Elixys Client-Server JSON

Table of Contents

[Overview 3](#_Toc292627448)

[Data structures 4](#_Toc292627449)

[Time and date 4](#_Toc292627450)

[Button constants 4](#_Toc292627451)

[Button 4](#_Toc292627452)

[Tab 4](#_Toc292627453)

[Configuration 5](#_Toc292627454)

[Sequence details 5](#_Toc292627455)

[Reagent 6](#_Toc292627456)

[Server state 6](#_Toc292627457)

[Sequence components 7](#_Toc292627458)

[Cassette 7](#_Toc292627459)

[Add 8](#_Toc292627460)

[Evaporate 9](#_Toc292627461)

[Transfer 10](#_Toc292627462)

[Elute 11](#_Toc292627463)

[React 12](#_Toc292627464)

[Prompt 13](#_Toc292627465)

[Move 13](#_Toc292627466)

[Install 14](#_Toc292627467)

[Comment 14](#_Toc292627468)

[Activity 15](#_Toc292627469)

[GET /configuration 15](#_Toc292627470)

[GET /state 16](#_Toc292627471)

[HOME 16](#_Toc292627472)

[SELECTSEQUENCE 17](#_Toc292627473)

[VIEWSEQUENCE 18](#_Toc292627474)

[EDITSEQUENCE 18](#_Toc292627475)

[RUNSEQUENCE 19](#_Toc292627476)

[MANUALRUN 20](#_Toc292627477)

[PROMPT 21](#_Toc292627478)

[GET /sequence/[*Sequence ID*] 22](#_Toc292627479)

[GET /sequence/[*Sequence ID*]/component/[*Component ID*] 22](#_Toc292627480)

[GET /sequence/[*Sequence ID*]/reagent/[*Reagent ID*] 22](#_Toc292627481)

[POST /[*state*] 23](#_Toc292627482)

[HOME 23](#_Toc292627483)

[SELECTSEQUENCE 23](#_Toc292627484)

[VIEWSEQUENCE 23](#_Toc292627485)

[EDITSEQUENCE 24](#_Toc292627486)

[RUNSEQUENCE 24](#_Toc292627487)

[MANUALRUN 24](#_Toc292627488)

[PROMPT 24](#_Toc292627489)

[POST /sequence/[*Sequence ID*] 25](#_Toc292627490)

[POST /sequence/[*Sequence ID*]/component/[*Component ID*] 25](#_Toc292627491)

[POST /sequence/[*Sequence ID*]/component/[*Component ID*]/[*Insert Index*] 25](#_Toc292627492)

[POST /sequence/[*Sequence ID*]/reagent/[*Reagent ID*] 25](#_Toc292627493)

[DELETE /sequence/[*Sequence ID*]/component/[*Component ID*] 25](#_Toc292627494)

# Overview

This document describes the messages passed between the Flash client and Python server in the Elixys system. The system is designed such that the client is thin and contains little to no state. The primary purpose of the client is to display the state that the server dictates and relay any significant user input to the server. The server will maintain the state of each client in its database. All aspects of user access control are handled on the server, from controlling the options the client displays to the user to preventing rouge clients from performing actions if they do not have sufficient privileges.

All interactions between the client and server occur over HTTP. Below is a brief overview of these interactions:

* The client can load details about the Elixys system configuration from the server by calling:

**GET /configuration**

This is done immediately after the client connects to load the system version and supported functions.

* The client can obtain the current client state from the server by calling:

**GET /state**

This is done immediately after the client connects to determine the initial state of the application. It is also called once per second while the client is running and several times per second during a synthesis to get the latest state of the system.

* The client can load the sequence associated with a given sequence ID and save sequence details by calling:

**GET\_/sequence/[*Sequence ID*]**

**POST\_/sequence/[*Sequence ID*]**

These calls are used when the user is viewing, editing or running a sequence. The POST call returns the new client state.

