

# executeRemoteAction()

Use this function to execute a [JWT remote action](#).

### Syntax

```
executeRemoteAction(actionName) #Output: Text
```

### Examples

Parser expression	Description
<pre>executeRemoteAction("Great action")</pre>	This example executes the action called "Great action" and returns the whole JSON response.

### Additional information

#### Parameters used in this function

Parameter	Input (data type)	Description
actionName	<input type="text" value="TEXT"/>	Name of the action to execute (case sensitive)

### Output

The function returns a  representing the following JSON:

```
{
  "status": 200 // Returned status code from the executed action
  "body": any // Response body that was returned from the executed action
}
```

Variant where you can specify a **JMESPath**.

### Syntax


```
executeRemoteAction(actionName, JMESPath) #Output: Text
```

### Examples

Parser expression	Description
<code>executeRemoteAction("Great action", "body")</code>	This example executes the action called "Great action" and returns only the body part of the JSON response.
<code>executeRemoteAction("Great action", "body.data")</code>	This example executes the action called "Great action" and returns only the data part of the JSON response's body object.
<code>executeRemoteAction("Great action", "status")</code>	This example executes the action called "Great action" and returns only the status code of the JSON response.

## Additional information

### Parameters used in this function

Parameter	Input (data type)	Description
actionName	<input type="text" value="TEXT"/>	Name of the action to execute (case sensitive)
JMESPath	<input type="text" value="TEXT"/>	<p>This parameter holds the <a href="#">JMESPath</a> which is applied to the returned json object .</p> <p> The text is not checked for syntactical correctness when defining it in the <a href="#">parser expression editor</a>, i.e. if it does not hold a <a href="#">JMESPath</a>, this is only detected during runtime.</p>

## Output

The function returns a  representing the following JSON (depending on the provided JMESPath):

```
{
  "status": 200 // Returned status code from the executed action
  "body": any // Response body that was returned from the executed action
}
```

Variant where you can specify additional **parameters**.

Syntax
<code>executeRemoteAction(actionName, parameterList) #Output: Text</code>

## Examples

Parser expression	Description
-------------------	-------------

<pre>executeRemoteAction("Great action", ["query": "Demo", "path": "projects"])</pre>	<p>This example executes the action called "Great action" and returns the whole json response. In addition, the parameters "query" and "path" will be added to the request.</p>
<pre>executeRemoteAction("POST action", ["body": "{ \"userId\": 12, \"name\": \"John Doe\" }"])</pre>	<p>This example executes the action called "POST action" and returns the whole json response. In addition, the action's request body will be populated with the provided JSON.</p>

## Additional information

### Parameters used in this function

Parameter	Input (data type)	Description
actionName	TEXT	Name of the action to execute (case sensitive)
parameter List	PARAMETER LIST	<p>The parameter list's syntax is <b>["key": "value", "key2": "anotherValue", "key3": 12, ...]</b>.</p> <p>The <b>key</b> must always be a text and unique within the parameter list.</p> <p>The <b>value</b> can be of type text or number. It is also possible to enter there parser expressions that return either a text or a number.</p> <p>The parameter list can be used to overwrite parameters set in the action configuration like the following:</p> <ul style="list-style-type: none"> <li>• If a key/ value pair is defined in the list and there is a query param with that key, the value for that param will be overwritten by the paramter list's value</li> <li>• If a key/ value pair is defined in the list and there is a path param with that key, the value for that param will be overwritten by the parameter list's value</li> <li>• If a key/ value pair is defined in the list and there is neither a query or path param with that key, this pair will be added as additional query param to the request</li> <li>• If the key is "body" and there is neither a query or path param with that key, this parameter will be used to set the action body</li> </ul>

## Output

The function returns a  representing the following JSON:

```
{
  "status": 200 // Returned status code from the executed action
  "body": any // Response body that was returned from the executed action
}
```

Variant where you can specify additional **parameters** and another **connection**.

