

# Apache Ranger 0.6 - REST APIs for Service Definition, Service and Policy Management

- Service Definition APIs
  - Get Service Definition by id
  - Get Service Definition by name
  - Create Service Definition
  - Update Service Definition by id
  - Update Service Definition by name
  - Delete Service Definition by id
  - Delete Service Definition by name
  - Search Service Definitions
- Service APIs
  - Get Service by id
  - Get Service by name
  - Create Service
  - Update Service by id
  - Update Service by name
  - Delete Service by id
  - Delete Service by name
  - Search Services
- Policy APIs
  - Get Policy by id
  - Search Policies
  - Get Policy by service-name and policy-name
  - Create Policy
  - Apply Policy
  - Update Policy by id
  - Update Policy by service-name and policy-name
  - Delete policy by id
  - Delete policy by service-name and policy-name
  - Search Policies in a Service

## Service Definition APIs

### Get Service Definition by id

API Name	Get Service Definition
Request Type	GET
Request URL	service/public/v2/api/servicedef/{id}
Request Params	
Response	<div>• HDFS Example Response:</div> <div><pre>{   "id": 1,   "guid": "0d047247-bafe-4cf8-8e9b-d5d377284b2d",   "isEnabled": true,   "createTime": 1472533134184,   "updateTime": 1472533134274,   "version": 1,   "name": "hdfs",   "implClass": "org.apache.ranger.services.hdfs.RangerServiceHdfs",   "label": "HDFS Repository",   "description": "HDFS Repository",   "options": {},   "configs": [{</pre></div>

```

        "itemId": 1,
        "name": "username",
        "type": "string",
        "subType": "",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Username"
    }, {
        "itemId": 2,
        "name": "password",
        "type": "password",
        "subType": "",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Password"
    }, {
        "itemId": 3,
        "name": "fs.default.name",
        "type": "string",
        "subType": "",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Namenode URL"
    }, {
        "itemId": 4,
        "name": "hadoop.security.authorization",
        "type": "bool",
        "subType": "YesTrue:NoFalse",
        "mandatory": true,
        "defaultValue": "false",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Authorization Enabled"
    }, {
        "itemId": 5,
        "name": "hadoop.security.authentication",
        "type": "enum",
        "subType": "authnType",
        "mandatory": true,
        "defaultValue": "simple",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Authentication Type"
    }, {
        "itemId": 6,
        "name": "hadoop.security.auth_to_local",
        "type": "string",
        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 7,
        "name": "dfs.datanode.kerberos.principal",
        "type": "string",
        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 8,

```

```

        "name": "dfs.namenode.kerberos.principal",
        "type": "string",
        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 9,
        "name": "dfs.secondary.namenode.kerberos.principal",
        "type": "string",
        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 10,
        "name": "hadoop.rpc.protection",
        "type": "enum",
        "subType": "rpcProtection",
        "mandatory": false,
        "defaultValue": "authentication",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "RPC Protection Type"
    }, {
        "itemId": 11,
        "name": "commonNameForCertificate",
        "type": "string",
        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Common Name for Certificate"
    }],
    "resources": [{
        "itemId": 1,
        "name": "path",
        "type": "path",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": true,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerPathResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Resource Path",
        "description": "HDFS file or directory path"
    }],
    "accessTypes": [{
        "itemId": 1,
        "name": "read",
        "label": "Read",
        "impliedGrants": []
    }, {
        "itemId": 2,
        "name": "write",
        "label": "Write",
        "impliedGrants": []
    }, {
        "itemId": 3,

```

```

        "name": "execute",
        "label": "Execute",
        "impliedGrants": []
    }],
    "policyConditions": [],
    "contextEnrichers": [],
    "enums": [{
        "itemId": 2,
        "name": "rpcProtection",
        "elements": [{
            "itemId": 1,
            "name": "authentication",
            "label": "Authentication"
        }, {
            "itemId": 2,
            "name": "integrity",
            "label": "Integrity"
        }, {
            "itemId": 3,
            "name": "privacy",
            "label": "Privacy"
        }
    ]],
    "defaultIndex": 0
}, {
    "itemId": 1,
    "name": "authnType",
    "elements": [{
        "itemId": 1,
        "name": "simple",
        "label": "Simple"
    }, {
        "itemId": 2,
        "name": "kerberos",
        "label": "Kerberos"
    }
    ],
    "defaultIndex": 0
}],
    "dataMaskDef": {
        "maskTypes": [],
        "accessTypes": [],
        "resources": []
    },
    "rowFilterDef": {
        "accessTypes": [],
        "resources": []
    }
}

```

## Get Service Definition by name

API Name	Get Service Definition
Request Type	GET
Request URL	service/public/v2/api/servicedef/name/{name}
Request Params	
Response	<p>• Hive Example Response:</p> <pre> {     "id": 3,     "guid": "3e1afb5a-184a-4e82-9d9c-87a5cacc243c",     "isEnabled": true, </pre>

```

"createTime": 1472533135494,
"updateTime": 1472533237053,
"version": 2,
"name": "hive",
"implClass": "org.apache.ranger.services.hive.RangerServiceHive",
"label": "Hive Server2",
"description": "Hive Server2",
"options": {},
"configs": [{
  "itemId": 1,
  "name": "username",
  "type": "string",
  "mandatory": true,
  "validationRegex": "",
  "validationMessage": "",
  "uiHint": "",
  "label": "Username"
}, {
  "itemId": 2,
  "name": "password",
  "type": "password",
  "mandatory": true,
  "validationRegex": "",
  "validationMessage": "",
  "uiHint": "",
  "label": "Password"
}, {
  "itemId": 3,
  "name": "jdbc.driverClassName",
  "type": "string",
  "mandatory": true,
  "defaultValue": "org.apache.hive.jdbc.HiveDriver",
  "validationRegex": "",
  "validationMessage": "",
  "uiHint": ""
}, {
  "itemId": 4,
  "name": "jdbc.url",
  "type": "string",
  "mandatory": true,
  "defaultValue": "",
  "validationRegex": "",
  "validationMessage": "",
  "uiHint": ""
}, {
  "itemId": 5,
  "name": "commonNameForCertificate",
  "type": "string",
  "mandatory": false,
  "validationRegex": "",
  "validationMessage": "",
  "uiHint": "",
  "label": "Common Name for Certificate"
}],
"resources": [{
  "itemId": 1,
  "name": "database",
  "type": "string",
  "level": 10,
  "mandatory": true,
  "lookupSupported": true,
  "recursiveSupported": false,
  "excludesSupported": true,
  "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
  "matcherOptions": {
    "wildCard": "true",
    "ignoreCase": "true"
  },
  "validationRegex": "",
  "validationMessage": "",

```

```

        "uiHint": "",
        "label": "Hive Database",
        "description": "Hive Database"
    }, {
        "itemId": 3,
        "name": "udf",
        "type": "string",
        "level": 20,
        "parent": "database",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Hive UDF",
        "description": "Hive UDF"
    }, {
        "itemId": 2,
        "name": "table",
        "type": "string",
        "level": 20,
        "parent": "database",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Hive Table",
        "description": "Hive Table"
    }, {
        "itemId": 4,
        "name": "column",
        "type": "string",
        "level": 30,
        "parent": "table",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Hive Column",
        "description": "Hive Column"
    }
  ],
  "accessTypes": [{
    "itemId": 1,
    "name": "select",
    "label": "select",

```

```

        "impliedGrants": []
    }, {
        "itemId": 2,
        "name": "update",
        "label": "update",
        "impliedGrants": []
    }, {
        "itemId": 3,
        "name": "create",
        "label": "Create",
        "impliedGrants": []
    }, {
        "itemId": 4,
        "name": "drop",
        "label": "Drop",
        "impliedGrants": []
    }, {
        "itemId": 5,
        "name": "alter",
        "label": "Alter",
        "impliedGrants": []
    }, {
        "itemId": 6,
        "name": "index",
        "label": "Index",
        "impliedGrants": []
    }, {
        "itemId": 7,
        "name": "lock",
        "label": "Lock",
        "impliedGrants": []
    }, {
        "itemId": 8,
        "name": "all",
        "label": "All",
        "impliedGrants": ["select", "update", "create", "drop", "alter", "index",
"lock"]
    }],
    "policyConditions": [],
    "contextEnrichers": [],
    "enums": [],
    "dataMaskDef": {
        "maskTypes": [{
            "itemId": 1,
            "name": "MASK",
            "label": "Redact",
            "description": "Replace lowercase with 'x', uppercase with 'X', digits
with '0'",
            "transformer": "mask({col})",
            "dataMaskOptions": {}
        }, {
            "itemId": 2,
            "name": "MASK_SHOW_LAST_4",
            "label": "Partial mask: show last 4",
            "description": "Show last 4 characters; replace rest with 'x'",
            "transformer": "mask_show_last_n({col}, 4, 'x', 'x', 'x', -1, '1')",
            "dataMaskOptions": {}
        }, {
            "itemId": 3,
            "name": "MASK_SHOW_FIRST_4",
            "label": "Partial mask: show first 4",
            "description": "Show first 4 characters; replace rest with 'x'",
            "transformer": "mask_show_first_n({col}, 4, 'x', 'x', 'x', -1, '1')",
            "dataMaskOptions": {}
        }, {
            "itemId": 4,
            "name": "MASK_HASH",
            "label": "Hash",
            "description": "Hash the value",
            "transformer": "mask_hash({col})",
            "dataMaskOptions": {}
        }
    ]
}

```

```

    }, {
      "itemId": 5,
      "name": "MASK_NULL",
      "label": "Nullify",
      "description": "Replace with NULL",
      "dataMaskOptions": {}
    }, {
      "itemId": 6,
      "name": "MASK_NONE",
      "label": "Unmasked (retain original value)",
      "description": "No masking",
      "dataMaskOptions": {}
    }, {
      "itemId": 12,
      "name": "MASK_DATE_SHOW_YEAR",
      "label": "Date: show only year",
      "description": "Date: show only year",
      "transformer": "mask({col}, 'x', 'x', 'x', -1, '1', 1, 0, -1)",
      "dataMaskOptions": {}
    }, {
      "itemId": 13,
      "name": "CUSTOM",
      "label": "Custom",
      "description": "Custom",
      "dataMaskOptions": {}
    }
  ],
  "accessTypes": [{
    "itemId": 1,
    "name": "select",
    "label": "select",
    "impliedGrants": []
  }],
  "resources": [{
    "itemId": 1,
    "name": "database",
    "type": "string",
    "level": 10,
    "mandatory": true,
    "lookupSupported": true,
    "recursiveSupported": false,
    "excludesSupported": false,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
      "wildCard": "false",
      "ignoreCase": "true"
    },
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "{ \"singleValue\":true }",
    "label": "Hive Database",
    "description": "Hive Database"
  }, {
    "itemId": 2,
    "name": "table",
    "type": "string",
    "level": 20,
    "parent": "database",
    "mandatory": true,
    "lookupSupported": true,
    "recursiveSupported": false,
    "excludesSupported": false,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
      "wildCard": "false",
      "ignoreCase": "true"
    },
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "{ \"singleValue\":true }",

```



```

        "label": "Hive Table",
        "description": "Hive Table"
    }, {
        "itemId": 4,
        "name": "column",
        "type": "string",
        "level": 30,
        "parent": "table",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "false",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "{ \"singleValue\":true }",
        "label": "Hive Column",
        "description": "Hive Column"
    }
  ],
  "rowFilterDef": {
    "accessTypes": [{
      "itemId": 1,
      "name": "select",
      "label": "select",
      "impliedGrants": []
    }],
    "resources": [{
      "itemId": 1,
      "name": "database",
      "type": "string",
      "level": 10,
      "mandatory": true,
      "lookupSupported": true,
      "recursiveSupported": false,
      "excludesSupported": false,
      "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
      "matcherOptions": {
        "wildCard": "false",
        "ignoreCase": "true"
      },
      "validationRegex": "",
      "validationMessage": "",
      "uiHint": "{ \"singleValue\":true }",
      "label": "Hive Database",
      "description": "Hive Database"
    }],
    {
      "itemId": 2,
      "name": "table",
      "type": "string",
      "level": 20,
      "parent": "database",
      "mandatory": true,
      "lookupSupported": true,
      "recursiveSupported": false,
      "excludesSupported": false,
      "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
      "matcherOptions": {
        "wildCard": "false",
        "ignoreCase": "true"
      },
      "validationRegex": "",
      "validationMessage": "",
      "uiHint": "{ \"singleValue\":true }",

