



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Platform SDK Developer's Guide

E-Mail Server

12/21/2025

E-Mail Server

The `EspEmailProtocol` allows communication to be established between a client application and an ESP-based E-mail Server.

Tip

Note that Platform SDK also provides the deprecated `EmailProtocol` class along with `EspEmailProtocol`. The `EmailProtocol` cannot be used with E-mail Servers of release 8.0.2 and higher. For newer e-mail servers, use `EspEmailProtocol`.

E-mail Server interfaces with the enterprise mail server and the Genesys Web API Server, bringing in new e-mail interactions and sending out replies or other outbound messages. For better understanding of e-mail server, see the [eServices documentation](#).

The `EspEmailProtocol` can be used in a web-based application, for example to submit new e-mail interaction that user filled out via the web form. You can find advanced samples with `EspEmailProtocol` in the [Web API Client Developer's Guide](#).

The sample below shows how to connect to E-mail Server and submit a new e-mail interaction.

Java

Open Connection to Server

Before using the Web Media Platform SDK, you will need to import the appropriate packages. Since we will be using the `EmailServer` protocol, we will use the following import statements:

```
import com.genesyslab.platform.commons.protocol.Endpoint;
import com.genesyslab.platform.commons.protocol.Message;
import com.genesyslab.platform.webmedia.protocol.EspEmailProtocol;
import com.genesyslab.platform.webmedia.protocol.espemail.EmailAttachment;
import com.genesyslab.platform.webmedia.protocol.espemail.EmailAttachmentList;
import com.genesyslab.platform.webmedia.protocol.espemail.events.EventCreateWebEmailIn;
import com.genesyslab.platform.webmedia.protocol.espemail.events.EventError;
import com.genesyslab.platform.webmedia.protocol.espemail.requests.RequestCreateWebEmailIn;
```

Now create a `Protocol` object and open a connection to the server.

```
EspEmailProtocol emailProtocol = new EspEmailProtocol(new Endpoint(hostname, port));
emailProtocol.open();
```

Submit a New E-mail Interaction

Create a `RequestCreateWebEmailIn` message to provide the e-mail message content and submit a new e-mail interaction.

```
RequestCreateWebEmailIn req = RequestCreateWebEmailIn.create();
req.setFirstName(firstName);
req.setLastName(lastName);
req.setFromAddress(emailAddress);
req.setSubject(subject);
req.setText(body);
req.setMailbox(replayFrom);
```

It is possible to attach data to an e-mail using `EmailAttachmentList` and `EmailAttachment` objects.

```
EmailAttachmentList attachmentList= new EmailAttachmentList();
EmailAttachment attachment = new EmailAttachment();
attachment.setContentType("image/png");
attachment.setFileName("picture.png");
attachment.setContent(content);
attachmentList.add(attachment);

req.setAttachments(attachmentList);
```

Send your request to the E-mail Server. If the operation is completed successfully, the server will respond with an `EventCreateWebEmailIn` message. This message contains the ID of the new interaction.

```
Message response = emailProtocol.request(req);
if (response instanceof EventCreateWebEmailIn) {
    EventCreateWebEmailIn eventAck = (EventCreateWebEmailIn)response;
    interactionId = eventAck.getNewInteractionId();
}
else if (response instanceof EventError) {
    EventError eventError = (EventError) response;
    //handle error
}
```

Disconnecting from E-mail Server

Finally, when you are finished communicating with the E-mail Server, you should close the connection to minimize resource utilization.

```
emailProtocol.close();
```

.NET

Open Connection to Server

Before using the Web Media Platform SDK, you should include using statements that allow access to types from the Platform SDK Commons and Web Media namespaces. For the EspEmail protocol, we use the following statements:

```
using Genesyslab.Platform.Commons.Collections;
using Genesyslab.Platform.Commons.Connection;
using Genesyslab.Platform.Commons.Protocols;
using Genesyslab.Platform.WebMedia.Protocols;
using Genesyslab.Platform.WebMedia.Protocols.EspEmail;
using Genesyslab.Platform.WebMedia.Protocols.EspEmail.Requests;
using Genesyslab.Platform.WebMedia.Protocols.EspEmail.Events;
```

Now create a Protocol object and open a connection to the server.

```
EspEmailProtocol espProtocol = new EspEmailProtocol(new Endpoint(hostname, port));
emailProtocol.Open();
```

Submit a New E-mail Interaction

Create a RequestCreateWebEmailIn message to provide the e-mail message content and submit a new e-mail interaction.

```
RequestCreateWebEmailIn req = RequestCreateWebEmailIn.Create();
req.FirstName = firstName;
req.LastName = lastName;
req.FromAddress = emailAddress;
req.Subject = subject;
req.Text = body;
req.Mailbox = replayFrom;
```

It is possible to attach data to an e-mail using EmailAttachmentList and EmailAttachment objects.

```
EmailAttachmentList attachmentList= new EmailAttachmentList();
EmailAttachment attachment = new EmailAttachment();
attachment.ContentType = "image/png";
attachment.FileName = fileName;
attachment.Content = File.ReadAllBytes(file);
attachmentList.Add(attachment);
req.Attachments = attachmentList;
```

Send your request to the E-mail Server. If the operation is completed successfully, the server will respond with an EventCreateWebEmailIn message. This message contains the ID of the new interaction.

```
IMessage response = emailProtocol.Request(req);

if (response is EventCreateWebEmailIn)
{
    EventCreateWebEmailIn eventAck = (EventCreateWebEmailIn)response;
    intearctionId = eventAck.NewInteractionId;
}
else if (response is EventError)
```

```
{  
    EventError eventError = (EventError) response;  
    //handle error  
}
```

Disconnecting from E-mail Server

Finally, when you are finished communicating with the E-mail Server, you should close the connection to minimize resource utilization.

```
emailProtocol.Close();
```