### Syntax

```
executeRemoteAction(actionName, parameterList, connectionName) #Output: Text
```

## Examples

Parser expression	Description
<pre>executeRemoteAction("Great action", [ "query": "Demo", "path": "projects"], "Some other connection")</pre>	This example executes the action called "Great action" and returns the whole json response. In addition, the parameters "query" and "path" will be added to the request and the connection called "Some other connection" is being used.

## Additional information

### Parameters used in this function

Parameter	Input (data type)	Description
actionName	<input type="text" value="TEXT"/>	Name of the action to execute (case sensitive)
parameter List	<input type="text" value="PARAMETER LIST"/>	<p>The parameter list's syntax is [<b>"key": "value"</b>, <b>"key2": "anotherValue"</b>, <b>"key3": 12</b>, ...].</p> <p>The <b>key</b> must always be a text and unique within the parameter list.</p> <p>The <b>value</b> can be of type text or number. It is also possible to enter there parser expressions that return either a text or a number.</p> <p>The parameter list can be used to overwrite parameters set in the action configuration like the following:</p> <ul style="list-style-type: none"><li>• If a key/ value pair is defined in the list and there is a query param with that key, the value for that param will be overwritten by the paramter list's value</li><li>• If a key/ value pair is defined in the list and there is a path param with that key, the value for that param will be overwritten by the parameter list's value</li><li>• If a key/ value pair is defined in the list and there is neither a query or path param with that key, this pair will be added as additional query param to the request</li><li>• If the key is "body" and there is neither a query or path param with that key, this parameter will be used to set the action body</li></ul>
connectionName	<input type="text" value="TEXT"/>	Name of the connection to use for the action (case sensitive)

## Output

The function returns a  representing the following JSON:

```
{  
  "status": 200 // Returned status code from the executed action  
  "body": any // Response body that was returned from the executed action  
}
```

Variant where you can specify a **JMESPath** and additional **parameters**.

## Syntax


```
executeRemoteAction(actionName, JMESPath, parameterList) #Output: Text
```

## Examples

Parser expression	Description
<pre>executeRemoteAction("Great action", "body.data", [{"query": "Demo", "path": "projects"}])</pre>	This example executes the action called "Great action" and returns only the data part of the response's body. In addition, the parameters "query" and "path" will be added to the request and the connection called "Some other connection" is being used.

## Additional information

### Parameters used in this function

Parameter	Input (data type)	Description
actionName	<input type="text" value="TEXT"/>	Name of the action to execute (case sensitive)
JMESPath	<input type="text" value="TEXT"/>	This parameter holds the <a href="#">JMESPath</a> which is applied to the returned json object .   The text is not checked for syntactical correctness when defining it in the <a href="#">parser expression editor</a> , i.e. if it does not hold a <a href="#">JMESPath</a> , this is only detected during runtime.
parameter List	<input type="text" value="PARAMETER LIST"/>	The parameter list's syntax is <b>[ "key": "value", "key2": "anotherValue", "key3": 12, ... ]</b> .  The <b>key</b> must always be a text and unique within the parameter list.  The <b>value</b> can be of type text or number. It is also possible to enter there parser expressions that return either a text or a number.  The parameter list can be used to overwrite parameters set in the action configuration like the following: <ul style="list-style-type: none"><li>• If a key/ value pair is defined in the list and there is a query param with that key, the value for that param will be overwritten by the paramter list's value</li><li>• If a key/ value pair is defined in the list and there is a path param with that key, the value for that param will be overwritten by the parameter list's value</li><li>• If a key/ value pair is defined in the list and there is neither a query or path param with that key, this pair will be added as additional query param to the request</li><li>• If the key is "body" and there is neither a query or path param with that key, this parameter will be used to set the action body</li></ul>

## Output

The function returns a  representing the following JSON (depending on the provided JMESPath):

```
{
  "status": 200 // Returned status code from the executed action
  "body": any // Response body that was returned from the executed action
}
```

Variant where you can specify a **JMESPath** and another **connection**.

