

Reusing Cookies in HTTP Consumers

Last Modified on 07/02/2020 11:21 am EDT

The following configuration shows how to share a session and reuse the session cookie container between two activities.

In this scenario, Cora SeQuence sends two calls to a service. The first call gets an authentication token and the second call performs the action.

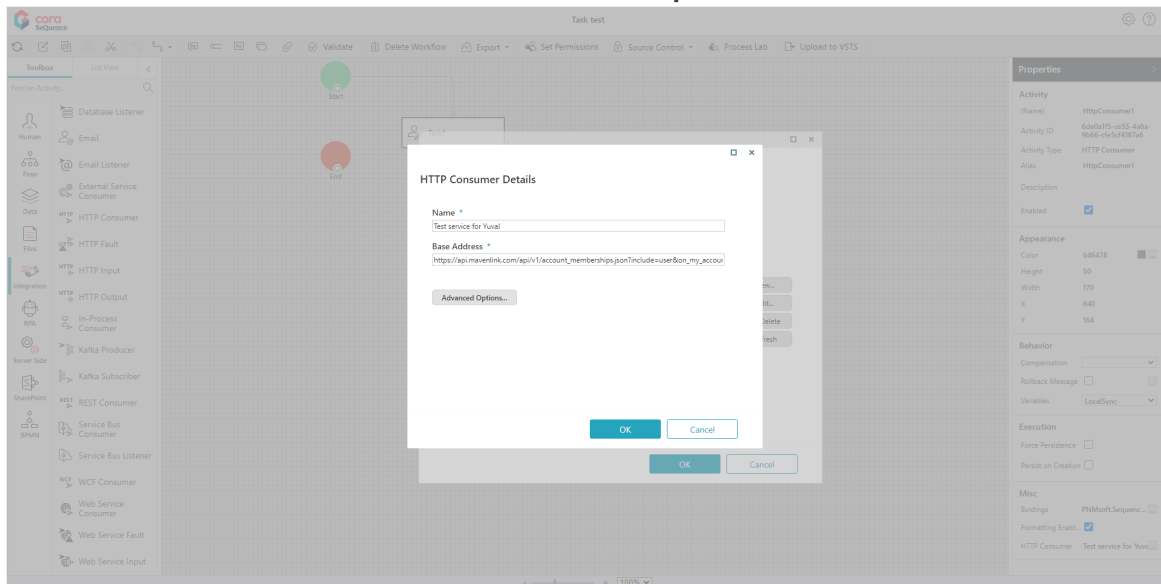
You need to set up one HTTP Consumer that captures the returned token (cookie) and then pushes it to a second HTTP Consumer, which sends additional calls to get the data.

NOTE

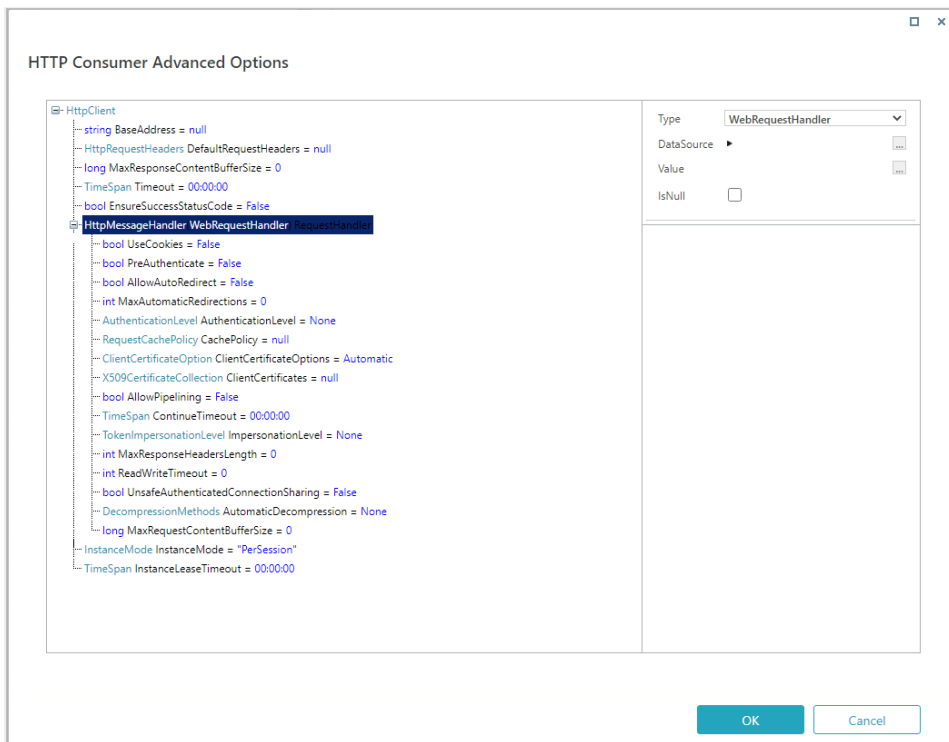
The following procedures show only the settings relevant for reusing cookies between activities. For more details on setting up HTTP Service activities, see [this article](#).

