

Build a custom connector with OAuth2 authentication to create a Google task list



In this example, you will create a **custom connector** that uses **OAuth2** to create a task list in Google Tasks. When designing the workflow, you will be prompted to select which Google account to use.

The complete [OpenAPI Specification](#) and icon for this example are available [here](#).

Summary

OAuth2 authentication in the OpenAPI Specification

Register an OAuth2.0 client ID and secret

Define OAuth2 authentication

Step 1: Create the basic OpenAPI Specification

Step 2: Create the security definitions object.

Step 3: Create the OAuth2 authentication definition

Step 4: Add the URLs and scopes

Step 5: Add a security array to the HTTP method object

Step 6: Add the security object reference to the array

The OpenAPI Specification

Create the task list creation workflow

Step 1: Add your custom connector

Step 2: Authorize the redirect URL

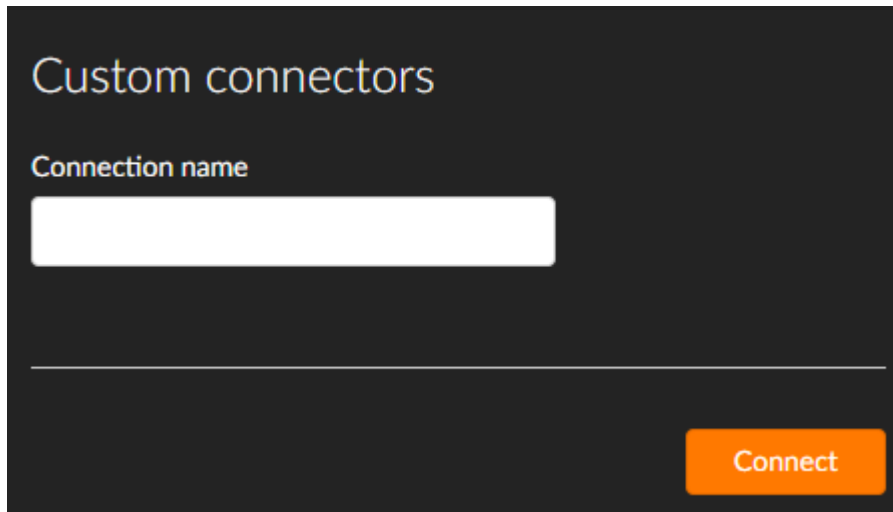
Step 3: Create the workflow

Step 4: Test the workflow

Summary

OAuth2 authentication requires Nintex Workflow Cloud to provide a client ID and a shared secret with the **API**. The API can then prompt the user to permit a defined scope of access to the user's account without having to give Nintex Workflow Cloud any authentication credentials. The client ID and shared secret must be registered with the API in advance.

When using custom connectors with OAuth2 authentication, you configure the client ID and secret on the security template when you import the custom connector, so that Nintex Workflow Cloud can identify itself securely to the API. When you create the **connection** in the Workflow designer, you will select which account to connect to.



Custom connectors

Connection name

Connect

OAuth2 authentication in the OpenAPI Specification

To add OAuth2 authentication to an OpenAPI Specification, you:

- Register a client ID and secret with the API you want to use
- Retrieve the redirect URLs from the client
- Define the OAuth2 authentication object inside the security definitions object
- Reference to this OAuth2 authentication object inside the HTTP method objects that require authentication

Register an OAuth2.0 client ID and secret

To import the completed custom connector, you must create the client ID and shared secret in a Google Developer account:

1. If you do not already have a Google Developer Account, create a free account [here](#).
2. Click the following link to [open the Google API console](#).
3. Click **Enable API** and type **Task API** in the search box.
4. Click **Enable** if this API is not already enabled.
5. Click **Credentials** in the left hand navigation pane.
6. Click **Create credentials** and **select OAuth client ID**.
7. Select **Web application**.
8. Type **Nintex Workflow Cloud connector for Tasks** in the Name field.
9. Click **Create**.
10. **Record the Client ID and Client Secret** somewhere secure. You will need these to import the connector into Nintex Workflow Cloud.

