

KRYON™

BE YOUR FUTURE

User Guide

Kryon Web Service API v5.25.1

This document contains Kryon Systems proprietary information. The information contained herein is confidential and cannot be distributed without the prior written approval of Kryon Systems Ltd.

© 2008-2018 Kryon Systems Ltd.
All rights reserved.

Document revision: 09-Oct-2018

Contents

CHAPTER 1: Introduction	3
CHAPTER 2: Architecture	4
CHAPTER 3: Authentication	5
CHAPTER 4: API Reference	6
Add Task	6
Add Task: WebRequest Parameters	7
Add Task: WebResponse Parameters	8
Get Status	10
Get Status: WebResponse Parameters	11
CHAPTER 5: Task Queue	14
CHAPTER 6: Testing	15

CHAPTER 1: Introduction

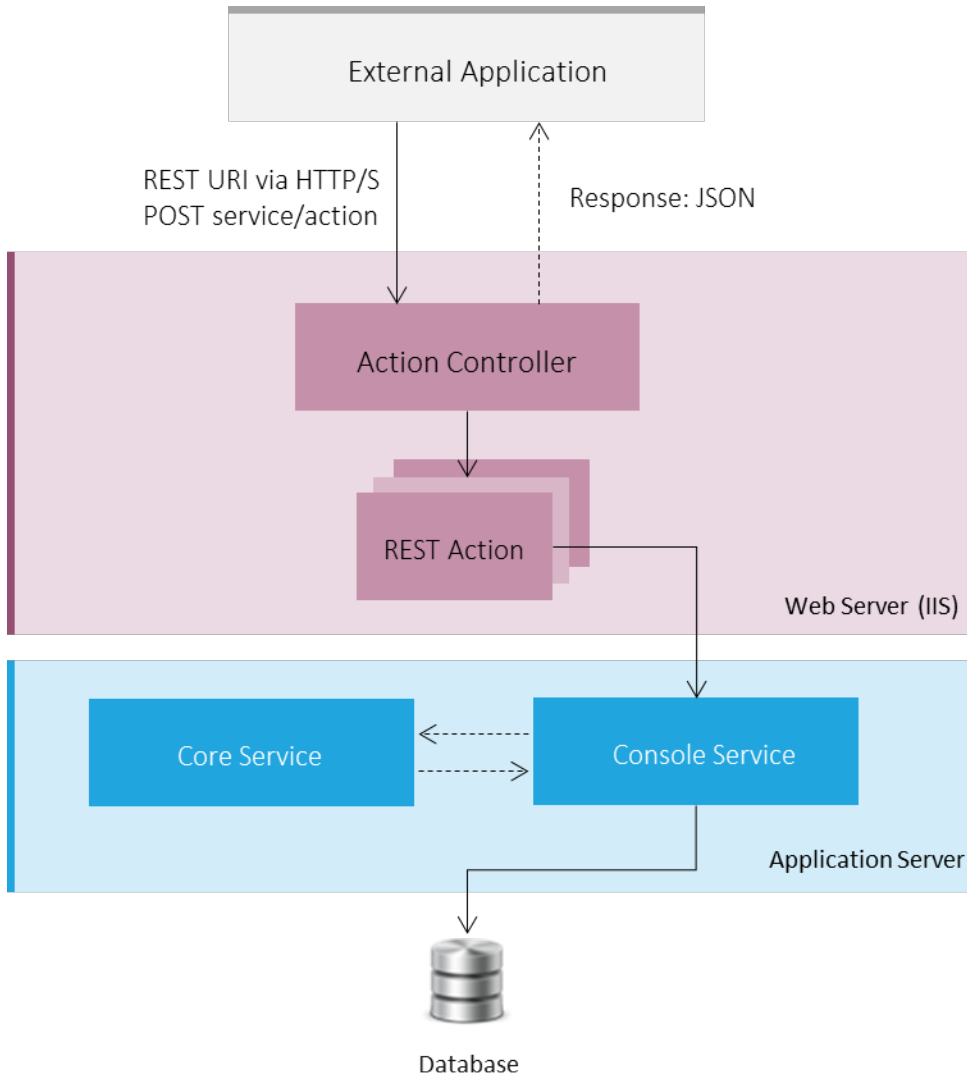
The Kryon Web Service API allows external applications to invoke/monitor the RPA tasks of the Kryon solution.

It utilizes the ASP.NET MVC (Web API) model with 2 methods:

- POST - creates a new task in the task queue
- GET - retrieves task status

All incoming requests are processed by the Kryon Application Server and responses are sent back to the calling application. If a process fails, an error response is sent to the external application along with error details.

CHAPTER 2: Architecture



CHAPTER 3: Authentication

Each web request requires 2 custom entries in a header section:

- username - Kryon username
- pwd - Kryon password

Authentication is performed on the Kryon Application Server for every API call, using the provided Kryon account credentials (username and password).

In case of authentication failure, the appropriate response, along with the error code is sent to the calling application.