Setting up the HTTP Consumer that captures the token

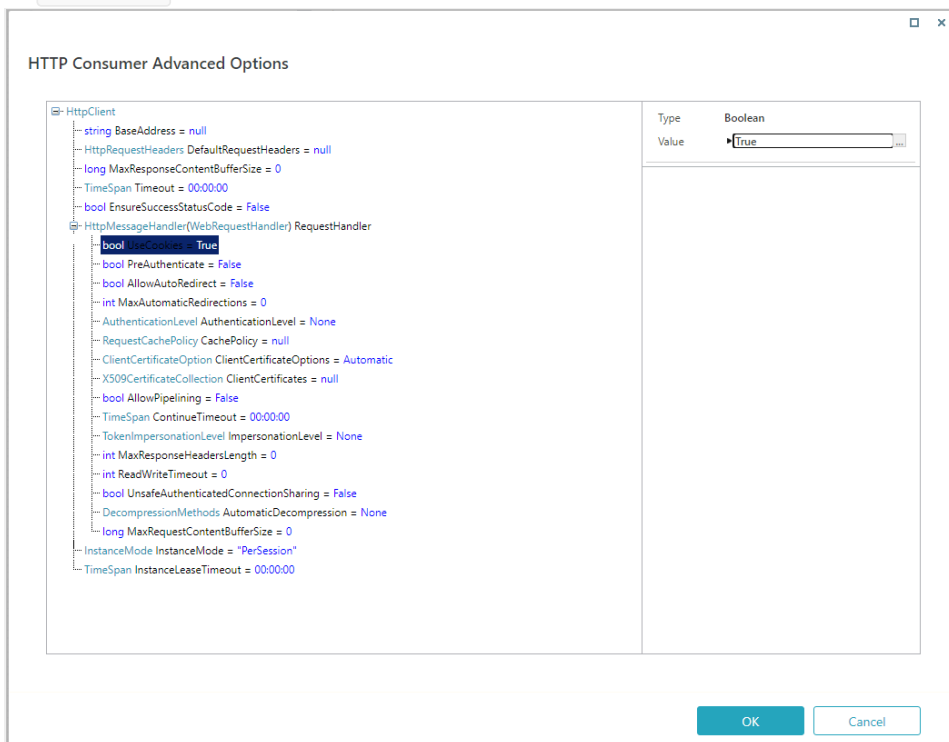
1. On the HTTP Consumer Details screen, click **Advanced Options**.