Define OAuth2 authentication

Step 1: Create the basic OpenAPI Specification

Create an OpenAPI Specification that:

- uses the **host** www.googleapis.com
- accepts **HTTPS** and produces **application/json**
- has a **post operation** to **/users/@me/lists** that passes the name of the task list in the **body** with a schema that lists the property of **title** as a string.
- **responds 200** with a schema that returns the title, id and link to the new task list, and the time it was updated.

Copy

```
{
  "swagger": "2.0",
  "info": {
    "version": "1.0.0",
    "title": "Google Tasks"
  },
  "host": "www.googleapis.com",
  "basePath": "/tasks/v1",
  "schemes": [
    "https"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/users/@me/lists": {
      "post": {
        "tags": [
          "Task"
        ],
        "summary": "Create a Task List",
        "description": "Create a Task List",
        "operationId": "createTaskList",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "task",
            "description": "Name of task list",
            "in": "body",
            "schema": {
              "properties": {
                "title": {
                  "type": "string"
                }
              }
            }
          },
          {
            "required": true
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "$ref": "#/definitions/list"
            }
          },
          "default": {
            "description": "Operation Failed"
          }
        }
      }
    }
  }
}
```

```

"definitions": {
  "list": {
    "type": "object",
    "properties": {
      "kind": {
        "type": "string"
      },
      "updated": {
        "type": "string",
        "format": "datetime"
      },
      "id": {
        "type": "string"
      },
      "selfLink": {
        "type": "string"
      },
      "title": {
        "type": "string"
      }
    }
  }
}

```

Step 2: Create the security definitions object.

Create the **securityDefinitions** object at the end of the OpenAPI Specification, after the **definitions** object but before the final closing brace. Make sure you add a comma to the end of the **definitions** object.

[Copy](#)

```

"definitions": {
  "list": {
    "type": "object",
    "properties": {
      "kind": {
        "type": "string"
      },
      "updated": {
        "type": "string",
        "format": "datetime"
      },
      "id": {
        "type": "string"
      },
      "selfLink": {
        "type": "string"
      },
      "title": {
        "type": "string"
      }
    }
  }
},
"securityDefinitions": {
}
}

```

Step 3: Create the OAuth2 authentication definition

Inside the **securityDefinitions** object, create an object to define the OAuth2 authentication. You can name the security object any name that is unique within the OpenAPI Specification. In this example, we have named it **oauth2**.

Add a **type** with a value of **oauth2** to define it as OAuth2 authentication, and add a property with the key of **flow** and a value of **accessToken** to use the access code method to authenticate.

[Copy](#)

```
"securityDefinitions": {
  "oauth2": {
    "type": "oauth2",
    "flow": "accessCode",
  }
}
```

Note: For security reasons, Nintex Xtensions only supports the **accessCode** flow.

Step 4: Add the URLs and scopes

Inside the **oauth2** object, add the properties for the URLs for the authentication stages and scopes provided by the API. For this example, add:

- **authorizationUrl** with a value of **https://accounts.google.com/o/oauth2/v2/auth**
- **tokenUrl** with a value of **https://www.googleapis.com/oauth2/v4/token**
- a **scopes** object, with a property for each resource you want to access, where:

the key is the resource you want to access:

https://www.googleapis.com/auth/tasks

the value is a description of the resource: **access to tasks**

Note: these properties are unique to this example. When creating OAuth2 authenticated custom connectors for other APIs, you will need to use the values provided by the API.

[Copy](#)

```
"securityDefinitions": {
  "oauth2": {
    "type": "oauth2",
    "flow": "accessCode",
    "authorizationUrl": "https://accounts.google.com/o/oauth2/v2/auth",
    "tokenUrl": "https://www.googleapis.com/oauth2/v4/token",
    "scopes": {
      "https://www.googleapis.com/auth/tasks": "access to tasks"
    }
  }
}
```

Step 5: Add a security array to the HTTP method object

Inside the HTTP method object, create a **security** array.

