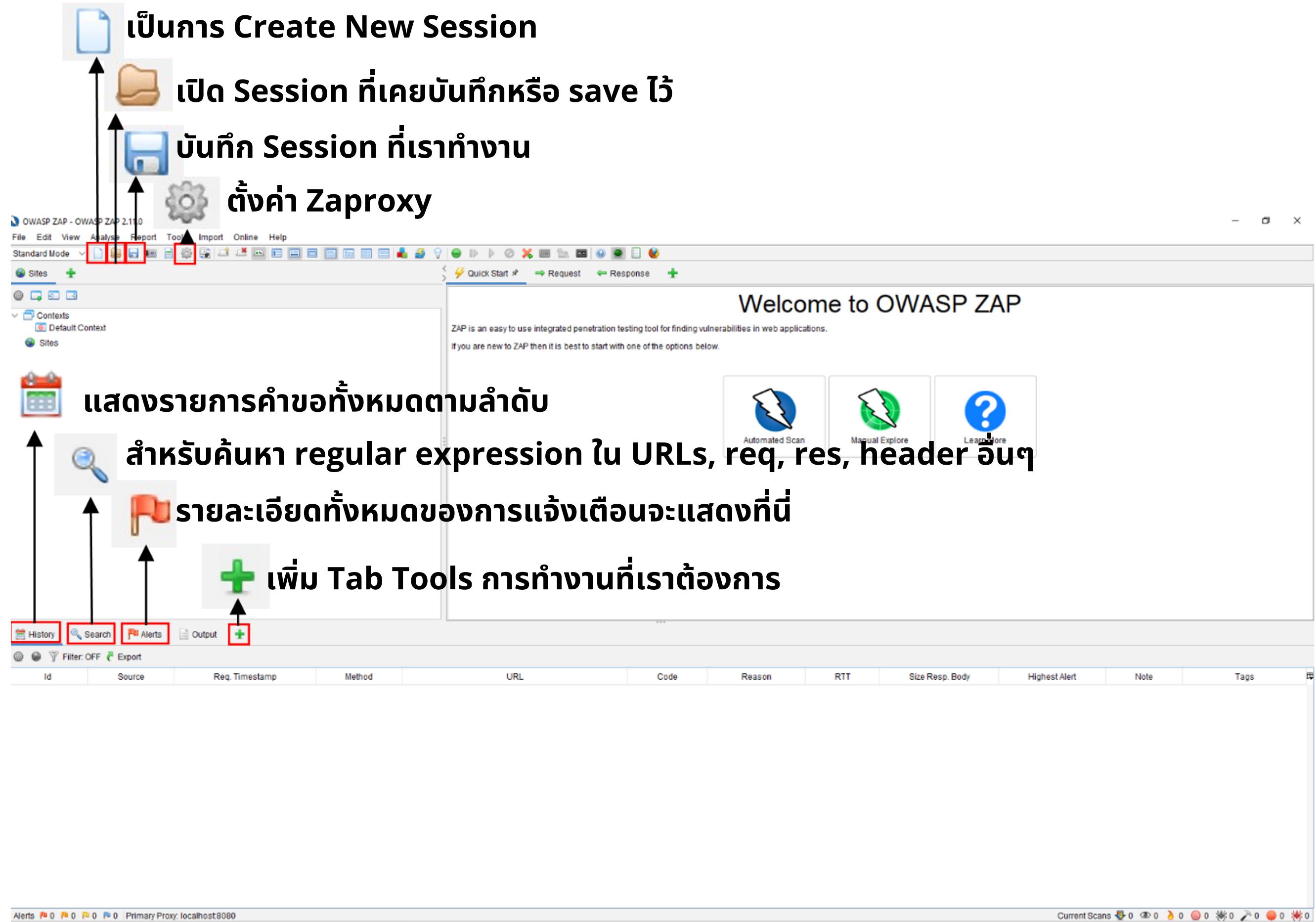
```
107     /**
108      * Main method
109      *
110      * @param args the arguments passed to the command line version
111      * @throws Exception if something wrong happens
112      */
113     public static void main(String[] args) throws Exception {
114         setCustomErrStream();
115
116         CommandLine cmdLine = null;
117         try {
118             cmdLine = new CommandLine(args != null ? Arrays.copyOf(args, args.length) : null);
119
120         } catch (final Exception e) {
121             // Cant use the CommandLine help here as the
122             // i18n messages wont have been loaded
123             System.out.println("Failed due to invalid parameters: " + Arrays.toString(args));
124             System.out.println(e.getMessage());
125             System.out.println("Use '-h' for more details.");
126             System.exit(1);
127         }
128     }
```

```
if (publisher == null) {
    throw new InvalidParameterException("Publisher must not be null");
}
if (eventTypes == null || eventTypes.length == 0) {
    throw new InvalidParameterException("At least one event type must be specified");
}
```

