 **Global Actions**

Global Actions

New Action

Enter Action Information

Save

Cancel

Action Type

Target Object

Standard Label Type

Label

Name

Description

Create Feed Item

Success Message

Icon

Create a Record

Create a Record

Send Email

Log a Call

Custom Visualforce

Custom Canvas

Lightning Component

Change Icon

Save

Cancel

Account Actions

New Action

Enter Action Information

Save

Cancel

Object Name

Action Type

Target Object

Standard Label Type

Label

Name

Description

Create Feed Item

Success Message

Icon

Account

Create a Record

Create a Record

Send Email

Log a Call

Custom Visualforce

Update a Record

Lightning Component

Flow

Save

Cancel

Quick Actions

Returns a list of global actions and object-specific actions. This resource is available in REST API version 28.0 and later. When working with actions, also refer to [sObject Quick Actions](#).

URI

`/services/data/vXX.X/quickActions/`

Formats

JSON, XML

HTTP Method

HEAD, GET, POST

Authentication

Authorization: Bearer *token*


Parameters

None required


Considerations

Add all required fields to an object before you create a quick action for that object. If you add a required field after creating a quick action, the field won't appear in the quick action's describe results. Then, when the quick action runs, the field won't be available and an error occurs for the missing field. If you don't want the required field to appear in the quick action's layout, set a default value for the field.


Example usage for getting global quick actions

```
1 curl https://yourInstance.salesforce.com/services/data/v51.0/quickActions/ -H "Authorization: Bearer token" 
```

Example for creating a contact using an action

```
1 curl https://yourInstance.salesforce.com/services/data/v51.0/quickActions/CreateContact -H 'Authorization: Bearer access_token' -H 'Content-Type: application/json' -d @newcontact.json 
```

Example JSON request body `newcontact.json` file

```
1 { 
2
3   "record" : { "LastName" : "Smith" }
4
5 }
```

sObject Quick Actions

Returns a list of actions and their details. This resource is available in REST API version 28.0 and later. When working with actions, also refer to [Quick Actions](#).

URI

To return a specific object's actions as well as global actions, use: `/services/data/vXX.X/objects/sObject/quickActions/`

To return a specific action, use `/services/data/vXX.X/objects/sObject/quickActions/action name`

To return a specific action's descriptive detail, use `/services/data/vXX.X/objects/sObject/quickActions/action name/describe/`

To return a specific action's default values, including default field values, use `/services/data/vXX.X/objects/sObject/quickActions/action name/defaultValues/`

In API version 28.0, to evaluate the default values for an action, use `/services/data/vXX.X/objects/sObject/quickActions/action name/defaultValues/parent id`

In API version 29.0 and greater, to evaluate the default values for an action, use `/services/data/vXX.X/objects/sObject/quickActions/action name/defaultValues/context id`
This returns the default values specific to the `context id` object.

Formats

JSON, XML

HTTP Method

HEAD, GET, POST

Authentication

Authorization: Bearer token

Parameters

None required

Example for getting Account actions

```
1 curl https://yourInstance.salesforce.com/services/data/v51.0/objects/Account/quickActions -H 'Authorization: Bearer token'
```

Example for creating a contact on Account using an action

```
1 curl https://yourInstance.salesforce.com/services/data/v51.0/objects/Account/quickActions/CreateContact -H 'Authorization: Bearer access_token' -H 'Content-Type: application/json' -d @newcontact.json'
```

Example JSON request body newcontact.json file

```
1 {
2
3   "contextId" : "001D000000JRSgf",
4   "record" : { "LastName" : "Smith" }
5
6 }
```

Considerations

- The resources return all actions—both global and standard—in addition to the ones requested.

Quick Action API

This component allows you to access methods for programmatically controlling actions on record pages in Lightning Experience. This component requires API version 43.0 and later.

Documentation Specification

A `lightning:quickActionAPI` component allows you to access methods for programmatically controlling quick actions on record pages. This component is supported in Lightning Experience only.

For example, if you have a custom component that displays Knowledge articles, you can use the `lightning:quickActionAPI` component to attach and send a Knowledge article from your custom component via the Email quick action on the case record page.

To access the methods, create an instance of the `lightning:quickActionAPI` component inside your Lightning component or page and assign an `aura:id` attribute to it.

```
<lightning:quickActionAPI aura:id="quickActionAPI"/>
```

This example creates two buttons that interact with the Update Case quick action on a case record page in Lightning Experience.

```
<aura:component implements="flexipage:availableForRecordHome" description="My Lightning  
  <lightning:quickActionAPI aura:id="quickActionAPI" />  
  <div>  
    <lightning:button label="Select Update Case Action" onclick="{!c.selectUpdateCaseAction}" />  
    <lightning:button label="Update Case Status Field" onclick="{!c.updateCaseStatusField}" />  
  </div>  
</aura:component>
```

The buttons call the following client-side controller.

```
{
  selectUpdateCaseAction : function( cmp, event, helper ) {
    var actionAPI = cmp.find("quickActionAPI");
    var args = {actionName: "Case.UpdateCase"};
    actionAPI.selectAction(args).then(function(result){
      //Action selected; show data and set field values
    }).catch(function(e){
      if(e.errors){
        //If the specified action isn't found on the page, show an error message
      }
    });
  },

  updateCaseStatusAction : function( cmp, event, helper ) {
    var actionAPI = cmp.find("quickActionAPI");
    var fields = {Status: {value: "Closed"}, Subject: {value: "Sets by lightning"};
    var args = {actionName: "Case.UpdateCase", entityName: "Case", targetFields: fields};
    actionAPI.setActionFieldValues(args).then(function(){
      actionAPI.invokeAction(args);
    }).catch(function(e){
      console.error(e.errors);
    });
  },
}
```