* The client can get a specific sequence component (i.e. cassette or unit operation), update, move or delete an existing component and insert a new component using the following calls:

**GET\_/sequence/[*Sequence\_ID*]/component/[*Component\_ID*]** **POST\_/sequence/[*Sequence\_ID*]/component/[*Component\_ID*]** **POST\_/sequence/[*Sequence\_ID*]/component/[*Component\_ID*]/[*Insert Index*]’**

**DELETE\_/sequence/[*Sequence\_ID*]/component/[*Component\_ID*]**

These calls are used when the user is viewing, editing or running a sequence. The POST and DELETE calls return the new client state.

* The client can get or update a specific sequence reagent using the following calls:

**GET\_/sequence/[*Sequence\_ID*]/reagent/[*Reagent\_ID*]** **POST\_/sequence/[*Sequence\_ID*]/reagent/[*Reaagent\_ID*]**

The number of reagents is fixed by the server based on the cassettes in use. The POST call returns the new client state.

* The client informs the server of any significant user action by calling:

**POST /[*state*]**

This call returns the new client state.

# Data structures

The data structures in this section are used throughout this document.

## Time and date

JSON does not dictate a standard format for times and dates, so the Elixys system will use the following standard:

1. Dates will be in the format “DD/MM/YYYY”.
2. Times will be in the format “HH:MM.SS”.

## Button constants

The following buttons are hardcoded into the client in certain screens because there is no foreseeable need for the server to dictate their presence. The following constant button IDs are recognized by the server on these screens:

1. BACK
2. NEXT
3. PREVIOUS
4. TRASH

## Button

Sent from the server to the client and describes a button that will be displayed to the user:

1. Text – The button text.
2. ID – Unique ID that is sent from the client to the server when the user clicks on the button.

**{**

**“type”:“button”,**

**“text”:”Create”,**

**“id”:”CREATE”**

**}**

## Tab

Sent from the server to the client and describes a tab in a tab control that will be displayed to the user:

1. Text – The tab text.
2. ID – Unique ID that is sent from the client to the server when the user clicks on the tab.

**{**

**“type”:“tab”,**

**“text”:”Saved Sequences”,**

**“id”:”SAVED”**

**}**

## Configuration

Sent from the server to the client and contains the Elixys system configuration:

1. Name – System name.
2. Version – System version.
3. Supported operations – Array of unit operation names that are supported by the underlying hardware.

**{**

**“type”:“configuration”,**

**“name”:“Mini cell 3”,**

**“version”:“2.0”,**

**"supportedoperations":**

**[**

**"Add",**

**"Evaporate",**

**"Transfer",**

**"Elute",**

**"React",**

**"Prompt",**

**"Install",**

**"Comment",**

**"Activity"**

**]**

**}**

## Sequence details

Sent from the server to the client and contains the details of a sequence:

1. Name – Sequence name.
2. Date – The date the sequence was created.
3. Time – The time the sequence was created.
4. Comment – Any comment associated with the sequence.
5. ID – Unique ID that is sent from the client to the server when the user selects the sequence.

**{**

**“type”:“sequencedetails”,**

**“name”:“FAC (high temp)”,**

**“time”:“8:00”,**

**“date”:“05/01/2012”,**

**“comment”:“Experimental FAC synthesis using high temperatures”,**

**“id”:“452”**

**}**

## Reagent

Describes a reagent and consists of the following:

1. Position – The reagent position.
2. Name – The short name of the reagent.
3. Name error – Contains a string describing the validation error or blank otherwise.
4. Description – The long description of the reagent.
5. Description error – Contains a string describing the validation error or blank otherwise.
6. ID – Unique ID that is sent from the client to the server to describe the reagent.

**{**

**“type”:”reagent”,**

**“position”:”3”,**

**“name”:”F-18”,**

**“nameerror”:””,**

**“description”:”[18F]F-, 10 mg Kryptofix (K222) and 1.0 mg potassium carbonate**

**(K2CO3) in acetonitrile (MeCN)”,**

**“descriptionerror”:””,**

**“id”:”928”**

**}**

## Server state

The current state of the server:

1. Running – Boolean value that indicates if a user is running the system. If this value is true then the subsequence values will be set.
2. Status – String describing the current system status.
3. Active unit – The active reaction unit.
4. Temperature – The temperature of the active reactor unit.
5. Activity – The last know radiation activity level of the active reactor unit in millicuries.
6. Activity time – The time the activity was last measured.
7. Video – URL of the video stream of the active reactor unit.

