Python and MATLAB API for NATS

Updated: 11.08, 2018

NATS Client API

No.	Type	Method and Description
1	EntityInterface	getEntityInterface() Returns a reference to the EntityInterface.
2	EnvironmentInterface	getEnvironmentInterface() Returns a reference to the EnvironmentInterface.
3	EquipmentInterface	getEquipmentInterface() Returns a reference to the EquipmentInterface.
4	SafetyMetricsInterface	getSafetyMetricsInterface() Returns a reference to the SafetyMetricsInterface.
5	SimulationInterface	getSimulationInterface() Returns a reference to the SimulationInterface.

SimulationInterface API

No.	Type	Method and Description
1	void	clear_trajectory() Clean up trajectory data.
2	float	get_curr_sim_time() Get the current simulation timestamp.
3	int	get_runtime_sim_status() Get the runtime status of the trajectory propagation. Value definition: NATS_SIMULATION_STATUS_READY = 0 NATS_SIMULATION_STATUS_START = 1 NATS_SIMULATION_STATUS_PAUSE = 2 NATS_SIMULATION_STATUS_RESUME = 3 NATS_SIMULATION_STATUS_STOP = 4 NATS_SIMULATION_STATUS_ENDED = 5 When the trajectory propagation finishes, the status will be changed to NATS_SIMULATION_STATUS_ENDED.
4	void	pause() Pause the trajectory propagation process.
5	void	resume() Resume the trajectory propagation process.
6	void	resume(long t_duration) Resume the trajectory propagation process and process data for certain seconds of duration time.
7	int	setupSimulation(long t_total_propagation_period, long t_step) Setup the trajectory propagation.

		Description of arguments: t_total_propagation_period: Total period of time of propagation in seconds. t_step: Time step of airborne traffic in seconds. For surface ground traffic, the propagation time step is set to 1 second.
8	void	start() Start the trajectory propagation process.
9	void	start(long t_duration) Start the trajectory propagation and process data for certain seconds of duration time.
10	void	startRealTime_singleUser() Start the real-time trajectory propagation in single-user mode. NATS Server runs trajectory propagation with 1-second time step using real-time clocking. External aircraft profile and state data can be sent from physical simulator to NATS Server. Please refer to the sample of XPlane simulator for the detail.
11	void	stop() Stop the trajectory propagation process.
12	void	write_trajectories(String output_file) Write trajectory data into file. File format supported: *.csv, *.kml, *.xml
13	int	public int externalSimulator_create_trajectory_profile_data(String ac_id, String ac_type, String origin_airport, String destination_airport, float cruise_altitude_ft, float cruise_tas_knots) Create trajectory profile data.
14	void	public void externalSimulator_inject_trajectory_state_data(String ac_id, double latitude_deg, double longitude_deg, double altitude_ft, double rocd_fps, double tas_knots, double tas_knots_ground, double course_deg, double fpa_deg, int sector_index, String flight_phase, long timestamp_utc_millisec) Inject trajectory state data.

Simulation Status Enum Values

Values

NATS_SIMULATION_STATUS_READY NATS_SIMULATION_STATUS_START NATS_SIMULATION_STATUS_PAUSE NATS_SIMULATION_STATUS_RESUME NATS_SIMULATION_STATUS_STOP NATS_SIMULATION_STATUS_ENDED

EquipmentInterface API

No.	Type	Method and Description
1	AircraftInterface	getAircraftInterface() Returns a reference to the AircraftInterface.

AircraftInterface API

No.	Type	Method and Description
1	int	load_aircraft(String trx_file, String mfl_file) Load aircraft data.
2	int	release_aircraft() Clean up aircraft data.
3	String[]	getAircraftIds(float minLatitude, float maxLatitude, float minLongitude, float maxLongitude, float minAltitude_ft, float maxAltitude_ft) Get qualified aircraft Id which satisfy the min/max range of latitude, longitude and/or altitude.
4	String[]	getAllAircraftId() Get complete aircraft Ids.
5	Aircraft	select_aircraft(String aircraft_id) Get an aircraft object by aircraft Id.
6	int	synchronize_aircraft_to_server(Aircraft aircraft) Push aircraft object to the server and synchronize the data. Return value indicating the server operation response. 0 means success. 1 means error.

