

API Documentation Sample

This document contains two parts. Part 1 contains two screenshots: GET Employee Rule by ID and a POST Employee Rule by ID command. Part 2 contains the entire sample.

Part 1: API Screenshots

1. GET Employee Rule by ID

Changelog

GET Employee Rule by ID

GET

http://localhost:3000/employee/{employeeid}

GET by employeeid {

returns employee JSON body

Request

Path Parameters

employeeid Integer

required

POST calls to ID 0 will create a new rule POST calls to ID != 0 will save data for an existing rule for partial updates.

Responses

200

OK

Body

application/json

id Integer

required

fname string

required

lname string

required

email string<email>

required

isactive boolean

required

title string

required

Parameters

employeeid* : Integer

Send API Request

Local host

Request Sample: Shell / cURL

```
curl --request GET \
--url http://localhost:3000/employee/employeeid \
--header 'Accept: application/json'
```

Response Example

```
1 {
2   "id": 0,
3   "fname": "Geoffrey",
4   "lname": "Reynolds",
5   "email": "greynolds@example.com",
6   "isactive": true,
7   "title": "Senior Technical Writer",
8   "DoB": "2019-08-24"
9 }
```

2. POST Employee Rule by ID

[Changelog](#)

POST Employee Rule by ID

POST `http://localhost:3000/employee/{employeeid}`

POST to `employeeid === 0` {
creates a *new* Employee record

POST to `employeeid !== 0` {
updates existing record and *requires* complete JSON body

Request

Path Parameters

employeeid integer required
POST calls to ID 0 will create a new rule POST calls to ID != 0 will save data for an existing rule for partial updates.

Body

application/json

Sample of all fields provided by JSON sample.

*Employeeid {
is a record locator.
array of:

<code>id</code>	integer	required
<code>fname</code>	string	required
<code>lname</code>	string	required
<code>email</code>	string<email>	required
<code>isactive</code>	boolean	required
<code>title</code>	string	required
<code>department</code>	string	required
<code>created</code>	string<date-time>	required

Parameters

employeeid* : integer

Body

1 []

Send API Request Local host

Request Sample: Shell / cURL

```
curl --request POST \
--url http://localhost:3000/employee/employeeid \
--header 'Accept: application/json, application/xml' \
--header 'Content-Type: application/json' \
--data '[]'
```

Response Example

```
1 {
2   "id": 0,
3   "fname": "Geoffrey",
4   "lname": "Reynolds",
5   "email": "greynolds@example.com",
6   "isactive": true,
7   "title": "Senior Technical Writer",
8   "DoB": "2019-08-24",
9   "created": "2019-08-24"
10 }
```

Responses

201

404

Created

Headers

Requested Employee ID string
Sample description

Body

application/json ▾

—	id	integer
—	lname	string
—	email	string<email>
—	DoB	string
—	isactive	boolean
—	department	string
—	created	string

Part 2: *API Code*

```
{
  "openapi": "3.1.0",
  "x-stoplight": {
    "id": "zq5wqfelgau0y"
  },
  "info": {
    "title": "Parts Authority API Sample 1",
    "version": "1.0",
    "summary": "Employee Information",
    "description": "This API provides REST endpoints for the creation, modification, and
deletion of employee records including id, status and demographic information.",
    "contact": {
      "name": "Geoff Reynolds",
      "url": "https://www.linkedin.com/in/drgeoffreyreynolds/",
      "email": "admin@gmail.com"
    },
    "license": {
      "name": "Apache",
      "url": "https://apache.org"
    },
    "termsOfService": "123.com/tos"
  },
  "servers": [
    {
      "url": "http://localhost:3000",
      "description": "Local host"
    }
  ],
  "paths": {
    "/employee/{employeeid}": {
      "parameters": [
        {
          "schema": {
            "format": "uuid",
            "type": "integer"
          },
          "name": "employeeid",
          "in": "path",
          "description": "POST calls to ID 0 will create a new rule POST calls to ID != 0 will
save data for an existing rule for partial updates.",
          "required": true
        }
      ]
    }
  }
}
```

```
],  
"get": {  
  "summary": "GET Employee Rule by ID",  
  "operationId": "get-employee-employeeid",  
  "responses": {  
    "200": {  
      "description": "OK",  
      "content": {  
        "application/json": {  
          "schema": {  
            "type": "object",  
            "properties": {  
              "id": {  
                "type": "integer"  
              },  
              "fname": {  
                "type": "string"  
              },  
              "lname": {  
                "type": "string"  
              },  
              "email": {  
                "type": "string",  
                "format": "email"  
              },  
              "isactive": {  
                "type": "boolean"  
              },  
              "title": {  
                "type": "string"  
              },  
              "DoB": {  
                "type": "string",  
                "format": "date"  
              },  
              "department": {  
                "type": "string"  
              },  
              "created": {  
                "type": "string",  
                "format": "date-time"  
              }  
            }  
          },  
          "required": [  

