| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV PACKAGE**](http://docs.google.com/javax/security/cert/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/sound/midi/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/security/sasl/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.security.sasl

Contains class and interfaces for supporting SASL.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**SaslClient**](http://docs.google.com/javax/security/sasl/SaslClient.html) | Performs SASL authentication as a client. |
| [**SaslClientFactory**](http://docs.google.com/javax/security/sasl/SaslClientFactory.html) | An interface for creating instances of SaslClient. |
| [**SaslServer**](http://docs.google.com/javax/security/sasl/SaslServer.html) | Performs SASL authentication as a server. |
| [**SaslServerFactory**](http://docs.google.com/javax/security/sasl/SaslServerFactory.html) | An interface for creating instances of SaslServer. |

| **Class Summary** | |
| --- | --- |
| [**AuthorizeCallback**](http://docs.google.com/javax/security/sasl/AuthorizeCallback.html) | This callback is used by SaslServer to determine whether one entity (identified by an authenticated authentication id) can act on behalf of another entity (identified by an authorization id). |
| [**RealmCallback**](http://docs.google.com/javax/security/sasl/RealmCallback.html) | This callback is used by SaslClient and SaslServer to retrieve realm information. |
| [**RealmChoiceCallback**](http://docs.google.com/javax/security/sasl/RealmChoiceCallback.html) | This callback is used by SaslClient and SaslServer to obtain a realm given a list of realm choices. |
| [**Sasl**](http://docs.google.com/javax/security/sasl/Sasl.html) | A static class for creating SASL clients and servers. |

| **Exception Summary** | |
| --- | --- |
| [**AuthenticationException**](http://docs.google.com/javax/security/sasl/AuthenticationException.html) | This exception is thrown by a SASL mechanism implementation to indicate that the SASL exchange has failed due to reasons related to authentication, such as an invalid identity, passphrase, or key. |
| [**SaslException**](http://docs.google.com/javax/security/sasl/SaslException.html) | This class represents an error that has occurred when using SASL. |

## Package javax.security.sasl Description

Contains class and interfaces for supporting SASL. This package defines classes and interfaces for SASL mechanisms. It is used by developers to add authentication support for connection-based protocols that use SASL.

#### SASL Overview

Simple Authentication and Security Layer (SASL) specifies a challenge-response protocol in which data is exchanged between the client and the server for the purposes of authentication and (optional) establishment of a security layer on which to carry on subsequent communications. It is used with connection-based protocols such as LDAPv3 or IMAPv4. SASL is described in [RFC 2222](http://www.ietf.org/rfc/rfc2222.txt).

There are various *mechanisms* defined for SASL. Each mechanism defines the data that must be exchanged between the client and server in order for the authentication to succeed. This data exchange required for a particular mechanism is referred to to as its *protocol profile*. The following are some examples of mechanims that have been defined by the Internet standards community.

* DIGEST-MD5 ([RFC 2831](http://www.ietf.org/rfc/rfc2831.txt)). This mechanism defines how HTTP Digest Authentication can be used as a SASL mechanism.
* Anonymous ([RFC 2245](http://www.ietf.org/rfc/rfc2245.txt)). This mechamism is anonymous authentication in which no credentials are necessary.
* External ([RFC 2222](http://www.ietf.org/rfc/rfc2222.txt)). This mechanism obtains authentication information from an external source (such as TLS or IPsec).
* S/Key ([RFC 2222](http://www.ietf.org/rfc/rfc2222.txt)). This mechanism uses the MD4 digest algorithm to exchange data based on a shared secret.
* GSSAPI ([RFC 2222](http://www.ietf.org/rfc/rfc2222.txt)). This mechanism uses the [GSSAPI](http://www.ietf.org/rfc/rfc2078.txt) for obtaining authentication information.

Some of these mechanisms provide both authentication and establishment of a security layer, others only authentication. Anonymous and S/Key do not provide for any security layers. GSSAPI and DIGEST-MD5 allow negotiation of the security layer. For External, the security layer is determined by the external protocol.

#### Usage

Users of this API are typically developers who produce client library implementations for connection-based protocols, such as LDAPv3 and IMAPv4, and developers who write servers (such as LDAP servers and IMAP servers). Developers who write client libraries use the SaslClient and SaslClientFactory interfaces. Developers who write servers use the SaslServer and SaslServerFactory interfaces.

Among these two groups of users, each can be further divided into two groups: those who *produce* the SASL mechanisms and those who *use* the SASL mechanisms. The producers of SASL mechanisms need to provide implementations for these interfaces, while users of the SASL mechanisms use the APIs in this package to access those implementations.

## Related Documentation

Please refer to the [Java SASL Programming Guide](http://docs.google.com/technotes/guides/security/sasl/sasl-refguide.html) for information on how to use this API.

**Since:** 1.5

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV PACKAGE**](http://docs.google.com/javax/security/cert/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/sound/midi/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/security/sasl/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).