Code	Meaning
0	No Error
1	User Not Found
2	User Inactive
3	Server Error
4	License Error
10	Unknown
11	User Locked
12	Password Change Required
13	Password Expired

CHAPTER 4: API Reference

Add Task

POST ConsoleServerAddress:API_port_number/*task/add*

- ConsoleServerAddress = IP Address or Name of Console Server (as defined when installing Kryon Application Server)
- API_port_number = Port number (i.e., binding) of the API website (as defined in IIS)
 - Example: SERVERNAME.COMPANYDOMAIN.LOCAL:8080/task/add

API controller name	Task
Method name	Add
Input	WebRequest (json)
Output	WebResponse (json)

Add Task: WebRequest Parameters

WizardCustomId	String	ID of the wizard to execute
NumberOfRuns	Long	Number of wizard runs per task; default is 1
SingleRunEstimation	Long	Time estimate for a single wizard run in minutes <i>(This is used only for display in Kryon Console and does not affect the wizard run in any way.)</i>
Variables	Array (String, String)	List of initial variable names and values to be populated into the wizard
QueuePriority	Integer	Priority in task queue: <ul style="list-style-type: none"> • 0 - normal (default) • 1 - high
GroupName	String	Robot group to which to assign the task; default is empty (any available robot)
MachineName	String	Robot name to which to assign the task; default is empty (any available robot)

Add Task: WebResponse Parameters

```
{  
  TaskId: <long>,  
  Status: <int>,  
  Error: <int>,  
  OutputData: <string>  
}
```

TaskId	Long	ID of the added task
Status	Integer	<i>(N/A – will be empty)</i>
Error	Integer	<i>(N/A – will be empty)</i>
OutputData	String	<i>(N/A – will be empty)</i>

Example

```
var request = {
  WizardCustomid: 'A_123',
  NumberOfRuns: 1,
  SingleRunEstimation: 20,
  Variables: [{Name: 'var1', Value: 'one'},
              {Name: 'var2', Value: 'two'}]};
$.ajax({
  url: '/task/add',
  data: JSON.stringify(request),
  type: 'post',
  crossDomain: true,
  contentType: 'application/json; charset=utf-8',
  beforeSend: function(request) {
    request.setRequestHeader("username",
    'username'),
    request.setRequestHeader("pwd",
    'password');
  }
})
.done(function(resp) {
  return resp;
})
.fail(function(error) {
  throw new Error("Error getting the
  data");
});
```

Get Status

GET ConsoleServerAddress:API_port_number/task/status?tid=taskid

- ConsoleServerAddress = IP Address or Name of Console Server (as defined when installing Kryon Application Server)
- API_port_number = Port number (i.e., binding) of the API website (as defined in IIS)
 - Example: SERVERNAME.COMPANYDOMAIN.LOCAL:8080/task/status?tid=taskid

API controller name	Task
Method name	GetStatus
Parameter	TaskId (long)
Output	WebResponse (json)

Get Status: WebResponse Parameters

```
{
  TaskId: <long>,
  Status: <int>,
  Error: <int>,
  OutputData: <string>
}
```

TaskId	Long	ID of the queried task
Status	Integer	Task status (<i>see table below</i>)
Error	Integer	Error code (<i>see table below</i>)
OutputData	String	<p>Output data reported by wizard</p> <p>In order for this output parameter to be returned, the relevant wizard must utilize the Report Wizard Output advanced command.</p> <ul style="list-style-type: none"> For additional details, see the document: <i>Advanced Commands Reference Guide (Report Wizard Output)</i>

Status Code	Meaning
0	Started
1	Stopped
2	Ended
3	Delayed
4	Inactive
5	Skipped
6	Queued
7	Faulty (<i>see table below</i>)

Error Code	Meaning
0	OK (no error)
1	General
2	Login
3	Expired
4	Failed To Create Task
5	Wizard Not Found
6	Failed To Get Status
7	Task Not Found

Example

```
$.ajax({ url: '/task/status?tid=1',
  type: 'get',
  crossDomain: true,
  beforeSend: function(request) {
    request.setRequestHeader("username",
      'username'),
    request.setRequestHeader("pwd",
      'password');}
  })
  .done(function(resp) {
    return resp;
  })
  .fail(function(error) {
    throw new Error("Error getting the data");
  });
```

CHAPTER 5: Task Queue

The task creation process is managed by the Kryon Server according to robot availability.

If no robot is available to immediately execute a task, the task is created and placed in queue with the status **Queued**.

A queued task is saved for up to 24 hours. If no robot becomes available to perform the task during that time, the task status is changed to **Expired**, and the task is permanently removed from the queue.

CHAPTER 6: Testing

The Web Service API can be tested using the provided **tester.html** file, which allows you to test the two API methods:

- The default location for this file is `C:\Program Files\Kryon Application Server\Kryon Web Server 64bit\WebAPI`

Add Task

Wizard custom id:

Number of runs:

Single run estimation:

Queue priority:

Machine Name:

Group Name:

User Name:

Password:

Use **variables** as parameters for the task's wizard.

Variable name	Value	
<input type="checkbox"/> var1	2	<input type="button" value="Add"/>
<input type="checkbox"/> var2	3	



NOTE

Prior to testing, you should ensure that a custom wizard ID exists. (You can use the tester to define a custom wizard ID if required.)

Get Status

Use the tester to add task IDs for status monitoring:

Id ^	Date ↕	Status ↕
52	"2015-07-19T07:24:39.027Z"	6

« Prev 1 Next »

Get Status



NOTE

Upon manual page refresh (F5), task information is lost and can no longer be monitored from the tester (only from Kryon Console or via custom API implementation).