Elixys Client-Server JSON

Table of Contents

[Overview 4](#_Toc331442508)

[Data structures 5](#_Toc331442509)

[Time and date 5](#_Toc331442510)

[Button 5](#_Toc331442511)

[Tab 5](#_Toc331442512)

[Column 6](#_Toc331442513)

[Disallowed reagent position 6](#_Toc331442514)

[Configuration 7](#_Toc331442515)

[User 8](#_Toc331442516)

[Sequence metadata 8](#_Toc331442517)

[Sequence component 9](#_Toc331442518)

[Reagent 9](#_Toc331442519)

[Vacuum state 10](#_Toc331442520)

[Liquid sensor state 10](#_Toc331442521)

[Pressure regulator state 10](#_Toc331442522)

[Valve state 10](#_Toc331442523)

[Reagent robot position 11](#_Toc331442524)

[Reagent robot state 11](#_Toc331442525)

[Reactor position 11](#_Toc331442526)

[Reactor state 12](#_Toc331442527)

[Prompt state 13](#_Toc331442528)

[Client state 13](#_Toc331442529)

[Run state 14](#_Toc331442530)

[Hardware state 15](#_Toc331442531)

[Server state 15](#_Toc331442532)

[Validation 16](#_Toc331442533)

[GET /configuration 16](#_Toc331442534)

[GET /state 17](#_Toc331442535)

[HOME 17](#_Toc331442536)

[SELECT\_SAVEDSEQUENCES and SELECT\_RUNHISTORY 18](#_Toc331442537)

[VIEW 18](#_Toc331442538)

[EDIT 18](#_Toc331442539)

[RUN 18](#_Toc331442540)

[GET /runstate 19](#_Toc331442541)

[GET component 19](#_Toc331442542)

[Cassette 20](#_Toc331442543)

[Add 21](#_Toc331442544)

[Evaporate 22](#_Toc331442545)

[Transfer 23](#_Toc331442546)

[React 24](#_Toc331442547)

[Prompt 25](#_Toc331442548)

[Install 25](#_Toc331442549)

[Comment 26](#_Toc331442550)

[Trap F18 26](#_Toc331442551)

[Elute F18 27](#_Toc331442552)

[Initialize 27](#_Toc331442553)

[Mix 28](#_Toc331442554)

[Move 28](#_Toc331442555)

[ExternalAdd 29](#_Toc331442556)

[Summary 29](#_Toc331442557)

[GET /sequence/[*Sequence ID*] 30](#_Toc331442558)

[GET /sequence/[*Sequence ID*]/component/[*Component ID*] 30](#_Toc331442559)

[GET /sequence/[*Sequence ID*]/reagent/[*Reagent ID 1*].[*Reagent ID 2*] 30](#_Toc331442560)

[POST /[*state*] 30](#_Toc331442561)

[HOME 31](#_Toc331442562)

[SELECT 31](#_Toc331442563)

[VIEW 31](#_Toc331442564)

[EDIT 31](#_Toc331442565)

[RUN 32](#_Toc331442566)

[PROMPT 32](#_Toc331442567)

[POST component 33](#_Toc331442568)

[Add 34](#_Toc331442569)

[Evaporate 34](#_Toc331442570)

[Transfer 35](#_Toc331442571)

[React 35](#_Toc331442572)

[Prompt 36](#_Toc331442573)

[Install 36](#_Toc331442574)

[Comment 36](#_Toc331442575)

[Trap F18 37](#_Toc331442576)

[Elute F18 37](#_Toc331442577)

[Initialize 37](#_Toc331442578)

[Mix 38](#_Toc331442579)

[Move 38](#_Toc331442580)

[ExternalAdd 38](#_Toc331442581)

[POST /sequence/[*Sequence ID*] 39](#_Toc331442582)

[POST /sequence/[*Sequence ID*]/component/[*Component ID*] 39](#_Toc331442583)

[POST /sequence/[*Sequence ID*]/component/[*Component ID*]/[*Insert ID*] 39](#_Toc331442584)

[POST /sequence/[*Sequence ID*]/reagent/[*Reagent ID*] 39](#_Toc331442585)

[DELETE /sequence/[*Sequence ID*]/component/[*Component ID*] 39](#_Toc331442586)

# 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 dictating 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 state from the server by calling:

**GET /state**

This is called at regular intervals while the client is running and more frequently 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 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 state.

* The client can get one or more reagents and update a specific reagent using the following calls:

**GET\_/sequence/[*Sequence\_ID*]/reagent/[*Reagent\_ID 1*].[Reagent ID 2].[Reagent ID N]**

**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 state.

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

**POST /[*state*]**

This call returns the new state.

* Small screen devices that are only concerned about the run state can call:

**GET /runstate**

# 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 MySQL timestamp format:

YYYY-MM-DD HH:MM:SS

## Button

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

1. ID – Unique ID that is sent from the client to the server when the user clicks on the button.
2. Text – Text to display on the face of the button.
3. Enabled – Flag that indicates if the button is enabled.
4. Selection required – Flag that indicates if this button is only enabled if the Enabled flag is set and an item is selected.