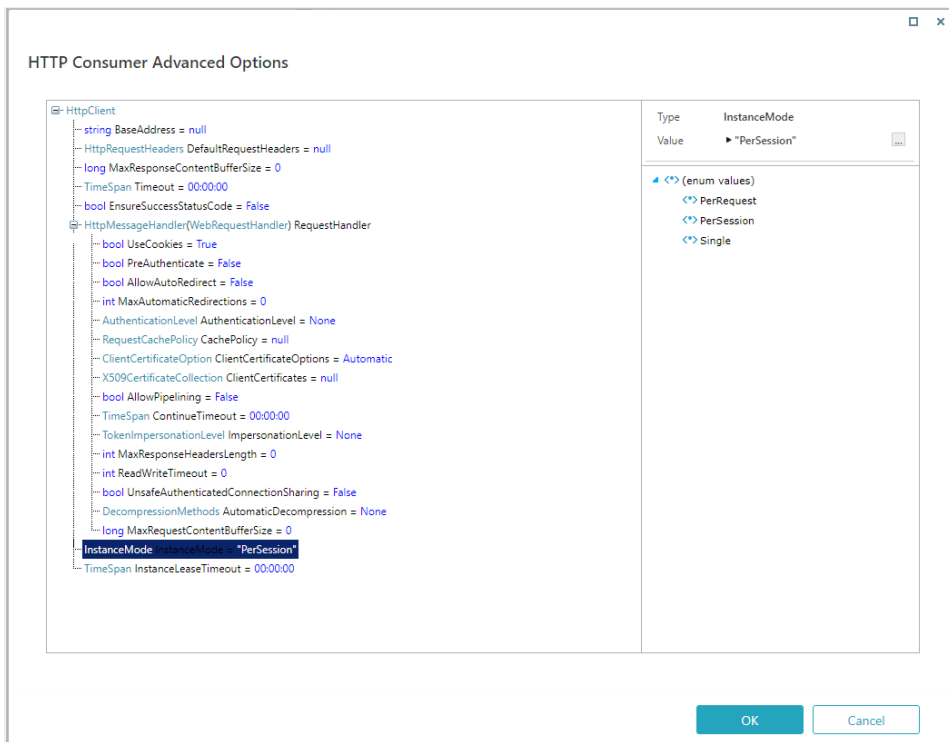
2. Set the type of the `HttpMessageHandler` node to **webRequestHandler**.



3. Set `UseCookies` to **True**.

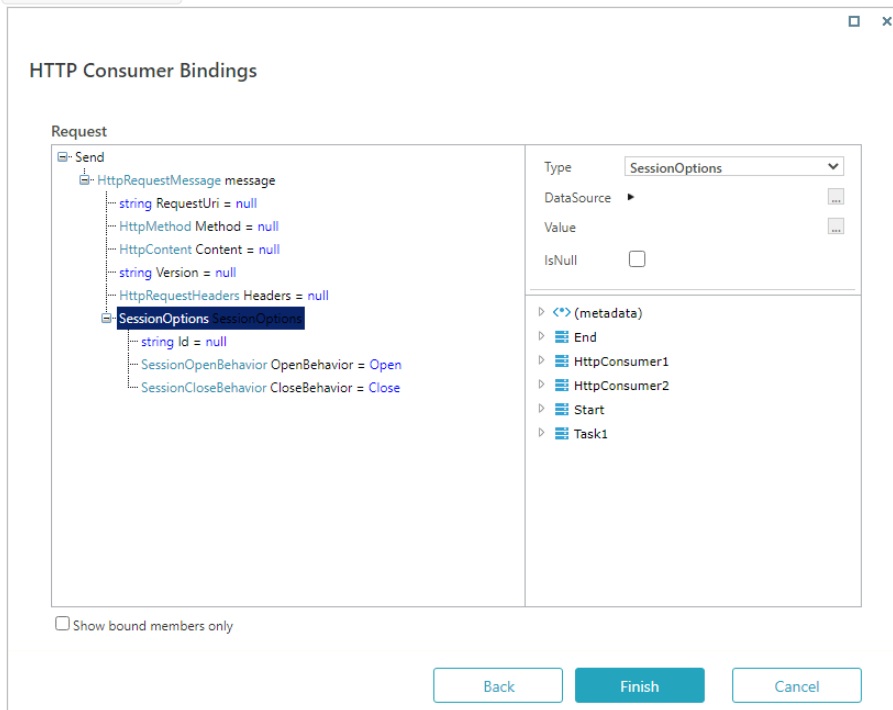


4. Set the value of `InstanceMode` to **"PerSession"**.



5. In the HTTP Consumer Bindings screen, clear the **IsNull** checkbox for the following nodes:

- a. `HttpRequestMessage`
- b. `SessionOptions`



6. Depending on the persistence mode, configure one of the following options for the session `Id` :

- o Default mode: Set it to the workflow instance ID as string
- o InMemory mode: Set it to a variable with a predefined random value. For example,

```
Guid.NewGuid().ToString("N")
```

7. Change `SessionCloseBehavior` to **"KeepOpen"**.

HTTP Consumer Bindings

Request

```

    Send
    - HttpRequestMessage message
      - string RequestUri = null
      - HttpMethod Method = null
      - HttpContent Content = null
      - string Version = null
      - HttpRequestHeaders Headers = null
      - SessionOptions SessionOptions
        - string Id = wf.WorkflowInstanceId.ToString()
        - SessionOpenBehavior OpenBehavior = Open
        - SessionCloseBehavior CloseBehavior = "KeepOpen"
  