Path: zaproxy/zaproxy/zap/src/main/java/org/zaproxy/zap/ZAP.java
zaproxy/zaproxy/zap/src/main/java/org/zaproxy/zap/eventBus/SimpleEventBus.java

APPROPRIATENESS RECOGNIZABILITY





SECURITY



Tactic: Encrypt data, Identity actors

```
/**  
 * Encodes the User in a String. Fields that contain strings are Base64 encoded.  
 *  
 * @param user the user  
 * @return the string  
 */  
  
public static String encode(User user) {  
    StringBuilder out = new StringBuilder();  
    out.append(user.id).append(FIELD_SEPARATOR);  
    out.append(user.isEnabled()).append(FIELD_SEPARATOR);  
    out.append(Base64.encodeBase64String(user.name.getBytes())).append(FIELD_SEPARATOR);  
    out.append(user.getContext().getAuthenticationMethod().getType().getUniqueIdentifier())  
        .append(FIELD_SEPARATOR);  
    out.append(user.authenticationCredentials.encode(FIELD_SEPARATOR));  
    if (log.isDebugEnabled()) log.debug("Encoded user: " + out.toString());  
    return out.toString();  
}
```

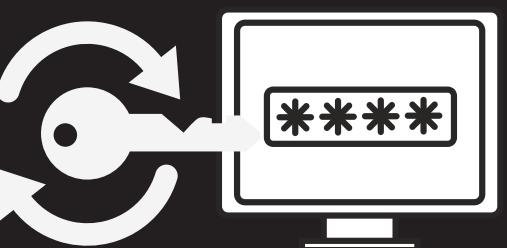
มีการเข้ารหัส User ที่เราส่งไปเพื่อนำไปใช้ใน Web session



PATH: zaproxy/zap/src/main/java/org/zaproxy/zap/users/User.java

```
/**  
 * Modifies a message so its Request Header/Body matches the web session corresponding to this  
 * user.  
 *  
 * @param message the message  
 */  
  
public void processMessageToMatchUser(HttpMessage message) {  
    // If the user is not yet authenticated, authenticate now  
    // Make sure there are no simultaneous authentications for the same user  
    synchronized (this) {  
        if (this.requiresAuthentication()) {  
            this.authenticate();  
            if (this.requiresAuthentication()) {  
                log.info("Authentication failed for user: " + name);  
                return;  
            }  
        }  
        processMessageToMatchAuthenticatedSession(message);  
    }  
  
/**  
 * Modifies a message so its Request Header/Body matches the web session corresponding to this  
 * user.  
 *  
 * @param message the message  
 */  
  
public void processMessageToMatchAuthenticatedSession(HttpMessage message) {  
    getContext()  
        .getSessionManagementMethod()  
        .processMessageToMatchSession(message, authenticatedSession);  
}
```

เป็นการแปลงข้อมูล User เป็น HttpMessage เพื่อไปใช้เช็คสิทธิเข้าถึงใน Web session



Tactic: Encrypt data, Identity actors

```
public static HttpMessage prepareMessage(  
    SessionManagementMethod sessionManagementMethod, User user) {  
  
    HttpMessage msg = new HttpMessage();  
    // Make sure the message will be sent with a good WebSession that can record the changes  
    if (user.getAuthenticatedSession() == null)  
        user.setAuthenticatedSession(sessionManagementMethod.createEmptyWebSession());  
    msg.setRequestingUser(user);  
  
    return msg;  
}
```

prepareMessage จะทำการ
setAuthenticatedSession ให้ User

```
public void setAuthenticatedSession(WebSession session) {  
    this.authenticatedSession = session;  
}
```

WebSession authenticate จะสร้าง
Websession สำหรับ User ที่ส่งไปเท่านั้น



```
@Override  
public WebSession authenticate(  
    SessionManagementMethod sessionManagementMethod,  
    AuthenticationCredentials credentials,  
    User user)  
throws UnsupportedAuthenticationCredentialsException {  
  
    WebSession session = user.getAuthenticatedSession();  
    if (session == null) session = sessionManagementMethod.createEmptyWebSession();  
  
    // type check  
    if (!(credentials instanceof UsernamePasswordAuthenticationCredentials)) {  
        user.getAuthenticationState()  
            .setLastAuthFailure(  
                "Credentials not UsernamePasswordAuthenticationCredentials");  
        throw new UnsupportedAuthenticationCredentialsException(  
            "Form based authentication method only supports "  
            + UsernamePasswordAuthenticationCredentials.class.getSimpleName());  
    }  
    UsernamePasswordAuthenticationCredentials userCredentials =  
        (UsernamePasswordAuthenticationCredentials) credentials;  
  
    AuthScope stateAuthScope = null;  
    NTCredentials stateCredentials = null;  
    try {  
        stateAuthScope =  
            new AuthScope(  
                this.hostname,  
                this.port,  
                (this.realm == null || this.realm.isEmpty())  
                    ? AuthScope.ANY_REALM  
                    : this.realm);  
        stateCredentials =  
            new NTCredentials(  
                userCredentials.getUsername(),  
                userCredentials.getPassword(),  
                InetAddress.getLocalHost().getCanonicalHostName(),  
                this.realm);  
        session.getHttpState().setCredentials(stateAuthScope, stateCredentials);  
    } catch (UnknownHostException e1) {  
        user.getAuthenticationState().setLastAuthFailure(e1.getMessage());  
        log.error(e1.getMessage(), e1);  
    }  
    user.getAuthenticationState().setLastAuthFailure("");  
    return session;  
}
```