Aircraft Instance API

No.	Type	Method and Description
1	int	delay_departure(int seconds) Postpone the departure time of the current aircraft for certain seconds. If the aircraft already departed, the departure time will not be changed.
2	String	getAcid() Get aircraft ID. Example: UA555
3	float	getAltitude_ft() Get current altitude in feet.
4	float	getCruise_alt_ft() Get cruise altitude in feet.

5	float	getCruise_tas_knots() Get cruise speed.
6	float	getDeparture_time_sec() Get departure time in second.
7	float	getDestination_airport_elevation_ft() Get elevation of the destination airport.
8	int	getFlight_phase() Get current flight phase. Flight phase is presented as integer type. Please refer to "Flight Phase Enum Values" for the detail value definition.
9	float[]	getFlight_plan_latitude_array() Get array of latitude of the flight plan.
10	int	getFlight_plan_length() Get number of records of the flight plan.
11	float[]	getFlight_plan_longitude_array() Get array of longitude of the flight plan.
12	String[]	getFlight_plan_waypoint_name_array() Get array of waypoint names of the flight plan.
13	String[ ]	getFlight_plan_alt_desc_array() Get array of flight plan altitude description.
14	double[ ]	getFlight_plan_alt_1_array() Get array of flight plan altitude 1.
15	double[ ]	getFlight_plan_alt_2_array() Get array of flight plan altitude 2.
16	double[ ]	getFlight_plan_speed_limit_array() Get array of flight plan speed limit.
17	String[ ]	getFlight_plan_speed_limit_desc_array() Get array of flight plan speed limit description.
18	float	getFpa_rad() Get flight path angle.
19	float	getCourse_rad() Get current course.
20	int	getLanded_flag() Get flag value indicating if the aircraft is landed.
21	float	getLatitude_deg() Get current latitude degree.
22	float	getLongitude_deg() Get current longitude degree.
23	float	getOrigin_airport_elevation_ft() Get elevation of the origin airport.
24	float	getRocd_fps() Get rate of climb or descent in feet per second.
25	int	getSector_index() Get current sector index.
26	int	getTarget_waypoint_index() Get array index of the flight plan data of the target waypoint.
27	String	getTarget_waypoint_name()

		Get target waypoint name.
28	float	getTas_knots() Get current speed.
29	int	getToc_index() Get the flight plan array index of the top-of-climb waypoint.
30	int	getTod_index() Get the flight plan array index of the top-of-descent waypoint.
31	void	setAltitude_ft(float altitude_ft) Set new value of altitude in feet.
32	void	setCruise_alt_ft(float cruise_alt_ft) Set new value of cruise altitude in feet.
33	void	setCruise_tas_knots(float cruise_tas_knots) Set new value of cruise speed.
34	void	setFlight_plan_latitude_deg(int index, float latitude_deg) Set latitude of the n-th waypoint.
35	void	setFlight_plan_longitude_deg(int index, float longitude_deg) Set longitude of the n-th waypoint.
36	void	setCourse_rad(float course_rad) Set new value of course.
37	void	setLatitude_deg(float latitude_deg) Set new value of latitude.
38	void	setLongitude_deg(float longitude_deg) Set new value of longitude.
39	void	setRocd_fps(float rocd_fps) Set new value of rate of climb or descent in feet per second.
40	void	setTarget_waypoint_latitude_deg(float latitude_deg) Set new value to target waypoint latitude.
41	void	setTarget_waypoint_longitude_deg(float longitude_deg) Set new value to target waypoint longitude.
42	void	setTas_knots(float tas_knots) Set new value of speed.

Flight Phase Enum Values

Values
FLIGHT_PHASE_ORIGIN_GATE FLIGHT_PHASE_PUSHBACK FLIGHT_PHASE_RAMP_DEPARTING FLIGHT_PHASE_TAXI_DEPARTING FLIGHT_PHASE_RUNWAY_THRESHOLD_DEPARTING FLIGHT_PHASE_TAKEOFF