**{**

**“type”:“button”,**

**“text”:”CREATE SEQUENCE”,**

**“id”:”CREATE”,**

**“enabled”:True,**

**“selectionrequired”:True**

**}**

## 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”**

**}**

## Column

Sent from the server to the client and describes a column in a table:

1. Data field – The name of the data field.
2. Display text – The text to display at the top of the column.
3. Percent width – The percent of the total table width that should be allocated to this column.
4. Sortable flag – True if the user can sort by this column, false otherwise.
5. Sort mode – The sort mode. Possible values are “up” and “down” of the table data is currently sorted by this column and “none” if the data cannot be sorted by this column.

**{**

**"type":"column",**

**"data":"name",**

**"display":"Name",**

**"percentwidth":35,**

**"sortable":True,**

**"sortmode":"down",**

**}**

## Disallowed reagent position

Sent from the server to the client and describes a disallowed reagent position:

1. Cassette – The number of the cassette.
2. Reagent – The reagent position that is disallowed.

**{**

**"type":"disallowedreagentposition",**

**"cassette":1,**

**"reagent":1**

**}**

## Configuration

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

1. Name – System name.
2. Version – System version.
3. Debug – Flag that indicates if the server is running in debug mode.
4. Supported operations – Array of unit operation names that are supported by the underlying hardware.
5. Reactors – Number of reactors in the system.
6. Reagents per reactor – Number of reagents associated with each reactor.
7. Columns per reactor – Number of columns associated with each reactor.
8. Disallowed reagent positions – Array of reagent positions that are not allowed.

**{**

**“type”:“configuration”,**

**“name”:“Mini cell 3”,**

**“version”:“2.0”,**

**“debug”:”false”,**

**"supportedoperations":**

**[**

**"ADD",**

**"EVAPORATE",**

**"TRANSFER",**

**"REACT",**

**"PROMPT",**

**"INSTALL",**

**"COMMENT",**

**"TRAPF18",**

**"ELUTEF18",**

**"INITIALIZE",**

**"MIX",**

**"MOVE",**

**"EXTERNALADD"**

**]**

**"reactors":3,**

**"reagentsperreactor":11,**

**"columnsperreactor":2,**

**“disallowedreagentpositions”:**

**{**

**(Disallowed reagent positions are described above)**

**}**

**}**

## User

Describes the current user and consists of the following:

1. User name – The name of the current user for display purposes.
2. First name – The user’s first name.
3. Last name – The user’s last name.
4. Access level – The current user’s access level in the system (e.g. “Administrator”). This is for display purposes only as all permissions are enforced on the server side.

**{**

**“type”:”user”,**

**"username":"hsimpson",**

**“firstname”:”Homer”,**

**“lastname”:”Simpson”,**

**"accesslevel":"Administrator"**

**}**

## Sequence metadata

Sent from the server to the client and contains the sequence metadata:

1. Name – Sequence name.
2. Time – Time the sequence was created.
3. Date – Date 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.
6. Creator – Name of the user that created the sequence.
7. Components – Number of components in this sequence.
8. Valid – Flag that indicates if the sequence is valid.
9. Dirty – Flag that indicates that the sequence has been flagged for validation.

**{**

**“type”:“sequencemetadata”,**

**“name”:“FAC (high temp)”,**

**“time”:“08:00.12”,**

**“date”:”03/19/2012”,**

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

**“id”:452,**

**“creator”:”devel”,**

**“components”:17,**

**“valid”:true,**

**“dirty”:false**

**}**

## Sequence component

Sent from the server to the client and contains metadata for a sequence component:

1. Component ID – Unique ID that the client uses to refer to the component.
2. Component type – The type of unit operation.
3. Note – Short text string to display with the component.
4. Validation error – Boolean value that specifies if the component has a validation error.

**{**

**“type”:”sequencecomponent”,**

**“componenttype”:”ADD”,**

**"id":100,**

**“note”:”F-18”,**

**"validationerror":false**

**}**

## Reagent

Describes a reagent and consists of the following:

1. Reagent ID – Unique ID that the client uses to describe the reagent to the server.
2. Component ID – The unique ID of the cassette where this reagent resides.
3. Position – The reagent position in the cassette.
4. Name – The short name of the reagent.
5. Name validation – Contains a string describing the name validation.
6. Description – The long description of the reagent.
7. Description validation – Contains a string describing the description validation.

**{**

**"type": "reagent",**

**"reagentid": 2,**

**"componentid": 1,**

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

**"name": "MeCN-1",**

**"namevalidation": "type=string; required=true",**

**"description": "1.5mL MeCN - 271004-12X100ML"**

**"descriptionvalidation": "type=string",**

**},**

## Vacuum state

Describes the state of a vacuum system and consists of the following:

1. On – Flag indicating if the vacuum is on.
2. Vacuum – The actual vacuum in kPa.

**{**

**“type”:”vacuumstate”,**

**“on”:True,**

**“vacuum”:-14.3**

**}**

## Liquid sensor state

Describes the state of a liquid sensor and consists of the following:

1. Name – String describing the liquid sensor.
2. Liquid present – Flag indicating if liquid is present.

**{**

**“type”:”liquidsensorstate”,**

**“name”:”LS1”,**

**“liquidpresent”:False**

**}**

## Pressure regulator state

Describes the state of a pressure regulator and consists of the following:

1. Name – String describing the pressure regulator.
2. Pressure – The actual pressure in PSI.

**{**

**“type”:”pressureregulatorstate”,**

**“name”:”Gas”,**

**“pressure”:4.3**

**}**

## Valve state

Describes the nitrogen valve states and consists of the following:

1. Gas transfer valve – Boolean value that indicates if the gas transfer valve is open.
2. F18 load valve – Boolean value that indicates if the F18 load valve is open.
3. HPLC valve – Position of the HPLC valve. Possible values are “load” and “inject”.

