# Guide for developing rules

This guide explains how to write extension rules for the PolicyLoader ZAP add on.

**1 - Writing rules:**
The rule interface is defined as follows:

```
public interface Rule {
   String getName();
   String getDescription();
   boolean isViolated(HttpMessage msg);
}
```

- *String getName()* : The name of your rule. This name is used to represent the rule in the report and in alerts.
- *String getDescription()* : A description for the rule.
- *boolean isViolated(HttpMessage msg)* : A violation test. The method takes an HttpMessage as defined in *org.parosproxy.paros.network.HttpMessage* . We note that an HttpMessage contains both the request and response messages, it is up to the developer to enforce his rule on either or both of them.

To write new rules, the developer implements the Rule interface.

**2 - Compiling rules:**
To write a new policy, the developer creates a new directory <rules_dir> and places the rules source code files inside it.
The rules can be compiled with the following commands:
*javac -cp <ZAP.jar> <rules_dir>*.java*
*jar cvf  <policyname>.jar <rules_dir>*

Where :
<ZAP.jar> the ZAP program in a JAR format.
<policyname> : the name of your policy.

We note that in our case <ZAP.jar> is contained in  “Group17/zaproxy/zap/build/libs/” after having executed  “*gradlew clean build*” .

**3 - Loading policies to ZAP :**
The developer is able to load his policies to ZAP by going to tools -> Policy Loader and selecting the .jar files of his policies. He should be presented with an alert message notifying him of the success (or failure) of the operation.