

使用SAP云平台工作流

目标：了解SAP云平台工作流的设计、定义，并掌握 **exclusive gateways, script tasks 和 service tasks**的使用。

SAP云平台工作流入门

准备创建工作流

1. 打开云平台驾驶舱

选中Neo 试用

The screenshot shows the SAP Cloud Platform Cockpit interface. On the left, there's a sidebar with '主页' (Home), '区域' (Regions), and '服务' (Services). The main content area has a header '您的 SAP Cloud Platform 试用' (Your SAP Cloud Platform Trial) with a background image of a hot air balloon over mountains. Two trial options are shown in boxes: 'Cloud Foundry 试用' (Cloud Foundry Trial) and 'Neo 试用' (Neo Trial). The 'Neo 试用' box is highlighted with a red border. Below the trials, there's a section for '您好 Developer' (Hello Developer) with a brief introduction and a '入门' (Getting Started) button. To the right, there are sections for '环境' (Environment) and '后续步骤' (Next Steps) with various links like '部署第一个应用程序' (Deploy first application) and '启动 SAP Web IDE' (Start SAP Web IDE).

选中服务

SAP Cloud Platform Cockpit

主页 (欧洲 (Rot) - 试用) / 欧洲 (Rot) - 试用 / p1942966529trial

子账户: p1942966529trial - 服务

所有: 40

所有类别 搜索

主数据

数据质量服务 未启用 嵌入数据质量服务以验证地址并增强地理编码。

全球化

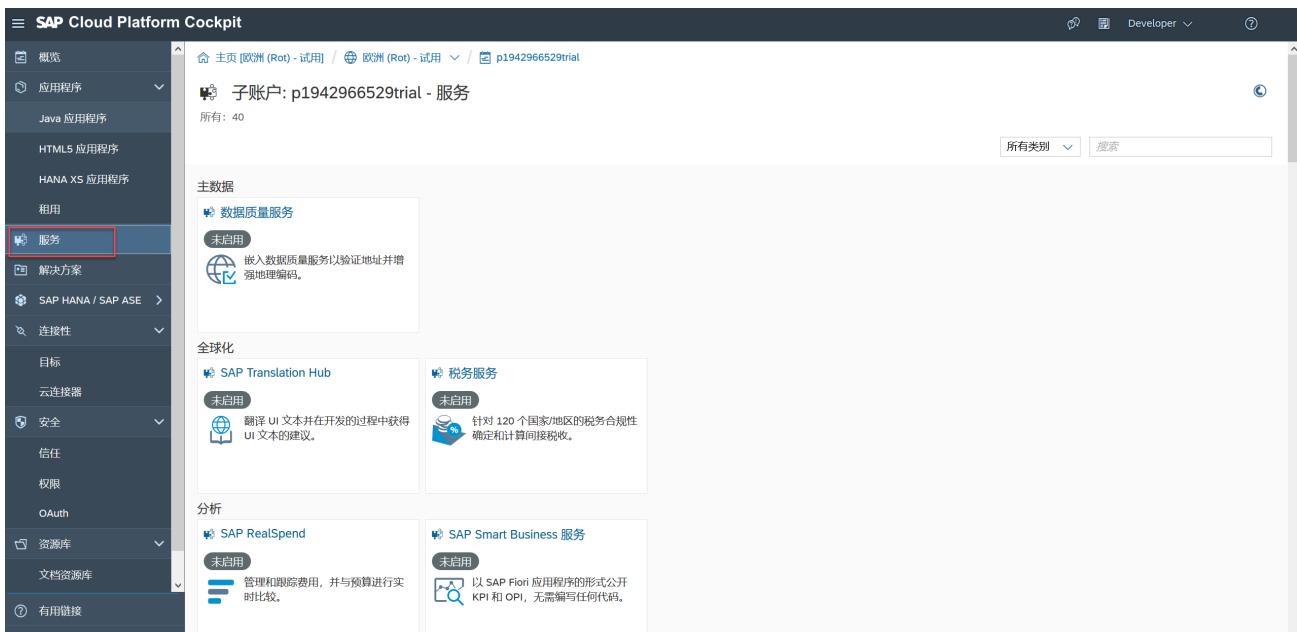
SAP Translation Hub 未启用 翻译 UI 文本并在开发的过程中获得 UI 文本的建议。

