

JAVAMAIL API - POP3 SERVERS

http://www.tutorialspoint.com/javamail_api/javamail_api_pop3_servers.htm

Copyright © tutorialspoint.com

Post Office Protocol *POP* is an application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection. POP supports simple download-and-delete requirements for access to remote mailboxes. A POP3 server listens on well-known port 110.

Package **com.sun.mail.pop3** is a POP3 protocol provider for the JavaMail API that provides access to a POP3 message store. The table below lists the classes in this package:

Name	Description
POP3Folder	A POP3 Folder <i>canonlybe " INBOX "</i> .
POP3Message	A POP3 Message.
POP3SSLStore	A POP3 Message Store using SSL.
POP3Store	A POP3 Message Store.

Some points to be noted above this provider:

- POP3 provider supports only a single folder named **INBOX**. Due to the limitations of the POP3 protocol, many of the JavaMail API capabilities like event notification, folder management, flag management, etc. are not allowed.
- The POP3 provider is accessed through the JavaMail APIs by using the protocol name *pop3* or a URL of the form *pop3://user:password@host:port/INBOX*".
- POP3 supports no permanent flags. For example the *Flags.Flag.RECENT* flag will never be set for POP3 messages. It's up to the application to determine which messages in a POP3 mailbox are *new*.
- POP3 does not support the *Folder.expunge* method. To delete and expunge messages, set the *Flags.Flag.DELETED* flag on the messages and close the folder using the *Folder.close* method.
- POP3 does not provide a *received date*, so the *getReceivedDate* method will return null.
- When the headers of a POP3 message are accessed, the POP3 provider uses the *TOP* command to fetch all headers, which are then cached.
- When the content of a POP3 message is accessed, the POP3 provider uses the *RETR* command to fetch the entire message.
- The *POP3Message.invalidate* method can be used to invalidate cached data without closing the folder.

The POP3 protocol provider supports the following properties, which may be set in the JavaMail Session object. The properties are always set as strings; the Type column describes how the string is interpreted.

Name	Type	Description
mail.pop3.user	String	Default user name for POP3.
mail.pop3.host	String	The POP3 server to connect to.
mail.pop3.port	int	The POP3 server port to connect to, if the connect method doesn't explicitly specify one. Defaults to

mail.pop3.connectiontimeout	int	Socket connection timeout value in milliseconds. Default is infinite timeout.
mail.pop3.timeout	int	Socket I/O timeout value in milliseconds. Default is infinite timeout.
mail.pop3.rsetbeforequit	boolean	Send a POP3 RSET command when closing the folder, before sending the QUIT command. Default is false.
mail.pop3.message.class	String	Class name of a subclass of <code>com.sun.mail.pop3.POP3Message</code> . The subclass can be used to handle <i>forexample</i> non-standard Content-Type headers. The subclass must have a public constructor of the form <code>MyPOP3Message Folderf, intmsgno</code> throws <code>MessagingException</code> .
mail.pop3.localaddress	String	Local address <i>hostname</i> to bind to when creating the POP3 socket. Defaults to the address picked by the <code>Socket</code> class.
mail.pop3.localport	int	Local port number to bind to when creating the POP3 socket. Defaults to the port number picked by the <code>Socket</code> class.
mail.pop3.apop.enable	boolean	If set to true, use APOP instead of USER/PASS to login to the POP3 server, if the POP3 server supports APOP. APOP sends a digest of the password rather than the clear text password. Defaults to false.
mail.pop3.socketFactory	Socket Factory	If set to a class that implements the <code>javax.net.SocketFactory</code> interface, this class will be used to create POP3 sockets.
mail.pop3.socketFactory.class	String	If set, specifies the name of a class that implements the <code>javax.net.SocketFactory</code> interface. This class will be used to create POP3 sockets.
mail.pop3.socketFactory.fallback	boolean	If set to true, failure to create a socket using the specified socket factory class will cause the socket to be created using the <code>java.net.Socket</code> class. Defaults to true.
mail.pop3.socketFactory.port	int	Specifies the port to connect to when using the specified socket factory. If not set, the default port will be used.
mail.pop3.ssl.enable	boolean	If set to true, use SSL to connect and use the SSL port by default. Defaults to false for the "pop3" protocol and true for the "pop3s" protocol.
mail.pop3.ssl.checkserveridentity	boolean	If set to true, check the server identity as specified by RFC 2595. Defaults to false.
mail.pop3.ssl.trust	String	If set, and a socket factory hasn't been specified, enables use of a <code>MailSSLSocketFactory</code> . If set to "*", all hosts are trusted. If set to a whitespace separated list of hosts, those hosts are trusted. Otherwise, trust depends on the certificate the server presents.
mail.pop3.ssl.socketFactory	SSL	If set to a class that extends the