PATH: zaproxy/zap/src/main/java/org/zaproxy/zap/authentication/AuthenticationHelper.java

Let's see! Design Patterns



Creational Design Patterns

- Builder Pattern
- Singleton Pattern
- Factory Method

Structural Design Patterns

- Adapter Pattern

Behavioral Design Patterns

- Command Pattern

Creational Design Patterns

- Singleton Pattern
-

- Builder Pattern
-

- Factory Method
-

Singleton Pattern

- **zaproxy/zap/src/main/java/org/parosproxy/paros/model/Model.java**
- **Line: 213 - 238**
- **Link:**
<https://github.com/zaproxy/zaproxy/blob/main/zap/src/main/java/org/parosproxy/paros/model/Model.java#L213-L238>
- **Usage:**
<https://github.com/zaproxy/zaproxy/blob/main/zap/src/main/java/org/zaproxy/zap/utils/ApiUtils.java>

Code

```
public class Model {  
  
    private static Model model = null;  
  
    public static Model getSingleton() {  
        if (model == null) {  
            // ZAP: Changed to use the method createSingleton().  
            createSingleton();  
        }  
        return model;  
    }  
  
    // ZAP: Added method.  
    private static synchronized void createSingleton() {  
        if (model == null) {  
            model = new Model();  
        }  
    }  
}
```

Path: zaproxy/zap/src/main/java/org/parosproxy/paros/model/Model.java

Usage

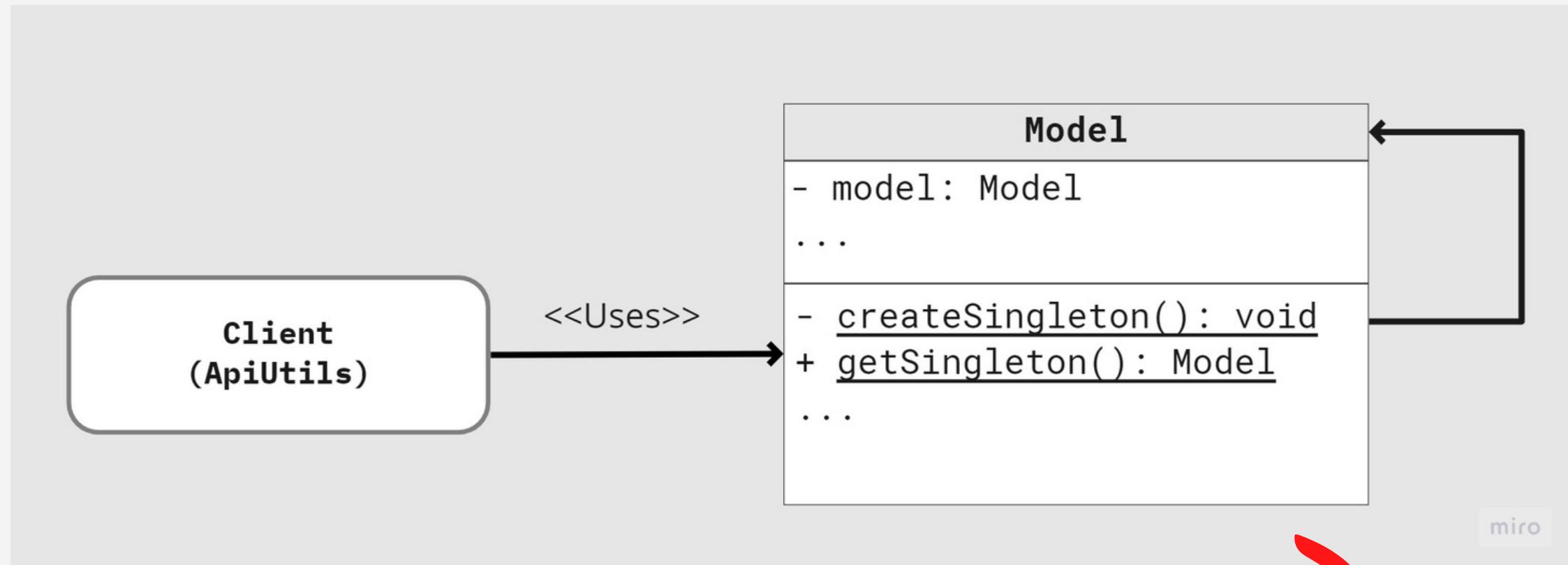
```
/** Utils for manipulating API calls and parameters. */
public final class ApiUtils {
```

```
    public static Context getContextByParamId(JSONObject params, String contextIdParamName)
        throws ApiException {
        int contextId = getIntParam(params, contextIdParamName);
        Context context = Model.singleton().getSession().getContext(contextId);
        if (context == null) {
            throw new ApiException(Type.CONTEXT_NOT_FOUND, contextIdParamName);
        }
        return context;
    }
```