**{**

**“type”:”serverstate”,**

**“running”:”true”,**

**“status”:”Reacting, 8:23 minutes”,**

**“activeunit”:”1”,**

**“temperature”:”112.0”,**

**“activity”:”15.5”,**

**“activitytime”:”8:12.03”,**

**“video”:”rtmp://192.168.0.200:1935/Elixys/mp4:unit1camera.mp4”**

**}**

## Sequence components

Describes an individual component of a sequence:

1. Component Type:
   1. CASSETTE
   2. ADD
   3. EVAPORATE
   4. TRANSFER
   5. ELUTE
   6. REACT
   7. PROMPT
   8. MOVE
   9. INSTALL
   10. COMMENT
   11. ACTIVITY
2. Name – The display name of the component.
3. ID – Unique ID that is used by the client to refer to the component when communicating with the server.
4. Additional details – All of the above component types contain additional information as documented in the next sections.

### Cassette

Describes the configuration of a cassette. Contains the following in addition to the base data members above:

1. Used flag – Boolean value that indicates if this cassette is used.
2. Reagents – Array of reagent IDs.

**{**

**“type”:”component”,**

**“componenttype”:”CASSETTE”,**

**“name”:”Cassette 1”,**

**“id”:”1923”,**

**“used”:”true”,**

**“reagents”:**

**[**

**”847”,**

**”848”,**

**”849”,**

**”850”**

**]**

**}**

### Add

Describes the reagent addition unit operation. Contains the following in addition to the base data members above:

1. Unit – The reactor unit.
2. Unit options – Array of possible units for the user to select from.
3. Unit error – Contains a string describing the validation error or blank otherwise.
4. Reagent – The reagent ID.
5. Reagent options – Array of possible reagents for the user to select from.
6. Reagent error – Contains a string describing the validation error or blank otherwise.

**{**

**“type”:”component”,**

**“componenttype”:”ADD”,**

**“name”:”Add F-18”,**

**“id”:”1924”,**

**“unit”:”1”,**

**“unitoptions”:**

**[**

**”1”,**

**”2”,**

**”3”**

**],**

**“uniterror”:””,**

**“reagent”:”344”,**

**“reagentoptions”:**

**[**

**”211”,**

**”212”,**

**”213”**

**],**

**“reagenterror”:””**

**}**

### Evaporate

Describes the evaporation unit operation. Contains the following in addition to the base data members above:

1. Unit – The reactor unit.
2. Unit options – Array of possible units for the user to select from.
3. Unit error – Contains a string describing the validation error or blank otherwise.
4. Duration – The length of the reaction in the time format.
5. Duration error – Contains a string describing the validation error or blank otherwise.
6. Evaporation temperature – The evaporation temperature in Celsius.
7. Evaporation temperature error – Contains a string describing the validation error or blank otherwise.
8. Final temperature – The final temperature in Celsius.
9. Final temperature error – Contains a string describing the validation error or blank otherwise.
10. Stir speed – The stir speed in rotations per minute.
11. Stir speed error – Contains a string describing the validation error or blank otherwise.

**{**

**"type":"component",**

**"componenttype":"EVAPORATE",**

**"name":"Evaporate",**

**"id":"441",**

**"unit":"1",**

**"unitoptions":**

**[**

**"1",**

**"2",**

**"3"**

**],**

**"uniterror":"",**

**"duration":"00:05.00",**

**"durationerror":"",**

**"evaporationtemperature":"165.0",**

**"evaporationtemperatureerror":"",**

**"finaltemperature":"35.0",**

**"finaltemperatureerror":"",**

**"stirspeed":"500",**

**"stirspeederror":""**

**}**

### Transfer

Describes the transfer unit operation. Contains the following in addition to the base data members above:

1. Unit – The reactor unit.
2. Unit options – Array of possible units for the user to select from.
3. Unit error – Contains a string describing the validation error or blank otherwise.
4. Target – The target ID.
5. Target options – Array of possible targets for the user to select from.
6. Target error – Contains a string describing the validation error or blank otherwise.

