JAVAMAIL API - OVERVIEW

The JavaMail API provides a platform-independent and protocol-independent framework to build mail and messaging applications. The JavaMail API provides a set of abstract classes defining objects that comprise a mail system. It is an optional package *standardextension* for reading, composing, and sending electronic messages.

JavaMail provides elements that are used to construct an interface to a messaging system, including system components and interfaces. While this specification does not define any specific implementation, JavaMail does include several classes that implement RFC822 and MIME Internet messaging standards. These classes are delivered as part of the JavaMail class package.

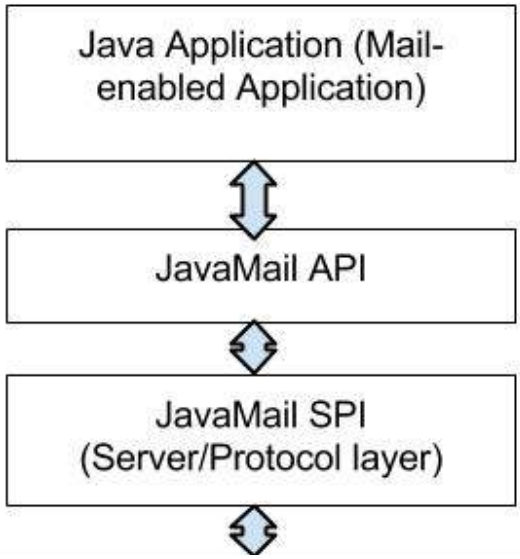
Following are some of the protocols supported in JavaMail API:

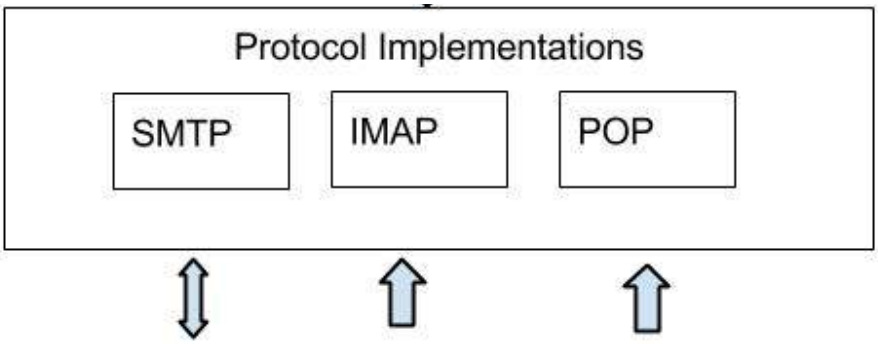
- **SMTP**: Acronym for **Simple Mail Transfer Protocol**. It provides a mechanism to deliver email.
- **POP**: Acronym for **Post Office Protocol**. POP is the mechanism most people on the Internet use to get their mail. It defines support for a single mailbox for each user. RFC 1939 defines this protocol.
- **IMAP**: Acronym for **Internet Message Access Protocol**. It is an advanced protocol for receiving messages. It provides support for multiple mailbox for each user, in addition to, mailbox can be shared by multiple users. It is defined in RFC 2060.
- **MIME**: Acronym for **Multipurpose Internet Mail Extensions**. . It is not a mail transfer protocol. Instead, it defines the content of what is transferred: the format of the messages, attachments, and so on. There are many different documents that take effect here: RFC 822, RFC 2045, RFC 2046, and RFC 2047. As a user of the JavaMail API, you usually don't need to worry about these formats. However, these formats do exist and are used by your programs.
- **NNTP and Others**:There are many protocols that are provided by third-party providers. Some of them are Network News Transfer Protocol *NNTP*, Secure Multipurpose Internet Mail Extensions *S/MIME* etc.

Details of these will be covered in the subsequent chapters.

Architecture

As said above the java application uses JavaMail API to compose, send and receive emails.The following figure illustrates the architecture of JavaMail:





The abstract mechanism of JavaMail API is similar to other J2EE APIs, such as JDBC, JNDI, and JMS. As seen the architecture diagram above, JavaMail API is divided into two main parts:

- An application-independent part: An application-programming interface *API* is used by the application components to send and receive mail messages, independent of the underlying provider or protocol used.
- A service-dependent part: A service provider interface *SPI* speaks the protocol-specific languages, such as SMTP, POP, IMAP, and Network News Transfer Protocol *NNTP*. It is used to plug in a provider of an e-mail service to the J2EE platform.