Path: zaproxy/zap/src/main/java/org/zaproxy/zap/utils/ApiUtils.java

UML Diagram



```
public static Model getSingleton() {
    if (model == null) {
        // ZAP: Changed to use the method createSingleton().
        createSingleton();
    }
    return model;
}

// ZAP: Added method.
private static synchronized void createSingleton() {
    if (model == null) {
        model = new Model();
    }
}
```

Builder Pattern

- **zaproxy/zap/src/main/java/org/zaproxy/zap/network/HttpRequestConfig.java**
- **Line: 129 - 226**
- **Link:**
<https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zap/src/main/java/org/zaproxy/zap/network/HttpRequestConfig.java#L128-L226>
- **Usage:**
<https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zap/src/main/java/org/parosproxy/paros/core/proxy/ProxyThread.java>

Code

```
public class HttpRequestConfig {

    /**
     * Constant that indicates that no value was set (thus using the value defined by the sender
     * implementation).
     *
     * @since 2.7.0
     */
    public static final int NO_VALUE_SET = -1;

    private final boolean followRedirects;
    private final HttpRedirectionValidator redirectionValidator;
    private final boolean notifyListeners;
    private final int soTimeout;
```

```
/** Builder of {@link HttpRequestConfig}. */
public static class Builder {

    private boolean followRedirects;
    private HttpRedirectionValidator redirectionValidator;
    private boolean notifyListeners;
    private int soTimeout;

    private Builder() {
        this.followRedirects = false;
        this.redirectionValidator = DefaultHttpRedirectionValidator.INSTANCE;
        this.notifyListeners = true;
        this.soTimeout = NO_VALUE_SET;
    }

    private Builder(HttpRequestConfig config) {
        this.followRedirects = config.isFollowRedirects();
        this.redirectionValidator = config.getRedirectionValidator();
        this.notifyListeners = config.isNotifyListeners();
    }

    public HttpRequestConfig build() {
        return new HttpRequestConfig(
            followRedirects, redirectionValidator, notifyListeners, soTimeout);
    }
}
```

Path: zaproxy/zap/src/main/java/org/zaproxy/zap/network/HttpRequestConfig.java

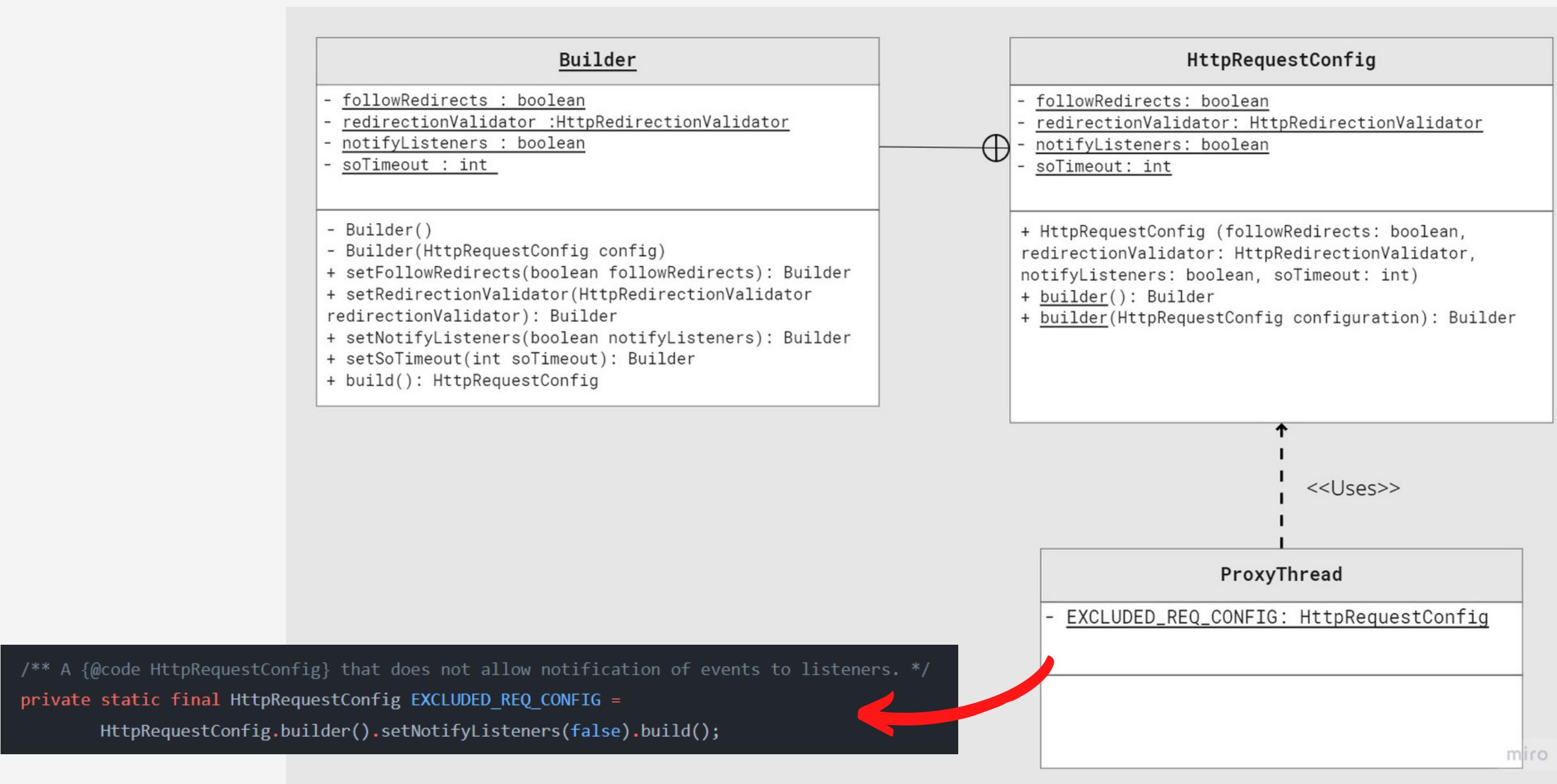
Usage

```
public class ProxyThread implements Runnable {
```

```
    /** A {@code HttpRequestConfig} that does not allow notification of events to listeners. */
    private static final HttpRequestConfig EXCLUDED_REQ_CONFIG =
        HttpRequestConfig.builder().setNotifyListeners(false).build();
```