	Socket Factory	javax.net.ssl.SSLSocketFactory class, this class will be used to create POP3 SSL sockets.
mail.pop3.ssl.socketFactory.class	String	If set, specifies the name of a class that extends the javax.net.ssl.SSLSocketFactory class. This class will be used to create POP3 SSL sockets.
mail.pop3.ssl.socketFactory.port	int	Specifies the port to connect to when using the specified socket factory. If not set, the default port will be used.
mail.pop3.ssl.protocols	string	Specifies the SSL protocols that will be enabled for SSL connections. The property value is a whitespace separated list of tokens acceptable to the javax.net.ssl.SSLSocket.setEnabledProtocols method.
mail.pop3.starttls.enable	boolean	If true, enables the use of the STLS command <i>if supported by the server</i> to switch the connection to a TLS-protected connection before issuing any login commands. Defaults to false.
mail.pop3.starttls.required	boolean	If true, requires the use of the STLS command. If the server doesn't support the STLS command, or the command fails, the connect method will fail. Defaults to false.
mail.pop3.socks.host	string	Specifies the host name of a SOCKS5 proxy server that will be used for connections to the mail server.
mail.pop3.socks.port	string	Specifies the port number for the SOCKS5 proxy server.
mail.pop3.disabletop	boolean	If set to true, the POP3 TOP command will not be used to fetch message headers. Defaults to false.
mail.pop3.forgettopheaders	boolean	If set to true, the headers that might have been retrieved using the POP3 TOP command will be forgotten and replaced by headers retrieved as part of the POP3 RETR command. Defaults to false.
mail.pop3.filecache.enable	boolean	If set to true, the POP3 provider will cache message data in a temporary file rather than in memory. Messages are only added to the cache when accessing the message content. Message headers are always cached in memory <i>ondemand</i> . The file cache is removed when the folder is closed or the JVM terminates. Defaults to false.
mail.pop3.filecache.dir	String	If the file cache is enabled, this property can be used to override the default directory used by the JDK for temporary files.
mail.pop3.cachewriteto	boolean	Controls the behavior of the writeTo method on a POP3 message object. If set to true, and the message content hasn't yet been cached, and ignoreList is null, the message is cached before being written. Otherwise, the message is streamed directly to the output stream without being cached. Defaults to false.
mail.pop3.keepmessagecontent	boolean	If this property is set to true, a hard reference to the cached content will be kept, preventing the

memory from being reused until the folder is closed or the cached content is explicitly invalidated *using the `invalidate` method*. Defaults to false.

In general, applications should not use the classes in this package directly. Instead, they should use the APIs defined by *javax.mail* package *and subpackages*. Applications should never construct instances of *POP3Store* or *POP3Folder* directly. Instead, they should use the `Session` method `getStore` to acquire an appropriate `Store` object, and from that acquire `Folder` objects.

Examples to use POP3 server is demonstrated in chapter [Checking Emails](#).

Loading [MathJax]/jax/output/HTML-CSS/jax.js