Node js code:

var soap = require('soap');

var url = 'http://sstracking.com/Services/WCF/V04/SSTracking.svc?wsdl';

var args = {"UserName":"aaron",

"Password":"Aaron@2018"};

exports.finduserdetails = (req,res) => {

soap.createClient(url,function (err, client) {

client.setSecurity(new soap.BasicAuthSecurity('aaron', 'aaron@2018'));

console.log(client);

client.ValidateLogin(args,function(err,result){

console.log(result);

})

});

res.render('pages/profile')

}

**Output of console.log(result):**

{ ValidateLoginResult:

{ ErrorMessage: 'Object reference not set to an instance of an object.',

IsValidUser: 'false' } }

**Output of console.log(client):**

Client {

domain: null,

\_events: {},

\_eventsCount: 0,

\_maxListeners: undefined,

wsdl:

WSDL {

uri: 'http://sstracking.com/Services/WCF/V04/SSTracking.svc?wsdl',

callback: [Function],

\_includesWsdl: [ [Object], [Object], [Object], [Object] ],

WSDL\_CACHE:

{ 'http://sstracking.com/Services/WCF/V04/SSTracking.svc?wsdl': [Circular],

'http://sstracking.com/Services/WCF/V04/SSTracking.svc?xsd=xsd0': [Object],

'http://sstracking.com/Services/WCF/V04/SSTracking.svc?xsd=xsd2': [Object],

'http://sstracking.com/Services/WCF/V04/SSTracking.svc?xsd=xsd1': [Object],

'http://sstracking.com/Services/WCF/V04/SSTracking.svc?xsd=xsd3': [Object] },

\_originalIgnoredNamespaces: undefined,

options:

{ ignoredNamespaces: [Array],

valueKey: '$value',

xmlKey: '$xml',

escapeXML: true,

returnFault: false,

handleNilAsNull: false,

namespaceArrayElements: true,

wsdl\_headers: undefined,

wsdl\_options: undefined,

ignoreBaseNameSpaces: false,

forceSoap12Headers: false,

customDeserializer: undefined,

useEmptyTag: false,

attributesKey: 'attributes',

envelopeKey: 'soap',

preserveWhitespace: false },

definitions:

subElement {

xmlns: [Object],

schemaXmlns: {},

valueKey: '$value',

xmlKey: '$xml',

ignoredNamespaces: [Array],

'$name': 'SSTracking',

'$targetNamespace': 'http://tempuri.org/',

messages: [Object],

portTypes: [Object],

bindings: [Object],

services: [Object],

schemas: [Object],

descriptions: [Object] },

xml: '’,

services: { SSTracking: [Object] },

xmlnsInEnvelope: ' xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:tns="http://tempuri.org/" xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"' },

streamAllowed: undefined,

normalizeNames: undefined,

GenerateTrip: [Function],

CancelTrip: [Function],

ValidateLogin: [Function],

PoiDetails: [Function],

UpdatePOI: [Function],

CUS\_Digital\_Trips: [Function],

CUS\_Digital\_Passenger\_List: [Function],

SSTracking:

{ BasicHttpBinding\_ISSTracking:

{ GenerateTrip: [Function],

CancelTrip: [Function],

ValidateLogin: [Function],

PoiDetails: [Function],

UpdatePOI: [Function],

CUS\_Digital\_Trips: [Function],

CUS\_Digital\_Passenger\_List: [Function] } },

httpClient:

HttpClient {

\_request:

{ [Function: request]