```

	<pre>        "label": "Hive Table",         "description": "Hive Table"       }     ]   } }</pre>
--	---

Create Service Definition

API Name	Create Service Definition
Request Type	Post
Request URL	service/public/v2/api/servicedef
Request Params	<div>application/json</div> <div>• Example:</div> <div><pre>{   "accessTypes": [     {       "impliedGrants": [],       "itemId": 1,       "label": "select",       "name": "select"     },     {       "impliedGrants": [],       "itemId": 2,       "label": "update",       "name": "update"     },     {       "impliedGrants": [],       "itemId": 3,       "label": "Create",       "name": "create"     },     {       "impliedGrants": [],       "itemId": 4,       "label": "Drop",       "name": "drop"     },     {       "impliedGrants": [         "select",         "update",         "create",         "drop"       ],       "itemId": 5,       "label": "All",       "name": "all"     }   ],   "configs": [     {       "itemId": 1,       "label": "Username",       "mandatory": true,       "name": "username",       "type": "string",       "uiHint": "",       "validationMessage": "",       "validationRegex": ""     }   ] }</pre></div>

```

        {
            "itemId": 2,
            "label": "Password",
            "mandatory": true,
            "name": "password",
            "type": "password",
            "uiHint": "",
            "validationMessage": "",
            "validationRegex": ""
        }
    ],
    "contextEnrichers": [],
    "description": "Test Component",
    "enums": [],
    "implClass": "org.apache.ranger.services.test.RangerServiceTest",
    "isEnabled": true,
    "label": "Test Component",
    "name": "test",
    "options": {},
    "policyConditions": [],
    "resources": [
        {
            "description": "Root Of Resource Hierarchy for Test Component",
            "excludesSupported": true,
            "itemId": 1,
            "label": "Test Root Resource",
            "level": 10,
            "lookupSupported": true,
            "mandatory": true,
            "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
            "matcherOptions": {
                "ignoreCase": "true",
                "wildCard": "true"
            },
            "name": "root",
            "recursiveSupported": false,
            "type": "string",
            "uiHint": "",
            "validationMessage": "",
            "validationRegex": ""
        },
        {
            "description": "Sub Resource for Test Component",
            "excludesSupported": true,
            "itemId": 2,
            "label": "Test sub resource",
            "level": 20,
            "lookupSupported": true,
            "mandatory": true,
            "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
            "matcherOptions": {
                "ignoreCase": "true",
                "wildCard": "true"
            },
            "name": "sub",
            "parent": "root",
            "recursiveSupported": false,
            "type": "string",
            "uiHint": "",
            "validationMessage": "",
            "validationRegex": ""
        }
    ],
    "version": 1
}

```

**Response**

200-Application/json

```

{
  "accessTypes": [
    {
      "impliedGrants": [],
      "itemId": 1,
      "label": "select",
      "name": "select"
    },
    {
      "impliedGrants": [],
      "itemId": 2,
      "label": "update",
      "name": "update"
    },
    {
      "impliedGrants": [],
      "itemId": 3,
      "label": "Create",
      "name": "create"
    },
    {
      "impliedGrants": [],
      "itemId": 4,
      "label": "Drop",
      "name": "drop"
    },
    {
      "impliedGrants": [
        "select",
        "update",
        "create",
        "drop"
      ],
      "itemId": 5,
      "label": "All",
      "name": "all"
    }
  ],
  "configs": [
    {
      "itemId": 1,
      "label": "Username",
      "mandatory": true,
      "name": "username",
      "type": "string",
      "uiHint": "",
      "validationMessage": "",
      "validationRegex": ""
    },
    {
      "itemId": 2,
      "label": "Password",
      "mandatory": true,
      "name": "password",
      "type": "password",
      "uiHint": "",
      "validationMessage": "",
      "validationRegex": ""
    }
  ],
  "contextEnrichers": [],
  "createTime": 1451347300617,
  "createdBy": "Admin",
  "description": "Test Component",
  "enums": [],
  "guid": "f889f2d3-920a-4504-9905-809bbc417902",
  "id": 101,
  "implClass": "org.apache.ranger.services.test.RangerServiceTest",
  "isEnabled": true,

```

```

"label": "Test Component",
"name": "test",
"options": {},
"policyConditions": [],
"resources": [
  {
    "description": "Root Of Resource Hierarchy for Test Component",
    "excludesSupported": true,
    "itemId": 1,
    "label": "Test Root Resource",
    "level": 10,
    "lookupSupported": true,
    "mandatory": true,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
      "ignoreCase": "true",
      "wildCard": "true"
    },
    "name": "root",
    "recursiveSupported": false,
    "type": "string",
    "uiHint": "",
    "validationMessage": "",
    "validationRegex": ""
  },
  {
    "description": "Sub Resource for Test Component",
    "excludesSupported": true,
    "itemId": 2,
    "label": "Test sub resource",
    "level": 20,
    "lookupSupported": true,
    "mandatory": true,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
      "ignoreCase": "true",
      "wildCard": "true"
    },
    "name": "sub",
    "parent": "root",
    "recursiveSupported": false,
    "type": "string",
    "uiHint": "",
    "validationMessage": "",
    "validationRegex": ""
  }
],
"updateTime": 1451347300618,
"updatedBy": "Admin",
"version": 1,
"dataMaskDef": {
  "maskTypes": [],
  "accessTypes": [],
  "resources": []
},
"rowFilterDef": {
  "accessTypes": [],
  "resources": []
}
}

```

## Update Service Definition by id

API Name	Update Service Definition
Request Type	PUT

Request URL	service/public/v2/api/servicedef/{id}
Request Params	<div>Application/json</div> <div>• Example:</div> <div><pre>{   "accessTypes": [     {       "impliedGrants": [],       "itemId": 1,       "label": "select",       "name": "select"     },     {       "impliedGrants": [],       "itemId": 2,       "label": "update",       "name": "update"     },     {       "impliedGrants": [],       "itemId": 3,       "label": "Create",       "name": "create"     },     {       "impliedGrants": [],       "itemId": 4,       "label": "Drop",       "name": "drop"     },     {       "impliedGrants": [],       "itemId": 5,       "label": "Index",       "name": "index"     },     {       "impliedGrants": [         "select",         "update",         "create",         "drop",         "index"       ],       "itemId": 6,       "label": "All",       "name": "all"     }   ],   "configs": [     {       "itemId": 1,       "label": "Username",       "mandatory": true,       "name": "username",       "type": "string",       "uiHint": "",       "validationMessage": "",       "validationRegex": ""     },     {       "itemId": 2,       "label": "Password",       "mandatory": true,       "name": "password",       "type": "password",       "uiHint": "",       "validationMessage": ""     }   ] }</pre></div>

	<pre>        "validationRegex": ""     } ], "contextEnrichers": [], "description": "Test Component", "enums": [], "implClass": "org.apache.ranger.services.test.RangerServiceTest", "isEnabled": true, "label": "Test Component", "name": "test", "options": {}, "policyConditions": [], "resources": [     {         "description": "Root Of Resource Hierarchy for Test Component",         "excludesSupported": true,         "itemId": 1,         "label": "Test Root Resource",         "level": 10,         "lookupSupported": true,         "mandatory": true,         "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",         "matcherOptions": {             "ignoreCase": "true",             "wildCard": "true"         },         "name": "root",         "recursiveSupported": false,         "type": "string",         "uiHint": "",         "validationMessage": "",         "validationRegex": ""     },     {         "description": "Sub Resource for Test Component",         "excludesSupported": true,         "itemId": 2,         "label": "Test sub resource",         "level": 20,         "lookupSupported": true,         "mandatory": true,         "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",         "matcherOptions": {             "ignoreCase": "true",             "wildCard": "true"         },         "name": "sub",         "parent": "root",         "recursiveSupported": false,         "type": "string",         "uiHint": "",         "validationMessage": "",         "validationRegex": ""     } ] }</pre>
Response	<div>200-Application/json</div> <div><pre>{   "accessTypes": [     {       "impliedGrants": [],       "itemId": 1,       "label": "select",       "name": "select"     }   ], }</pre></div>

```

    {
      "impliedGrants": [],
      "itemId": 2,
      "label": "update",
      "name": "update"
    },
    {
      "impliedGrants": [],
      "itemId": 3,
      "label": "Create",
      "name": "create"
    },
    {
      "impliedGrants": [],
      "itemId": 4,
      "label": "Drop",
      "name": "drop"
    },
    {
      "impliedGrants": [],
      "itemId": 5,
      "label": "Index",
      "name": "index"
    },
    {
      "impliedGrants": [
        "select",
        "update",
        "create",
        "drop",
        "index"
      ],
      "itemId": 6,
      "label": "All",
      "name": "all"
    }
  ],
  "configs": [
    {
      "itemId": 1,
      "label": "Username",
      "mandatory": true,
      "name": "username",
      "type": "string",
      "uiHint": "",
      "validationMessage": "",
      "validationRegex": ""
    },
    {
      "itemId": 2,
      "label": "Password",
      "mandatory": true,
      "name": "password",
      "type": "password",
      "uiHint": "",
      "validationMessage": "",
      "validationRegex": ""
    }
  ],
  "contextEnrichers": [],
  "createTime": 1451347301000,
  "createdBy": "Admin",
  "description": "Test Component",
  "enums": [],
  "guid": "f889f2d3-920a-4504-9905-809bbc417902",
  "id": 101,
  "implClass": "org.apache.ranger.services.test.RangerServiceTest",
  "isEnabled": true,
  "label": "Test Component",
  "name": "test",
  "options": {}

```



```

"policyConditions": [],
"resources": [
  {
    "description": "Root Of Resource Hierarchy for Test Component",
    "excludesSupported": true,
    "itemId": 1,
    "label": "Test Root Resource",
    "level": 10,
    "lookupSupported": true,
    "mandatory": true,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
      "ignoreCase": "true",
      "wildCard": "true"
    },
    "name": "root",
    "recursiveSupported": false,
    "type": "string",
    "uiHint": "",
    "validationMessage": "",
    "validationRegex": ""
  },
  {
    "description": "Sub Resource for Test Component",
    "excludesSupported": true,
    "itemId": 2,
    "label": "Test sub resource",
    "level": 20,
    "lookupSupported": true,
    "mandatory": true,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
      "ignoreCase": "true",
      "wildCard": "true"
    },
    "name": "sub",
    "parent": "root",
    "recursiveSupported": false,
    "type": "string",
    "uiHint": "",
    "validationMessage": "",
    "validationRegex": ""
  }
],
"updateTime": 1451351474321,
"updatedBy": "Admin",
"version": 2,
  "dataMaskDef": {
    "maskTypes": [],
    "accessTypes": [],
    "resources": []
  },
  "rowFilterDef": {
    "accessTypes": [],
    "resources": []
  }
}

```

## Update Service Definition by name

API Name	Update Service Definition
Request Type	PUT
Request URL	service/public/v2/api/servicedef/{name}

<b>Request Params</b>	<b>Application/json</b>  • <b>Example:</b>
<b>Response</b>	200-Application/json

## Delete Service Definition by id

<b>API Name</b>	<b>Delete Service Definition</b>
<b>Request Type</b>	<b>DELETE</b>
<b>Request URL</b>	<b>service/public/v2/api/servicedef/{id}</b>
<b>Request Param</b>	
<b>Response</b>	204-No Content

## Delete Service Definition by name

<b>API Name</b>	<b>Delete Service Definition</b>
<b>Request Type</b>	<b>DELETE</b>
<b>Request URL</b>	<b>service/public/v2/api/servicedef/name/{name}</b>
<b>Request Param</b>	
<b>Response</b>	204-No Content