**{**

**"type":"component",**

**"componenttype":"TRANSFER",**

**"name":"Transfer",**

**"id":"442",**

**"unit":"1",**

**"unitoptions":**

**[**

**"1",**

**"2",**

**"3"**

**]**

**"uniterror":"",**

**"target":"321",**

**"targetoptions":**

**[**

**"321",**

**]**

**"targeterror":"",**

**}**

### Elute

Describes the elution unit operation. Contains the following in addition to the base data members above:

1. Unit – The reactor unit.
2. Unit options – Array of possible units for the user to select from.
3. Unit error – Contains a string describing the validation error or blank otherwise.
4. Reagent – The reagent ID.
5. Reagent options – Array of possible reagents for the user to select from.
6. Reagent error – Contains a string describing the validation error or blank otherwise.
7. Target – The target ID.
8. Target options – Array of possible targets for the user to select from.
9. Target error – Contains a string describing the validation error or blank otherwise.

**{**

**"type":"component",**

**"componenttype":"ELUTE",**

**"name":"Elute",**

**"id":"443",**

**"unit":"2",**

**"unitoptions":**

**[**

**"1",**

**"2",**

**"3"**

**],**

**"uniterror":"",**

**"reagent":"12",**

**"reagentoptions":**

**[**

**"12",**

**"23",**

**"34"**

**],**

**"reagenterror":"",**

**"target":"321",**

**"targetoptions":**

**[**

**"321",**

**]**

**"targeterror":"",**

**}**

### React

Describes the reaction unit operation. Contains the following in addition to the base data members above:

1. Unit – The reactor unit.
2. Unit options – Array of possible units for the user to select from.
3. Unit error – Contains a string describing the validation error or blank otherwise.
4. Position – The react position.
5. Position options – Array of possible positions for the user to select from.
6. Duration – The length of the reaction in the time format.
7. Duration error – Contains a string describing the validation error or blank otherwise.
8. Reaction temperature – The reaction temperature in Celsius.
9. Reaction temperature error – Contains a string describing the validation error or blank otherwise.
10. Final temperature – The final temperature in Celsius.
11. Final temperature error – Contains a string describing the validation error or blank otherwise.
12. Stir speed – The stir speed in rotations per minute.
13. Stir speed error – Contains a string describing the validation error or blank otherwise.

**{**

**"type":"component",**

**"componenttype":"REACT",**

**"name":"React",**

**"id":"444",**

**"unit":"2",**

**"unitoptions":**

**[**

**"1",**

**"2",**

**"3"**

**],**

**"uniterror":"",**

**"position":"1",**

**"positionoptions":**

**[**

**"1",**

**"2"**

**],**

**"duration":"12:00.04",**

**"durationerror":"",**

**"reactiontemperature":"500.0",**

**"reactiontemperatureerror":"",**

**"finaltemperature":"35.0",**

**"finaltemperatureerror":"",**

**"stirspeed":"5000",**

**"stirspeederror":"",**

**}**

### Prompt

Describes the prompt unit operation. Contains the following in addition to the base data members above:

1. Message – Text to display to the user.
2. Message error – Contains a string describing the validation error or blank otherwise.

**{**

**"type":"component",**

**"componenttype":"PROMPT",**

**"name":"Prompt",**

**"id":"445",**

**"message":"Please take a sample for analysis",**

**"messageerror":"",**

**}**

### Move

Describes the move unit operation. Contains the following in addition to the base data members above:

1. Unit – The reactor unit.
2. Unit options – Array of possible units for the user to select from.
3. Unit error – Contains a string describing the validation error or blank otherwise.
4. Position – The react position.
5. Position options – Array of possible positions for the user to select from.
6. State – The reactor state.
7. State options – Array of possible states for the user to select from.

**{**

**"type":"component",**

**"componenttype":"MOVE",**

**"name":"Move",**

**"id":"445",**

**"unit":"1",**

**"unitoptions":**

**[**

**"1",**

**"2",**

**"3"**

**],**

**"uniterror":"",**

**"position":"2",**

**"positionoptions":**

**[**

**"1",**

**"2",**

**]**

**"state":"OPEN",**

**"stateoptions":**

**[**

**"OPEN",**

**"CLOSE"**

**]**

**}**

### Install

Describes the install unit operation. Contains the following in addition to the base data members above:

1. Unit – The reactor unit.
2. Unit options – Array of possible units for the user to select from.
3. Unit error – Contains a string describing the validation error or blank otherwise.
4. Message – Text to display to the user.
5. Message error – Contains a string describing the validation error or blank otherwise.

