

# Validator Source Overview

# Table of Contents

The ValidatorAPI defines methods to access its database of AVPs and check if the AVP and message have the proper structure.

The Validator is currently message oriented. This means that it declares methods that center on message consistency checks. The class containing all validation logic is `org.jdiameter.common.impl.validation.DiameterMessageValidator`. It exposes the following methods:

*public boolean isOn();*

Simple method to determine if the `Validator` is enabled.

*public ValidatorLevel getSendLevel();*

Returns the validation level of outgoing messages. It can have one of the following values: `OFF`, `MESSAGE`, `ALL`.

*public ValidatorLevel getReceiveLevel()*

Returns the validation level of incoming messages. It can have one of the following values: `OFF`, `MESSAGE`, `ALL`.

*public void validate(Message msg, boolean incoming) throws JAvpNotAllowedException*

Performs validation on a message. Based on the `incoming` flag, the correct validation level is applied. If validation fails, an exception with details is thrown.

*public void validate(Message msg, ValidatorLevel validatorLevel) throws JAvpNotAllowedException*

Performs validation on messages with a specified level. It is a programatical way to allow different levels of validation from those configured. If validation fails, a `JAvpNotAllowedException` with details is thrown.



The current implementation provides more methods, however those are out of scope for this documentation.

A simple example of a Validator use case is shown below:

### Example 1. Validator Message Check Example

The example below is pseudo-code.

```
...
boolean isRequest = true;
boolean isIncoming = false;

DiameterMessageValidator messageValidator = DiameterMessageValidator.
getInstance();
Message message = createMessage(UserDataRequest.MESSAGE_CODE, isRequest,
    applicationId);

//add AVPs
...
//perform check
try{
    messageValidator.validate(message, isIncoming);
}
catch(JAvpNotAllowedException e) {
    System.err.println("Failed to validate ..., avp code: " + e.getAvpCode() + "
    avp vendor:" + e.getVendorId() + ", message:" + e.getMessage());
}
```