税务服务 未启用 针对 120 个国家/地区的税务合规性确定和计算间接税收。

分析

SAP RealSpend 未启用 管理和跟踪费用，并与预算进行实时比较。

SAP Smart Business 服务 未启用 以 SAP Fiori 应用程序的形式公开 KPI 和 OPI，无需编写任何代码。



启用门户

SAP Cloud Platform Cockpit

主页 (欧洲 (Rot) - 试用) / 欧洲 (Rot) - 试用 / p1942966529trial

所有: 40

所有类别 搜索

用户体验

SAP Build 未启用 创建基于最终用户反馈的交互式原型，无需编写代码。

UI 主题设计器 未启用 将公司品牌应用到基于 SAPUI5 技术的应用程序。

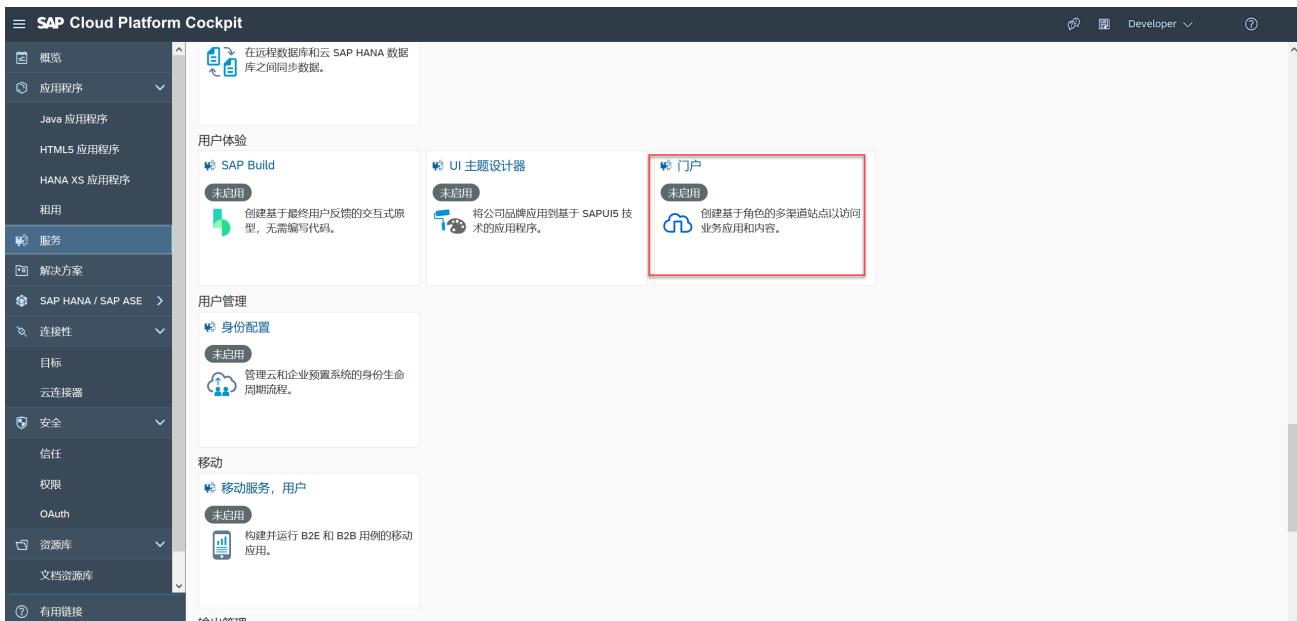
门户 未启用 创建基于角色的多渠道站点以访问业务应用和内容。

用户管理

身份配置 未启用 管理云和企业预置系统的身份生命周期流程。

移动

移动服务, 用户 未启用 构建并运行 B2E 和 B2B 用例的移动应用。



启用工作流

The screenshot shows the SAP Cloud Platform Cockpit interface. On the left, there is a sidebar with various navigation options like Overview, Applications, Services, Solutions, Connectivity, Goals, Cloud Connectors, Security, Resource Catalog, and Documentation. The 'Services' section is currently selected. In the main content area, there are several cards representing different extensions. One card, 'Workflow', is highlighted with a red border. Other visible cards include 'SAP RealSpend', 'SAP Smart Business Service', 'SAP Jam', 'Business Rules', 'Feedback Service', 'Gameification', 'Git Service', 'Java Debugging', 'Rapid Application Development ...', and 'SAP Web IDE Full Stack'. Each card has a status indicator (e.g., Enabled or Beta) and a brief description.