FLIGHT_PHASE_CLIMBOUT

FLIGHT_PHASE_HOLD_IN_DEPARTURE_PATTERN

FLIGHT_PHASE_CLIMB_TO_CRUISE_ALTITUDE

FLIGHT_PHASE_TOP_OF_CLIMB

FLIGHT_PHASE_CRUISE

FLIGHT_PHASE_HOLD_IN_ENROUTE_PATTERN

FLIGHT_PHASE_TOP_OF_DESCENT

FLIGHT_PHASE_INITIAL_DESCENT

FLIGHT_PHASE_HOLD_IN_ARRIVAL_PATTERN

FLIGHT_PHASE_APPROACH

FLIGHT_PHASE_FINAL_APPROACH

FLIGHT_PHASE_GO_AROUND

FLIGHT_PHASE_TOUCHDOWN

FLIGHT_PHASE_LAND

FLIGHT_PHASE_EXIT_RUNWAY

FLIGHT_PHASE_TAXI_ARRIVING

FLIGHT_PHASE_RUNWAY_CROSSING

FLIGHT_PHASE_RAMP_ARRIVING

FLIGHT_PHASE_DESTINATION_GATE

FLIGHT_PHASE_LANDED

EnvironmentInterface API

No.	Type	Method and Description
1	void	load_rap(String wind_dir) Load wind RAP file.
2	int	release_rap()

		Clean up RAP data.
3	AirportInterface	getAirportInterface() Returns a reference to the AirportInterface.
4	TerrainInterface	getTerrainInterface() Returns a reference to the TerrainInterface.
5	TerminalAreaInterface	getTerminalAreaInterface() Returns a reference to the TerminalAreaInterface.
6	WeatherInterface	getWeatherInterface() Returns a reference to the WeatherInterface.

AirportInterface API

No.	Type	Method and Description
1	Airport	select_airport(String airport_code) Get an Airport object instance by a given airport code.
2	String	getArrivalAirport(String acid) Get arrival airport of the requested aircraft.
3	String	getDepartureAirport(String acid) Get departure airport of the requested aircraft.
4	double[]	getLocation(String airport_code) Get latitude and longitude of the requested airport. Return an array containing latitude and longitude.
5	String	getClosestAirport(double latitude, double longitude) Get airport code of the closest airport to the given position.
6	String[]	getAirportsWithinMiles(double lat_deg, double lon_deg, double miles) Get all airports within the given location and mile range.
7	String	getFullName(String airportid) Get full airport name of the given airport code.
8	Object[]	getAllRunways(String airport_code) Get all runway of the given airport. The returned data is an array. Each element is an array of: - Runway name - Waypoint Id
9	String[]	getRunwayExits(String airport_code, String runway_id) Get all runway exits of the given airport code and runway id.
10	Object[]	getLayout_node_map(String airport_code) Get mapping of node and sequence number of a given airport. The returned data is an array. Each array element is an array of: - Waypoint node Id - Node sequence number
11	Object[]	getLayout_node_data(String airport_code) Get waypoint node data of a given airport.

		The returned data is an array. Each array element is an array of: - Node sequence number - Latitude - Longitude
12	Object[]	getLayout_links(String airport_code) Get links of waypoint nodes of a given airport. The returned data is an array. Each array element is an array of: - Node 1 sequence number - Node 2 sequence number
13	String[]	getSurface_taxi_plan(String acid, String airport_code) Get surface taxi plan of the given aircraft Id and airport code. Return Array of all waypoint Ids in the order of visiting.
14	int	generate_surface_taxi_plan(String acid, String airport_code, String startNode_waypoint_id, String endNode_waypoint_id, String runway_name) Generate taxi plan and load it in the program. The function arguments: acid: Aircraft Id airport_code: Airport code startNode_waypoint_id: Starting waypoint Id endNode_waypoint_id: Ending waypoint Id runway_name: Name of runway Notice This function doesn't need to specify V2 or touchdown point as parameters. Return value 0 means success. 1 means error.
15	int	setUser_defined_surface_taxi_plan(String acid, String airport_code, String[] user_defined_waypoint_ids) Set user-defined surface taxi plan and load it in the program. Return value 0 means success. 1 means error.
16	String[]	get_taxi_route_from_A_To_B(String acid, String airport_code, String startNode_waypoint_id, String endNode_waypoint_id) Get generate taxi route from waypoint A to B. This function only returns an array of waypoint Ids. No taxi plan will be loaded in the program.
17	String	getDepartureRunway(String acid) Get departure runway of the given aircraft. If the departure taxi plan does not exist, no result will be returned.

