

# Define the responses

In the previous topic, you added a [parameter](#) to your [operation](#), to send the airport code to the Airport-Data API in order to retrieve information about that airport. Now you will define the response you expect from the [API](#), including the format of the returned data. The returned data structure appears as configuration fields for this [action](#) in the Workflow designer.

Summary

Responses in the OpenAPI Specification

Define the responses

Step 1: Create the responses object

Step 2: Create a response

Step 3: Describe the response

Step 4: Define the response data

The OpenAPI Specification

Next steps

## Summary

A **response** is a reply from the API to a specific operation. It includes the **HTTP** status code, and a definition of any returned data.

## Responses in the OpenAPI Specification

Responses are defined in the `responses` object within each operation. You must use the status codes and variable names defined by the API.

Some APIs return a lot of data in their response. To avoid cluttering the Workflow designer configuration screens with fields, only define the data you plan to use. Anything you do not define will not be available to you in the Workflow designer.

## Define the responses

To define a response, you define:

- The HTTP status code that identifies the response.
- The description of the response. This is for documentation purposes only within the [OpenAPI Specification](#), and does not appear in the workflow.
- The definition of any data delivered by the response.

### Step 1: Create the responses object

Each operation in an OpenAPI Specification defines its own **responses** object within the HTTP method object. The **responses** object is usually added after the **parameters** array, before the closing brace of the HTTP method object.

[Copy](#)

```
"get": {
  "summary": "Get airport info",
  "description": "Get airport info",
  "operationId": "getAirportInfo",
  "produces": [
    "application/json"
  ],
  "parameters": [
    {
      "name": "iata",
      "description": "Airport code",
      "in": "query",
      "type": "string"
    }
  ],
  "responses": {
  }
}
```

### Step 2: Create a response

Each response is defined as an object, with a key of the HTTP status code returned by the API. Add an opening and closing brace to hold the definition of the response.

[Copy](#)

```
"responses": {
  "200": {
  }
}
```

Defining more than one response

### Step 3: Describe the response

Add a **description** to help anyone reading the OpenAPI Specification understand what the response is used for, and a **schema** object that defines the **type of response as an object**.

[Copy](#)

```
"responses": {
  "200": {
    "description": "OK",

```

```
"schema": {  
  "type": "object",  
}  
}
```

#### Step 4: Define the response data

Create a **properties** object to define the data returned. Inside the **properties** object, list each return variable as an object with the variable's name as the key, and the data **type** of the variable inside.

[Copy](#)

```
"responses": {  
  "200": {  
    "description": "OK",  
    "schema": {  
      "type": "object",  
      "properties": {  
        "name": {  
          "type": "string"  
        },  
        "location": {  
          "type": "string"  
        },  
        "country": {  
          "type": "string"  
        }  
      }  
    }  
  }  
}
```

Nintex Workflow Cloud uses the properties to create the **Result** configuration fields for this action in the Workflow designer.

**Get airport info**

**Data**

**Airport Code**

**Result Name**

Select a variable

**Result Location**

Select a variable

**Result Country**

Select a variable

## The OpenAPI Specification

The OpenAPI Specification is complete. You could import this OpenAPI Specification as-is and use it in your workflows. But using references can make building and maintaining your OpenAPI Specification much easier. Next, you will add a reference to the response schema to the OpenAPI Specification.

[Copy](#)

```
{
  "swagger": "2.0",

  "info": {
    "version": "1.0.0",
    "title": "Airport Data"
    "description": "Retrieves location information based on airport codes"
  },

  "host": "www.airport-data.com",
  "basePath": "/api",
  "schemes": [
    "http"
  ],

  "produces": [
    "application/json"
  ],

  "paths" : {
    "/ap_info.json": {
      "get": {
        "summary": "Get airport info",
        "description": "Get airport info",
        "operationId": "getAirportInfo",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "iata",
            "description": "Airport code",
            "type": "string",
            "in": "query"
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "object",
              "properties": {
                "name": {
                  "type": "string"
                },
                "location": {
                  "type": "string"
                },
                "country": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Next steps

[Streamline with references](#)

Resources

- Search Resources
- SDKs
- Learning Center
- Product Help
- Status

© 2018 Nintex Global Ltd. | [Legal](#) | [Security](#) | [Privacy](#)

Connect

- Community
- Partner Central
- Nintex Xchange
- Blog