```

| Type | SessionCloseBehavior |
|--|----------------------|
| Value | ► "KeepOpen" |
| ◀ ◻ (enum values) <ul style="list-style-type: none"> ◻ Close ◻ KeepOpen | |
| ◀ ◻ (metadata) <ul style="list-style-type: none"> ◻ CompletedAt ◻ CreatedAt ▶ ◻ CreatedBy ◻ DisplayField ◻ MasterWorkflowInstanceId ▶ ◻ OwnedBy ◻ SourceActivityInstanceId ◻ SourceWorkflowInstanceId ◻ Stage ▶ ◻ Statistics ◻ UpdatedAt ▶ ◻ UpdatedBy | |

Show bound members only

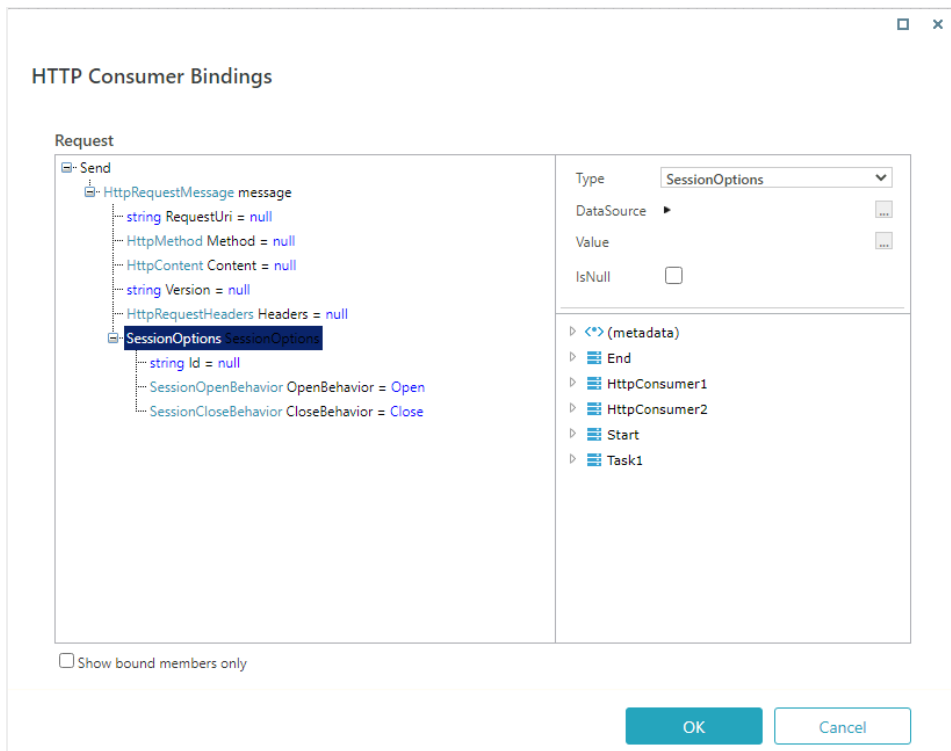
Back Finish Cancel

Setting up the HTTP Consumer that accesses the data

1. On the HTTP Consumer Bindings screen, select the clear the `HttpRequestMessage` node, and clear the **IsNull** checkbox.

The screenshot shows the 'HTTP Consumer Bindings' dialog box. In the 'Request' section, the 'HttpRequestMessage' node is selected. The 'Type' dropdown is set to 'HttpRequestMessage'. The 'Value' field is empty, and the 'IsNull' checkbox is checked. The 'Properties' panel on the right shows details for 'HTTP Consumer3', including its name, ID, type, and description. The 'Enabled' checkbox is checked, and the 'Variables' dropdown is set to 'LocalSync'.

2. Select the `SessionOptions` node, and clear the **IsNull** checkbox.



3. Set the SessionOptions properties, as follows:
 - a. Session `Id`: workflow instance ID as string
 - b. `SessionOpenBehavior`: **"Reuse"**
 - c. `SessionCloseBehavior`: **"Close"**

NOTE

The session ID must match the session ID of the first HTTP Consumer.