## Search Service Definitions

<b>API Name</b>	<b>Search Service Definitions</b>
<b>Request Type</b>	<b>GET</b>
<b>Request URL</b>	<b>service/public/v2/api/servicedef</b>
<b>Request Params</b>	<b>Query Params</b> pageSize int <i>The page size required</i> startIndex int <i>The start record index</i> serviceType string <i>The service definition names("hdfs","hive","hbase","knox","storm","solr","kafka","yarn","atlas","kms","nifi")</i> isEnabled boolean <i>The enabled status : true if enabled; false otherwise</i> Example : pageSize=25&startIndex=0
<b>Response</b>	200-Application/json • <b>Example:</b> <pre>[{   "id": 1,   "guid": "0d047247-bafe-4cf8-8e9b-d5d377284b2d",   "isEnabled": true,   "createTime": 1472533134184,   "updateTime": 1472533134274,   "version": 1,   "name": "hdfs",</pre>

```

"implClass": "org.apache.ranger.services.hdfs.RangerServiceHdfs",
"label": "HDFS Repository",
"description": "HDFS Repository",
"options": {},
"configs": [{
    "itemId": 1,
    "name": "username",
    "type": "string",
    "subType": "",
    "mandatory": true,
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Username"
}, {
    "itemId": 2,
    "name": "password",
    "type": "password",
    "subType": "",
    "mandatory": true,
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Password"
}, {
    "itemId": 3,
    "name": "fs.default.name",
    "type": "string",
    "subType": "",
    "mandatory": true,
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Namenode URL"
}, {
    "itemId": 4,
    "name": "hadoop.security.authorization",
    "type": "bool",
    "subType": "YesTrue:NoFalse",
    "mandatory": true,
    "defaultValue": "false",
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Authorization Enabled"
}, {
    "itemId": 5,
    "name": "hadoop.security.authentication",
    "type": "enum",
    "subType": "authnType",
    "mandatory": true,
    "defaultValue": "simple",
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Authentication Type"
}, {
    "itemId": 6,
    "name": "hadoop.security.auth_to_local",
    "type": "string",
    "subType": "",
    "mandatory": false,
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": ""
}, {
    "itemId": 7,
    "name": "dfs.datanode.kerberos.principal",
    "type": "string",
    "subType": "",
    "mandatory": false,

```

```

        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 8,
        "name": "dfs.namenode.kerberos.principal",
        "type": "string",
        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 9,
        "name": "dfs.secondary.namenode.kerberos.principal",
        "type": "string",
        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 10,
        "name": "hadoop.rpc.protection",
        "type": "enum",
        "subType": "rpcProtection",
        "mandatory": false,
        "defaultValue": "authentication",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "RPC Protection Type"
    }, {
        "itemId": 11,
        "name": "commonNameForCertificate",
        "type": "string",
        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Common Name for Certificate"
    }
  ],
  "resources": [{
    "itemId": 1,
    "name": "path",
    "type": "path",
    "level": 10,
    "mandatory": true,
    "lookupSupported": true,
    "recursiveSupported": true,
    "excludesSupported": false,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerPathResourceMatcher",
    "matcherOptions": {
      "wildCard": "true",
      "ignoreCase": "false"
    },
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Resource Path",
    "description": "HDFS file or directory path"
  }],
  "accessTypes": [{
    "itemId": 1,
    "name": "read",
    "label": "Read",
    "impliedGrants": []
  }, {
    "itemId": 2,
    "name": "write",

```

```

        "label": "Write",
        "impliedGrants": []
    }, {
        "itemId": 3,
        "name": "execute",
        "label": "Execute",
        "impliedGrants": []
    }],
    "policyConditions": [],
    "contextEnrichers": [],
    "enums": [{
        "itemId": 2,
        "name": "rpcProtection",
        "elements": [{
            "itemId": 1,
            "name": "authentication",
            "label": "Authentication"
        }, {
            "itemId": 2,
            "name": "integrity",
            "label": "Integrity"
        }, {
            "itemId": 3,
            "name": "privacy",
            "label": "Privacy"
        }
    ]],
    "defaultIndex": 0
}, {
    "itemId": 1,
    "name": "authnType",
    "elements": [{
        "itemId": 1,
        "name": "simple",
        "label": "Simple"
    }, {
        "itemId": 2,
        "name": "kerberos",
        "label": "Kerberos"
    }
    ],
    "defaultIndex": 0
}],
    "dataMaskDef": {
        "maskTypes": [],
        "accessTypes": [],
        "resources": []
    },
    "rowFilterDef": {
        "accessTypes": [],
        "resources": []
    }
}, {
    "id": 2,
    "guid": "d6cealf0-2509-4791-8fc1-7b092399ba3b",
    "isEnabled": true,
    "createTime": 1472533134858,
    "updateTime": 1472533134861,
    "version": 1,
    "name": "hbase",
    "implClass": "org.apache.ranger.services.hbase.RangerServiceHBase",
    "label": "HBase",
    "description": "HBase",
    "options": {},
    "configs": [{
        "itemId": 1,
        "name": "username",
        "type": "string",
        "subType": "",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }

```

```

    "label": "Username"
  }, {
    "itemId": 2,
    "name": "password",
    "type": "password",
    "subType": "",
    "mandatory": true,
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Password"
  }, {
    "itemId": 3,
    "name": "hadoop.security.authentication",
    "type": "enum",
    "subType": "authnType",
    "mandatory": true,
    "defaultValue": "simple",
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": ""
  }, {
    "itemId": 4,
    "name": "hbase.master.kerberos.principal",
    "type": "string",
    "subType": "",
    "mandatory": false,
    "defaultValue": ""
  }, {
    "itemId": 5,
    "name": "hbase.security.authentication",
    "type": "enum",
    "subType": "authnType",
    "mandatory": true,
    "defaultValue": "simple",
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": ""
  }, {
    "itemId": 6,
    "name": "hbase.zookeeper.property.clientPort",
    "type": "int",
    "subType": "",
    "mandatory": true,
    "defaultValue": "2181",
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": ""
  }, {
    "itemId": 7,
    "name": "hbase.zookeeper.quorum",
    "type": "string",
    "subType": "",
    "mandatory": true,
    "defaultValue": "",
    "validationRegex": "",
    "validationMessage": ""
  }, {
    "itemId": 8,
    "name": "zookeeper.znode.parent",
    "type": "string",
    "subType": "",
    "mandatory": true,
    "defaultValue": "/hbase",
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": ""
  }, {
    "itemId": 9,
    "name": "commonNameForCertificate",
    "type": "string",

```

```

        "subType": "",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Common Name for Certificate"
    }},
    "resources": [{
        "itemId": 1,
        "name": "table",
        "type": "string",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "HBase Table",
        "description": "HBase Table"
    }, {
        "itemId": 2,
        "name": "column-family",
        "type": "string",
        "level": 20,
        "parent": "table",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "HBase Column-family",
        "description": "HBase Column-family"
    }, {
        "itemId": 3,
        "name": "column",
        "type": "string",
        "level": 30,
        "parent": "column-family",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "HBase Column",
        "description": "HBase Column"
    }},
    "accessTypes": [{
        "itemId": 1,
        "name": "read",
        "label": "Read",

```

```

        "impliedGrants": []
    }, {
        "itemId": 2,
        "name": "write",
        "label": "Write",
        "impliedGrants": []
    }, {
        "itemId": 3,
        "name": "create",
        "label": "Create",
        "impliedGrants": []
    }, {
        "itemId": 4,
        "name": "admin",
        "label": "Admin",
        "impliedGrants": ["read", "write", "create"]
    }],
    "policyConditions": [],
    "contextEnrichers": [],
    "enums": [{
        "itemId": 1,
        "name": "authnType",
        "elements": [{
            "itemId": 1,
            "name": "simple",
            "label": "Simple"
        }, {
            "itemId": 2,
            "name": "kerberos",
            "label": "Kerberos"
        }],
        "defaultIndex": 0
    }],
    "dataMaskDef": {
        "maskTypes": [],
        "accessTypes": [],
        "resources": []
    },
    "rowFilterDef": {
        "accessTypes": [],
        "resources": []
    }
}, {
    "id": 5,
    "guid": "84b481b5-f23b-4f71-b8b6-ab33977149ca",
    "isEnabled": true,
    "createTime": 1472533136447,
    "updateTime": 1472533136448,
    "version": 1,
    "name": "knox",
    "implClass": "org.apache.ranger.services.knox.RangerServiceKnox",
    "label": "Knox Gateway",
    "description": "Knox Gateway",
    "options": {},
    "configs": [{
        "itemId": 1,
        "name": "username",
        "type": "string",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Username"
    }, {
        "itemId": 2,
        "name": "password",
        "type": "password",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }

```



```

        "label": "Password"
    }, {
        "itemId": 3,
        "name": "knox.url",
        "type": "string",
        "mandatory": true,
        "defaultValue": "",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 4,
        "name": "commonNameForCertificate",
        "type": "string",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Common Name for Certificate"
    }],
    "resources": [{
        "itemId": 1,
        "name": "topology",
        "type": "string",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Knox Topology",
        "description": "Knox Topology"
    }, {
        "itemId": 2,
        "name": "service",
        "type": "string",
        "level": 20,
        "parent": "topology",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Knox Service",
        "description": "Knox Service"
    }],
    "accessTypes": [{
        "itemId": 1,
        "name": "allow",
        "label": "Allow",
        "impliedGrants": []
    }],
    "policyConditions": [{
        "itemId": 1,
        "name": "ip-range",
        "evaluator": "org.apache.ranger.plugin.conditionevaluator.RangerIpMatcher",
        "evaluatorOptions": {}
    }],

```

```

        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "IP Address Range",
        "description": "IP Address Range"
    }},
    "contextEnrichers": [],
    "enums": [],
    "dataMaskDef": {
        "maskTypes": [],
        "accessTypes": [],
        "resources": []
    },
    "rowFilterDef": {
        "accessTypes": [],
        "resources": []
    }
}, {
    "id": 6,
    "guid": "2a60f427-edcf-4e20-834c-a9a267b5b963",
    "isEnabled": true,
    "createTime": 1472533136716,
    "updateTime": 1472533136717,
    "version": 1,
    "name": "storm",
    "implClass": "org.apache.ranger.services.storm.RangerServiceStorm",
    "label": "Storm",
    "description": "Storm",
    "options": {},
    "configs": [{
        "itemId": 1,
        "name": "username",
        "type": "string",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Username"
    }, {
        "itemId": 2,
        "name": "password",
        "type": "password",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Password"
    }, {
        "itemId": 3,
        "name": "nimbus.url",
        "type": "string",
        "mandatory": true,
        "defaultValue": "",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Nimbus URL"
    }, {
        "itemId": 4,
        "name": "commonNameForCertificate",
        "type": "string",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Common Name for Certificate"
    }],
    "resources": [{
        "itemId": 1,
        "name": "topology",
        "type": "string",

```

```

        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Storm Topology",
        "description": "Storm Topology"
    }],
    "accessTypes": [{
        "itemId": 1,
        "name": "submitTopology",
        "label": "Submit Topology",
        "impliedGrants": ["fileUpload", "fileDownload"]
    }, {
        "itemId": 2,
        "name": "fileUpload",
        "label": "File Upload",
        "impliedGrants": []
    }, {
        "itemId": 5,
        "name": "fileDownload",
        "label": "File Download",
        "impliedGrants": []
    }, {
        "itemId": 6,
        "name": "killTopology",
        "label": "Kill Topology",
        "impliedGrants": []
    }, {
        "itemId": 7,
        "name": "rebalance",
        "label": "Rebalance",
        "impliedGrants": []
    }, {
        "itemId": 8,
        "name": "activate",
        "label": "Activate",
        "impliedGrants": []
    }, {
        "itemId": 9,
        "name": "deactivate",
        "label": "Deactivate",
        "impliedGrants": []
    }, {
        "itemId": 10,
        "name": "getTopologyConf",
        "label": "Get Topology Conf",
        "impliedGrants": []
    }, {
        "itemId": 11,
        "name": "getTopology",
        "label": "Get Topology",
        "impliedGrants": []
    }, {
        "itemId": 12,
        "name": "getUserTopology",
        "label": "Get User Topology",
        "impliedGrants": []
    }, {
        "itemId": 13,
        "name": "getTopologyInfo",
        "label": "Get Topology Info",
        "impliedGrants": []
    }