2. 设置工作流编辑器

进入工作流服务，打开SAP Web IDE 全栈

The screenshot shows the configuration page for the SAP Cloud Platform Workflow service. At the top, it displays the service name and its status: '已启用' (Enabled). Below this, there are two main sections: '服务描述' (Service Description) and '可用性' (Availability). The '服务描述' section contains a note about prerequisites and a detailed description of the service's purpose. The '可用性' section lists supported regions and cloud foundries. At the bottom, there are sections for '采取措施' (Measures Taken), '附加资源' (Additional Resources), and links for '查看 API' (View API) and 'SAP Web IDE 全栈' (SAP Web IDE Full Stack). A red box highlights the 'SAP Web IDE Full Stack' link.

选中Extensions，找到Workflow Editor并启用，保存更改

The screenshot shows the SAP Web IDE interface with the 'Global Preferences' menu open. Under 'Workspace Preferences', the 'Extensions' option is selected and highlighted with a red box. The main content area displays a list of available extensions:

- SAP HANA Database Development Tools**: Version 1.1.0, OFF. Description: These tools enable you to develop SAP HANA Database (HDB) modules and database artifacts, use More Information.
- Mobile Services App Development Tools**: Version V19.05, OFF. Description: Mobile Services App Development Tools enable developers to build end to end mobile applications. More Information.
- SAP SE**: Automatically loads these extensions: SAP HANA Database Explorer
- Workflow Editor**: Version V1.52.0, OFF. Description: You can model and deploy workflows that help in automating process steps. More Information.
- OME**: OData Model Editor, Version V1.7.0, OFF. Description: OData Model Editor. More Information.

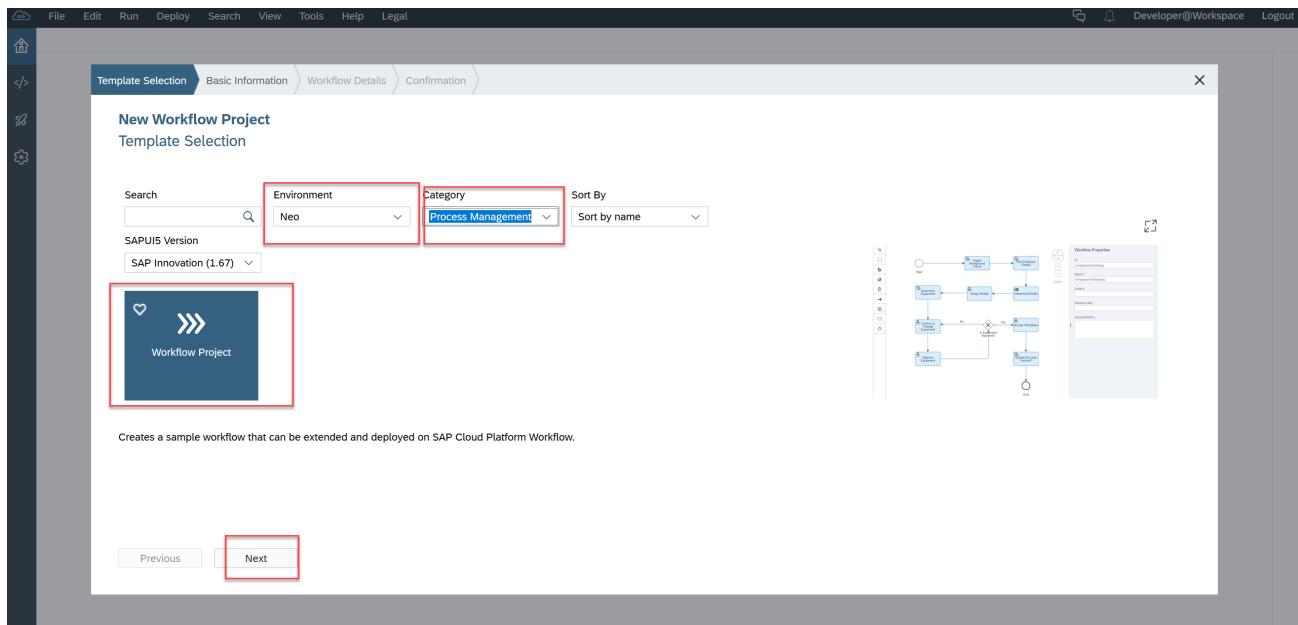
A 'Save' button is located at the bottom left of the extension list.

