

Validating XML and JSON API schemas

NetYCE supports its API's using two methods. The original implementation used strictly XML payloads supporting http 'post' or ip-socket based transactions over port 8888.

As of version 7.0, NetYCE extended its internal JSON API (for the web front-end) to include the API calls of the original implementation. The JSON based interface used port 8080 and also solely supports http 'posts'. It will use JSON formatted payloads by default, but will accept XML formatted requests as well. The responses likewise can be requested to be XML formatted as well.

XML

Validate request

The following

xsd

can be used to validate requests:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:complexType name="head">
    <xs:attribute name="passwd" type="xs:string" use="required"/>
    <xs:attribute name="task_name" type="xs:string" use="required"/>
    <xs:attribute name="task_type" type="xs:string" use="required"/>
    <xs:attribute name="userid" type="xs:string" use="required"/>
    <xs:attribute name="error" type="xs:string"/>
    <xs:attribute name="req_app" type="xs:string"/>
    <xs:attribute name="req_host" type="xs:string"/>
    <xs:attribute name="status" type="xs:string"/>
  </xs:complexType>
  <xs:complexType name="ext_attr">
    <xs:attribute name="rfc" type="xs:string"/>
    <xs:attribute name="CI" type="xs:string"/>
  </xs:complexType>
  <xs:complexType name="host">
    <xs:sequence>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="ext_attr" type="ext_attr"/>
      </xs:sequence>
      <xs:element name="subnet_addr" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="comment" type="xs:string" use="required"/>
    <xs:attribute name="host_domain" type="xs:string" use="required"/>
    <xs:attribute name="host_name" type="xs:string" use="required"/>
    <xs:attribute name="record_type" type="xs:string" use="required"/>
  </xs:complexType>
```