**{**

**"type":"component",**

**"componenttype":"INSTALL",**

**"name":"Install",**

**"id":"445",**

**"unit":"1",**

**"unitoptions":**

**[**

**"1",**

**"2",**

**"3"**

**],**

**"uniterror":"",**

**"message":"Take a radiation measurement",**

**"messageerror":""**

**}**

### Comment

Describes the comment unit operation. Contains the following in addition to the base data members above:

1. Comment – Comment the user added to the sequence.
2. Comment error – Contains a string describing the validation error or blank otherwise.

**{**

**"type":"component",**

**"componenttype":"COMMENT",**

**"name":"Comment",**

**"id":"445",**

**"comment":"Bromination and cytosine coupling",**

**"commenterror":""**

**}**

### Activity

Describes the radiation activity measurement unit operation. Contains the following in addition to the base data members above:

1. Unit – The reactor unit.
2. Unit options – Array of possible units for the user to select from.
3. Unit error – Contains a string describing the validation error or blank otherwise.

**{**

**"type":"component",**

**"componenttype":"ACTIVITY",**

**"name":"Measure activity",**

**"id":"445",**

**"unit":"1",**

**"unitoptions":**

**[**

**"1",**

**"2",**

**"3"**

**],**

**"uniterror":""**

**}**

# GET /configuration

This operation returns the Elixys system configuration.

**{**

**(Configuration is described above)**

**}**

# GET /state

This request is sent from the client to the server to fetch the current client state. The server responds with the following:

1. User name – The name of the current user for display purposes.
2. User access level – The current user’s access level in the system (e.g. “Tech”). This is for display purposes only as all permissions are enforced on the server side.
3. Server state – The current state of the server.
4. Client state – The page the user is currently viewing. Possible values are:
   1. HOME
   2. SELECTSEQUENCE
   3. VIEWSEQUENCE
   4. EDITSEQUENCE
   5. RUNSEQUENCE
   6. MANUALRUN
   7. PROMPT
5. Additional information – Page-specific information is returned as documented in the sections below.

## HOME

The home page contains the following additional information:

1. Buttons – Array of buttons to display on the screen.

**{**

**"type":"state",**

**"username":"Homer",**

**"useraccesslevel":"Administrator",**

**"serverstate":**

**[**

**(Server state is described above)**

**],**

**"clientstate":"HOME",**

**"buttons":**

**[**

**(Buttons are described above)**

**]**

**}**

## SELECTSEQUENCE

The Select Sequence page contains the following additional information:

1. Tabs – Array of tabs to show in the tab control.
2. Tab ID – The unique ID of the currently selected tab.
3. Navigation buttons – Array of buttons to display in the upper right corner.
4. Option buttons – Array of buttons to display in the bottom center of the tab panel.
5. Sequences – Array of sequence details to display in the tab panel.

**{**

**"type":"state",**

**"username":"Homer",**

**"useraccesslevel":"Administrator",**

**"serverstate":**

**[**

**(Server state is described above)**

**],**

**"clientstate":"SELECTSEQUENCE",**

**"tabs":**

**[**

**(Tabs are described above)**

**],**

**"tabid":"14",**

**"navigationbuttons":**

**[**

**(Buttons are described above)**

**]**

**"optionbuttons":**

**[**

**(Buttons are described above)**

**]**

**"sequences":**

**[**

**(Sequence details are described above)**

**]**

**}**

## VIEWSEQUENCE

The View Sequence page contains the following additional information:

1. Navigation buttons – Array of buttons to display in the upper right corner.
2. Sequence ID – The unique ID of the sequence the user is currently viewing.
3. Component ID – The unique ID of the component in the sequence the user is currently viewing.

