

glogin

glogin Component

Available in Camel 2.3

The `glogin` component is used by Camel applications outside Google App Engine (GAE) for programmatic login to GAE applications. It is part of the [Camel Components for Google App Engine](#). [Security-enabled GAE applications](#) normally redirect the user to a login page. After submitting username and password for authentication, the user is redirected back to the application. That works fine for applications where the client is a browser. For all other applications, the login process must be done programmatically. All the [necessary steps](#) for programmatic login are implemented by the `glogin` component. These are

1. Get an authentication token from [Google Accounts](#) via the [ClientLogin API](#).
2. Get an authorization cookie from Google App Engine's login API.

The authorization cookie must then be send with subsequent HTTP requests to the GAE application. It expires after 24 hours and must then be renewed.

URI format

```
glogin://hostname[:port][?options]
```

The `hostname` is either the internet hostname of a GAE application (e.g. `camelcloud.appspot.com`) or the name of the host where the [development server](#) is running (e.g. `localhost`). The `port` is only used when connecting to a development server (i.e. when `devMode=true`, see [options](#)) and defaults to 8080.

Options

Name	Default Value	Required	Description
<code>clientName</code>	<code>apache-camel-2.x</code>	false	A client name with recommended (but not required) format <code><organization>-<appname>-<version></code> .
<code>userName</code>	<code>null</code>	true (can alternatively be set via <code>GLoginBinding.GLOGIN_USER_NAME</code> message header)	Login username (an email address).
<code>password</code>	<code>null</code>	true (can alternatively be set via <code>GLoginBinding.GLOGIN_PASSWORD</code> message header)	Login password.
<code>devMode</code>	<code>false</code>	false	If set to <code>true</code> a login to a development server is attempted.
<code>devAdmin</code>	<code>false</code>	false	If set to <code>true</code> a login to a development server in admin role is attempted.

Message headers

Name	Type	Message	Description
<code>GLoginBinding.GLOGIN_HOST_NAME</code>	String	in	Overrides the hostname defined in the endpoint URI.
<code>GLoginBinding.GLOGIN_USER_NAME</code>	String	in	Overrides the <code>userName</code> option.
<code>GLoginBinding.GLOGIN_PASSWORD</code>	String	in	Overrides the <code>password</code> option.
<code>GLoginBinding.GLOGIN_TOKEN</code>	String	out	Contains the authentication token obtained from Google Accounts . Login to a development server does not set this header.
<code>GLoginBinding.GLOGIN_COOKIE</code>	String	out	Contains the application-specific authorization cookie obtained from Google App Engine (or a development server).

Message body

The `glogin` component doesn't read or write message bodies.

Usage

The following JUnit test show an example how to login to a development server as well as to a deployed GAE application located at <http://camelcloud.appspot.com>.

GLoginTest.java

```
import org.apache.camel.Exchange;
import org.apache.camel.Processor;
import org.apache.camel.ProducerTemplate;
import org.junit.Ignore;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import static org.apache.camel.component.gae.login.GLoginBinding.*;
import static org.junit.Assert.*;

public class GLoginTest {

    private ProducerTemplate template = ...

    @Test
    public void testDevLogin() {
        Exchange result = template.request("glogin://localhost:8888?userName=test@example.org&devMode=true",
null);
        assertNotNull(result.getOut().getHeader(GLOGIN_COOKIE));
    }

    @Test
    public void testRemoteLogin() {
        Exchange result = template.request("glogin://camelcloud.appspot.com", new Processor() {
            public void process(Exchange exchange) throws Exception {
                exchange.getIn().setHeader(GLOGIN_USER_NAME, "replaceme@gmail.com");
                exchange.getIn().setHeader(GLOGIN_PASSWORD, "replaceme");
            }
        });
        assertNotNull(result.getOut().getHeader(GLOGIN_COOKIE));
    }
}
```

The resulting authorization cookie from login to a development server looks like

```
ahlogincookie=test@example.org:false:11223191102230730701;Path=/
```

The resulting authorization cookie from login to a deployed GAE application looks (shortened) like

```
ACSID=AJKiYcE...XxhH9P_jR_V3; expires=Sun, 07-Feb-2010 15:14:51 GMT; path=/
```

Dependencies

Maven users will need to add the following dependency to their `pom.xml`.

pom.xml

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-gae</artifactId>
  <version>x.x.x</version>
</dependency>
```

See Also

- [Configuring Camel](#)
- [Component](#)
- [Endpoint](#)
- [Getting Started](#)