依据模板创建工作流项目

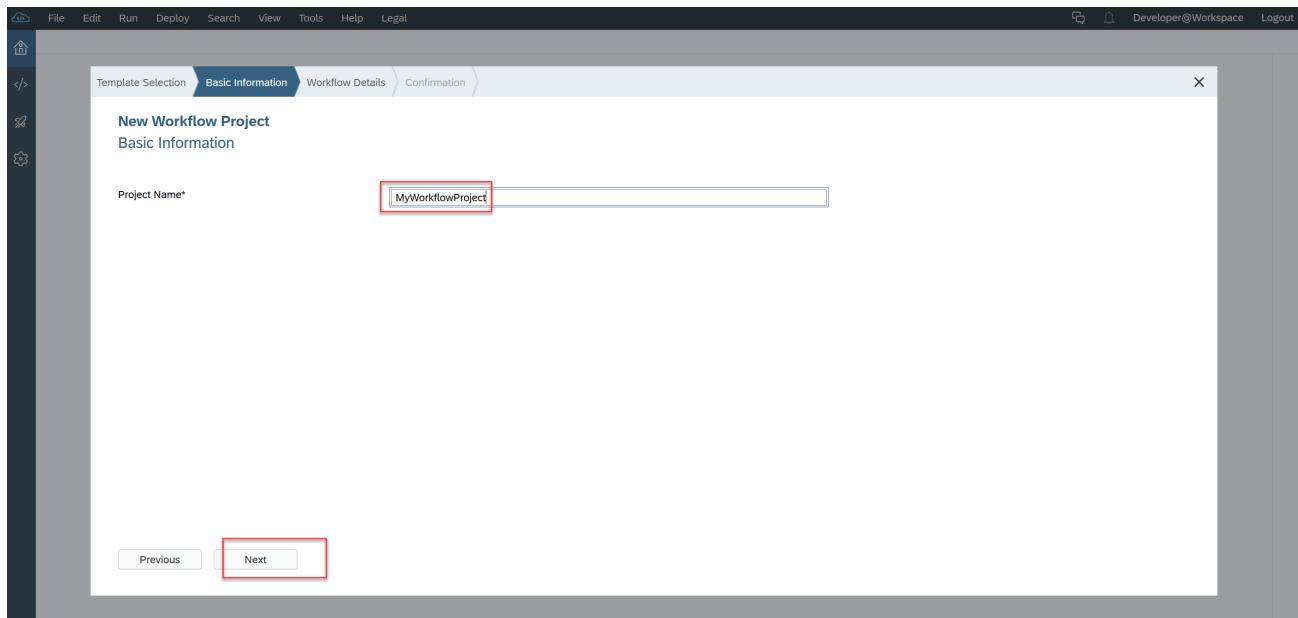
1. 选中 New Project from Template

The screenshot shows the SAP Web IDE home page. In the 'Create a Project' section, the 'New Project from Template' option is highlighted with a red box. Other options include 'Quick Start with Layout Editor', 'New Project from Sample Application', and 'New Extension Project'. To the right, there's a 'What's New' section with a video thumbnail and a 'Helpful Links' sidebar with links like SAP Web IDE Homepage, SAP Web IDE Documentation, Ask a Question on SAP Community, SAP Fiori Design Guidelines, SAP Web IDE SDK, SAPUI5 SDK, Tutorials, and SAP Cloud Platform Cockpit.

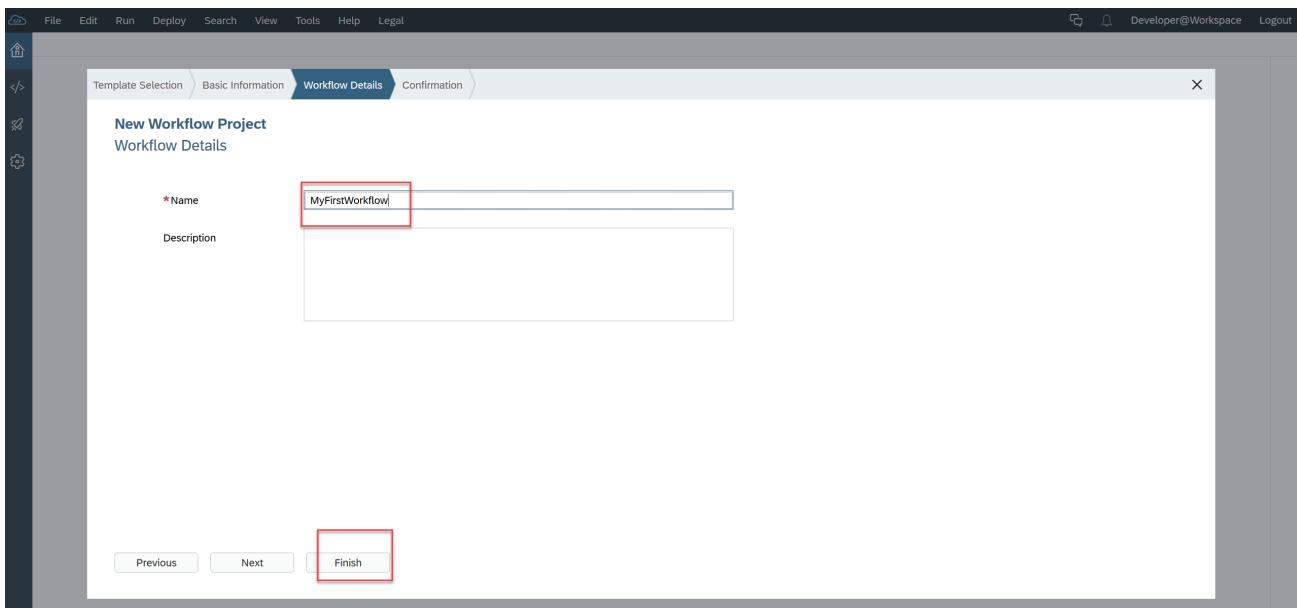
依据筛选条件Environment -> Neo, Category -> Business Process Management, 选中Workflow Project, 点击Next