Path: zaproxy/zap/src/main/java/org/parosproxy/paros/core/proxy/ProxyThread.java

UML Diagram



Factory Method

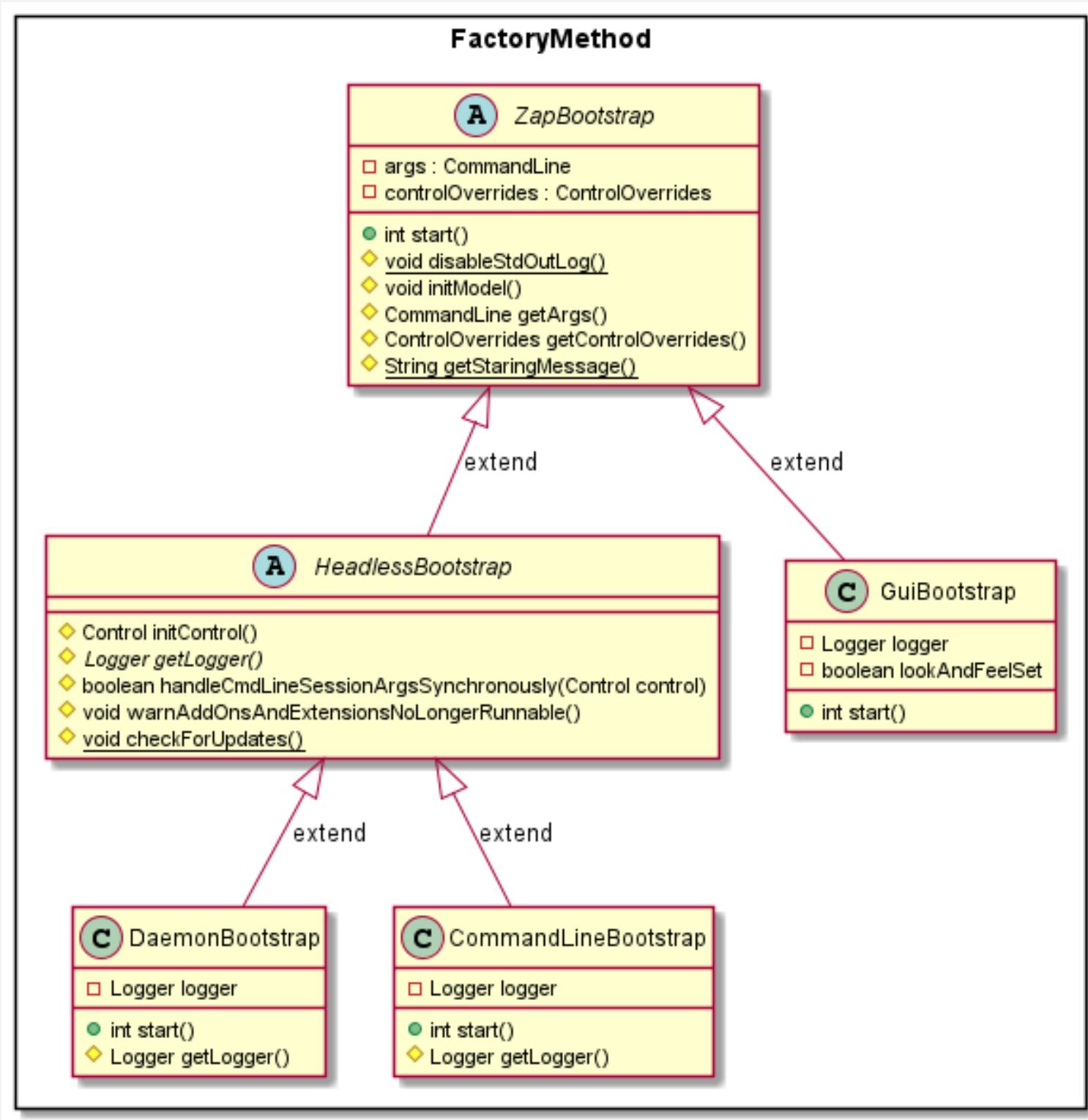
- **Path:** zap/src/main/java/org/zaproxy/zap/**ZapBootstrap.java**
- **Link:**
<https://github.com/zaproxy/zaproxy/blob/ba55831c0ee1667586275307cbc14be1528442ca/zap/src/main/java/org/zaproxy/zap/ZapBootstrap.java>
- **Sub Class:**
 - zap/src/main/java/org/zaproxy/zap/**HeadlessBootstrap.java**
 - zap/src/main/java/org/zaproxy/zap/**CommandLineBootstrap.java**
 - zap/src/main/java/org/zaproxy/zap/**DaemonBootstrap.java**
 - zap/src/main/java/org/zaproxy/zap/**GuiBootstrap.java**
- **Usage:**
 - zap/src/main/java/org/zaproxy/zap/**ZAP.java**
 - **Line:** 159-172