**{**

**“type”:”valvestate”,**

**“gastransfervalve”:False,**

**“f18loadvalve”:True,**

**“hplcvalve”:”load”**

**}**

## Reagent robot position

Describes the position of the reagent robot and consists of the following:

1. Name 1 – First position name (e.g. “Reactor 1”).
2. Name 2 – Second position name (e.g. “Elute”).

**{**

**”type”:”reagentrobotposition”,**

**"name1”:”Reactor 1”,**

**"name2”:”Reagent 5”,**

**}**

## Reagent robot state

Describes the state of the reagent robot and consists of the following:

1. Position – The current position of the reagent robot.
2. Gripper – String that specifies the state of the gripper. Possible values are “Open”, “Closed” and “Indeterminate”. Note that “Closed” indicates a vial is between the fingers while “Indeterminate” indicates no vial present.
3. Gripper actuator – String that specifies the state of the gripper actuator. Possible values are “Up”, “Down” and “Indeterminate”.
4. Gas transfer actuator – String that specifies the state of the gas transfer actuator. Possible values are “Up”, “Down” and “Indeterminate”.

**{**

**“type”:”reagentrobotstate”,**

**"position":(Reagent robot position is described above),**

**“gripper”:”Open”,**

**“gripperactuator”:”Up”,**

**“gastransferactuator”:”Up”**

**}**

## Reactor position

Describes the position of a reactor and consists of the following:

1. Horizontal – String describing the horizontal position, e.g. “Add”, “Evaporate”.
2. Vertical – String describing the vertical position. Possible values are “Up”, “Down” and “Indeterminate”.

**{**

**"type":"reactorposition",**

**"horizontal":"React1",**

**"vertical":"Up"**

**}**

## Reactor state

Describes the state of a reactor and consists of the following:

1. Number – The reactor number.
2. Temperature – The actual temperature of the reactor in Celsius.
3. Position – The reactor position.
4. Activity – The last know radiation activity level of the reactor in millicuries.
5. Activity time – The time the activity was measured.
6. Transfer position – String that describes the transfer stopcock position. Possible values are “Waste”, “Out” and “Indeterminate”.
7. Stir speed – Speed of the stir motor.
8. Video – URL of the video stream.

Reactor 1 will have the following additional fields:

1. Column position – String that describes the position of the column stopcocks. Possible values are “Load”, “Elute” and “Indeterminate”.

**{**

**"type":"reactorstate",**

**"number":1,**

**"temperature":165.0,**

**"position":(Reactor position is described above),**

**"activity":645.1,**

**"activitytime":"18:23.31",**

**"transferposition":"Waste",**

**"stirspeed":500,**

**“video”:”rtmp://192.168.1.100/Elixys/mp4:reactor1.mp4”,**

**"columnposition":"Load"**

**}**

## Prompt state

The current state of the Prompt modal dialog box:

1. Show – Boolean value that specifies if the prompt dialog is shown.
2. Title – Title of the prompt dialog.
3. Text 1 – The first text to display to the user.
4. Edit 1 – True if the first edit box is to be displayed.
5. Edit 1 default – Default text to display in the first edit box.
6. Edit 1 validation – Validation string for the text entered by the user in the first edit box.
7. Text 2 – The second text to display to the user or empty if not used.
8. Edit 2 – True if the second edit box is to be displayed.
9. Edit 2 default – Default text to display in the second edit box.
10. Edit 2 validation – Validation string for the text entered by the user in the second edit box.
11. Buttons – Array of buttons to display to the user.

**{**

**"type":"promptstate",**

**“show”:”true”,**

**“title”:”Abort”,**

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

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

**“edit1default:”:””,**

**“edit1validation”:”type=string; required=true”,**

**"text2":"",**

**"edit2":”false",**

**“edit2default:”:””,**

**“edit2validation”:””,**

**"buttons":**

**[**

**{Buttons are described above}**

**]**

**}**

## Client state

The current state of the client:

1. Screen – Screen the client is currently on.
2. Prompt – Describes any prompt that is being displayed.
3. Sequence ID – ID of the sequence that the client is on.
4. Component ID – ID of the component that the client is on.
5. Last select screen – The last select screen the user viewed (possible values are “SAVED” or “HISTORY”).