```
<xs:complexType name="request" mixed="true">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:choice>
      <xs:element name="host" type="host"/>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="custom" type="custom"/>
      </xs:sequence>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="action_type" type="xs:string"/>
  <xs:attribute name="client_type" type="xs:string"/>
  <xs:attribute name="service_class" type="xs:string"/>
  <xs:attribute name="service_task" type="xs:string"/>
  <xs:attribute name="service_type" type="xs:string"/>
  <xs:attribute name="servername" type="xs:string"/>
  <xs:attribute name="action" type="xs:string"/>
  <xs:attribute name="command" type="xs:string"/>
  <xs:attribute name="detail" type="xs:string"/>
  <xs:attribute name="nodename" type="xs:string"/>
  <xs:attribute name="filename" type="xs:string"/>
</xs:complexType>
<xs:element name="task" type="task"/>
<xs:complexType name="task">
  <xs:sequence>
    <xs:element name="head" type="head"/>
    <xs:element name="request" type="request"/>
  </xs:sequence>
  <xs:attribute name="response" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="custom">
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="type" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:string" use="required"/>
</xs:complexType>
</xs:schema>
```

Validate response

The following

xsd

can be used to validate responses

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:complexType name="head">
    <xs:attribute name="error" type="xs:string" use="required"/>
    <xs:attribute name="passwd" type="xs:string" use="required"/>
```

```
<xs:attribute name="req_host" type="xs:string" use="required"/>
<xs:attribute name="status" type="xs:string" use="required"/>
<xs:attribute name="task_id" type="xs:string" use="required"/>
<xs:attribute name="task_name" type="xs:string" use="required"/>
<xs:attribute name="task_type" type="xs:string" use="required"/>
<xs:attribute name="userid" type="xs:string" use="required"/>
<xs:attribute name="req_app" type="xs:string"/>
</xs:complexType>
<xs:complexType name="ext_attr">
  <xs:attribute name="rfc" type="xs:string"/>
  <xs:attribute name="CI" type="xs:string"/>
</xs:complexType>
<xs:complexType name="host">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="ext_attr" type="ext_attr"/>
  </xs:sequence>
  <xs:attribute name="comment" type="xs:string" use="required"/>
  <xs:attribute name="host_domain" type="xs:string" use="required"/>
  <xs:attribute name="host_name" type="xs:string" use="required"/>
  <xs:attribute name="record_type" type="xs:string" use="required"/>
  <xs:attribute name="subnet_addr" type="xs:string"/>
  <xs:attribute name="disable" type="xs:string"/>
  <xs:attribute name="host_fqdn" type="xs:string"/>
  <xs:attribute name="ib_status" type="xs:string"/>
  <xs:attribute name="ib_status_msg" type="xs:string"/>
  <xs:attribute name="new_ip_addr" type="xs:string"/>
  <xs:attribute name="request_id" type="xs:string"/>
  <xs:attribute name="ttl" type="xs:string"/>
</xs:complexType>
<xs:complexType name="request">
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:choice>
      <xs:element name="host" type="host"/>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="custom" type="custom"/>
      </xs:sequence>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="action_type" type="xs:string"/>
  <xs:attribute name="auth_agent" type="xs:string" use="required"/>
  <xs:attribute name="task_module" type="xs:string" use="required"/>
  <xs:attribute name="task_sub" type="xs:string" use="required"/>
  <xs:attribute name="user_level" type="xs:string" use="required"/>
  <xs:attribute name="client_type" type="xs:string"/>
  <xs:attribute name="service_class" type="xs:string"/>
  <xs:attribute name="service_task" type="xs:string"/>
  <xs:attribute name="service_type" type="xs:string"/>
  <xs:attribute name="action" type="xs:string"/>
  <xs:attribute name="command" type="xs:string"/>
  <xs:attribute name="detail" type="xs:string"/>
  <xs:attribute name="servername" type="xs:string"/>
</xs:complexType>
```

```
</xs:complexType>
<xs:complexType name="response">
  <xs:choice>
    <xs:element name="host" type="host"/>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="info" type="xs:string"/>
    </xs:sequence>
    <xs:sequence>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="alias" type="alias"/>
      </xs:sequence>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="custom" type="custom"/>
      </xs:sequence>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="log" type="xs:string"/>
      </xs:sequence>
    </xs:sequence>
  </xs:choice>
  <xs:attribute name="auth_agent" type="xs:string"/>
  <xs:attribute name="client_type" type="xs:string"/>
  <xs:attribute name="service_class" type="xs:string"/>
  <xs:attribute name="service_task" type="xs:string"/>
  <xs:attribute name="service_type" type="xs:string"/>
  <xs:attribute name="task_module" type="xs:string"/>
  <xs:attribute name="task_sub" type="xs:string"/>
  <xs:attribute name="user_level" type="xs:string"/>
  <xs:attribute name="servername" type="xs:string"/>
</xs:complexType>
<xs:element name="task" type="task"/>
<xs:complexType name="task">
  <xs:sequence>
    <xs:element name="head" type="head"/>
    <xs:element name="request" type="request"/>
    <xs:element name="response" type="response"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="custom">
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="type" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="alias">
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="type" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:string" use="required"/>
</xs:complexType>
</xs:schema>
```