Code

```
private static ZapBootstrap createZapBootstrap(CommandLine cmdLineArgs) {
    ZapBootstrap bootstrap;
    if (cmdLineArgs.isGUI()) {
        ZAP.processType = ProcessType.desktop;
        bootstrap = new GuiBootstrap(cmdLineArgs);
    } else if (cmdLineArgs.isDaemon()) {
        ZAP.processType = ProcessType.daemon;
        bootstrap = new DaemonBootstrap(cmdLineArgs);
    } else {
        ZAP.processType = ProcessType.cmdline;
        bootstrap = new CommandLineBootstrap(cmdLineArgs);
    }
    return bootstrap;
}
```

Path: zap/src/main/java/org/zaproxy/zap/ZAP.java

UML Diagram



Structural Design Patterns

- Adapter Pattern

Adapter

- **Path:**

zap/src/main/java/org/zaproxy/zap/view/messagelocation/MessageLocationProducerFocusListenerAdapter.java

- **Link:**

<https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zap/src/main/java/org/zaproxy/zap/view/messagelocation/MessageLocationProducerFocusListenerAdapter.java>

- **Usage:**

- HttpRequestHeaderPanelSyntaxHighlightTextView.java
- HttpRequestAllPanelSyntaxHighlightTextView.java
- HttpRequestBodyPanelSyntaxHighlightTextView.java

Code

```
private final List<MessageLocationProducerFocusListener> focusListeners;
private final MessageLocationProducer source;

public MessageLocationProducerFocusListenerAdapter(MessageLocationProducer source) {
    this.source = source;
    focusListeners = new ArrayList<>(5);
}
```

```
public boolean hasFocusListeners() {
    return !focusListeners.isEmpty();
}

public void addFocusListener(MessageLocationProducerFocusListener focusListener) {
    focusListeners.add(focusListener);
}

public void removeFocusListener(MessageLocationProducerFocusListener focusListener) {
    focusListeners.remove(focusListener);
}
```