**{**

**“type”:”clientstate”,**

**"screen": "HOME",**

**“prompt”:{Prompt state is described above}**

**"sequenceid": 1,**

**"componentid": 32**

**“lastselectscreen”: “SAVED”,**

**}**

## Run state

The current run state of the server:

1. Status – String describing the current system status.
2. Running – Flag that indicates if the system is running a sequence.
3. Description – String describing the current unit operation.
4. Username – Name of the user that is operating the system.
5. Sequence ID – ID of the sequence that the system is running.
6. Component ID – ID of the component that the system is currently on.
7. Prompt – Describes any prompt that is being displayed in association with the run.
8. Time – Amount of time either remaining or elapsed.
9. Time description – String indicating if the time field contains time remaining or elapsed.
10. User alert – String to display to the user or empty if none.
11. Button – Button to display to the user or blank if none.
12. Waiting for input – Flag that indicates if the system is waiting for user input.
13. Run complete – Flag that indicates if the run is complete.

**{**

**“type”:”runstate”,**

**“status”:”Moving reactor”,**

**“running”:True,**

**“description”:”Adding MeCN to reaction 1 using 2 psi nitrogen for 15 seconds”,**

**“username”:”hsimpson”,**

**“sequenceid”:4,**

**“componentid”:6,**

**“prompt”:{Prompt state is described above},**

**"time":”00:03:14”,**

**"timedescription":”TIME REMAINING”,**

**"useralert":”Run will pause after the current unit operation.”,**

**"unitoperationbutton":{Button is described above},**

**"waitingforuserinput":False,**

**"runcomplete":False**

**}**

## Hardware state

The current hardware state of the server:

1. Cooling – Specifies if the cooling system is on.
2. Vacuum – The actual vacuum in kPa.
3. Liquid sensors – Array of liquid sensors.
4. Pressure regulators – Array of pressure regulators.
5. Valves – Details of the nitrogen valves.
6. Reagent Robot – Details of the reagent robot.
7. Reactors – Array of reactors.

**{**

**“type”:”hardwarestate”,**

**“cooling”:False,**

**“vacuum”:**

**[**

**{Vacuum state is described above}**

**]**

**“liquidsensors”:**

**[**

**{Liquid sensor state is described above}**

**]**

**“pressureregulators”:**

**[**

**{Pressure regulator state is described above}**

**]**

**“valves”:**

**[**

**{Valve state is described above}**

**]**

**“reagentrobot”:{Reagent robot state is described above},**

**“reactors”:**

**[**

**{Reactor state is described above}**

**]**

**}**

## Server state

The current state of the server:

1. Run state – The current run state of the server.
2. Hardware state – The current hardware state of the server

**{**

**“type”:”serverstate”,**

**“runstate”:{Run state is described above},**

**“hardwarestate”:{Hardware state is described above}**

**}**

## Validation

Each user-editable field in the state is accompanies by a validation string. This string is semicolon-delimited and can consist of the following options:

1. Type – Describes the type of field. Possible types and the required fields that must accompany each are:
   1. enum-number – An enumeration of number values specified in the “values” field. Presented to the user as a list of options to choose from.
   2. enum-reagent – An enumeration of reagents IDs specified in the “values” field. Presented to the user as a list of reagent short names to choose from.
   3. enum-target – An enumeration of target IDs specified in the “values” field. Presented to the user as a list of target names to choose from.
   4. string – Specifies a string.
2. Values – Describes a fixed set of comma-separated values. The manner in which the values are interpreted depends on the type.
3. Minimum – The minimum allowed value. The manner in which this is interpreted depends on the type.
4. Maximum – The maximum allowed value. The manner in which this is interpreted depends on the type.
5. Required – Flag that determines if this field is required.

# 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 state. The server responds with the following:

1. User – The current user.
2. Timestamp – Integer representing the epoch time when the state was generated. The purposed of this field is to allow the client to only use the most up-to-date state.
3. Server state – The current state of the server.
4. Client state – The state of the client. This includes which screen the user is currently viewing which in turn indicates additional fields that will be set. Possible screens are:
   1. HOME
   2. SELECT\_SAVEDSEQUENCES
   3. SELECT\_RUNHISTORY
   4. VIEW
   5. EDIT
   6. RUN
5. Buttons – Array of navigation buttons to display.
6. Screen-specific details – Additional information is returned as documented in the following sections.

**{**

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

**“user”:**

**{**

**(User is described above)**

**}**

**“timestamp”:”1339128018.78”,**

**"serverstate":**

**{**

**(Server state is described above)**

**},**

**"clientstate":**

**{**

**(Client state is described above)**

**}**

**"buttons":**

**{**

**(Buttons are described above)**

**}**

**}**

## HOME

The Home screen does not contain any additional information.

## SELECT\_SAVEDSEQUENCES and SELECT\_RUNHISTORY

The Select Saved Sequence and Select Run History pages contain 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. Columns – Array of table columns.
4. Sequences – Array of sequence metadata to display in the tab panel.

**{**

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

**“user”:**

**{**

**(User is described above)**

**}**

**"serverstate":**

**{**

**(Server state is described above)**

**},**

**"clientstate":**

**{**

**(Client state is described above)**

**}**

**"buttons":**

**{**

**(Buttons are described above)**

**}**

**"tabs":**

**{**

**(Tabs are described above)**

**},**

**"tabid":14,**

**“columns”:**

**{**

**(Columns are described above)**

**}**

**"sequences":**

**{**

**(Sequence metadata are described above)**

**}**

**}**

## VIEW

The View Sequence screen does not contain any additional information.

## EDIT

The Edit Sequence screen does not contain any additional information.