18	String	getArrivalRunway(String acid) Get arrival runway of the given aircraft. If the arrival taxi plan does not exist, no result will be returned.
19	double	getTaxi_tas_knots(String acid) Get surface taxi speed in tas knots of the given aircraft.
20	void	setTaxi_tas_knots(String acid, double tas_knots) Set surface taxi speed in tas knots of the given aircraft.

Airport Instance API

No.	Type	Method and Description
1	String	getCode() Get airport code.
2	float	getElevation() Get elevation of the airport in feet.
3	float	getLatitude() Get latitude of the airport.
4	float	getLongitude() Get longitude of the airport.
5	String	getName() Get airport full name.

TerminalAreaInterface API

No.	Type	Method and Description
1	String[]	getAllApproaches(String airport_code) Get all Approach procedures of the given airport.
2	String[]	getAllSids(String airport_code) Get all SID procedures of the given airport.
3	String[]	getAllStars(String airport_code) Get all STAR procedures of the given airport.
4	String	getCurrentApproach(String acid) Get current Approach procedure of the given airport on the given aircraft flight.
5	String	getCurrentSid(String acid) Get current SID procedure of the given airport on the given aircraft flight.
6	String	getCurrentStar(String acid) Get current STAR procedure of the given airport on the given aircraft flight.
7	String[]	getProcedure_leg_names(String proc_type, String proc_name, String airport_code) Get leg names of the given airport code, procedure type and procedure name. Arguments: proc_type: Procedure type. The valid values are only limited to "SID", "STAR" and "APPROACH". proc_name: Name of procedure. airport_code: Airport code.

8	String[]	getWaypoints_in_procedure_leg(String proc_type, String proc_name, String airport_code, String proc_leg_name) Get waypoints of the given airport code, procedure type, procedure name and leg name. Arguments: proc_type: Procedure type. The valid values are only limited to "SID", "STAR" and "APPROACH". proc_name: Name of procedure. airport_code: Airport code. proc_leg_name: Name of procedure leg.
9	double[]	getWaypoint_Latitude_Longitude_deg(String waypoint_name) Get latitude and longitude degree of a given waypoint.
10	double	getProcedure_alt_1(String proc_type, String proc_name, String airport_code, String proc_leg_name, String proc_wp_name) Get alt 1 value of the given airport code, procedure type, procedure name, leg name and waypoint name.
11	double	getProcedure_alt_2(String proc_type, String proc_name, String airport_code, String proc_leg_name, String proc_wp_name) Get alt 2 value of the given airport code, procedure type, procedure name, leg name and waypoint name.
12	double	getProcedure_speed_limit(String proc_type, String proc_name, String airport_code, String proc_leg_name, String proc_wp_name) Get speed limit of the given airport code, procedure type, procedure name, leg name and waypoint name.
13	String	getProcedure_alt_desc(String proc_type, String proc_name, String airport_code, String proc_leg_name, String proc_wp_name) Get alt description of the given airport code, procedure type, procedure name, leg name and waypoint name.
14	String	getProcedure_speed_limit_desc(String proc_type, String proc_name, String airport_code, String proc_leg_name, String proc_wp_name) Get speed limit description of the given airport code, procedure type, procedure name, leg name and waypoint name.

TerrainInterface API

No.	Type	Method and Description
1	double	getElevation(double latDeg, double lonDeg) Returns the terrain elevation (in feet above sea level) at the specified latitude and longitude (degrees).
2	double[]	getElevationAreaStats(double minLatDeg, double maxLatDeg, double minLonDeg, double maxLonDeg) Returns an array of coarse statistical information calculated from using terrain elevation data for the specified region.
3	double[]	getElevationAreaStatsM(double minLatDeg, double maxLatDeg, double minLonDeg, double maxLonDeg)

		Returns an array of coarse statistical information calculated from using terrain elevation data for the specified region.
4	double	getElevationM(double latDeg, double lonDeg) Returns the terrain elevation (in meters above sea level) at the specified latitude and longitude (degrees).
5	double[]	getElevationMapBounds() Returns the minimum and maximum latitude and longitude bounds of the data used to interpolate elevation data.
6	double[]	getElevationMapBoundsRad() Returns the minimum and maximum latitude and longitude bounds of the data used to interpolate elevation data.
7	int	getElevationMapHeight() Returns the height (in pixels) of the image map which is used to interpolate terrain elevation data.
8	int	getElevationMapWidth() Returns the width (in pixels) of the image map which is used to interpolate terrain elevation data.
9	double	getElevationRad(double latRad, double lonRad) Returns the terrain elevation (in feet above sea level) at the specified latitude and longitude (radians).