#### Syntax


```
executeRemoteAction(actionName, JMESPath, connectionName) #Output: Text
```

## Examples

Parser expression	Description
<pre>executeRemoteAction("Great action", "body.data", "Different connection")</pre>	This example executes the action called "Great action" and returns the whole JSON response. In addition, the connection called "Different connection" is being used.

## Additional information

### Parameters used in this function

Parameter	Input (data type)	Description
actionName	<input type="text" value="TEXT"/>	Name of the action to execute (case sensitive)
JMESPath	<input type="text" value="TEXT"/>	This parameter holds the <a href="#">JMESPath</a> which is applied to the returned json object .   The text is not checked for syntactical correctness when defining it in the <a href="#">parser expression editor</a> , i.e. if it does not hold a <a href="#">JMESPath</a> , this is only detected during runtime.
connectionName	<input type="text" value="TEXT"/>	Name of the connection to use for the action (case sensitive)

## Output

The function returns a  representing the following JSON (depending on the provided JMESPath):

```
{  
  "status": 200 // Returned status code from the executed action  
  "body": any // Response body that was returned from the executed action  
}
```

Variant where you can specify a **JMESPath**, a **connection** and additional **parameters**.

#### Syntax


```
executeRemoteAction(actionName, JMESPath, connectionName, parameterList) #Output: Text
```

## Examples

Parser expression	Description
<pre>executeRemoteAction("Great action", "body.data", "cool connection", ["query": "Demo", "path": "projects"])</pre>	This example executes the action called "Great action" and returns only the data part of the response's body. In addition, the parameters "query" and "path" will be added to the request and the connection called "cool connection" is being used.

## Additional information

### Parameters used in this function

Parameter	Input (data type)	Description
actionName	<input type="text" value="TEXT"/>	Name of the action to execute (case sensitive)
JMESPath	<input type="text" value="TEXT"/>	This parameter holds the <a href="#">JMESPath</a> which is applied to the returned json object.   The text is not checked for syntactical correctness when defining it in the <a href="#">parser expression editor</a> , i.e. if it does not hold a <a href="#">JMESPath</a> , this is only detected during runtime.
connectionName	<input type="text" value="TEXT"/>	Name of the connection to use for the action (case sensitive)
parameterList	<input type="text" value="PARAMETER LIST"/>	The parameter list's syntax is <b>[<i>"key"</i>: <i>"value"</i>, <i>"key2"</i>: <i>"anotherValue"</i>, <i>"key3"</i>: 12, ...]</b> .  The <b>key</b> must always be a text and unique within the parameter list.  The <b>value</b> can be of type text or number. It is also possible to enter there parser expressions that return either a text or a number.  The parameter list can be used to overwrite parameters set in the action configuration like the following: <ul style="list-style-type: none"><li>• If a key/ value pair is defined in the list and there is a query param with that key, the value for that param will be overwritten by the parameter list's value</li><li>• If a key/ value pair is defined in the list and there is a path param with that key, the value for that param will be overwritten by the parameter list's value</li><li>• If a key/ value pair is defined in the list and there is neither a query or path param with that key, this pair will be added as additional query param to the request</li><li>• If the key is "body" and there is neither a query or path param with that key, this parameter will be used to set the action body</li></ul>



## Output

The function returns a  representing the following JSON (depending on the provided JMESPath):

```
{  
  "status": 200 // Returned status code from the executed action  
  "body": any // Response body that was returned from the executed action  
}
```



Use cases and examples

Use case	JWT feature	Workflow function	Field type	Automated action	Parser functions
Display reporters of linked Cloud issues			Text		<code>executeRemoteAction()</code> <code>findPattern()</code> <code>toStringList()</code> <code>toString()</code> <code>distinct()</code>
Sum of Tempo billable hours			Number		<code>executeRemoteAction()</code> <code>toNumberList()</code> <code>sum()</code> <code>toNumber()</code>