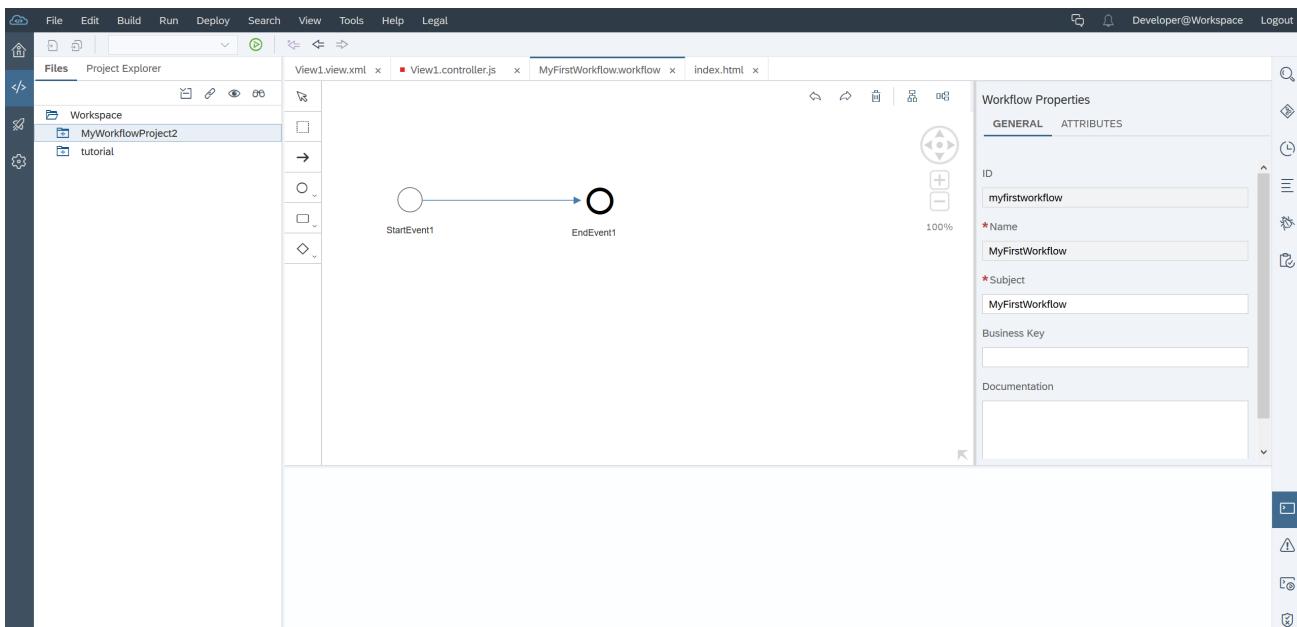
输入项目名**MyWorkflowProject**, 点击**Next**



输入工作流名 **MyFirstWorkflow** , 点击 **Finish**

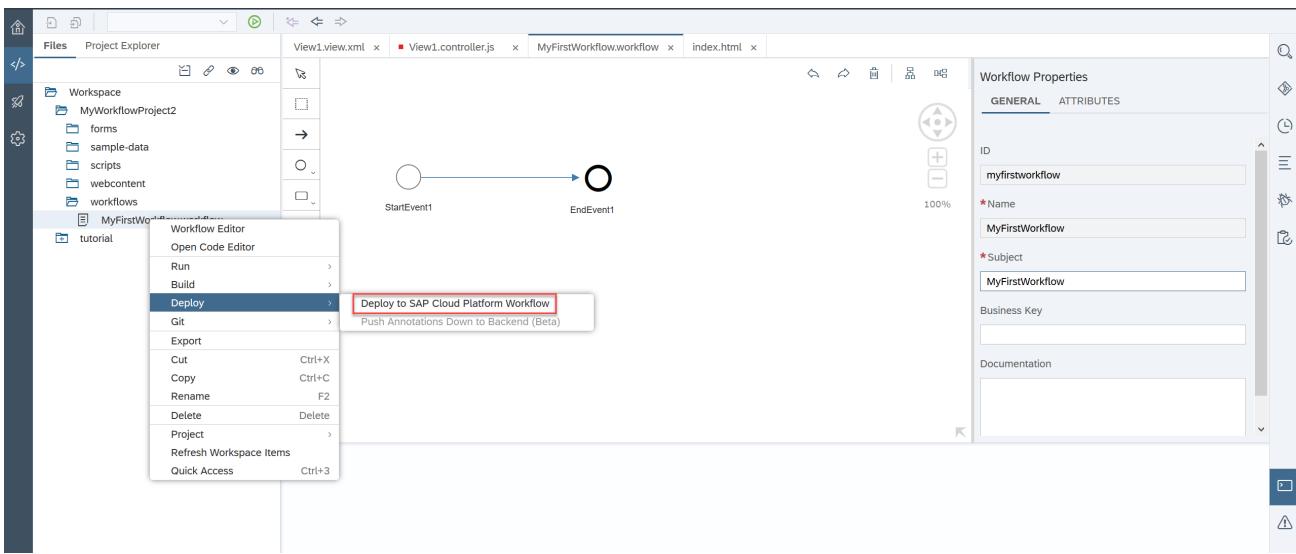


工作流的项目就创建好了

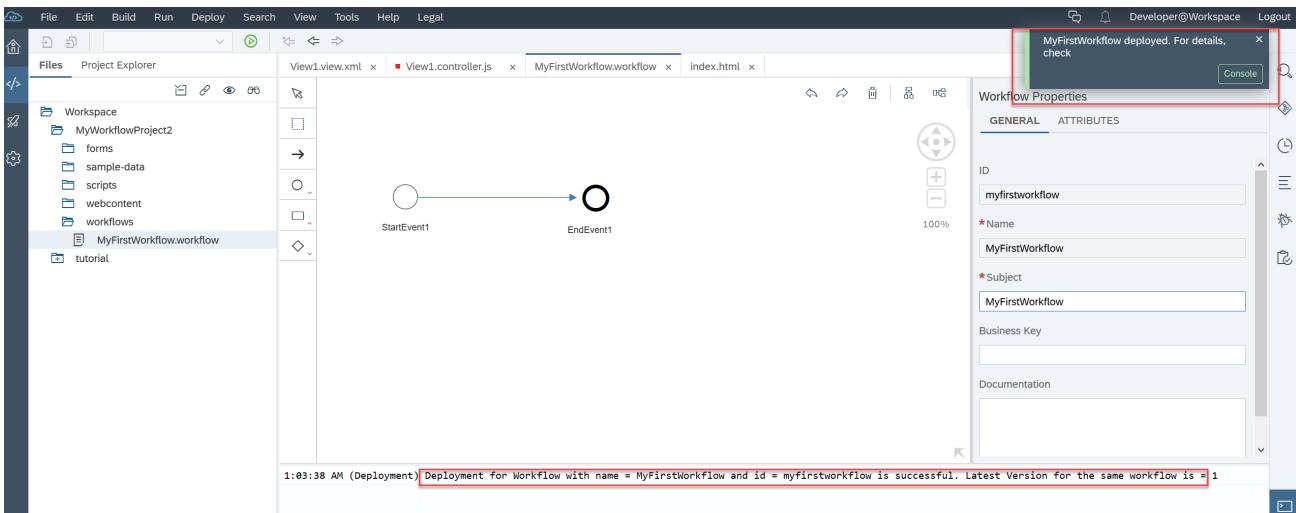


2. 部署你的工作流

右击工作流文件，选择 Deploy | Deploy to SAP Cloud Platform Workflow