```

```

        "id",
        "fname",
        "lname",
        "email",
        "isActive",
        "title",
        "DoB"
    ]
},
"examples": {
    "example-1": {
        "value": {
            "id": 0,
            "fname": "Geoffrey",
            "lname": "Reynolds",
            "email": "greynolds@example.com",
            "isActive": true,
            "title": "Senior Technical Writer",
            "DoB": "2019-08-24"
        }
    }
}
}
}
}
},
"description": "***GET by employeeid** { <br>\n*returns* employee JSON body"
},
"post": {
    "summary": "POST Employee Rule by ID",
    "operationId": "post-employee-employeeid",
    "responses": {
        "201": {
            "description": "Created",
            "content": {
                "application/json": {
                    "schema": {
                        "type": "object",
                        "properties": {
                            "id": {
                                "type": "integer"
                            },
                            "lname": {
                                "type": "string"
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```
    },
    "email": {
      "type": "string",
      "format": "email"
    },
    "DoB": {
      "type": "string"
    },
    "isactive": {
      "type": "boolean"
    },
    "department": {
      "type": "string"
    },
    "created": {
      "type": "string"
    }
  }
},
"examples": {
  "example-1": {
    "value": {
      "id": 0,
      "fname": "Geoffrey",
      "lname": "Reynolds",
      "email": "greynolds@example.com",
      "isactive": true,
      "title": "Senior Technical Writer",
      "DoB": "2019-08-24",
      "created": "2019-08-24"
    }
  }
}
},
"application/xml": {
  "schema": {
    "type": "array",
    "items": {
      "type": "object",
      "properties": {
        "": {
          "type": "string"
        }
      }
    }
  }
}
```

```

        }
      }
    },
    "headers": {
      "Requested Employee ID": {
        "schema": {
          "type": "string"
        },
        "description": "Sample description"
      }
    }
  },
  "404": {
    "description": "Not Found",
    "content": {
      "application/json": {
        "schema": {
          "type": "object",
          "properties": {}
        }
      }
    },
    "headers": {
      "Employee not found": {
        "schema": {
          "type": "string"
        },
        "description": "Make sure you entered the correct ID."
      }
    }
  },
  "requestBody": {
    "content": {
      "application/json": {
        "schema": {
          "type": "array",
          "items": {
            "$ref": "#/components/schemas/employeeid"
          }
        }
      },
      "examples": {
        "example-1": {

```



```

        "value": []
      }
    }
  },
  "description": "Sample of all fields provided by JSON sample."
},
"description": "***POST to employeeid === 0** { <br>\n creates a *new* Employee
record\n\n**POST to employeeid !== 0** { <br>\n *updates* existing record and *requires*
complete JSON body"
},
"delete": {
  "summary": "DELETE Employee ID",
  "operationId": "delete-employee-employeeid",
  "responses": {
    "200": {
      "description": "OK",
      "content": {
        "application/json": {
          "schema": {
            "type": "string"
          },
          "examples": {
            "Employee ID deleted.": {
              "value": "string"
            }
          }
        }
      }
    },
    "404": {
      "description": "Not Found"
    }
  },
  "description": "***DELETE by employeeid** { <br>\n*removes* matching employee id
record."
}
},
"components": {
  "schemas": {
    "employeeid": {
      "title": "employeeid",
      "x-stoplight": {

```

```
    "id": "pj6r9valkgn0y"
  },
  "description": "***Employeeid* ( </br>\nis a record locator.",
  "examples": [
    {
      "id": "string"
    }
  ],
  "type": "object",
  "properties": {
    "id": {
      "type": "integer"
    },
    "fname": {
      "type": "string"
    },
    "lname": {
      "type": "string"
    },
    "email": {
      "type": "string",
      "format": "email"
    },
    "isactive": {
      "type": "boolean"
    },
    "title": {
      "type": "string"
    },
    "department": {
      "type": "string"
    },
    "created": {
      "type": "string",
      "format": "date-time"
    }
  },
  "required": [
    "id",
    "fname",
    "lname",
    "email",
    "isactive",
    "title",
```

```
    "department",  
    "created"  
  ]  
}  
}  
}
```