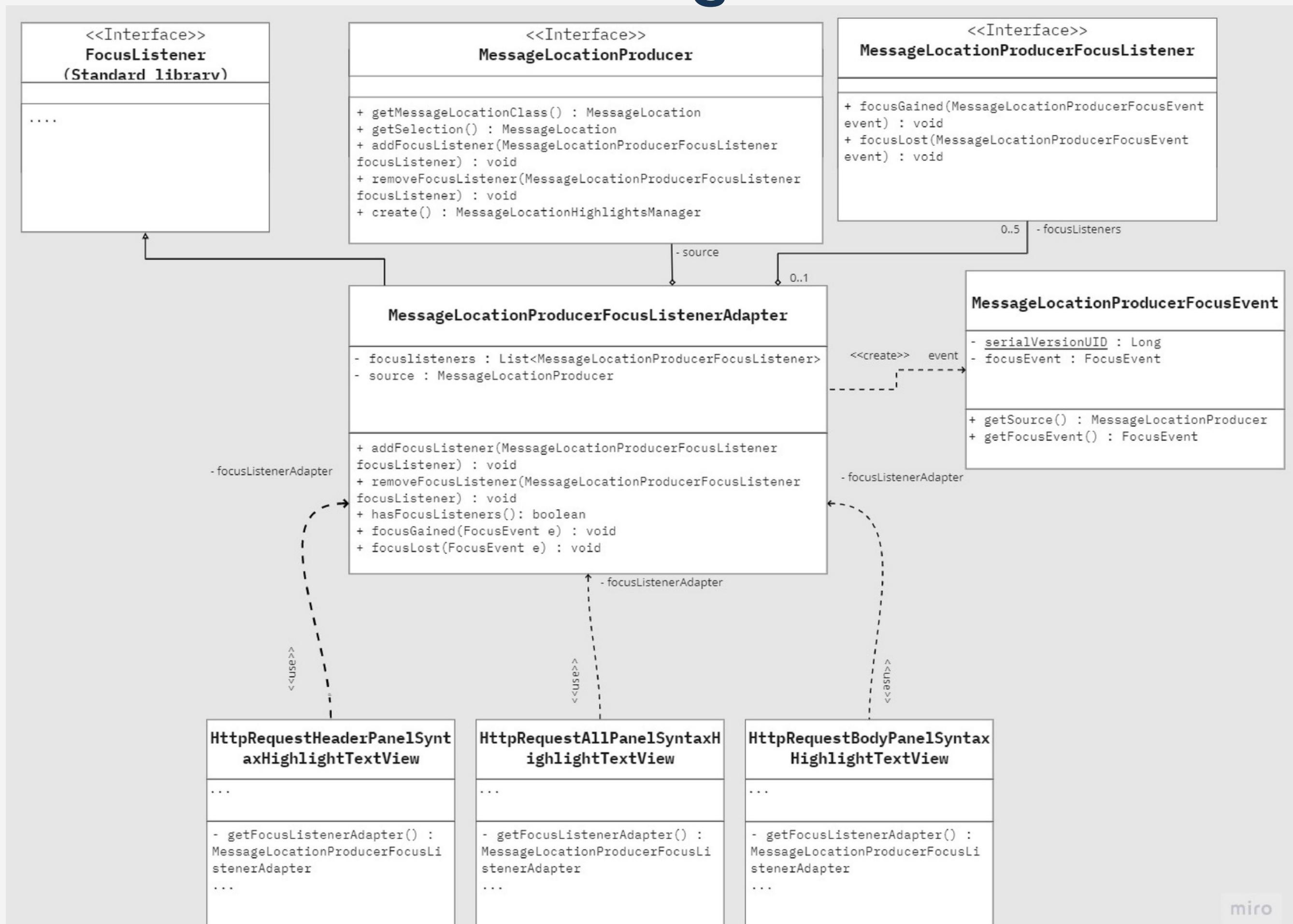
```
@Override
public void focusGained(FocusEvent e) {
    if (focusListeners.isEmpty()) {
        return;
    }

    MessageLocationProducerFocusEvent event = new MessageLocationProducerFocusEvent(source, e);
    for (MessageLocationProducerFocusListener focusListener : focusListeners) {
        focusListener.focusGained(event);
    }
}

@Override
public void focusLost(FocusEvent e) {
    if (focusListeners.isEmpty()) {
        return;
    }

    MessageLocationProducerFocusEvent event = new MessageLocationProducerFocusEvent(source, e);
    for (MessageLocationProducerFocusListener focusListener : focusListeners) {
        focusListener.focusLost(event);
    }
}
```

UML Diagram



Usage

```
private MessageLocationProducerFocusListenerAdapter focusListenerAdapter;
```



```
@Override  
public void addFocusListener(MessageLocationProducerFocusListener focusListener) {  
    getFocusListenerAdapter().addFocusListener(focusListener);  
}  
  
@Override  
public void removeFocusListener(MessageLocationProducerFocusListener focusListener) {  
    getFocusListenerAdapter().removeFocusListener(focusListener);  
  
    if (!getFocusListenerAdapter().hasFocusListeners()) {  
        getHttpPanelTextArea().removeFocusListener(focusListenerAdapter);  
        focusListenerAdapter = null;  
    }  
}  
  
private MessageLocationProducerFocusListenerAdapter getFocusListenerAdapter() {  
    if (focusListenerAdapter == null) {  
        focusListenerAdapter = new MessageLocationProducerFocusListenerAdapter(this);  
        getHttpPanelTextArea().addFocusListener(focusListenerAdapter);  
    }  
    return focusListenerAdapter;  
}
```