**{**

**"type":"state",**

**"username":"Homer",**

**"useraccesslevel":"Administrator",**

**"serverstate":**

**[**

**(Server state is described above)**

**],**

**"clientstate":"VIEWSEQUENCE",**

**"navigationbuttons":**

**[**

**(Buttons are described above)**

**]**

**"sequenceid":"65",**

**"componentid":422"**

**}**

## EDITSEQUENCE

The Edit Sequence page contains the following additional information:

1. Navigation buttons – Array of buttons to display in the upper right corner.
2. Sequence ID – The unique ID of the sequence the user is currently viewing.
3. Component ID – The unique ID of the component in the sequence the user is currently viewing.

**{**

**"type":"state",**

**"username":"Homer",**

**"useraccesslevel":"Administrator",**

**"serverstate":**

**[**

**(Server state is described above)**

**],**

**"clientstate":"EDITSEQUENCE",**

**"navigationbuttons":**

**[**

**(Buttons are described above)**

**]**

**"sequenceid":"65",**

**"componentid":422"**

**}**

## RUNSEQUENCE

The Run Sequence page contains the following additional information:

1. Navigation button – Button to display in the upper right corner.
2. Sequence ID – The unique ID of the sequence the user is currently running.
3. Component ID – The unique ID of the component in the sequence the user is currently running.

**{**

**"type":"state",**

**"username":"Homer",**

**"useraccesslevel":"Administrator",**

**"serverstate":**

**[**

**(Server state is described above)**

**],**

**"clientstate":"RUNSEQUENCE",**

**"navigationbuttons":**

**[**

**(Buttons are described above)**

**]**

**"sequenceid":"65",**

**"componentid":422"**

**}**

## MANUALRUN

The Manual Run page contains the following additional information:

1. Step – Describes the current step in the manual run:
   1. SELECT – The user is prompted to select the next operation.
   2. CONFIGURE – The user is prompted to configure the operation.
   3. RUN – The operation is being performed.
2. Navigation buttons – Array of buttons to display in the upper right corner.
3. Sequence ID – The unique ID of the sequence the user is currently running.
4. Select operation text – The text to display to the user if this is the SELECT step.
5. Component ID – The unique ID of the component in the sequence. This component is the one the user must configure in the CONFIGURE step and the one the system is running in the RUN step.

**{**

**"type":"state",**

**"username":"Homer",**

**"useraccesslevel":"Administrator",**

**"serverstate":**

**[**

**(Server state is described above)**

**],**

**"clientstate":"MANUALRUN",**

**"step":"SELECT",**

**"navigationbuttons":**

**[**

**(Buttons are described above)**

**]**

**"sequenceid":"65",**

**"selectoperationtext":"Addition of acetonitrile to unit 1 completed**

**successfully.\n\nPlease select the next operation from the toolbox.",**

**"componentid":"422"**

**}**

## PROMPT

The Prompt modal dialog box contains the following additional information:

1. Text 1 – The first text to display to the user.
2. Edit 1 – True if the first edit box is to be displayed.
3. Text 2 – The second text to display to the user or empty if not used.
4. Edit 2 – True if the second edit box is to be displayed.
5. Buttons – Array of buttons to display to the user.

**{**

**"type":"state",**

**"username":"Homer",**

**"useraccesslevel":"Administrator",**

**"serverstate":**

**[**

**(Server state is described above)**

**],**

**"clientstate":"PROMPT",**

**"text1":"Are you sure you want to abort the current operation?",**

**"edit1":"false",**

**"text2":"",**

**"edit2":false",**

**"buttons":**

**[**

**(Buttons are described above)**

**]**

**}**

# GET /sequence/[*Sequence ID*]

This operation returns the following for the given sequence ID:

1. Sequence details – Details of the sequence.
2. Sequence components – Array of the following:
   1. Component name – Short display name for this component.
   2. Component ID – Unique ID that the client uses to refer to the component.
   3. Validation error – Boolean value that specifies if the component has a validation error.

**{**

**"type":"sequence",**

**"details":**

**[**

**(Sequence details are described above)**

**],**

**"components":**

**[**

**{**

**"name":"Add F-18",**

**"id":"100",**

**"validationerror":"false"**

**},**

**{**

**"name":"Evaporate",**

**"id":"101",**

**"validationerror":"false"**

**},**

**{**

**"name":"React",**

**"id":"102",**

**"validationerror":"false"**

**}**

**]**

**}**

# GET /sequence/[*Sequence ID*]/component/[*Component ID*]

This operation returns the component details for the given sequence and component IDs.