部署成功后会提示用户



添加User Task到工作流中

1. 将工作流添加到主页中

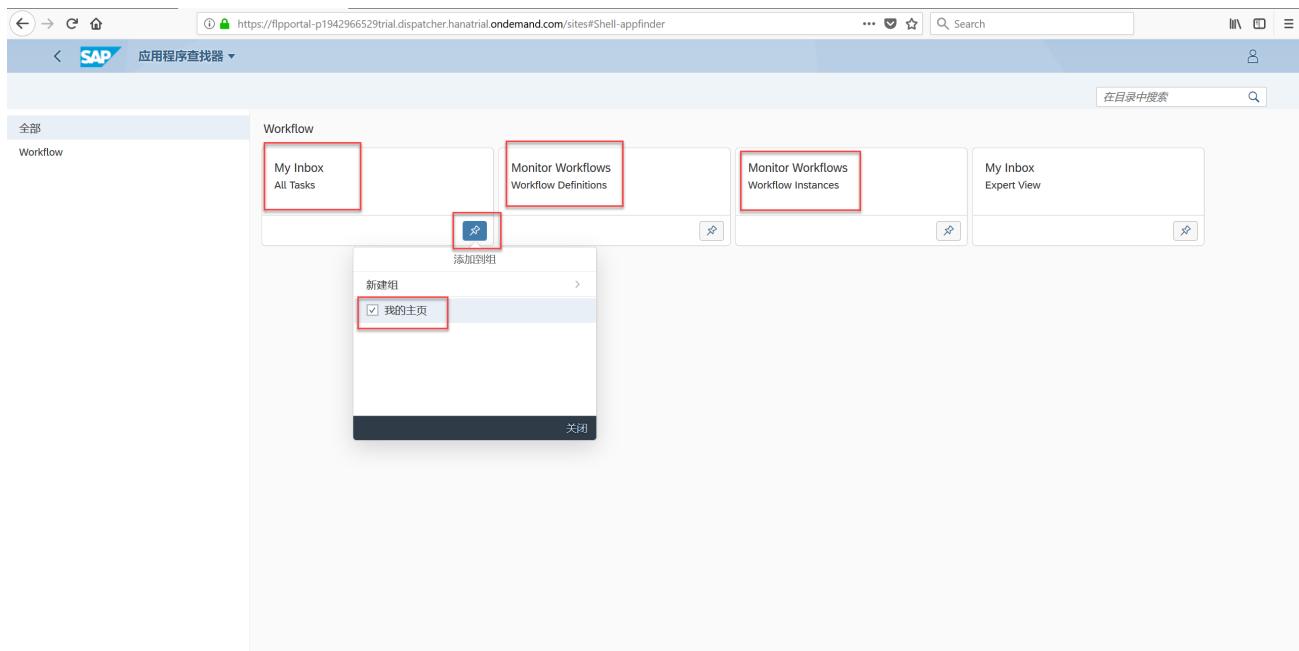
在工作流服务中，选中Fiori 快速启动板（缺省站点）

The screenshot shows the SAP Cloud Platform Cockpit interface. At the top, it displays the title "SAP Cloud Platform Cockpit". Below the header, there's a navigation bar with links like "主页 [欧洲 (Rot) - 试用]" and "工作流". The main content area is titled "服务: 工作流 - 概览". It includes sections for "服务描述" (Service Description), "可用性" (Availability), "采取措施" (Actions Taken), and "附加资源" (Additional Resources). A red box highlights the "Fiori 快速启动板 (缺省站点)" link under "采取措施".