## RUN

The Run Sequence screen does not contain any additional information.

# GET /runstate

This request is sent from small-screen clients to the server to fetch the current run state. It returns either the run state if the system is running a sequence or the home state otherwise.

# GET component

Used by the server to describe an individual component of a sequence to the client. Contains the following base fields plus additional component-specific detail documented in the sections below:

1. Component Type:
   1. CASSETTE
   2. ADD
   3. EVAPORATE
   4. TRANSFER
   5. ELUTE
   6. REACT
   7. PROMPT
   8. INSTALL
   9. COMMENT
   10. TRAPF18
   11. ELUTEF18
   12. INITIALIZE
   13. MIX
   14. MOVE
   15. EXTERNALADD
   16. SUMMARY
2. ID – Unique ID that is used by the client to refer to the component when communicating with the server.
3. Name – The display name of the component.
4. Validation error – Flag that indicates that one of the fields of this component are invalid.

## Cassette

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

1. Reactor – The reactor associated with this component.
2. Reagents – Array of reagents.

**{**

**“type”:”component”,**

**“componenttype”:”CASSETTE”,**

**“id”:1,**

**“name”:”Cassette 1”,**

**“sequenceid”:1,**

**“reactor”:1,**

**"validationerror":false,**

**“reagents”:**

**[**

**(Reagents are described above)**

**]**

**}**

## Add

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

1. Reactor – The reactor associated with this component.
2. Reactor validation – Contains a string describing the reactor validation.
3. Reagent – The reagent to add to the reactor.
4. Reagent validation – Contains a string describing the reagent validation. Included in this string are the IDs of the other reagents that the user can choose from in Edit mode.
5. Delivery position – Reagent delivery position. Possible values are 1 and 2.
6. Delivery position validation – Contains a string describing the delivery position validation.
7. Delivery time – Time to deliver the reagent in seconds. This value will override the default if set or ignored if zero.
8. Delivery time validation – Contains a string describing the delivery time validation.
9. Delivery pressure – Pressure in PSI to use when delivering the reagent. This value will override the default if set or ignored if zero.
10. Delivery pressure validation – Contains a string describing the delivery pressure validation.

**{**

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

**"componenttype":"ADD",**

**"id":11,**

**"name":"Add MeCN-1"**

**"sequenceid":1,**

**"reactor":1,**

**"reactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"deliverypressure":5,**

**"deliverypressurevalidation":"type=number; min=0; max=15",**

**"deliverytime":10,**

**"deliverytimevalidation":"type=number; min=0; max=10",**

**"deliveryposition":2,**

**"deliverypositionvalidation":"type=enum-number; values=1,2; required=true",**

**"reagent":(Reagent details are described above),**

**"reagentvalidation":"type=enum-reagent; values=1,2,3,4,5,6,7,8; required=true",**

**"validationerror":false,**

**}**

## Evaporate

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

1. Reactor – The reactor associated with this component.
2. Reactor validation – Contains a string describing the reactor validation.
3. Duration – The length of the evaporation in seconds.
4. Duration validation – Contains a string describing the reactor validation.
5. Evaporation temperature – The evaporation temperature in Celsius.
6. Evaporation temperature validation – Contains a string describing the evaporation temperature validation.
7. Final temperature – The final temperature in Celsius.
8. Final temperature validation – Contains a string describing the final temperature validation.
9. Stir speed – The stir speed in arbitrary units.
10. Stir speed validation – Contains a string describing the stir speed validation.
11. Stop at temperature flag – Flag that indicates if stir bar will stop once the evaporation temperature is reached.
12. Stop at temperature flag validation – Contains a string descripting the stop at temperature flag validation.
13. Evaporation pressure – Nitrogen pressure in PSI to use when evaporating. This value will override the default if set or ignored if zero.
14. Evaporation pressure validation – Contains a string describing the evaporation pressure validation.

**{**

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

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

**"id":10,**

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

**"sequenceid":1,**

**"reactor":1,**

**"reactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"evaporationtemperature":110,**

**"evaporationtemperaturevalidation":"type=number; min=20; max=200; required=true",**

**"evaporationpressure":10,**

**"evaporationpressurevalidation":"type=number; min=0; max=25",**

**"finaltemperature":30,**

**"finaltemperaturevalidation":"type=number; min=20; max=200; required=true",**

**"duration":480,**

**"durationvalidation":"type=number; min=0; max=7200; required=true",**

**"stirspeed":450,**

**"stirspeedvalidation":"type=number; min=0; max=5000; required=true"**

**"stopattemperature":0,**

**"stopattemperaturevalidation":"type=enum-number; values=0,1; required=true",**

**"validationerror":false,**

**}**

## Transfer

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

1. Source reactor – The source reactor.
2. Source reactor validation – Contains a string describing the source reactor validation.
3. Target reactor – The target reagent.
4. Target reactor validation – Contains a string describing the target reactor validation.
5. Mode – The transfer mode, either “Trap” or “Elute”.
6. Mode validation – Contains a string describing the mode validation.
7. Pressure – Nitrogen pressure in PSI to use when transferring. This value will override the default if set or ignored if zero.
8. Pressure validation – Contains a string describing the pressure validation.
9. Duration – The length of the evaporation in seconds.
10. Duration validation – Contains a string describing the duration validation.