HttpRequestHeaderPanelSyntaxHighlightTextView.java

Behavioral Design Patterns

- Command Pattern

Command

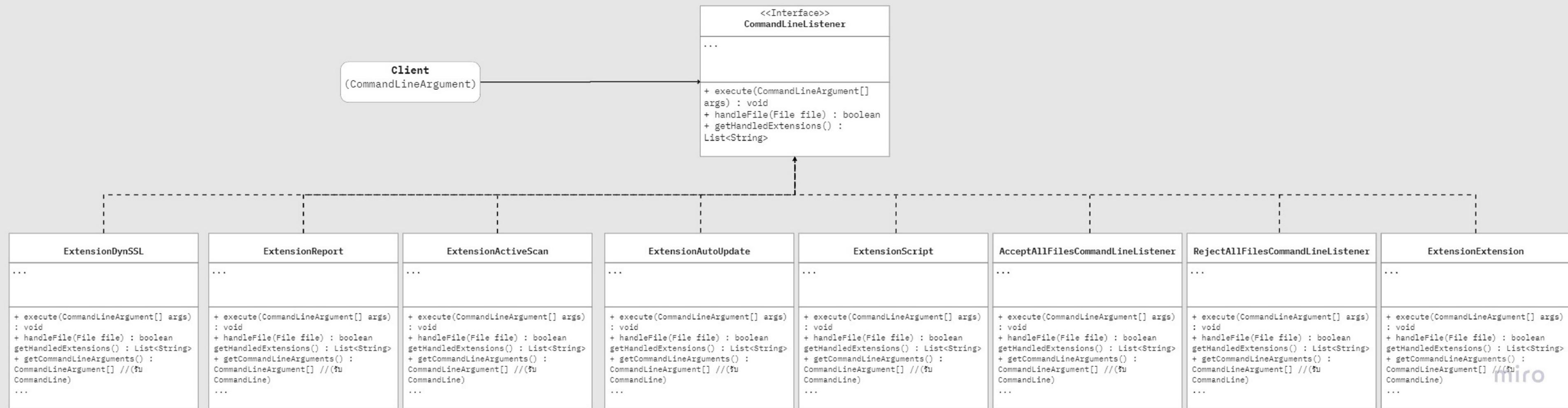
- **Path:** zap/src/main/java/org/parosproxy/paros/extension/CommandLineListener.java
- **Link:**
<https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zap/src/main/java/org/parosproxy/paros/extension/CommandLineListener.java>
- **Usage:**
 - ExtensionReport.java
 - ExtensionActiveScan.java
 - ExtensionAutoUpdate.java
 - ExtensionDynSSL.java
 - ExtensionExtension.java
 - ExtensionScript.java
 - CommandLineUnitTest.java (2 Class)

Code

```
public interface CommandLineListener {  
    /**  
     * execute the command line using the argument provided.  
     *  
     * @param args the command line arguments  
     */  
    void execute(CommandLineArgument[] args);  
  
    /**  
     * Handle the specified file (in whatever way is appropriate). This will only be called for  
     * files specified on the command line without switches and which match one of the extensions  
     * returned by getHandledExtensions()  
     *  
     * @param file the file provided through the command line  
     * @return true if the listener handled the file, false otherwise  
     */  
    boolean handleFile(File file);  
  
    /**  
     * Get the list of extensions this listener can handle  
     *  
     * @return a {@code List} with the handled extensions  
     */  
    List<String> getHandledExtensions();  
}
```

<https://github.com/zaproxy/zaproxy/blob/8ac4b10d3d9d816957985694c2b7cf127d1364d3/zap/src/main/java/org/parosproxy/paros/extension/CommandLineListener.java>

UML Diagram



Usage

```
@Override  
public void execute(CommandLineArgument[] args) {  
  
    if (arguments[ARG_LAST_SCAN_REPORT_IDX].isEnabled()) {  
        CommandLineArgument arg = arguments[ARG_LAST_SCAN_REPORT_IDX];  
        ReportLastScan report = new ReportLastScan();  
        // ZAP: Removed unnecessary cast.  
        String fileName = arg.getArguments().get(0);  
        try {  
            report.generate(fileName, ReportLastScan.ReportType.HTML);  
            CommandLine.info("Last Scan Report generated at " + fileName);  
        } catch (Exception e) {  
            CommandLine.error(e.getMessage(), e);  
        }  
    } else {  
        return;  
    }  
}  
  
private CommandLineArgument[] getCommandLineArguments() {  
    arguments[ARG_LAST_SCAN_REPORT_IDX] =  
        new CommandLineArgument(  
            "-last_scan_report",  
            1,  
            null,  
            "",  
            Constant.messages.getString("report.cmdline.gen.help"));  
    return arguments;  
}
```

```
@Override  
public boolean handleFile(File file) {  
    // Cant handle any files  
    return false;  
}  
  
@Override  
public List<String> getHandledExtensions() {  
    // Cant handle any extensions  
    return null;  
}
```

ExtensionReport.java

Thank you!

Feel free to rate us! & to approach us
if you have any questions.