```

```

    }, {
      "itemId": 14,
      "name": "uploadNewCredentials",
      "label": "Upload New Credential",
      "impliedGrants": []
    }],
    "policyConditions": [],
    "contextEnrichers": [],
    "enums": [],
    "dataMaskDef": {
      "maskTypes": [],
      "accessTypes": [],
      "resources": []
    },
    "rowFilterDef": {
      "accessTypes": [],
      "resources": []
    }
  }, {
    "id": 4,
    "guid": "5b710438-edcf-4e20-834c-a9a267b5b963",
    "isEnabled": true,
    "createTime": 1472533136998,
    "updateTime": 1472533137000,
    "version": 1,
    "name": "yarn",
    "implClass": "org.apache.ranger.services.yarn.RangerServiceYarn",
    "label": "YARN",
    "description": "YARN",
    "options": {},
    "configs": [{
      "itemId": 1,
      "name": "username",
      "type": "string",
      "mandatory": true,
      "validationRegex": "",
      "validationMessage": "",
      "uiHint": "",
      "label": "Username"
    }, {
      "itemId": 2,
      "name": "password",
      "type": "password",
      "mandatory": true,
      "validationRegex": "",
      "validationMessage": "",
      "uiHint": "",
      "label": "Password"
    }, {
      "itemId": 3,
      "name": "yarn.url",
      "type": "string",
      "mandatory": true,
      "defaultValue": "",
      "validationRegex": "",
      "validationMessage": "",
      "uiHint": "",
      "label": "YARN REST URL"
    }, {
      "itemId": 4,
      "name": "hadoop.security.authentication",
      "type": "enum",
      "subType": "authnType",
      "mandatory": false,
      "defaultValue": "simple",
      "validationRegex": "",
      "validationMessage": "",
      "uiHint": "",
      "label": "Authentication Type"
    }, {
      "itemId": 5,

```

```

        "name": "commonNameForCertificate",
        "type": "string",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Common Name for Certificate"
    }},
    "resources": [{
        "itemId": 1,
        "name": "queue",
        "type": "string",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": true,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerPathResourceMatcher",
        "matcherOptions": {
            "pathSeparatorChar": ".",
            "wildCard": "true",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Queue",
        "description": "Queue"
    }},
    "accessTypes": [{
        "itemId": 1,
        "name": "submit-app",
        "label": "submit-app",
        "impliedGrants": []
    }, {
        "itemId": 2,
        "name": "admin-queue",
        "label": "admin-queue",
        "impliedGrants": ["submit-app"]
    }},
    "policyConditions": [],
    "contextEnrichers": [],
    "enums": [{
        "itemId": 1,
        "name": "authnType",
        "elements": [{
            "itemId": 1,
            "name": "simple",
            "label": "Simple"
        }, {
            "itemId": 2,
            "name": "kerberos",
            "label": "Kerberos"
        }
    ]},
    "defaultIndex": 0
    }},
    "dataMaskDef": {
        "maskTypes": [],
        "accessTypes": [],
        "resources": []
    },
    "rowFilterDef": {
        "accessTypes": [],
        "resources": []
    }
}, {
    "id": 9,
    "guid": "019a2d10-690f-4912-ae0-c77fe7608d54",
    "isEnabled": true,
    "createTime": 1472533137152,
    "updateTime": 1472533137154,

```

```

"version": 1,
"name": "kafka",
"implClass": "org.apache.ranger.services.kafka.RangerServiceKafka",
"label": "Kafka",
"description": "Apache Kafka",
"options": {},
"configs": [{
    "itemId": 1,
    "name": "username",
    "type": "string",
    "mandatory": true,
    "label": "Username"
}, {
    "itemId": 2,
    "name": "password",
    "type": "password",
    "mandatory": true,
    "label": "Password"
}, {
    "itemId": 3,
    "name": "zookeeper.connect",
    "type": "string",
    "mandatory": true,
    "defaultValue": "localhost:2181",
    "label": "Zookeeper Connect String"
}, {
    "itemId": 4,
    "name": "commonNameForCertificate",
    "type": "string",
    "mandatory": false,
    "label": "Ranger Plugin SSL CName"
}],
"resources": [{
    "itemId": 1,
    "name": "topic",
    "type": "string",
    "level": 1,
    "mandatory": true,
    "lookupSupported": true,
    "recursiveSupported": false,
    "excludesSupported": true,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
        "wildCard": "true",
        "ignoreCase": "true"
    },
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Topic",
    "description": "Topic"
}],
"accessTypes": [{
    "itemId": 1,
    "name": "publish",
    "label": "Publish",
    "impliedGrants": ["describe"]
}, {
    "itemId": 2,
    "name": "consume",
    "label": "Consume",
    "impliedGrants": ["describe"]
}, {
    "itemId": 5,
    "name": "configure",
    "label": "Configure",
    "impliedGrants": ["describe"]
}, {
    "itemId": 6,
    "name": "describe",
    "label": "Describe",

```

```

        "impliedGrants": []
    }, {
        "itemId": 8,
        "name": "create",
        "label": "Create",
        "impliedGrants": []
    }, {
        "itemId": 9,
        "name": "delete",
        "label": "Delete",
        "impliedGrants": []
    }, {
        "itemId": 7,
        "name": "kafka_admin",
        "label": "Kafka Admin",
        "impliedGrants": ["publish", "consume", "configure", "describe", "create",
"delete"]
    }],
    "policyConditions": [{
        "itemId": 1,
        "name": "ip-range",
        "evaluator": "org.apache.ranger.plugin.conditionevaluator.RangerIpMatcher",
        "evaluatorOptions": {},
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "IP Address Range",
        "description": "IP Address Range"
    }],
    "contextEnrichers": [],
    "enums": [],
    "dataMaskDef": {
        "maskTypes": [],
        "accessTypes": [],
        "resources": []
    },
    "rowFilterDef": {
        "accessTypes": [],
        "resources": []
    }
}, {
    "id": 8,
    "guid": "65ea5d7a-24ba-49bb-8a21-a5da6b7becdb",
    "isEnabled": true,
    "createTime": 1472533137375,
    "updateTime": 1472533137376,
    "version": 1,
    "name": "solr",
    "implClass": "org.apache.ranger.services.solr.RangerServiceSolr",
    "label": "SOLR",
    "description": "Solr",
    "options": {},
    "configs": [{
        "itemId": 100,
        "name": "username",
        "type": "string",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Username"
    }, {
        "itemId": 200,
        "name": "password",
        "type": "password",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Password"
    }, {

```

```

        "itemId": 400,
        "name": "solr.url",
        "type": "string",
        "mandatory": true,
        "defaultValue": "",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Solr URL"
    }, {
        "itemId": 500,
        "name": "commonNameForCertificate",
        "type": "string",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Ranger Plugin SSL CName"
    }],
    "resources": [{
        "itemId": 100,
        "name": "collection",
        "type": "string",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Solr Collection",
        "description": "Solr Collection"
    }],
    "accessTypes": [{
        "itemId": 100,
        "name": "query",
        "label": "Query",
        "impliedGrants": []
    }, {
        "itemId": 200,
        "name": "update",
        "label": "Update",
        "impliedGrants": []
    }, {
        "itemId": 300,
        "name": "others",
        "label": "Others",
        "impliedGrants": []
    }, {
        "itemId": 900,
        "name": "solr_admin",
        "label": "Solr Admin",
        "impliedGrants": ["query", "update", "others"]
    }],
    "policyConditions": [{
        "itemId": 100,
        "name": "ip-range",
        "evaluator": "org.apache.ranger.plugin.conditionevaluator.RangerIpMatcher",
        "evaluatorOptions": {},
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "IP Address Range",
        "description": "IP Address Range"
    }],

```



```

"contextEnrichers": [],
"enums": [],
"dataMaskDef": {
    "maskTypes": [],
    "accessTypes": [],
    "resources": []
},
"rowFilterDef": {
    "accessTypes": [],
    "resources": []
}
}, {
    "id": 11,
    "guid": "5fb8620c-139c-4f3b-91ba-8c7cbb1dadf3",
    "isEnabled": true,
    "createTime": 1472533137527,
    "updateTime": 1472533137529,
    "version": 1,
    "name": "atlas",
    "implClass": "org.apache.ranger.services.atlas.RangerServiceAtlas",
    "label": "Atlas Metadata Server",
    "description": "Atlas Metadata Server",
    "options": {},
    "configs": [{
        "itemId": 1,
        "name": "username",
        "type": "string",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Username"
    }, {
        "itemId": 2,
        "name": "password",
        "type": "password",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Password"
    }, {
        "itemId": 3,
        "name": "atlas.rest.address",
        "type": "string",
        "mandatory": true,
        "defaultValue": "http://localhost:21000",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 4,
        "name": "commonNameForCertificate",
        "type": "string",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Common Name for Certificate"
    }
    ],
    "resources": [{
        "itemId": 5,
        "name": "term",
        "type": "string",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {

```

```

        "wildCard": "true",
        "ignoreCase": "true"
    },
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Term",
    "description": "Term"
}, {
    "itemId": 4,
    "name": "taxonomy",
    "type": "string",
    "level": 10,
    "mandatory": true,
    "lookupSupported": true,
    "recursiveSupported": false,
    "excludesSupported": true,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
        "wildCard": "true",
        "ignoreCase": "true"
    },
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Taxonomy",
    "description": "Taxonomy"
}, {
    "itemId": 3,
    "name": "operation",
    "type": "string",
    "level": 10,
    "mandatory": true,
    "lookupSupported": true,
    "recursiveSupported": false,
    "excludesSupported": true,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
        "wildCard": "true",
        "ignoreCase": "true"
    },
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Operations",
    "description": "Admin Operations"
}, {
    "itemId": 2,
    "name": "type",
    "type": "string",
    "level": 10,
    "mandatory": true,
    "lookupSupported": true,
    "recursiveSupported": false,
    "excludesSupported": true,
    "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
    "matcherOptions": {
        "wildCard": "true",
        "ignoreCase": "true"
    },
    "validationRegex": "",
    "validationMessage": "",
    "uiHint": "",
    "label": "Type",
    "description": "Type"
}, {
    "itemId": 1,
    "name": "entity",
    "type": "string",
    "level": 10,
    "mandatory": true,

```

```

        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Entity",
        "description": "Entity"
    }],
    "accessTypes": [{
        "itemId": 1,
        "name": "read",
        "label": "read",
        "impliedGrants": []
    }, {
        "itemId": 2,
        "name": "create",
        "label": "create",
        "impliedGrants": []
    }, {
        "itemId": 3,
        "name": "update",
        "label": "update",
        "impliedGrants": []
    }, {
        "itemId": 4,
        "name": "delete",
        "label": "delete",
        "impliedGrants": []
    }, {
        "itemId": 5,
        "name": "all",
        "label": "All",
        "impliedGrants": ["read", "create", "update", "delete"]
    }],
    "policyConditions": [],
    "contextEnrichers": [],
    "enums": [],
    "dataMaskDef": {
        "maskTypes": [],
        "accessTypes": [],
        "resources": []
    },
    "rowFilterDef": {
        "accessTypes": [],
        "resources": []
    }
}, {
    "id": 100,
    "guid": "0d047248-baff-4cf9-8e9e-d5d377284b2e",
    "isEnabled": true,
    "createTime": 1472533137752,
    "updateTime": 1472533142359,
    "version": 11,
    "name": "tag",
    "implClass": "org.apache.ranger.services.tag.RangerServiceTag",
    "label": "TAG",
    "description": "TAG Service Definition",
    "options": {
        "ui.pages": "tag-based-policies"
    },
    "configs": [],
    "resources": [{
        "itemId": 1,
        "name": "tag",
        "type": "string",

```

```

        "level": 1,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "false",
            "ignoreCase": "false"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "{ \"singleValue\":true }",
        "label": "TAG",
        "description": "TAG"
    }],
    "accessTypes": [{
        "itemId": 1002,
        "name": "hdfs:read",
        "label": "Read",
        "impliedGrants": []
    }, {
        "itemId": 1003,
        "name": "hdfs:write",
        "label": "Write",
        "impliedGrants": []
    }, {
        "itemId": 1004,
        "name": "hdfs:execute",
        "label": "Execute",
        "impliedGrants": []
    }, {
        "itemId": 2003,
        "name": "hbase:read",
        "label": "Read",
        "impliedGrants": []
    }, {
        "itemId": 2004,
        "name": "hbase:write",
        "label": "Write",
        "impliedGrants": []
    }, {
        "itemId": 2005,
        "name": "hbase:create",
        "label": "Create",
        "impliedGrants": []
    }, {
        "itemId": 2006,
        "name": "hbase:admin",
        "label": "Admin",
        "impliedGrants": ["hbase:create", "hbase:write", "hbase:read"]
    }, {
        "itemId": 3004,
        "name": "hive:select",
        "label": "select",
        "impliedGrants": []
    }, {
        "itemId": 3005,
        "name": "hive:update",
        "label": "update",
        "impliedGrants": []
    }, {
        "itemId": 3006,
        "name": "hive:create",
        "label": "Create",
        "impliedGrants": []
    }, {
        "itemId": 3007,
        "name": "hive:drop",
        "label": "Drop",
        "impliedGrants": []
    }