**{**

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

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

**"id":22,**

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

**"sequenceid":1,**

**"sourcereactor":1,**

**"sourcereactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"targetreactor":2,**

**"targetreactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"mode":"Trap",**

**"modevalidation":"type=enum-string; values=Trap,Elute; required=true",**

**"pressure":3,**

**"pressurevalidation":"type=number; min=0; max=25",**

**"duration":30,**

**"durationvalidation":"type=number; min=0; max=7200; required=true",**

**"validationerror":false**

**}**

## React

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

1. Reactor – The reactor associated with this component.
2. Reactor validation – Contains a string describing the reactor validation.
3. Position – The react position.
4. Position validation – Contains a string describing the position validation.
5. Duration – The length of the reaction in the seconds.
6. Duration validation – Contains a string describing the reactor validation.
7. Reaction temperature – The reaction temperature in Celsius.
8. Reaction temperature validation – Contains a string describing the reaction temperature validation.
9. Final temperature – The final temperature in Celsius.
10. Final temperature validation – Contains a string describing the final temperature validation.
11. Cooling delay – Number of seconds to continue cooling after the reactor reaches the final temperature.
12. Cooling delay validation – Contains a string describing the cooling temperature validation.
13. Stir speed – The stir speed in arbitrary units.
14. Stir speed validation – Contains a string describing the stir speed validation.
15. Stop at temperature flag – Flag that indicates if stir bar will stop once temperature is reached.
16. Stop at temperature flag validation – Contains a string descripting the stop at temperature flag validation.

**{**

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

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

**"id":20,**

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

**"sequenceid":1,**

**"reactor":1,**

**"reactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"reactiontemperature":160,**

**"reactiontemperaturevalidation":"type=number; min=20; max=200; required=true",**

**"position":1,**

**"positionvalidation":"type=enum-number; values=1,2; required=true",**

**"duration":900,**

**"durationvalidation":"type=number; min=0; max=7200; required=true",**

**"finaltemperature":25,**

**"finaltemperaturevalidation":"type=number; min=20; max=200; required=true",**

**"coolingdelay":60,**

**"coolingdelayvalidation":"type=number; min=0; max=7200; required=true",**

**"stirspeed":450,**

**"stirspeedvalidation":"type=number; min=0; max=5000; required=true"**

**"stopattemperature":0,**

**"stopattemperaturevalidation":"type=enum-number; values=0,1; required=true",**

**"validationerror":false,**

**}**

## Prompt

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

1. Message – Text to display to the user.
2. Message validation – Contains a string describing the message validation.

**{**

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

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

**"id":5,**

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

**"sequenceid":1,**

**"message":"Deliver F-18 from the cyclotron to vial",**

**"messagevalidation":"type=string; required=true",**

**"validationerror":false,**

**}**

## Install

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

1. Reactor – The reactor associated with this component.
2. Reactor validation – Contains a string describing the reactor validation.
3. Message – Text to display to the user.
4. Message validation – Contains a string describing the message validation.

**{**

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

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

**"id":9,**

**"name":"Install"**

**"sequenceid":1,**

**"reactor":1,**

**"reactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"message":"Take activity measurement of the vial in reactor 1",**

**"messagevalidation":"type=string; required=true",**

**"validationerror":false,**

**}**

## Comment

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

1. Comment – User-specified comment.
2. Comment validation – Contains a string describing the comment validation.

**{**

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

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

**“id”:1,**

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

**"validationerror": false,**

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

**"commentvalidation":"type=string"**

**}**

## Trap F18

Describes the trap F18 unit operation. Contains the following in addition to the base data members above:

1. Reactor – The reactor associated with this component.
2. Cyclotron flag – Flag indicating if a cyclotron will push the F18 (1) or if the system will push the F18 (0).
3. Cyclotron flag validation – Contains a string describing the cyclotron flag validation.
4. Trap time – The length to trap in seconds. Not used when the cyclotron flag is set.
5. Trap time validation – Contains a string describing the trap time validation.
6. Trap pressure – Nitrogen pressure in PSI to use when trapping. This value will override the default if set or ignored if zero. Not used when the cyclotron flag is set.
7. Trap pressure validation – Contains a string describing the trap pressure validation.

**{**

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

**"componenttype":"TRAPF18",**

**"id":6,**

**"name":"Trap F18",**

**"sequenceid":1,**

**"reactor":1,**

**“cyclotronflag”:0,**

**"cyclotronflagvalidation":"type=enum-number; values=0,1; required=true",**

**"traptime":60,**

**"traptimevalidation":"type=number; min=0; max=7200; required=true",**

**"trappressure":5,**

**"trappressurevalidation":"type=number; min=0; max=25",**

**"validationerror":false**

**}**

## Elute F18

Describes the elute F18 unit operation. Contains the following in addition to the base data members above:

1. Reactor – The reactor associated with this component.
2. Elute time – The length to elute in seconds.
3. Elute time validation – Contains a string describing the elute time validation.
4. Elute pressure – Nitrogen pressure in PSI to use when eluting. This value will override the default if set or ignored if zero.
5. Elute pressure validation – Contains a string describing the elute pressure validation.
6. Reagent – The reagent to use to elute the F18.
7. Reagent validation – Contains a string describing the reagent validation. Included in this string are the IDs of the other reagents that the user can choose from in Edit mode.

