RESTful JSON API

We include the following overview of our application programming interface in order to explain the calls and interfaces that we are employing in our program. This is aimed to give a comprehensive overview of all calls and explain their meanings.

Objects

appState

Description:

appState is what the server sends to the client in order to notify the client of the current state of the application.

Form:

```
appState={
        simulation:simulation
        device:device
        simulations_list:[ simulation_description ]
}
```

Explanation:

1. simulation is a simulation object
2. device is the device object which belongs to the current user
3. simulation_list is an array of simulation_description objects, each object contains the name of a simulation, the number of devices in that simulation and number of networks.

simulation_description

Description:

A bare-bones description of a of the simulations in the application used for viewing before registering to one.

Form:

```
simulation_description={
        simulation_id : string
        simulation_name: string
        num_devices: int
        num_networks: int
}
```

Explanation:

1. simulation_id : the id of the simulation object
2. simulation_name: the name of this simulation
3. num_devices: the number of devices currently in this simulation
4. num_networks: the number of networks currently in this simulation

simulation

Description:

An object which describes a simulation.

Form:

```
simulation={
        _id: string //the unique ID of the simulation
        num_devices: int
        num_networks: int
        simulation_population: int
        simulation_name: string
        tokenMethod: string
        partition_list: [partition]
        apps : [app_spec]
        rdts: [rdt_spec]
        activity_logs: String
        }
```

Explanation:

1. _id: the unique identifier of this simulation
2. num_devices: the number of devices in this simulation
3. num_networks: the number of networks in this simulation
4. simulation_population: the number of verified users in this simulation
5. simulation_name: the name of this simulation
6. tokenMethod: The method by which tokens are propagated to users
7. partition_list: Holds an array of all partitions, which hold a list of networks
8. apps: An array of application specifications imported into the simulation
9. rdts: An array of rdt specifications imported into the simulation
10. activity_logs: holds all of the activities which occurred on the server.

Partition

Description:

Holds a list of all partitions in the simulation

Form:

```
partition={
        _id: string,
        partition_name: string,
        network_list: [Network]
}
```

Explanation:

1. _id: the unique identifier for this partition
2. network_list: an array of all Network objects

Network

Description:

Represents a network within the simulation
Form:
Network={
    _id: string
    network_name: string
    network_type: string
    partition: string
    device_list:[device]
}
Explanation:

1. _id: the unique identifier of this network
2. network_name: the name of this network
3. network_type: displays what kind of network this is. For example, wifi, GSM.
4. partition: The name of the partition which this device belongs to.
5. device_list: An array of all devices within this network

App_spec
Description:
A description of the application imported into the framework
Form:
app_spec={
    _id: string,
    name: string,
    description: string,
    version: string,
    main : string,
    rdt_list: [String]
}
Explanation:

1. _id: the unique identifier for this application specification
2. name: the name of the application
3. description: the description of what the application does
4. main: the main file of the application. That is the landing page for application
5. rdt_list: a list of the rdts the application would like to use, just a list of names

RDT_spec
Description:
A description of the application imported into the framework
Form:
rdt_spec={
    _id: string,
    name: string,
    description: string,

version: string,
main : string,
}

Explanation:

1. _id: the unique identifier for this RDT specification
2. name: the name of the RDT
3. description: the description of what the RDT does
4. version: the version number of the RDT
5. main: the main file of the RDT. That is the file for initializing the rdts

Device

Description:

The representation of a device within a simulation.

Form:

Device={
_id: string
token: string
email: string
verified: boolean
current_partition: string
current_network: string
registeredOn: string
admin: boolean
apps : [app_spec]
networks_created: [string]
current_simulation: string
current_device_name: string
activity: string
}

Explanation:

1. _id: the unique id of the device
2. token: the unique token assigned to this device. This is the unique identifier of this device
3. email: the email of the user using
4. verified: a boolean value specified whether this device has been verified with the simulation.
5. current_parition: the name of the current partition which this device is a member of .
6. current_network: the name of the network which this device is a member of.
7. registeredOn: The date which this device was verified on.
8. admin: whether the device is an admin or not.
9. apps : The list of apps the device has access to
10. networks_created: a list of the names of networks which this device has created.

11. current_simulation: the name of the current simulation which this device is a part of.
12. current_device_name: the name of this device
13. activity: the activity log of this device.

Simulation_history

Description:

A list of ‘snapshots’ of the simulation, one for each event which has occurred to be viewed, and accompanied by the logs.

Form:

```
States={
        simulation_id: string
        state: [ history_state  ]
}
```

Explanation:

1. simulation_id: the id of the simulation this states object represents
2. state : an array of state_objects

history_state

Description:

The state or ‘snapshot’ of what the simulation looks like at that point in time. Used to accompany viewing logs.

Form:

```
state_object ={
        timestamp: string
        simulation: simulation_object
}
```

Explanation:

1. timestamp: the timestamp of when this history_state was current
2. simulation: the simulation object recorded at this timestamp

event_queue_wrapper

Description:

The list of events which have occurred on the client since the previous sync which are to be sent to the server.

Form:

```
event_queue_wrapper={
        eventQueue:[event]
        token:string
        simulationName :string

}
```

Explanation:

1. eventQueue is an array of event objects (outlined below).
2. token is a string representing the unique token of that device
3. simulationName is a string representation of the name of the simulation which that device is a member of.

event

Description:

An event which occurred on the client side which must be handled by the server.

Form:

```
event={
        route: string
        event_data: {event_data_object}
        time_stamp: string
}
```

Explanation:

1. route is a string indicating how the router on the server side should handle this event. It is of the form "/type/event". The different routes are detailed below.
2. event_data is the information about the event to be handled by the server.
3. time_stamp is the time at which this event occurred.

file

Description:

A generic file. It contains a type, and data in the form of a text string.

Form:

```
file={
        name: string,
        type: ('RDT'/'APP'),
        data: string
}
```

Routes

The following is a list of all of the routes which are used to handle the events in the event queue passed to the server.

/create/Simulation

```
        event_data={
                admin_token: string
                num_devices: int
                num_networks: int
                simulation_population: 0? what is this?
                simulation_name: string
```

tokenMethod: string (We should just use email for now)
config_map: config_map
activity_logs: ‘’ why do we need this?
}

/create/Network

event_data={
network_name: string
simulation_id: string
}

/create/Device

event_data={
simulation_id: string
device_name: string
}

/move/Device/Network

event_data={
network_id: string
simulation_id: string
device_token: string
}

/move/Device/Freelist

event_data={
simulation_id: string
device_token: string
}

/merge/Partitions

event_data={
partition_a_id: string
partition_b_id: string
simulation_id: string
}

/authenticate/authToken

event_data={
token: string,
simulation_id: string
}

/deploy/App

event_data={
app_id: String
device_id : String
}

/launch/App

event_data={
app_id: String
device_id : String
}

/reverse/deploy/App

event_data={
app_id: String
device_id : String
}

/divide/Partitions

event_data={
partition_id: string
split_networks_list: [network_id]
simulation_id: string
}

Explanation:
partition_id: the unique id of the partition
split_networks: an array of unique_id's of networks which are being removed from this partition and put into a new partition
simulation_id: the unique id of this simulation

/upload/

event_data={
name: String,
spec: Object,
simulation_id : String
files:[**file**]
}

Explanation: This is called when a simulation admin wishes to upload an RDT, TestScript or an Application to their simulation. The server handles the file types and placing them in the proper directories.