```

```

    }, {
      "itemId": 3008,
      "name": "hive:alter",
      "label": "Alter",
      "impliedGrants": []
    }, {
      "itemId": 3009,
      "name": "hive:index",
      "label": "Index",
      "impliedGrants": []
    }, {
      "itemId": 3010,
      "name": "hive:lock",
      "label": "Lock",
      "impliedGrants": []
    }, {
      "itemId": 3011,
      "name": "hive:all",
      "label": "All",
      "impliedGrants": ["hive:lock", "hive:index", "hive:alter", "hive:drop", "hive:
create", "hive:update", "hive:select"]
    }, {
      "itemId": 7008,
      "name": "kms:create",
      "label": "Create",
      "impliedGrants": []
    }, {
      "itemId": 7009,
      "name": "kms:delete",
      "label": "Delete",
      "impliedGrants": []
    }, {
      "itemId": 7010,
      "name": "kms:rollover",
      "label": "Rollover",
      "impliedGrants": []
    }, {
      "itemId": 7011,
      "name": "kms:setkeymaterial",
      "label": "Set Key Material",
      "impliedGrants": []
    }, {
      "itemId": 7012,
      "name": "kms:get",
      "label": "Get",
      "impliedGrants": []
    }, {
      "itemId": 7013,
      "name": "kms:getkeys",
      "label": "Get Keys",
      "impliedGrants": []
    }, {
      "itemId": 7014,
      "name": "kms:getmetadata",
      "label": "Get Metadata",
      "impliedGrants": []
    }, {
      "itemId": 7015,
      "name": "kms:generateeeek",
      "label": "Generate EEK",
      "impliedGrants": []
    }, {
      "itemId": 7016,
      "name": "kms:decrypteeek",
      "label": "Decrypt EEK",
      "impliedGrants": []
    }, {
      "itemId": 5006,
      "name": "knox:allow",
      "label": "Allow",
      "impliedGrants": []
    }
  ]
}

```

```

}, {
  "itemId": 6007,
  "name": "storm:submitTopology",
  "label": "Submit Topology",
  "impliedGrants": ["storm:fileDownload", "storm:fileUpload"]
}, {
  "itemId": 6008,
  "name": "storm:fileUpload",
  "label": "File Upload",
  "impliedGrants": []
}, {
  "itemId": 6011,
  "name": "storm:fileDownload",
  "label": "File Download",
  "impliedGrants": []
}, {
  "itemId": 6012,
  "name": "storm:killTopology",
  "label": "Kill Topology",
  "impliedGrants": []
}, {
  "itemId": 6013,
  "name": "storm:rebalance",
  "label": "Rebalance",
  "impliedGrants": []
}, {
  "itemId": 6014,
  "name": "storm:activate",
  "label": "Activate",
  "impliedGrants": []
}, {
  "itemId": 6015,
  "name": "storm:deactivate",
  "label": "Deactivate",
  "impliedGrants": []
}, {
  "itemId": 6016,
  "name": "storm:getTopologyConf",
  "label": "Get Topology Conf",
  "impliedGrants": []
}, {
  "itemId": 6017,
  "name": "storm:getTopology",
  "label": "Get Topology",
  "impliedGrants": []
}, {
  "itemId": 6018,
  "name": "storm:getUserTopology",
  "label": "Get User Topology",
  "impliedGrants": []
}, {
  "itemId": 6019,
  "name": "storm:getTopologyInfo",
  "label": "Get Topology Info",
  "impliedGrants": []
}, {
  "itemId": 6020,
  "name": "storm:uploadNewCredentials",
  "label": "Upload New Credential",
  "impliedGrants": []
}, {
  "itemId": 4005,
  "name": "yarn:submit-app",
  "label": "submit-app",
  "impliedGrants": []
}, {
  "itemId": 4006,
  "name": "yarn:admin-queue",
  "label": "admin-queue",
  "impliedGrants": ["yarn:submit-app"]
}, {

```

```

        "itemId": 9010,
        "name": "kafka:publish",
        "label": "Publish",
        "impliedGrants": ["kafka:describe"]
    }, {
        "itemId": 9011,
        "name": "kafka:consume",
        "label": "Consume",
        "impliedGrants": ["kafka:describe"]
    }, {
        "itemId": 9014,
        "name": "kafka:configure",
        "label": "Configure",
        "impliedGrants": ["kafka:describe"]
    }, {
        "itemId": 9015,
        "name": "kafka:describe",
        "label": "Describe",
        "impliedGrants": []
    }, {
        "itemId": 9017,
        "name": "kafka:create",
        "label": "Create",
        "impliedGrants": []
    }, {
        "itemId": 9018,
        "name": "kafka:delete",
        "label": "Delete",
        "impliedGrants": []
    }, {
        "itemId": 9016,
        "name": "kafka:kafka_admin",
        "label": "Kafka Admin",
        "impliedGrants": ["kafka:delete", "kafka:create", "kafka:describe", "kafka:
configure", "kafka:consume", "kafka:publish"]
    }, {
        "itemId": 8108,
        "name": "solr:query",
        "label": "Query",
        "impliedGrants": []
    }, {
        "itemId": 8208,
        "name": "solr:update",
        "label": "Update",
        "impliedGrants": []
    }, {
        "itemId": 8308,
        "name": "solr:others",
        "label": "Others",
        "impliedGrants": []
    }, {
        "itemId": 8908,
        "name": "solr:solr_admin",
        "label": "Solr Admin",
        "impliedGrants": ["solr:others", "solr:update", "solr:query"]
    }, {
        "itemId": 11012,
        "name": "atlas:read",
        "label": "read",
        "impliedGrants": []
    }, {
        "itemId": 11013,
        "name": "atlas:create",
        "label": "create",
        "impliedGrants": []
    }, {
        "itemId": 11014,
        "name": "atlas:update",
        "label": "update",
        "impliedGrants": []
    }, {

```

```

        "itemId": 11015,
        "name": "atlas:delete",
        "label": "delete",
        "impliedGrants": []
    }, {
        "itemId": 11016,
        "name": "atlas:all",
        "label": "All",
        "impliedGrants": ["atlas:delete", "atlas:update", "atlas:create", "atlas:read"]
    }],
    "policyConditions": [{
        "itemId": 1,
        "name": "accessed-after-expiry",
        "evaluator": "org.apache.ranger.plugin.conditionevaluator.
RangerScriptTemplateConditionEvaluator",
        "evaluatorOptions": {
            "scriptTemplate": "ctx.isAccessedAfter('expiry_date');"
        },
        "uiHint": "{ \"singleValue\":true }",
        "label": "Accessed after expiry_date (yes/no)?",
        "description": "Accessed after expiry_date? (yes/no)"
    }],
    "contextEnrichers": [{
        "itemId": 1,
        "name": "TagEnricher",
        "enricher": "org.apache.ranger.plugin.contextenricher.RangerTagEnricher",
        "enricherOptions": {
            "tagRetrieverClassName": "org.apache.ranger.plugin.contextenricher.
RangerAdminTagRetriever",
            "tagRefresherPollingInterval": "60000"
        }
    }],
    "enums": [],
    "dataMaskDef": {
        "maskTypes": [],
        "accessTypes": [],
        "resources": []
    },
    "rowFilterDef": {
        "accessTypes": [],
        "resources": []
    }
}, {
    "id": 3,
    "guid": "3e1afb5a-184a-4e82-9d9c-87a5cacc243c",
    "isEnabled": true,
    "createTime": 1472533135494,
    "updateTime": 1472533237053,
    "version": 2,
    "name": "hive",
    "implClass": "org.apache.ranger.services.hive.RangerServiceHive",
    "label": "Hive Server2",
    "description": "Hive Server2",
    "options": {},
    "configs": [{
        "itemId": 1,
        "name": "username",
        "type": "string",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Username"
    }, {
        "itemId": 2,
        "name": "password",
        "type": "password",
        "mandatory": true,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",

```



```

        "label": "Password"
    }, {
        "itemId": 3,
        "name": "jdbc.driverClassName",
        "type": "string",
        "mandatory": true,
        "defaultValue": "org.apache.hive.jdbc.HiveDriver",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 4,
        "name": "jdbc.url",
        "type": "string",
        "mandatory": true,
        "defaultValue": "",
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": ""
    }, {
        "itemId": 5,
        "name": "commonNameForCertificate",
        "type": "string",
        "mandatory": false,
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Common Name for Certificate"
    }],
    "resources": [{
        "itemId": 1,
        "name": "database",
        "type": "string",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Hive Database",
        "description": "Hive Database"
    }, {
        "itemId": 3,
        "name": "udf",
        "type": "string",
        "level": 20,
        "parent": "database",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Hive UDF",
        "description": "Hive UDF"
    }, {
        "itemId": 2,
        "name": "table",

```

```

        "type": "string",
        "level": 20,
        "parent": "database",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Hive Table",
        "description": "Hive Table"
    }, {
        "itemId": 4,
        "name": "column",
        "type": "string",
        "level": 30,
        "parent": "table",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": true,
        "matcher": "org.apache.ranger.plugin.resourcematcher.RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "true",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "",
        "label": "Hive Column",
        "description": "Hive Column"
    }],
    "accessTypes": [{
        "itemId": 1,
        "name": "select",
        "label": "select",
        "impliedGrants": []
    }, {
        "itemId": 2,
        "name": "update",
        "label": "update",
        "impliedGrants": []
    }, {
        "itemId": 3,
        "name": "create",
        "label": "Create",
        "impliedGrants": []
    }, {
        "itemId": 4,
        "name": "drop",
        "label": "Drop",
        "impliedGrants": []
    }, {
        "itemId": 5,
        "name": "alter",
        "label": "Alter",
        "impliedGrants": []
    }, {
        "itemId": 6,
        "name": "index",
        "label": "Index",
        "impliedGrants": []
    }, {
        "itemId": 7,
        "name": "lock",

```

```

        "label": "Lock",
        "impliedGrants": []
    }, {
        "itemId": 8,
        "name": "all",
        "label": "All",
        "impliedGrants": ["select", "update", "create", "drop", "alter", "index", "lock"]
    }],
    "policyConditions": [],
    "contextEnrichers": [],
    "enums": [],
    "dataMaskDef": {
        "maskTypes": [{
            "itemId": 1,
            "name": "MASK",
            "label": "Redact",
            "description": "Replace lowercase with 'x', uppercase with 'X', digits
with '0'",
            "transformer": "mask({col})",
            "dataMaskOptions": {}
        }, {
            "itemId": 2,
            "name": "MASK_SHOW_LAST_4",
            "label": "Partial mask: show last 4",
            "description": "Show last 4 characters; replace rest with 'x'",
            "transformer": "mask_show_last_n({col}, 4, 'x', 'x', 'x', -1, '1')",
            "dataMaskOptions": {}
        }, {
            "itemId": 3,
            "name": "MASK_SHOW_FIRST_4",
            "label": "Partial mask: show first 4",
            "description": "Show first 4 characters; replace rest with 'x'",
            "transformer": "mask_show_first_n({col}, 4, 'x', 'x', 'x', -1, '1')",
            "dataMaskOptions": {}
        }, {
            "itemId": 4,
            "name": "MASK_HASH",
            "label": "Hash",
            "description": "Hash the value",
            "transformer": "mask_hash({col})",
            "dataMaskOptions": {}
        }, {
            "itemId": 5,
            "name": "MASK_NULL",
            "label": "Nullify",
            "description": "Replace with NULL",
            "dataMaskOptions": {}
        }, {
            "itemId": 6,
            "name": "MASK_NONE",
            "label": "Unmasked (retain original value)",
            "description": "No masking",
            "dataMaskOptions": {}
        }, {
            "itemId": 12,
            "name": "MASK_DATE_SHOW_YEAR",
            "label": "Date: show only year",
            "description": "Date: show only year",
            "transformer": "mask({col}, 'x', 'x', 'x', -1, '1', 1, 0, -1)",
            "dataMaskOptions": {}
        }, {
            "itemId": 13,
            "name": "CUSTOM",
            "label": "Custom",
            "description": "Custom",
            "dataMaskOptions": {}
        }
    ],
    "accessTypes": [{
        "itemId": 1,
        "name": "select",
        "label": "select",

