Recast

Liquit.Applications.List

Last Modified on 2026-09-24

The *Liquit.Applications.List* function returns all applications that are available to the user. Whenever an application has been made available in the Application Workspace it will be listed in the response of this function. The Application Workspace API autodetects whether or not the Agent is available and adjusts its response according to it. For example, a local application will not be enabled when the Agent is not running.

Request parameters

Name	Description	Value
sort	Determines the order in which the API returns the applications: • 0 User-defined ordering, as determined by their position • 1 Name ascending • 2 Name descending • 3 By launch count (available for Application Workspace 3.4 and later) • 4 By last launch (available for Application Workspace 3.4 and later)	<int>
refresh	Defines if the action is a refresh.	```
search	Allows you to search by name	<string>
tags	If its value is true, tags will be included	<boolean>
teams	If its value is true, teams will be included	<boolean>
event	Define the type of event: • **None** no action is executed. • **Refresh** triggers actions defined in refresh events. • **LiquitLogon** triggers the actions that would be executed at Application Workspace logon.	<string>

Return parameters

The following properties are returned:

Name	Description	Value
autoLaunch	Whether or not auto launch is enabled for the application	<boolean>
categories	Returns an array of categories in which the application is included	<array[]>
editable	Determines whether or not the end-user is privileged to edit the package	<boolean>
enabled	Determines whether or not the application is enabled (e.g. local applications will not be enabled when no Agent is running.)	<boolean>
favorite	Will return true if the end-user added this application to their favorites	<boolean>
icon	Returns the URL of the application icon	<URL>
id	The id of the application	<guid>
name	The friendly name of the application	<string>
new	Will return true if the end-user has never started this application	<boolean>
position	The position of the application	<int>

Name	Description	Value
rating	This will return the average rating of the app calculated based on the rating given by all users	<int>
removable	Whether or not the end-user is allowed to remove the application from their workspace (e.g. forced applications will return false)	<boolean>
repair	Whether or not the repair action is available for the application	<boolean>
source	Returns an object with information about the applications: • **id** the ID of the source which can be either the ID of the zone or of a team • **type** the source type, which can be either "Zone" or "Team"	<object{}>
stage	The published stage: • Development • Test • Acceptance • Production	<string>
status	Returns the state of the application.	<string>

Example

Below you will find a small example of code to render the applications and their respective icons.

```
// Do Liquit API calls...
Liquit.Applications.List({

  sort:0

},function (fault, result) {

  // Check if the request failed.
  if (fault != null) {
    alert('Error ' + fault.code + ': ' + fault.message);
    return;
  }

  for (var i = 0; i < result.length; i++) {

    $('body').append(
      $('<div>')
        .attr({
          id: result[i].id
        })
        .addClass('app')
        .toggleClass('disabled', !result[i].enabled)
        .on('click', function (e) {

          // Get target html element.
          var target = $(e.currentTarget);

          // Verify application is disabled
          if (target.hasClass('disabled')) {
            alert('Application is disabled');
            return;
          }

          // Launch application by ID with callback returning progress information.
```

```
          // Launch application by ID with callback returning progress information.
          Liquit.Applications.Launch(target.attr('id'), null, function (e) {

            console.log(e);

            // *** e.state ***
            // Can be one of the following values:
            //
            // * 'success': The application has been successfully launched.
            // * 'inprogress': The application is being installed. Check the progress property to see how far along the install
ation is.
            // * 'failure': The application has failed to start. See the details property for more information about the error.
            //
            // *** e.progress ***
            // Gives an estimated percentage of the installation progress.
            //
            // *** e.details ***
            // Gives additional information about the failure to launch the application.
            // This is structured with a 'code' field and a 'message' field describing the error.
            //

            // Application started
            if (e.state == 'success')
              alert('application started');

            // Failed to start application.
            else if (e.state == 'failure')
              alert('application failed to start: ' + e.details.code + ': ' + e.details.message);

          });

        })
        .append(
          $('<img>')
            .addClass('icon')
            .attr({
              src: result[i].icon
            })
            .toggle(result[i].icon != null),
          $('<span>')
            .addClass('title')
            .text(result[i].name)
        )
      );

    }

});
```