const rules = [

{

condition: (id) => true,

attribute: 'minNumberOfAccountsAllowedDocuSign',

expected: 1,

message: 'In all action display IDs, the attribute name "minNumberOfAccountsAllowedDocuSign" should have value 1'

},

{

condition: (id) => id === 1060,

attribute: 'maxNumberOfAccountsAllowedDocuSign',

expected: 6,

message: 'For action display ID 1060, the attribute name "maxNumberOfAccountsAllowedDocuSign" should have value 6'

},

{

condition: (id) => id !== 1060,

attribute: 'maxNumberOfAccountsAllowedDocuSign',

expected: 1,

message: 'For all action display IDs except 1060, the attribute name "maxNumberOfAccountsAllowedDocuSign" should have value 1'

},

{

condition: (id) => true,

attribute: 'minNumberOfAccountsAllowedDigital',

expected: 1,

message: 'In all action display IDs, the attribute name "minNumberOfAccountsAllowedDigital" should have value 1'

},

{

condition: (id) => id === 1060,

attribute: 'maxNumberOfAccountsAllowedDigital',

expected: 12,

message: 'For action display ID 1060, the attribute name "maxNumberOfAccountsAllowedDigital" should have value 12'

},

{

condition: (id) => id !== 1060,

attribute: 'maxNumberOfAccountsAllowedDigital',

expected: 1,

message: 'For all action display IDs except 1060, the attribute name "maxNumberOfAccountsAllowedDigital" should have value 1'

}

];

// Normalize the response structure

const rawActions = Array.isArray(responseBody) ? responseBody : [responseBody];

const actions = rawActions.flatMap(obj =>

Array.isArray(obj) ? obj : Object.values(obj).filter(v => typeof v === 'object' && v?.attributes)

);

let messages = [];

let errors = [];

rules.forEach(rule => {

actions.forEach(action => {

const id = action.actionId;

if (!rule.condition(id)) return;

const attrs = action.attributes || {};

const actualValue = attrs[rule.attribute];

if (actualValue === undefined) {

messages.push(`❌ ActionId ${id}: attribute "${rule.attribute}" is missing. ${rule.message}`);

errors.push(`ActionId ${id} missing attribute "${rule.attribute}"`);

} else if (actualValue !== rule.expected) {

messages.push(`❌ ActionId ${id}: attribute "${rule.attribute}" has value ${actualValue}, expected ${rule.expected}. ${rule.message}`);

errors.push(`ActionId ${id} - Expected ${rule.expected} but got ${actualValue} for "${rule.attribute}"`);

} else {

messages.push(`✅ ActionId ${id}: attribute "${rule.attribute}" has correct value ${actualValue}. ${rule.message}`);

}

});

});

// Output all messages

messages.forEach(msg => console.log(msg));

// Final check

if (errors.length > 0) {

throw new Error(`Validation failed for ${errors.length} cases:\n` + errors.join('\n'));

}