```

```

        "impliedGrants": []
    }},
    "resources": [{
        "itemId": 1,
        "name": "database",
        "type": "string",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "false",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "{ \"singleValue\":true }",
        "label": "Hive Database",
        "description": "Hive Database"
    }, {
        "itemId": 2,
        "name": "table",
        "type": "string",
        "level": 20,
        "parent": "database",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "false",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "{ \"singleValue\":true }",
        "label": "Hive Table",
        "description": "Hive Table"
    }, {
        "itemId": 4,
        "name": "column",
        "type": "string",
        "level": 30,
        "parent": "table",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
            "wildCard": "false",
            "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "{ \"singleValue\":true }",
        "label": "Hive Column",
        "description": "Hive Column"
    }
    ]
},
"rowFilterDef": {
    "accessTypes": [{
        "itemId": 1,
        "name": "select",
        "label": "select",

```

```

    "impliedGrants": [
    ],
    "resources": [
      {
        "itemId": 1,
        "name": "database",
        "type": "string",
        "level": 10,
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
          "wildCard": "false",
          "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "{ \"singleValue\":true }",
        "label": "Hive Database",
        "description": "Hive Database"
      }, {
        "itemId": 2,
        "name": "table",
        "type": "string",
        "level": 20,
        "parent": "database",
        "mandatory": true,
        "lookupSupported": true,
        "recursiveSupported": false,
        "excludesSupported": false,
        "matcher": "org.apache.ranger.plugin.resourcematcher.
RangerDefaultResourceMatcher",
        "matcherOptions": {
          "wildCard": "false",
          "ignoreCase": "true"
        },
        "validationRegex": "",
        "validationMessage": "",
        "uiHint": "{ \"singleValue\":true }",
        "label": "Hive Table",
        "description": "Hive Table"
      }
    ]
  }
}

```

# Service APIs

## Get Service by id

API Name	Get Service
Request Type	GET
Request URL	service/public/v2/api/service/{id}
Request Params	

Response	<p>• HDFS Example Response:</p> <pre> {   "id": 3,   "guid": "e5d7f3d3-86ad-40ee-a5a8-1657834f0617",   "isEnabled": true,   "createdBy": "amb_ranger_admin",   "createTime": 1476568614715,   "updateTime": 1477112880894,   "version": 2,   "type": "hdfs",   "name": "c1l_hadoop",   "description": "hdfs repo",   "configs": {     "tag.download.auth.users": "hdfs",     "password": "*****",     "policy.download.auth.users": "hdfs",     "hadoop.security.authentication": "simple",     "hadoop.rpc.protection": "authentication",     "hadoop.security.authorization": "false",     "fs.default.name": "hdfs://localhost.localdomain:8020",     "hadoop.security.auth_to_local": "DEFAULT",     "ambari.service.check.user": "ambari-qa",     "username": "hadoop"   },   "policyVersion": 4,   "policyUpdateTime": 1477513761335,   "tagVersion": 1,   "tagUpdateTime": 1476543414721 } </pre>
----------	--

## Get Service by name

API Name	Get Service
Request Type	GET
Request URL	service/public/v2/api/service/name/{name}
Request Params	

<b>Response</b>	<b>• Hive Example :</b> <pre> {     "id": 1,     "guid": "279e3d18-f45a-4908-8910-38cc758d3a31",     "isEnabled": true,     "createdBy": "amb_ranger_admin",     "updatedBy": "Admin",     "createTime": 1472533426778,     "updateTime": 1476405850463,     "version": 3,     "type": "hive",     "name": "c1l_hive",     "description": "hive repo",     "tagService": "test_for_ancestor_match",     "configs": {         "tag.download.auth.users": "hive",         "password": "*****",         "policy.download.auth.users": "hive",         "policy.grantrevoke.auth.users": "hive",         "jdbc.driverClassName": "org.apache.hive.jdbc.HiveDriver",         "jdbc.url": "jdbc:hive2://localhost.localdomain:10000",         "ambari.service.check.user": "ambari-qa",         "username": "hive"     },     "policyVersion": 60,     "policyUpdateTime": 1476529340066,     "tagVersion": 2,     "tagUpdateTime": 1476407292853 } </pre>
-----------------	---

## Create Service

<b>API Name</b>	<b>Create Service</b>
<b>Request Type</b>	<b>Post</b>
<b>Request URL</b>	<b>service/public/v2/api/service</b>
<b>Request Params</b>	<b>application/json</b>  <b>• Example:</b> <pre> {     "configs": {         "password": "*****",         "username": "hadoop"     },     "description": "test service",     "isEnabled": true,     "name": "c1l_test",     "type": "test",     "version": 1 } </pre>

<b>Response</b>	200-Application/json <div> <pre> {   "configs": {     "password": "*****",     "username": "hadoop"   },   "createTime": 1451348710255,   "createdBy": "Admin",   "description": "test service",   "guid": "e72cb64d-66d7-4632-b5ae-c6966cb52105",   "id": 6,   "isEnabled": true,   "name": "c1l_test",   "tagVersion": 1,   "type": "test",   "updateTime": 1451348710256,   "updatedBy": "Admin",   "version": 1 } </pre> </div>
-----------------	---

## Update Service by id

<b>API Name</b>	<b>Update Service</b>
<b>Request Type</b>	<b>PUT</b>
<b>Request URL</b>	<b>service/public/v2/api/service/{id}</b>
<b>Request Params</b>	<b>Application/json</b> <b>• Example:</b>
<b>Response</b>	200-Application/json

## Update Service by name

<b>API Name</b>	<b>Update Service</b>
<b>Request Type</b>	<b>PUT</b>
<b>Request URL</b>	<b>service/public/v2/api/service/name/{name}</b>
<b>Request Params</b>	<b>Application/json</b> <b>• Example:</b> <div> <pre> {   "configs": {     "password": "*****",     "username": "admin"   },   "description": "test service",   "isEnabled": true,   "name": "c1l_test",   "type": "test" } </pre> </div>



<b>Response</b>	200-Application/json <div> <pre> {   "configs": {     "password": "*****",     "username": "admin"   },   "createTime": 1451348710000,   "createdBy": "Admin",   "description": "test service",   "guid": "e72cb64d-66d7-4632-b5ae-c6966cb52105",   "id": 6,   "isEnabled": true,   "name": "c1l_test",   "policyUpdateTime": 1451351474000,   "policyVersion": 3,   "tagVersion": 1,   "type": "test",   "updateTime": 1451352016713,   "updatedBy": "Admin",   "version": 5 } </pre> </div>
-----------------	---

## Delete Service by id

<b>API Name</b>	<b>Delete Service</b>
<b>Request Type</b>	<b>DELETE</b>
<b>Request URL</b>	<b>service/public/v2/api/service/{id}</b>
<b>Request Param</b>	
<b>Response</b>	204-No Content

## Delete Service by name

<b>API Name</b>	<b>Delete Service</b>
<b>Request Type</b>	<b>DELETE</b>
<b>Request URL</b>	<b>service/public/v2/api/service/name/{name}</b>
<b>Request Param</b>	
<b>Response</b>	204-No Content

## Search Services

<b>API Name</b>	<b>Search Services</b>
<b>Request Type</b>	<b>GET</b>
<b>Request URL</b>	<b>service/public/v2/api/service</b>

Request Params	<p><b>Query Parameters:</b></p> <p>pageSize int <i>The page size required</i></p> <p>startIndex int <i>The start record index</i></p> <p>serviceName string <i>The service name</i></p> <p>serviceNamePartial string <i>Partial service name</i></p> <p>serviceType string <i>The service types(such as "hdfs","hive","hbase","knox","storm", "atlas")</i></p> <p>isEnabled boolean <i>The enabled status (true/false): true is enabled, false otherwise</i></p> <p>Example :</p> <p>pageSize=25&amp;startIndex=0</p>
Response	<p>200-Application/json</p> <p>• <b>Example:</b></p> <pre>[{   "id": 1,   "guid": "279e3d18-f45a-4908-8910-38cc758d3a31",   "isEnabled": true,   "createdBy": "amb_ranger_admin",   "updatedBy": "Admin",   "createTime": 1472533426778,   "updateTime": 1476405850463,   "version": 3,   "type": "hive",   "name": "c11_hive",   "description": "hive repo",   "tagService": "test_for_ancestor_match",   "configs": {     "tag.download.auth.users": "hive",     "password": "*****",     "policy.download.auth.users": "hive",     "policy.grantrevoke.auth.users": "hive",     "jdbc.driverClassName": "org.apache.hive.jdbc.HiveDriver",     "jdbc.url": "jdbc:hive2://localhost.localdomain:10000",     "ambari.service.check.user": "ambari-qa",     "username": "hive"   },   "policyVersion": 60,   "policyUpdateTime": 1476529340066,   "tagVersion": 2,   "tagUpdateTime": 1476407292853 }, {   "id": 2,   "guid": "497f3cc2-a924-4e05-9a0c-31051f76c5b9",   "isEnabled": true,   "createdBy": "Admin",   "updatedBy": "Admin",   "createTime": 1476405815786,   "updateTime": 1476405815802,   "version": 1,   "type": "tag",   "name": "test_for_ancestor_match",   "description": "test_for_ancestor_matching_tags",   "configs": {},   "policyVersion": 8,   "policyUpdateTime": 1476529474141,   "tagVersion": 1,   "tagUpdateTime": 1476380615835 }, {   "id": 3,   "guid": "e5d7f3d3-86ad-40ee-a5a8-1657834f0617",   "isEnabled": true,</pre>

```

    "createdBy": "amb_ranger_admin",
    "createTime": 1476568614715,
    "updateTime": 1477112880894,
    "version": 2,
    "type": "hdfs",
    "name": "c1l_hadoop",
    "description": "hdfs repo",
    "configs": {
      "tag.download.auth.users": "hdfs",
      "password": "*****",
      "policy.download.auth.users": "hdfs",
      "hadoop.security.authentication": "simple",
      "hadoop.rpc.protection": "authentication",
      "hadoop.security.authorization": "false",
      "fs.default.name": "hdfs://localhost.localdomain:8020",
      "hadoop.security.auth_to_local": "DEFAULT",
      "ambari.service.check.user": "ambari-qa",
      "username": "hadoop"
    },
    "policyVersion": 4,
    "policyUpdateTime": 1477513761335,
    "tagVersion": 1,
    "tagUpdateTime": 1476543414721
  }, {
    "id": 4,
    "guid": "f1b48310-b531-4e86-bfe7-a9a67d6f0d75",
    "isEnabled": true,
    "createdBy": "amb_ranger_admin",
    "createTime": 1477115428518,
    "updateTime": 1477378568635,
    "version": 2,
    "type": "hbase",
    "name": "c1l_hbase",
    "description": "hbase repo",
    "configs": {
      "hbase.zookeeper.quorum": "localhost.localdomain",
      "tag.download.auth.users": "hbase",
      "password": "*****",
      "policy.download.auth.users": "hbase",
      "hadoop.security.authentication": "simple",
      "hbase.security.authentication": "simple",
      "policy.grantrevoke.auth.users": "hbase",
      "hbase.zookeeper.property.clientPort": "2181",
      "zookeeper.znode.parent": "/hbase-unsecure",
      "ambari.service.check.user": "ambari-qa",
      "username": "hbase"
    },
    "policyVersion": 7,
    "policyUpdateTime": 1477511916809,
    "tagVersion": 1,
    "tagUpdateTime": 1477090228525
  }
}

```

## Policy APIs

### Get Policy by id

API Name	Get Policy
Request Type	Get
Request URL	service/public/v2/api/policy/{id}
Request Params	