**{**

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

**"componenttype":"ELUTEF18",**

**"id":6,**

**"name":"Elute F18",**

**"sequenceid":1,**

**"reactor":1,**

**"elutetime":180,**

**"elutetimevalidation":"type=number; min=0; max=7200; required=true",**

**"elutepressure":5,**

**"elutepressurevalidation":"type=number; min=0; max=25",**

**"reagent":(Reagent details are described above),**

**"reagentvalidation":"type=enum-reagent; values=1,2,3,4,5,6,7,8; required=true",**

**"validationerror":false**

**}**

## Initialize

Describes the initialize unit operation. Contains no additional fields beyond the base data members described above.

## Mix

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

1. Reactor – The reactor associated with this component.
2. Reactor validation – Contains a string describing the reactor validation.
3. Mix time – The length to mix in seconds.
4. Mix time validation – Contains a string describing the mix time validation.
5. Stir speed – The stir speed in arbitrary units.
6. Stir speed validation – Contains a string describing the stir speed field.

**{**

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

**"componenttype":"MIX",**

**"id":24,**

**"name":"Mix",**

**"sequenceid":1,**

**"reactor":1,**

**"reactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"mixtime":20,**

**"mixtimevalidation":"type=number; min=0; max=7200; required=true",**

**"stirspeed":450,**

**"stirspeedvalidation":"type=number; min=0; max=5000; required=true",**

**"validationerror":false**

**}**

## Move

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

1. Reactor – The reactor associated with this component.
2. Reactor validation – Contains a string describing the reactor validation.
3. Position – The position to move to.
4. Position validation – Contains a string describing the position validation.

**{**

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

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

**"id":34,**

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

**"sequenceid":1,**

**"reactor":2,**

**"reactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"position":"Add",**

**"positionvalidation":"type=enum-string;**

**values=Install,Transfer,React1,Add,React2,Evaporate; required=true",**

**"validationerror":false**

**}**

## ExternalAdd

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

1. Reactor – The reactor associated with this component.
2. Reactor validation – Contains a string describing the reactor validation.
3. Reagent name – A string describing the reagent to add to the reactor.
4. Reagent name validation – Contains a string describing the reagent validation.
5. Message – Contains a message to be displayed to the user.

**{**

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

**"componenttype":"EXTERNALADD",**

**"id":11,**

**"name":"Add HBr"**

**"sequenceid":1,**

**"reactor":1,**

**"reactorvalidation":"type=enum-number; values=1,2,3; required=true",**

**"reagentname":"HBr",**

**"reagentvalidation":"type=string; required=true",**

**"message":"Externally add HBr to reactor 2.",**

**"validationerror":false,**

**}**

## Summary

Describes the summary unit operation. This unit operation is special in that it is only generated by the server and cannot be created or edited by the client. Contains the following in addition to the base data members:

1. Success flag – The reactor associated with this component.
2. Message – Contains a string describing the final run result.

**{**

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

**"componenttype":"SUMMARY",**

**"id":11,**

**"name":"Summary"**

**"sequenceid":1,**

**"successflag":0,**

**"message":"Failed to communicate with PLC",**

**"validationerror":false,**

**}**

# GET /sequence/[*Sequence ID*]

This operation returns the following for the given sequence ID:

1. Sequence metadata – Details of the sequence.
2. Sequence components – Array of the sequence components.

**{**

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

**"metadata":**

**{**

**(Sequence metadata are described above)**

**},**

**"components":**

**{**

**(Sequence components are described above)**

**}**

**}**

# 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 1*].[*Reagent ID 2*]

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

**{**

**“type”:”reagents”,**

**“reagents”:**

**[**

**(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"**

**}**

**}**

## SELECT

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

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

**POST /SELECT**

**{**

**"action":**

**{**

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

**"targetid":"BACK"**

**},**

**“sequenceid”:9000**

**}**

## VIEW

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 /VIEW**

**{**

**"action":**

**{**

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

**"targetid":"BACK"**

**}**

**}**

## EDIT

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 /EDIT**

**{**

**"action":**

**{**

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

**"targetid":"BACK"**

**}**

**}**

## RUN

The Run Sequence page does not send any additional information.

**POST /RUN**

**{**

**"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 component

Used by the client when updating or creating an individual sequence component on the server. Contains the following base fields plus additional component-specific detail documented in the sections below:

1. Component Type:
   1. ADD
   2. EVAPORATE
   3. TRANSFER
   4. REACT
   5. PROMPT
   6. INSTALL
   7. COMMENT
   8. TRAPF18
   9. ELUTEF18
   10. INITIALIZE
   11. MIX
   12. MOVE
   13. EXTERNALADD
   14. SUMMARY
2. ID – Unique ID of the component obtained from the server when updating an existing component or zero for a new component.
3. Name – The display name of the component. This field is optional and can be left blank.

## Add

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

1. Reactor – The reactor where the reagent will be added.
2. Reagent – The ID of the reagent to add to the reactor.
3. Delivery position – Reagent delivery position. Possible values are 1 and 2.
4. Delivery time – Time to deliver the reagent in seconds. This value will override the default if nonzero.
5. Delivery pressure – Pressure in PSI to use when delivering the reagent. This value will override the default if nonzero.