[Copy](#)

```
"paths": {
  "/users/@me/lists": {
    "post": {
      "tags": [
        "Task"
      ],
      "summary": "Create a Task List",
      "description": "Create a Task List",
      "operationId": "createTaskList",
      "produces": [
        "application/json"
      ],
      "security": [
      ],
    },
  },
}
```

Step 6: Add the security object reference to the array

Inside the array, reference the security object you defined earlier by using its name as the key, and adding an array of the resource paths you defined as keys in the **scopes** object.

[Copy](#)

```
"paths": {
  "/users/@me/lists": {
    "post": {
      "tags": [
        "Task"
      ],
      "summary": "Create a Task List",
      "description": "Create a Task List",
      "operationId": "createTaskList",
      "produces": [
        "application/json"
      ],
      "consumes": [
        "application/json"
      ],
      "security": [
        {
          "oauth2": [
            "https://www.googleapis.com/auth/tasks"
          ]
        }
      ],
    },
  },
}
```


The OpenAPI Specification

This is the complete OpenAPI Specification that OAuth2 to create a new task list in Google Tasks.

[Copy](#)

```
{
  "swagger": "2.0",
  "info": {
    "version": "1.0.0",
    "title": "Google Tasks"
  },
  "host": "www.googleapis.com",
  "basePath": "/tasks/v1",
  "schemes": [
    "https"
  ],
  "produces": [
    "application/json"
  ],
  "paths": {
    "/users/@me/lists": {
      "post": {
        "tags": [
          "Task"
        ],
        "summary": "Create a Task List",
        "description": "Create a Task List",
        "operationId": "createTaskList",
        "produces": [
          "application/json"
        ],
        "parameters": [
          {
            "name": "task",
            "description": "Name of task list",
            "in": "body",
            "schema": {
              "properties": {
                "title": {
                  "type": "string"
                }
              }
            }
          },
          {
            "required": true
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "$ref": "#/definitions/list"
            }
          },
          "default": {
            "description": "Operation Failed"
          }
        },
        "security": [
          {
            "oauth2": [
              "https://www.googleapis.com/auth/tasks"
            ]
          }
        ]
      }
    }
  },
  "definitions": {
    "list": {
      "type": "object",
```

```

    "properties": {
      "kind": {
        "type": "string"
      },
      "updated": {
        "type": "string",
        "format": "datetime"
      },
      "id": {
        "type": "string"
      },
      "selfLink": {
        "type": "string"
      },
      "title": {
        "type": "string"
      }
    }
  },
  "securityDefinitions": {
    "oauth2": {
      "type": "oauth2",
      "flow": "accessToken",
      "authorizationUrl": "https://accounts.google.com/o/oauth2/v2/auth",
      "tokenUrl": "https://www.googleapis.com/oauth2/v4/token",
      "scopes": {
        "https://www.googleapis.com/auth/tasks": "access to tasks"
      }
    }
  }
}

```

Create the task list creation workflow

Step 1: Add your custom connector

Import the OpenAPI Specification you created into Nintex Workflow Cloud:

1. Open your Nintex Workflow Cloud tenancy.
2. Click **Xtensions** in the dashboard to open the Xtensions page.
3. Click **+** in the Custom connector list.
4. Click **Choose a file**.
5. Navigate to the OpenAPI Specification on your computer.
6. Wait for Nintex Workflow Cloud to validate the file.
7. Click **Next**.
8. Nintex Workflow Cloud detects the OAuth 2.0 security template.
9. Type the Client ID in the **Client ID** field.
10. Type the shared secret in the **Client Secret** field.

11. Copy the **Redirect URL**: you will need to add this to your OAuth credentials.
12. Click **Next**.
13. Edit the **Name** of the connector, which becomes the name of the action group in the Workflow designer.
14. Edit the **Description** of the connector. This appears in the Custom connector list in the Xtensions page.
15. Select or upload an icon for the connector. This is displayed for each action or event in the Workflow designer
16. Click **Publish**.