get: [Function],

head: [Function],

options: [Function],

post: [Function],

put: [Function],

patch: [Function],

del: [Function],

delete: [Function],

jar: [Function],

cookie: [Function],

defaults: [Function],

forever: [Function],

Request: [Object],

initParams: [Function: initParams],

debug: [Getter/Setter] } },

GenerateTripAsync: [Function: ret],

CancelTripAsync: [Function: ret],

ValidateLoginAsync: [Function: ret],

PoiDetailsAsync: [Function: ret],

UpdatePOIAsync: [Function: ret],

CUS\_Digital\_TripsAsync: [Function: ret],

CUS\_Digital\_Passenger\_ListAsync: [Function: ret],

addSoapHeaderAsync: [Function: ret],

changeSoapHeaderAsync: [Function: ret],

getSoapHeadersAsync: [Function: ret],

clearSoapHeadersAsync: [Function: ret],

addHttpHeaderAsync: [Function: ret],

getHttpHeadersAsync: [Function: ret],

clearHttpHeadersAsync: [Function: ret],

addBodyAttributeAsync: [Function: ret],

getBodyAttributesAsync: [Function: ret],

clearBodyAttributesAsync: [Function: ret],

setEndpointAsync: [Function: ret],

describeAsync: [Function: ret],

setSecurityAsync: [Function: ret],

setSOAPActionAsync: [Function: ret],

setMaxListenersAsync: [Function: ret],

getMaxListenersAsync: [Function: ret],

emitAsync: [Function: ret],

addListenerAsync: [Function: ret],

onAsync: [Function: ret],

prependListenerAsync: [Function: ret],

onceAsync: [Function: ret],

prependOnceListenerAsync: [Function: ret],

removeListenerAsync: [Function: ret],

removeAllListenersAsync: [Function: ret],

listenersAsync: [Function: ret],

listenerCountAsync: [Function: ret],

eventNamesAsync: [Function: ret],

security: BasicAuthSecurity { \_username: 'aaron', \_password: 'aaron@2018', defaults: {} } }

**New API Output: (The above highlighted function missing in the new output. The new output doesn’t have a single function or operation to execute)**

Client {

domain: null,

\_events: {},

\_eventsCount: 0,

\_maxListeners: undefined,

wsdl:

WSDL {

uri: 'http://www.sstracking.com/Services/Mobile/V09/TrackingService.svc?wsdl',

callback: [Function],

\_includesWsdl: [ [Object], [Object], [Object], [Object] ],

WSDL\_CACHE:

{ 'http://www.sstracking.com/Services/Mobile/V09/TrackingService.svc?wsdl': [Circular],

'http://www.sstracking.com/Services/Mobile/V09/TrackingService.svc?xsd=xsd0': [Object],

'http://www.sstracking.com/Services/Mobile/V09/TrackingService.svc?xsd=xsd2': [Object],

'http://www.sstracking.com/Services/Mobile/V09/TrackingService.svc?xsd=xsd3': [Object],

'http://www.sstracking.com/Services/Mobile/V09/TrackingService.svc?xsd=xsd1': [Object] },

\_originalIgnoredNamespaces: undefined,

options:

{ ignoredNamespaces: [Array],

valueKey: '$value',

xmlKey: '$xml',

escapeXML: true,

returnFault: false,

handleNilAsNull: false,

namespaceArrayElements: true,

wsdl\_headers: undefined,

wsdl\_options: undefined,

ignoreBaseNameSpaces: false,

forceSoap12Headers: false,

customDeserializer: undefined,

useEmptyTag: false,

attributesKey: 'attributes',

envelopeKey: 'soap',

preserveWhitespace: false },

definitions:

subElement {

xmlns: [Object],

schemaXmlns: {},

valueKey: '$value',

xmlKey: '$xml',

ignoredNamespaces: [Array],

'$name': 'TrackingService',

'$targetNamespace': 'http://tempuri.org/',

messages: [Object],

portTypes: [Object],

bindings: {},

services: [Object],

schemas: [Object],

descriptions: [Object] },

xml: '',

services: { TrackingService: [Object] },

xmlnsInEnvelope: ' xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:tns="http://tempuri.org/" xmlns:msc="http://schemas.microsoft.com/ws/2005/12/wsdl/contract"' },

streamAllowed: undefined,

normalizeNames: undefined,

TrackingService: {},

httpClient:

HttpClient {

\_request:

{ [Function: request]

get: [Function],

head: [Function],

options: [Function],

post: [Function],

put: [Function],

patch: [Function],

del: [Function],

delete: [Function],

jar: [Function],

cookie: [Function],

defaults: [Function],

forever: [Function],

Request: [Object],

initParams: [Function: initParams],

debug: [Getter/Setter] } },

addSoapHeaderAsync: [Function: ret],

changeSoapHeaderAsync: [Function: ret],

getSoapHeadersAsync: [Function: ret],

clearSoapHeadersAsync: [Function: ret],

addHttpHeaderAsync: [Function: ret],

getHttpHeadersAsync: [Function: ret],

clearHttpHeadersAsync: [Function: ret],

addBodyAttributeAsync: [Function: ret],

getBodyAttributesAsync: [Function: ret],

clearBodyAttributesAsync: [Function: ret],

setEndpointAsync: [Function: ret],

describeAsync: [Function: ret],

setSecurityAsync: [Function: ret],

setSOAPActionAsync: [Function: ret],

setMaxListenersAsync: [Function: ret],

getMaxListenersAsync: [Function: ret],

emitAsync: [Function: ret],

addListenerAsync: [Function: ret],

onAsync: [Function: ret],

prependListenerAsync: [Function: ret],

onceAsync: [Function: ret],

prependOnceListenerAsync: [Function: ret],

removeListenerAsync: [Function: ret],

removeAllListenersAsync: [Function: ret],

listenersAsync: [Function: ret],

listenerCountAsync: [Function: ret],

eventNamesAsync: [Function: ret],

security: BasicAuthSecurity { \_username: 'aaron', \_password: 'aaron@2018', defaults: {} } }