**{**

**“type”:”component”,**

**“componenttype”:”ADD”,**

**“id”:1,**

**“name”:”Add K222”,**

**“reactor”:1,**

**“reagent”:14,**

**"deliveryposition":2,**

**"deliverytime":0,**

**"deliverypressure":0**

**}**

## Evaporate

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

1. Reactor – The reactor that will be evaporated.
2. Duration – The length of the evaporation in seconds.
3. Evaporation temperature – The evaporation temperature in Celsius.
4. Final temperature – The final temperature in Celsius.
5. Stir speed – The stir speed in arbitrary units.
6. Evaporation pressure – Nitrogen pressure in PSI to use when evaporating. This value will override the default if nonzero.

**{**

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

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

**“id”:1,**

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

**“reactor”:1,**

**"duration":600,**

**"evaporationtemperature":165.0,**

**"finaltemperature":35.0,**

**"stirspeed":500,**

**“evaporationpressure”:10,**

**}**

## Transfer

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

1. Source reactor – The source reactor associated with this component.
2. Target reactor – The target reagent associated with this component.
3. Mode – The transfer mode, either “Trap” or “Elute”.
4. Pressure – Nitrogen pressure in PSI to use when transferring. This value will override the default if nonzero.
5. Duration – Duration of the transfer in seconds.

**{**

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

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

**“id”:15,**

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

**“sourcereactor”:1,**

**"targetreactor":2,**

**"mode":"Trap",**

**"pressure":3,**

**"duration":30**

**}**

## React

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

1. Reactor – The reactor associated with this component.
2. Position – The react position.
3. Duration – The length of the reaction in seconds.
4. Reaction temperature – The reaction temperature in Celsius.
5. Final temperature – The final temperature in Celsius.
6. Cooling delay – The number of seconds to keep cooling after reaching the final temperature.
7. Stir speed – The stir speed in arbitrary units.

**{**

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

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

**“id”:1,**

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

**“reactor”:1,**

**"position":1,**

**"duration":300,**

**"reactiontemperature":165.0,**

**"finaltemperature":35.0,**

**“coolingdelay”:60,**

**"stirspeed":500**

**}**

## Prompt

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

1. Message – Text to display to the user.

**{**

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

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

**“id”:1,**

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

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

**}**

## Install

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

1. Reactor – The reactor associated with this component.
2. Message – Text to display to the user.

**{**

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

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

**“id”:1,**

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

**“reactor”:1,**

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

**}**

## Comment

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

1. Comment – User-specified comment.

**{**

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

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

**“id”:1,**

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

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

**}**

## Trap F18

Describes the trap F18 unit operation. Contains the following in addition to the base data members above:

1. Cyclotron flag – 1 if a cyclotron will push the F18 or 0 to have the system push.
2. Trap time – The length to trap in seconds. Ignored if the cyclotron flag is set.
3. Trap pressure – Nitrogen pressure in PSI to use when trapping. This value will override the default if set or ignored if zero. Will also be ignored if the cyclotron flag is set.

**{**

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

**"componenttype":"TRAPF18",**

**"id":6,**

**"name":"Trap F18",**

**“cycletronflag”:0,**

**"traptime":60,**

**"trappressure":5**

**}**

## Elute F18

Describes the elute F18 unit operation. Contains the following in addition to the base data members above:

1. Elute time – The length to elute in seconds.
2. Elute pressure – Nitrogen pressure in PSI to use when eluting. This value will override the default if set or ignored if zero.
3. Reagent – The ID of the reagent to add to the reactor.

**{**

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

**"componenttype":"ELUTEF18",**

**"id":6,**

**"name":"Elute F18",**

**"elutetime":180,**

**"elutepressure":5,**

**"reagent":4**

**}**

## Initialize

Describes the initialize unit operation. Contains no additional fields beyond the base data members described above.

## Mix

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

1. Reactor – The reactor associated with this component.
2. Mix time – The length to mix in seconds.
3. Stir speed – The stir speed in arbitrary units.

**{**

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

**"componenttype":"MIX",**

**"id":24,**

**"name":"Mix",**

**"reactor":1,**

**"mixtime":20,**

**"stirspeed":450**

**}**

## Move

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

1. Reactor – The reactor associated with this component.
2. Position – The position to move to.

**{**

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

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

**"id":34,**

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

**"reactor":2,**

**"position":"Add”**

**}**

## ExternalAdd

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

1. Reactor – The reactor associated with this component.
2. Reagent name – A string describing the reagent to add to the reactor.

**{**

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

**"componenttype":"EXTERNALADD",**

**"id":34,**

**"name":"External Add",**

**"reactor":2,**

**"reagentname”:”HBr”**

**}**

# POST /sequence/[*Sequence ID*]

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

**{**

**(Component details are described above)**

**}**

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

This operation is used by the client to insert the given component in the sequence at the insertion ID. Insert a new component by setting the component ID to zero and passing the new component in the body of the request. Move an existing component by specifying the ID of the component to move in which case the server will ignore anything passed in the body of the request.

**{**

**(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 from a sequence.