EntityInterface API

No.	Type	Method and Description
1	ControllerInterface	getControllerInterface() Returns a reference to the ControllerInterface.
2	PilotInterface	getPilotInterface() Returns a reference to the PilotInterface.

ControllerInterface API

No.	Type	Method and Description
1	int	setDelayPeriod(String acid, AircraftClearance aircraft_clearance, int seconds) Set delay period in seconds to a given aircraft clearance.

PilotInterface API

No.	Type	Method and Description
1	int	int setActionRepeat(String aircraftID, String repeatParameter) Repeat pilot action, based on the repeatParameter value. repeatParameter can have following values: 1. AIRSPEED 2. VERTICAL_SPEED 3. COURSE

2	int	int skipFlightPhase(String aircraftID, String flightPhase) Ignore flight phase transition, by skipping the mentioned flight phase. flightPhase can have any of the Flight Phase Enum Values. Eg. FLIGHT_PHASE_CLIMB_TO_CRUISE_ALTITUDE
3	int	int setWrongAction(String aircraftID, String originalChangeParameter, String wrongChangeParameter) Set the value of one parameter, erroneously to another. For example, the pilot can set magnitude of airspeed (170 kts) as course angle (170 degrees). These are following pairs of parameters that can be mutually interchanged: 1. AIRSPEED – COURSE 2. FLIGHT_LEVEL – AIRSPEED 3. COURSE – FLIGHT_LEVEL
4	int	int setActionReversal(String aircraftID, String changeParameter) Reverse a pilot action, by reverting the value of changeParameter. changeParameter can have following values: 1. AIRSPEED 2. VERTICAL_SPEED 3. COURSE
5	int	int setPartialAction(String aircraftID, String changeParameter, float originalTarget, float percentage) Execute only part of an action, by providing the original target value of parameter, and percentage of it to be performed by pilot, for the changeParameter. changeParameter can have following values: 1. AIRSPEED 2. VERTICAL_SPEED 3. COURSE
6	int	int skipChangeAction(String aircraftID, String skipParameter) Omit parameter change, by continuing to maintain current value for skipParameter. skipParameter can have following values: 1. AIRSPEED 2. VERTICAL_SPEED 3. COURSE
7	int	int setActionLag(String aircraftID, String lagParameter, float lagTimeConstant, float percentageError, float parameterTarget) Lag pilot action, by reaching certain percent of execution within a given time period. Following are the parameters: lagParameter: Paremeter to be lagged, can have following values: 1. AIRSPEED

		2. VERTICAL_SPEED 3. COURSE lagTimeConstant: To be provided in seconds. Eg. 10 seconds. percentageError: Error percentage for lag. For example, if 95% of the action is to be executed, percentage error would be 0.05. parameterTarget: Original parameter value to be reached.
8	Int	Int setFlightPlanReadError(String aircraftID, String errorParameter, float correctValue) If simulation has not started, the flight plan read from TRX can be changed using this function. This constitutes to error in reading the flight plan. Following are the parameters: errorParameter: Parameter with erroneous data. It can have any of the following values: 1. AIRSPEED 2. VERTICAL_SPEED 3. COURSE correctValue: This is the correct flight plan data that should have ideally be read.

AircraftClearance Enum Values

Values
AIRCRAFT_CLEARANCE_PUSHBACK
AIRCRAFT_CLEARANCE_TAXI_DEPARTING
AIRCRAFT_CLEARANCE_TAKEOFF
AIRCRAFT_CLEARANCE_ENTER_ARTC
AIRCRAFT_CLEARANCE_DESCENT_FROM_CRUISE
AIRCRAFT_CLEARANCE_ENTER_TRACON
AIRCRAFT_CLEARANCE_APPROACH
AIRCRAFT_CLEARANCE_TOUCHDOWN
AIRCRAFT_CLEARANCE_TAXI_LANDING
AIRCRAFT_CLEARANCE_RAMP_LANDING