Step 2: Authorize the redirect URL

The API will only accept requests that redirect to URLs that have already been authorised for the OAuth2 credentials. To authorise the redirect URL

1. Open the **Credentials** section of the Google Developer Console.
2. Edit the OAuth2 credentials you created earlier.

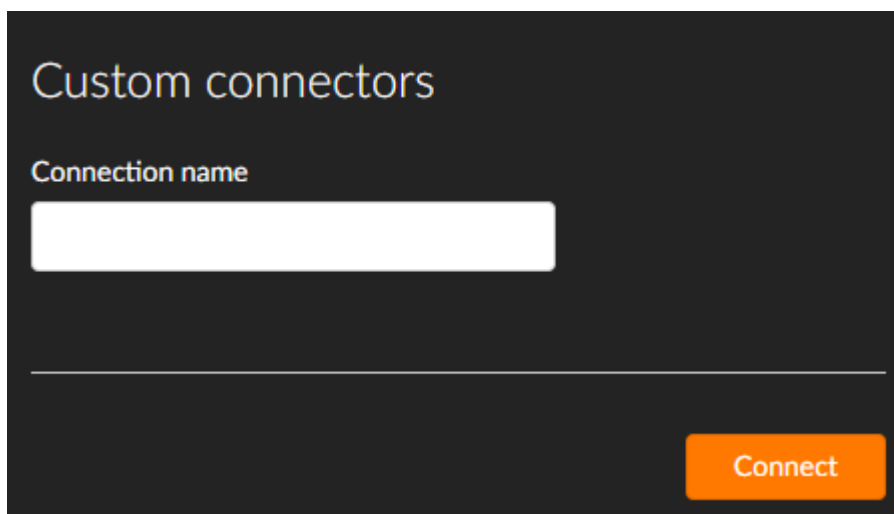
3. Paste the Redirect URL you copied into the Authorized redirect URIs field.
4. Click Save.

Step 3: Create the workflow

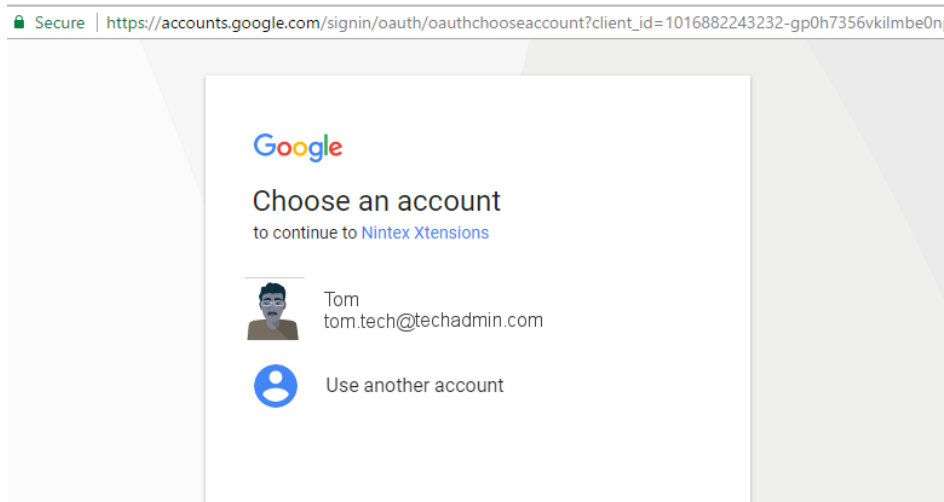
Create a workflow that prompts the user for the name of the new task list, and creates it in Google Tasks.

To create the workflow:

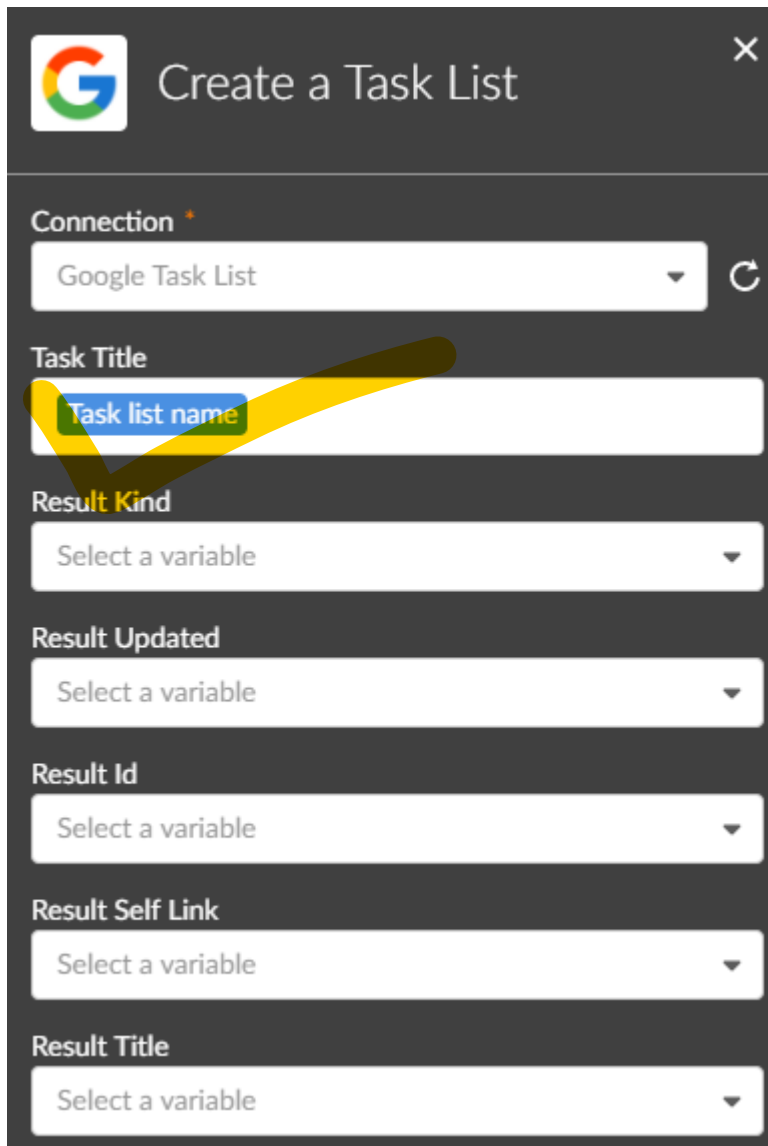
1. Click **Create workflow** in your Nintex Workflow Cloud tenancy.
 2. Configure the **Start event** to be a **Public web form** with one text variable for the name of the new task list.
- For more information on designing forms in Nintex Workflow Cloud, see [Design a form](#).
3. Drag the **Create a Task List** action after the **Start event**.
 4. Create the connection.



5. Select the Google account to connect to.



6. Click Allow.
7. Configure the **Create a Task List** action to use the new task list name from the web form as the Task Title.



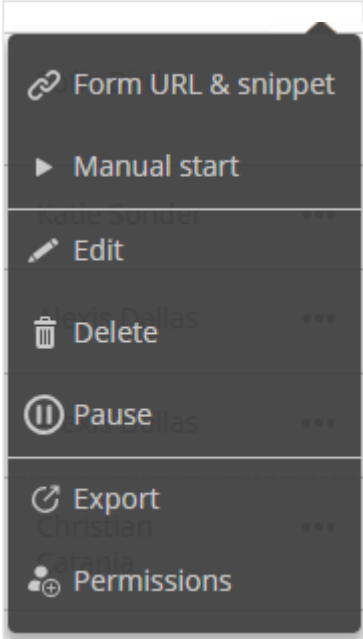
8. Publish the workflow.

Tip: If you want to troubleshoot a custom connector, select **Development** as the Assigned Use when you publish the workflow. Development workflows display more detailed error messages in their instance details. Republish your workflow as Production when you're ready to use it.

Step 4: Test the workflow

1. Click **Workflow** in your Nintex Workflow Cloud tenancy.

2. Click ... on the workflow you created.



- 3. Click **Manual start**.
- 4. Type the name of the task list you want to add in the form.
- 5. Click **Start** to create the task list in Google Tasks.
- 6. View your task lists in Google Tasks to confirm.

Resources

- [Search Resources](#)
- [SDKs](#)
- [Learning Center](#)
- [Product Help](#)
- [Status](#)

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

Connect

- [Community](#)
- [Partner Central](#)
- [Nintex Xchange](#)
- [Blog](#)