Response	<p>• Example:</p> <p>200 - Application/json</p> <pre> {   "allowExceptions": [],   "createTime": 1450757397000,   "createdBy": "amb_ranger_admin",   "denyExceptions": [],   "denyPolicyItems": [],     "dataMaskPolicyItems": [],     "rowFilterPolicyItems": [],   "description": "Default Policy for Service: cll_hadoop",   "guid": "4c2f7afb-23fa-45e9-9b41-29bdc7423b65",   "id": 1,   "isAuditEnabled": true,   "isEnabled": true,   "name": "cll_hadoop-1-20151222040957",   "policyItems": [     {       "accesses": [         {           "isAllowed": true,           "type": "read"         },         {           "isAllowed": true,           "type": "write"         },         {           "isAllowed": true,           "type": "execute"         }       ],       "conditions": [],       "delegateAdmin": true,       "groups": [],       "users": [         "ambari-qa"       ]     }   ],   "resourceSignature": "6f956063401eda656f1eae8870c1afac",   "resources": {     "path": {       "isExcludes": false,       "isRecursive": true,       "values": [         "/*"       ]     }   },   "service": "cll_hadoop",   "updateTime": 1450757398000,   "updatedBy": "amb_ranger_admin",   "version": 2 } </pre>
----------	--

## Search Policies

API Name	Search Policies
Request Type	Get
Request URL	service/public/v2/api/policy

Request Params	<p><u>Query Parameters:</u></p> <p>pageSize int <i>The page size required</i></p> <p>startIndex int <i>The start record index</i></p> <p>serviceType string <i>Name of the component (such as "hbase", "hdfs", etc)</i></p> <p>serviceTypeid int <i>Id of the component (such as 2 for "hbase", 1 for "hdfs", etc)</i></p> <p>serviceName string <i>The exact name of the service</i></p> <p>serviceid int <i>Id of the service</i></p> <p>policyName string <i>The exact Name of the policy</i></p> <p>policyNamePartial string <i>Partial name of the policy</i></p> <p>policyId int <i>The policy ID</i></p> <p>policyType int <i>Type of policy (0 - access, 1 - data-mask, 2 - row-filter)</i></p> <p>polResource string <i>The policy resource value. It may appear at any level (e.g. in hbase policy, as table, column-family or column value)</i></p> <p>resource:resource-type string <i>The policy resource value for given resource-type (e.g. for hive policy, database, udf, table or column)</i></p> <p>user string <i>The user name</i></p> <p>group string <i>The group name</i></p> <p>isRecursive boolean <i>The isRecursive property ("true" or "false")</i></p> <p>isEnabled boolean <i>The enable/disabled property ("true" or "false")</i></p> <p>Example =</p> <p>pageSize=25&amp;startIndex=0&amp;polResource=finance&amp;isEnabled=true</p>
-------------------	---

<b>Response</b>	200 - Application/json <div> <pre> [ {   "id": 114,   "isEnabled": true,   "version": 6,   "service": "c11_hive",   "name": "mask out finance.invoice.id",   "policyType": 1,   "description": "",   "isAuditEnabled": true,   "resources": {     "database": {       "values": ["finance"],       "isExcludes": false,       "isRecursive": false     },     "column": {       "values": ["id"],       "isExcludes": false,       "isRecursive": false     },     "table": {       "values": ["invoice_1"],       "isExcludes": false,       "isRecursive": false     }   },   "policyItems": [],   "denyPolicyItems": [],   "allowExceptions": [],   "denyExceptions": [],   "dataMaskPolicyItems": [ {     "accesses": [ {       "type": "select",       "isAllowed": true     } ],     "users": ["hive"],     "groups": [],     "conditions": [],     "delegateAdmin": false,     "dataMaskInfo": {       "dataMaskType": "MASK_NULL",       "conditionExpr": "",       "valueExpr": ""     }   } ],   "rowFilterPolicyItems": [] } ] </pre> </div>
-----------------	---

## Get Policy by service-name and policy-name

<b>API Name</b>	<b>Get Policy</b>
<b>Request Type</b>	<b>Get</b>
<b>Request URL</b>	<b>service/public/v2/api/service/{service-name}/policy/{policy-name}</b>
<b>Request Params</b>	
<b>Response</b>	<ul style="list-style-type: none"> <li><b>Hive Example Policy:</b></li> </ul> 200 - Application/json

```

{
  "allowExceptions": [],
  "createTime": 1450757992000,
  "createdBy": "amb_ranger_admin",
  "denyExceptions": [],
  "denyPolicyItems": [],
    "dataMaskPolicyItems": [],
    "rowFilterPolicyItems": [],
  "description": "Default Policy for Service: cll_hive",
  "guid": "d6218120-1b66-43e6-9fef-9c917a8e9e25",
  "id": 4,
  "isAuditEnabled": true,
  "isEnabled": true,
  "name": "cll_hive-2-20151222041952",
  "policyItems": [
    {
      "accesses": [
        {
          "isAllowed": true,
          "type": "select"
        },
        {
          "isAllowed": true,
          "type": "update"
        },
        {
          "isAllowed": true,
          "type": "create"
        },
        {
          "isAllowed": true,
          "type": "drop"
        },
        {
          "isAllowed": true,
          "type": "alter"
        },
        {
          "isAllowed": true,
          "type": "index"
        },
        {
          "isAllowed": true,
          "type": "lock"
        },
        {
          "isAllowed": true,
          "type": "all"
        }
      ],
      "conditions": [],
      "delegateAdmin": true,
      "groups": [],
      "users": [
        "ambari-qa"
      ]
    }
  ],
  "resourceSignature": "c834ed2b8c7462d2aa8bbffdb05226c8",
  "resources": {
    "database": {
      "isExcludes": false,
      "isRecursive": false,
      "values": [
        ""
      ]
    },
    "udf": {
      "isExcludes": false,

```

```
        "isRecursive": false,  
        "values": [  
            "*"   
        ]  
    },  
    "service": "c11_hive",  
    "updateTime": 1450757995000,  
    "updatedBy": "amb_ranger_admin",  
    "version": 2  
}
```

## Create Policy

API name	Create Policy
Request Type	POST
Request URL	service/public/v2/api/policy



**Request Params**

Application/json

```
{
  "allowExceptions": [],
  "denyExceptions": [],
    "dataMaskPolicyItems": [],
    "rowFilterPolicyItems": [],
  "denyPolicyItems": [],
  "description": "Policy for Service: cll_test",
  "isAuditEnabled": true,
  "isEnabled": true,
  "name": "cll_test-1",
  "policyItems": [
    {
      "accesses": [
        {
          "isAllowed": true,
          "type": "select"
        },
        {
          "isAllowed": true,
          "type": "update"
        },
        {
          "isAllowed": true,
          "type": "create"
        },
        {
          "isAllowed": true,
          "type": "drop"
        }
      ],
      "conditions": [],
      "delegateAdmin": true,
      "groups": ["public"],
      "users": [
      ]
    }
  ],
  "resources": {
    "root": {
      "isExcludes": false,
      "isRecursive": false,
      "values": [
        "abc"
      ]
    },
    "sub": {
      "isExcludes": false,
      "isRecursive": false,
      "values": [
        "*"
      ]
    }
  },
  "service": "cll_test",
  "version": 1
}
```

**Response**

200 - Application/json

```
{
  "allowExceptions": [],
  "createTime": 1451350456093,
  "createdBy": "Admin",
  "denyExceptions": [],
  "denyPolicyItems": [],
    "dataMaskPolicyItems": [],
    "rowFilterPolicyItems": [],
  "description": "Policy for Service: cll_test",
  "guid": "ff0b3c4a-6aa0-4803-9314-17f3b8950482",
  "id": 8,
  "isAuditEnabled": true,
  "isEnabled": true,
  "name": "cll_test-1",
  "policyItems": [
    {
      "accesses": [
        {
          "isAllowed": true,
          "type": "select"
        },
        {
          "isAllowed": true,
          "type": "update"
        },
        {
          "isAllowed": true,
          "type": "create"
        },
        {
          "isAllowed": true,
          "type": "drop"
        }
      ],
      "conditions": [],
      "delegateAdmin": true,
      "groups": [
        "public"
      ],
      "users": []
    }
  ],
  "resourceSignature": "8a2fac99ba72c687defacff39d6354fb",
  "resources": {
    "root": {
      "isExcludes": false,
      "isRecursive": false,
      "values": [
        "abc"
      ]
    },
    "sub": {
      "isExcludes": false,
      "isRecursive": false,
      "values": [
        "*"
      ]
    }
  },
  "service": "cll_test",
  "updateTime": 1451350456094,
  "updatedBy": "Admin",
  "version": 1
}
```

# Apply Policy

Create or update policy as specified by the policy specification in the argument. Input policy specification should not contain any Ranger conditions.

This API is used to update an existing Ranger policy that exactly matches resource specified in the input policy specification. If such policy does not exist, then it is created according to input policy specification and returned. If such policy exists, then its policy items are updated and/or created according to the input policy specification.

API Name	Apply policy
Request Type	POST
Request URL	service/public/v2/api/policy/apply
Existing Policy	<pre>{   "id": 11,   "isEnabled": true,   "version": 1,   "service": "c11_hadoop",   "name": "apply-policy-test",   "policyType": 0,   "isAuditEnabled": true,   "resources": {     "path": {       "values": ["/test"],       "isExcludes": false,       "isRecursive": true     }   },   "policyItems": [{     "accesses": [{       "type": "read",       "isAllowed": true     }],     "users": ["hive"],     "groups": [],     "conditions": [],     "delegateAdmin": false   }],   "denyPolicyItems": [],   "allowExceptions": [],   "denyExceptions": [],   "dataMaskPolicyItems": [],   "rowFilterPolicyItems": [] }</pre>

Request Params	Application/json
	<pre> {   "service": "c11_hadoop",   "name": "apply_policy_test_1",   "isAuditEnabled": true,   "resources": {     "path": {       "values": [         "/test"       ]     }   },   "policyItems": [{     "users": [       "ambari-qa"     ],     "groups": [],     "delegateAdmin": false,     "accesses": [{       "isAllowed": true,       "type": "execute"     }, {       "isAllowed": true,       "type": "write"     }, {       "isAllowed": true,       "type": "read"     }],     "conditions": []   }, {     "users": [       "hdfs"     ],     "groups": [       "hdfs"     ],     "delegateAdmin": true,     "accesses": [{       "isAllowed": true,       "type": "read"     }, {       "isAllowed": true,       "type": "write"     }, {       "isAllowed": true,       "type": "execute"     }],     "conditions": []   }] } </pre>
Response	200 - Application/json
	<pre> {   "id": 11,   "guid": "f59b4a52-f0b3-4c83-a6b1-2e2c932f0772",   "isEnabled": true,   "createdBy": "Admin",   "updatedBy": "Admin",   "createTime": 1477537351264,   "updateTime": 1477538961259,   "version": 2,   "service": "c11_hadoop",   "name": "apply-policy-test",   "policyType": 0, </pre>

```

"description": "Test policy",
"resourceSignature": "e2bcd0e02c24933b2e57c697555e877",
"isAuditEnabled": true,
"resources": {
    "path": {
        "values": ["/test"],
        "isExcludes": false,
        "isRecursive": true
    }
},
"policyItems": [{
    "accesses": [{
        "type": "execute",
        "isAllowed": true
    }, {
        "type": "write",
        "isAllowed": true
    }, {
        "type": "read",
        "isAllowed": true
    }],
    "users": ["ambari-qa"],
    "groups": [],
    "conditions": [],
    "delegateAdmin": false
}, {
    "accesses": [{
        "type": "read",
        "isAllowed": true
    }],
    "users": ["hive"],
    "groups": [],
    "conditions": [],
    "delegateAdmin": false
}, {
    "accesses": [{
        "type": "read",
        "isAllowed": true
    }, {
        "type": "write",
        "isAllowed": true
    }, {
        "type": "execute",
        "isAllowed": true
    }],
    "users": ["hdfs"],
    "groups": ["hdfs"],
    "conditions": [],
    "delegateAdmin": true
}],
"denyPolicyItems": [],
"allowExceptions": [],
"denyExceptions": [{
    "accesses": [{
        "type": "execute",
        "isAllowed": true
    }, {
        "type": "write",
        "isAllowed": true
    }, {
        "type": "read",
        "isAllowed": true
    }],
    "users": ["ambari-qa"],
    "groups": [],
    "conditions": [],
    "delegateAdmin": false
}, {
    "accesses": [{
        "type": "read",
        "isAllowed": true