JSON

Validating request

The following

JSON schema

can be used to validate requests

```
"definitions": {
  "custom": {
    "properties": {
      "name": {
        "type": "string"
      },
      "type": {
        "type": "string"
      },
      "value": {
        "type": "string"
      }
    },
    "required": [
      "name",
      "type",
      "value"
    ],
    "type": "object"
  },
  "task": {
    "properties": {
      "response": {
        "type": "string"
      },
      "head": {
        "$ref": "#/definitions/head"
      },
      "request": {
        "$ref": "#/definitions/request"
      }
    },
    "required": [
      "head",
      "request"
    ],
    "type": "object"
  },
  "requestCHOICE0": {
    "properties": {
      "action_type": {
```

```
    "type": "string"
  },
  "custom": {
    "$ref": "#/definitions/custom"
  },
  "client_type": {
    "type": "string"
  },
  "service_class": {
    "type": "string"
  },
  "service_task": {
    "type": "string"
  },
  "service_type": {
    "type": "string"
  },
  "servername": {
    "type": "string"
  },
  "action": {
    "type": "string"
  },
  "command": {
    "type": "string"
  },
  "detail": {
    "type": "string"
  },
  "nodename": {
    "type": "string"
  },
  "filename": {
    "type": "string"
  }
},
"type": "object"
},
"host": {
  "properties": {
    "comment": {
      "type": "string"
    },
    "subnet_addr": {
      "type": "string"
    },
    "host_domain": {
      "type": "string"
    },
    "host_name": {
      "type": "string"
    }
  }
}
```

```
    },
    "record_type": {
      "type": "string"
    }
  },
  "required": [
    "comment",
    "host_domain",
    "host_name",
    "record_type",
    "subnet_addr"
  ],
  "type": "object"
},
"ext_attr": {
  "properties": {
    "rfc": {
      "type": "string"
    },
    "CI": {
      "type": "string"
    }
  }
},
"type": "object"
},
"head": {
  "properties": {
    "passwd": {
      "type": "string"
    },
    "task_name": {
      "type": "string"
    },
    "task_type": {
      "type": "string"
    },
    "userid": {
      "type": "string"
    },
    "error": {
      "type": "string"
    },
    "req_app": {
      "type": "string"
    },
    "req_host": {
      "type": "string"
    },
    "status": {
      "type": "string"
    }
  }
}
```

```
    },
    "required": [
      "passwd",
      "task_name",
      "task_type",
      "userid"
    ],
    "type": "object"
  },
  "request": {
    "anyOf": [
      {
        "$ref": "#/definitions/requestCHOICE0"
      }
    ]
  }
},
"$ref": "#/definitions/task"
}
```

Validating response

The following

JSON schema

can be used to validate requests

```
{
  "custom": {
    "properties": {
      "name": {
        "type": "string"
      },
      "type": {
        "type": "string"
      },
      "value": {
        "type": "string"
      }
    },
    "required": [
      "name",
      "type",
      "value"
    ],
    "type": "object"
  },
  "task": {
    "properties": {
      "response": {
```

```
    "type": "string"
  },
  "head": {
    "$ref": "#/definitions/head"
  },
  "request": {
    "$ref": "#/definitions/request"
  }
},
"required": [
  "response",
  "head",
  "request"
],
"type": "object"
},
"requestCHOICE0": {
  "properties": {
    "action_type": {
      "type": "string"
    },
    "custom": {
      "$ref": "#/definitions/custom"
    },
    "client_type": {
      "type": "string"
    },
    "service_class": {
      "type": "string"
    },
    "service_task": {
      "type": "string"
    },
    "service_type": {
      "type": "string"
    },
    "servername": {
      "type": "string"
    },
    "action": {
      "type": "string"
    },
    "command": {
      "type": "string"
    },
    "detail": {
      "type": "string"
    },
    "nodename": {
      "type": "string"
    }
  }
}
```

```
    },
    "filename": {
      "type": "string"
    }
  },
  "type": "object",
  "required": [
    "custom"
  ]
},
"host": {
  "properties": {
    "comment": {
      "type": "string"
    },
    "subnet_addr": {
      "type": "string"
    },
    "host_domain": {
      "type": "string"
    },
    "host_name": {
      "type": "string"
    },
    "record_type": {
      "type": "string"
    }
  },
  "required": [
    "comment",
    "host_domain",
    "host_name",
    "record_type",
    "subnet_addr"
  ],
  "type": "object"
},
"ext_attr": {
  "properties": {
    "rfc": {
      "type": "string"
    },
    "CI": {
      "type": "string"
    }
  },
  "type": "object"
},
"head": {
  "properties": {
    "passwd": {
```

```
    "type": "string"
  },
  "task_name": {
    "type": "string"
  },
  "task_type": {
    "type": "string"
  },
  "userid": {
    "type": "string"
  },
  "error": {
    "type": "string"
  },
  "req_app": {
    "type": "string"
  },
  "req_host": {
    "type": "string"
  },
  "status": {
    "type": "string"
  }
},
"required": [
  "passwd",
  "task_name",
  "task_type",
  "userid"
],
"type": "object"
},
"request": {
  "anyOf": [
    {
      "$ref": "#/definitions/requestCHOICE0"
    }
  ]
}
}
```

From:

<https://wiki.netyce.com/> - **Technical documentation**

Permanent link:

<https://wiki.netyce.com/doku.php?id=maintenance:general:xml>

Last update: **2024/07/03 12:31**