**{**

**(Component details are described above)**

**}**

# GET /sequence/[*Sequence ID*]/reagent/[*Reagent ID*]

This operation returns the reagent details for the given sequence and reagent IDs.

**{**

**(Reagent details are described above)**

**}**

# POST /[*state*]

The client uses POST to inform the server of any significant user action on the page specified by *state*:

1. Action – The action performed by the user:
   1. Action – Describes the user action. Possible values are:
      1. BUTTONCLICK
      2. TABCLICK
   2. Target ID – ID of the target component the user acted on.
2. Additional information – Page-specific information as documented in the sections below.

The server always returns the new state of the client as a response to a POST.

## HOME

The Home page does not send any additional information.

**POST /HOME**

**{**

**"action":**

**{**

**"type":"BUTTONCLICK",**

**"targetid":"BACK"**

**}**

**}**

## SELECTSEQUENCE

The select sequence page sends the following in addition to the action:

1. Sequence ID – The unique ID of the currently selected sequence.

**POST /SELECTSEQUENCE**

**{**

**"action":**

**{**

**"type":"BUTTONCLICK",**

**"targetid":"BACK"**

**},**

**“id”:”9000”**

**}**

## VIEWSEQUENCE

The View Sequence page does not send any additional information. Each component in the sequence is displayed as a button. If the user clicks on one of these buttons then a BUTTONCLICK action will be sent to the server with the component ID for the button ID.

**POST /VIEWSEQUENCE**

**{**

**"action":**

**{**

**"type":"BUTTONCLICK",**

**"targetid":"BACK"**

**}**

**}**

## EDITSEQUENCE

The Edit Sequence page does not send any additional information. Each component in the sequence is displayed as a button. If the user clicks on one of these buttons then a BUTTONCLICK action will be sent to the server with the component ID for the button ID.

**POST /EDITSEQUENCE**

**{**

**"action":**

**{**

**"type":"BUTTONCLICK",**

**"targetid":"BACK"**

**}**

**}**

## RUNSEQUENCE

The Run Sequence page does not send any additional information.

**POST /RUNSEQUENCE**

**{**

**"action":**

**{**

**"type":"BUTTONCLICK",**

**"targetid":"BACK"**

**}**

**}**

## MANUALRUN

The Manual Run page does not send any additional information.

**POST /MANUALRUN**

**{**

**"action":**

**{**

**"type":"BUTTONCLICK",**

**"targetid":"BACK"**

**}**

**}**

## PROMPT

The Prompt modal dialog box contains the following additional information:

1. Edit 1 – The text from the first edit box.
2. Edit 2 – The text from the second edit box.

**POST /PROMPT**

**{**

**"action":**

**{**

**"type":"BUTTONCLICK",**

**"targetid":"BACK"**

**},**

**“edit1”:”New sequence name”,**

**“edit2”:””**

**}**

# POST /sequence/[*Sequence ID*]

This operation is used by the client to save sequence details to the server. Specify a sequence ID to update an existing sequence or leave it blank to create a new one.

**{**

**(Sequence details are described above)**

**}**

# POST /sequence/[*Sequence ID*]/component/[*Component ID*]

This operation is used by the client to update existing unit operations within a sequence.

**{**

**(Component details are described above)**

**}**

# POST /sequence/[*Sequence ID*]/component/[*Component ID*]/[*Insert Index*]

This operation is used by the client to insert the given component at an absolute position in the sequence. Specify a component ID to move an existing sequence or leave it blank to insert a new one.

**{**

**(Component details are described above)**

**}**

# POST /sequence/[*Sequence ID*]/reagent/[*Reagent ID*]

This operation is used by the client to update reagent details for an existing reagent.

**{**

**(Reagent details are described above)**

**}**

# DELETE /sequence/[*Sequence ID*]/component/[*Component ID*]

This operation is used by the client to delete an existing unit operation within a sequence.