```

	<pre>         }, {             "type": "write",             "isAllowed": true         }, {             "type": "execute",             "isAllowed": true         }],         "users": ["hdfs"],         "groups": ["hdfs"],         "conditions": [],         "delegateAdmin": true     }],     "dataMaskPolicyItems": [],     "rowFilterPolicyItems": [] } </pre>
--	---

### Update Policy by id

<b>API Name</b>	<b>update policy</b>
<b>Request Type</b>	<b>PUT</b>
<b>Request URL</b>	<b>service/public/v2/api/policy/{id}</b>
<b>Request Params</b>	Application /json <div> <pre> {   "id": 8,   "allowExceptions": [],   "denyExceptions": [],   "dataMaskPolicyItems": [],   "rowFilterPolicyItems": [],   "denyPolicyItems": [     {       "accesses": [         {           "isAllowed": true,           "type": "drop"         }       ],       "conditions": [],       "delegateAdmin": true,       "groups": [],       "users": [         "admin"       ]     }   ],   "description": "Policy for Service: cll_test",   "isAuditEnabled": true,   "isEnabled": true,   "name": "cll_test-1",   "policyItems": [     {       "accesses": [         {           "isAllowed": true,           "type": "select"         },         {           "isAllowed": true,           "type": "update"         }       ],       {         "isAllowed": true, </pre> </div>

	<pre>        "type": "create"       },       {         "isAllowed": true,         "type": "drop"       }     ],     "conditions": [],     "delegateAdmin": true,     "groups": ["public"],     "users": [     ]   } ], "resources": {   "root": {     "isExcludes": false,     "isRecursive": false,     "values": [       "abc"     ]   },   "sub": {     "isExcludes": false,     "isRecursive": false,     "values": [       "*"     ]   } }, "service": "c11_test", "version": 1 }</pre>
Response	<div>200 - Application/json</div> <div><pre>{   "allowExceptions": [],   "createTime": 1451350456000,   "createdBy": "Admin",   "denyExceptions": [],   "denyPolicyItems": [     {       "accesses": [         {           "isAllowed": true,           "type": "drop"         }       ],       "conditions": [],       "delegateAdmin": true,       "groups": [],       "users": [         "admin"       ]     }   ],   "dataMaskPolicyItems": [],   "rowFilterPolicyItems": [],   "description": "Policy for Service: c11_test",   "guid": "ff0b3c4a-6aa0-4803-9314-17f3b8950482",   "id": 8,   "isAuditEnabled": true,   "isEnabled": true,   "name": "c11_test-1",   "policyItems": [     { </pre></div>

```

        "accesses": [
            {
                "isAllowed": true,
                "type": "select"
            },
            {
                "isAllowed": true,
                "type": "update"
            },
            {
                "isAllowed": true,
                "type": "create"
            },
            {
                "isAllowed": true,
                "type": "drop"
            }
        ],
        "conditions": [],
        "delegateAdmin": true,
        "groups": [
            "public"
        ],
        "users": []
    }
},
"resourceSignature": "8a2fac99ba72c687defacff39d6354fb",
"resources": {
    "root": {
        "isExcludes": false,
        "isRecursive": false,
        "values": [
            "abc"
        ]
    },
    "sub": {
        "isExcludes": false,
        "isRecursive": false,
        "values": [
            "*"
        ]
    }
},
"service": "c11_test",
"updateTime": 1451955041580,
"updatedBy": "Admin",
"version": 3
}

```

## Update Policy by service-name and policy-name

API Name	update policy
Request Type	PUT
Request URL	service/public/v2/api/service/{service-name}/policy/{policy-name}
Request Params	<p>• Example:</p> <p>Application /json</p> <pre> {     "allowExceptions": [],     "denyExceptions": [], </pre>



```

"denyPolicyItems": [
  {
    "accesses": [
      {
        "isAllowed": true,
        "type": "drop"
      }
    ],
    "conditions": [],
    "delegateAdmin": true,
    "groups": [],
    "users": [
      "hadoop",
      "admin"
    ]
  }
],
"dataMaskPolicyItems": [],
"rowFilterPolicyItems": [],
"description": "Policy for Service: cll_test",
"isAuditEnabled": true,
"isEnabled": true,
"name": "cll_test-1",
"policyItems": [
  {
    "accesses": [
      {
        "isAllowed": true,
        "type": "select"
      },
      {
        "isAllowed": true,
        "type": "update"
      },
      {
        "isAllowed": true,
        "type": "create"
      },
      {
        "isAllowed": true,
        "type": "drop"
      }
    ],
    "conditions": [],
    "delegateAdmin": true,
    "groups": ["public"],
    "users": [
    ]
  }
],
"resources": {
  "root": {
    "isExcludes": false,
    "isRecursive": false,
    "values": [
      "abc"
    ]
  },
  "sub": {
    "isExcludes": false,
    "isRecursive": false,
    "values": [
      "*"
    ]
  }
},
"service": "cll_test",
"version": 1
}

```

**Response**

200 - Application/json

```
{
  "allowExceptions": [],
  "createTime": 1451350456000,
  "createdBy": "Admin",
  "denyExceptions": [],
  "denyPolicyItems": [
    {
      "accesses": [
        {
          "isAllowed": true,
          "type": "drop"
        }
      ],
      "conditions": [],
      "delegateAdmin": true,
      "groups": [],
      "users": [
        "hadoop",
        "admin"
      ]
    }
  ],
  "dataMaskPolicyItems": [],
  "rowFilterPolicyItems": [],
  "description": "Policy for Service: cll_test",
  "guid": "ff0b3c4a-6aa0-4803-9314-17f3b8950482",
  "id": 8,
  "isAuditEnabled": true,
  "isEnabled": true,
  "name": "cll_test-1",
  "policyItems": [
    {
      "accesses": [
        {
          "isAllowed": true,
          "type": "select"
        },
        {
          "isAllowed": true,
          "type": "update"
        },
        {
          "isAllowed": true,
          "type": "create"
        },
        {
          "isAllowed": true,
          "type": "drop"
        }
      ],
      "conditions": [],
      "delegateAdmin": true,
      "groups": [
        "public"
      ],
      "users": []
    }
  ],
  "resourceSignature": "8a2fac99ba72c687defacff39d6354fb",
  "resources": {
    "root": {
      "isExcludes": false,
      "isRecursive": false,
      "values": [
```

```

        "abc"
    ]
},
"sub": {
    "isExcludes": false,
    "isRecursive": false,
    "values": [
        "*"
    ]
}
},
"service": "c11_test",
"updateTime": 1451352707567,
"updatedBy": "Admin",
"version": 2
}

```

## Delete policy by id

API Name	Delete Policy
Request Type	DELETE
Request URL	service/public/v2/api/policy/{id}
Request Params	
Response	204 - No Content

## Delete policy by service-name and policy-name

API Name	Delete Policy
Request Type	DELETE
Request URL	service/public/v2/api/policy
Request Params	<p><u>Query Parameters:</u></p> <p>servicename string The name of service</p> <p>polycyname string The name of policy</p> <p>Example:</p> <p>servicename=service-name&amp;polycyname=policy-name</p>
Response	204 - No Content

## Search Policies in a Service

API Name	Search Policies in a Service
API Name	Search Policies in a Service
Request Type	GET
Request URL	service/public/v2/api/service/{service-name}/policy

<b>Request Params</b>	<p><u>Query Parameters:</u></p> <p>pageSize int <i>The page size required</i></p> <p>startIndex int <i>The start record index</i></p> <p>serviceType string <i>Name of the component (such as "hbase", "hdfs", etc)</i></p> <p>serviceTypeId int <i>Id of the component (such as 2 for "hbase", 1 for "hdfs", etc)</i></p> <p>serviceName string <i>The exact name of the service</i></p> <p>serviceId int <i>Id of the service</i></p> <p>policyName string <i>The exact Name of the policy</i></p> <p>policyNamePartial string <i>Partial name of the policy</i></p> <p>policyId int <i>The policy ID</i></p> <p>policyType int <i>Type of policy (0 - access, 1 - data-mask, 2 - row-filter)</i></p> <p>polResource string <i>The policy resource value. It may appear at any level (e.g. in hbase policy, as table, column-family or column value)</i></p> <p>resource:resource-type string <i>The policy resource value for given resource-type (e.g. for hive policy, database, udf, table or column)</i></p> <p>user string <i>The user name</i></p> <p>group string <i>The group name</i></p> <p>isRecursive boolean <i>The isRecursive property ("true" or "false")</i></p> <p>isEnabled boolean <i>The enable/disabled property ("true" or "false")</i></p> <p>Example =</p> <p>pageSize=25&amp;startIndex=0&amp;resource:database=finance</p>
<b>Response</b>	<p>200 - Application/json</p> <p><b>Example:</b></p> <pre>[   {     "allowExceptions": [],     "createTime": 1450757992000,     "createdBy": "amb_ranger_admin",     "denyExceptions": [],     "dataMaskPolicyItems": [],     "rowFilterPolicyItems": [],     "denyPolicyItems": [],     "description": "Default Policy for Service: c11_hive",     "guid": "4a322a05-c17f-4d6c-b291-94cae3e6c353",     "id": 3,     "isAuditEnabled": true,     "isEnabled": true,     "name": "c11_hive-1-20151222041951",     "policyItems": [       {         "accesses": [           {             "isAllowed": true,             "type": "select"           },           {             "isAllowed": true,             "type": "update"           },           {             "isAllowed": true,             "type": "create"           }         ]       }     ]   } ]</pre>

```

        },
        {
            "isAllowed": true,
            "type": "drop"
        },
        {
            "isAllowed": true,
            "type": "alter"
        },
        {
            "isAllowed": true,
            "type": "index"
        },
        {
            "isAllowed": true,
            "type": "lock"
        },
        {
            "isAllowed": true,
            "type": "all"
        }
    ],
    "conditions": [],
    "delegateAdmin": true,
    "groups": [],
    "users": [
        "ambari-qa"
    ]
}
],
"resourceSignature": "6e79c1c989c79b7e53af663d3bdc2de6",
"resources": {
    "column": {
        "isExcludes": false,
        "isRecursive": false,
        "values": [
            ""
        ]
    },
    "database": {
        "isExcludes": false,
        "isRecursive": false,
        "values": [
            ""
        ]
    },
    "table": {
        "isExcludes": false,
        "isRecursive": false,
        "values": [
            ""
        ]
    }
},
"service": "c11_hive",
"updateTime": 1450757994000,
"updatedBy": "amb_ranger_admin",
"version": 2
},
{
    "allowExceptions": [],
    "createTime": 1450757992000,
    "createdBy": "amb_ranger_admin",
    "denyExceptions": [],
    "denyPolicyItems": [],
    "dataMaskPolicyItems": [],
    "rowFilterPolicyItems": [],
    "description": "Default Policy for Service: c11_hive",
    "guid": "d6218120-1b66-43e6-9fef-9c917a8e9e25",
    "id": 4,
    "isAuditEnabled": true,

```

```

        "isEnabled": true,
        "name": "c11_hive-2-20151222041952",
        "policyItems": [
            {
                "accesses": [
                    {
                        "isAllowed": true,
                        "type": "select"
                    },
                    {
                        "isAllowed": true,
                        "type": "update"
                    },
                    {
                        "isAllowed": true,
                        "type": "create"
                    },
                    {
                        "isAllowed": true,
                        "type": "drop"
                    },
                    {
                        "isAllowed": true,
                        "type": "alter"
                    },
                    {
                        "isAllowed": true,
                        "type": "index"
                    },
                    {
                        "isAllowed": true,
                        "type": "lock"
                    },
                    {
                        "isAllowed": true,
                        "type": "all"
                    }
                ],
                "conditions": [],
                "delegateAdmin": true,
                "groups": [],
                "users": [
                    "ambari-qa"
                ]
            }
        ],
        "resourceSignature": "c834ed2b8c7462d2aa8bbffdb05226c8",
        "resources": {
            "database": {
                "isExcludes": false,
                "isRecursive": false,
                "values": [
                    "finance"
                ]
            },
            "udf": {
                "isExcludes": false,
                "isRecursive": false,
                "values": [
                    ""
                ]
            }
        },
        "service": "c11_hive",
        "updateTime": 1450757995000,
        "updatedBy": "amb_ranger_admin",
        "version": 2
    }
}
]

```