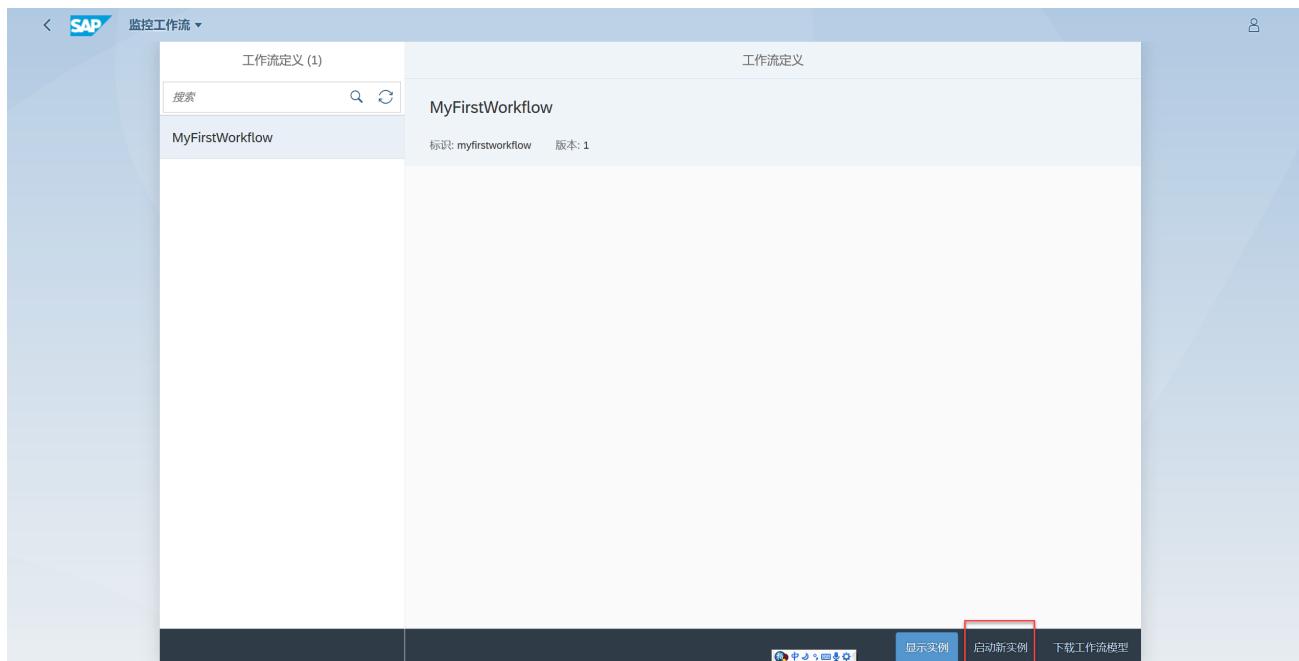
在SAP Fiori启动面板中，选中应用程序查找器

The screenshot shows the SAP Fiori launchpad. In the top right corner, there's a user menu labeled "Developer Qiu". A red box highlights the "应用程序查找器" (Application Navigator) icon in the dropdown menu.

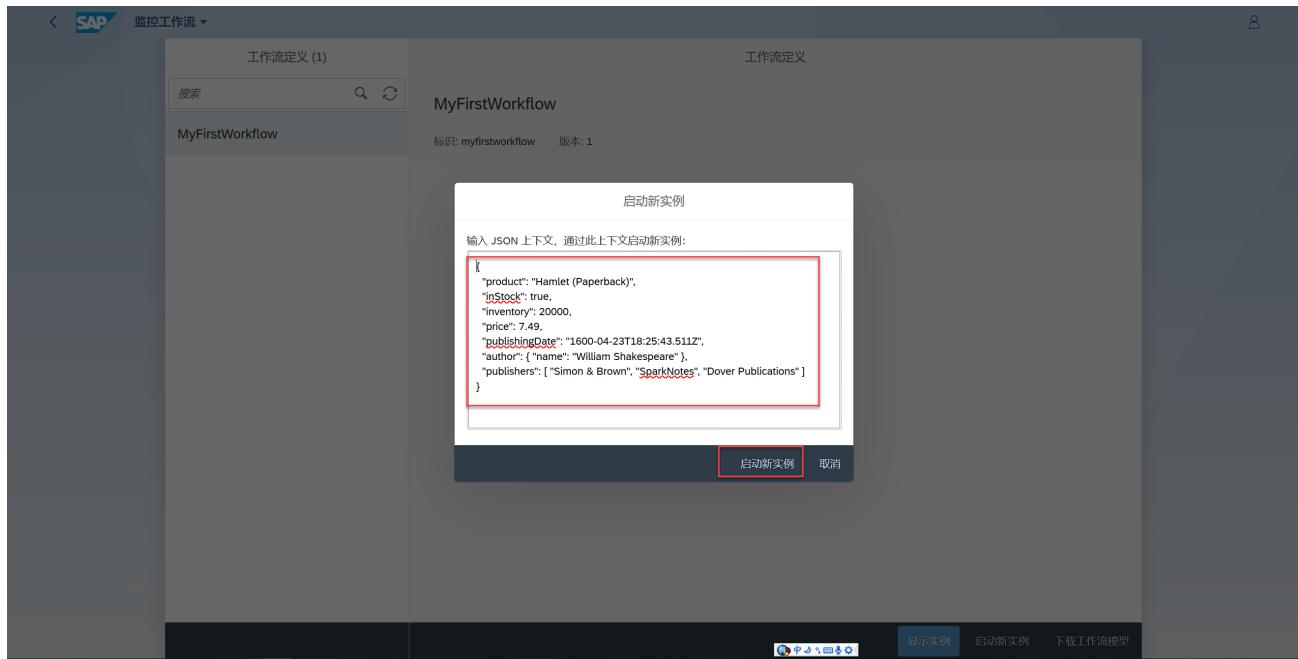
将My Inbox, Monitor Workflows (Definitions), Monitor Workflow (Instances)选中，添加到主页



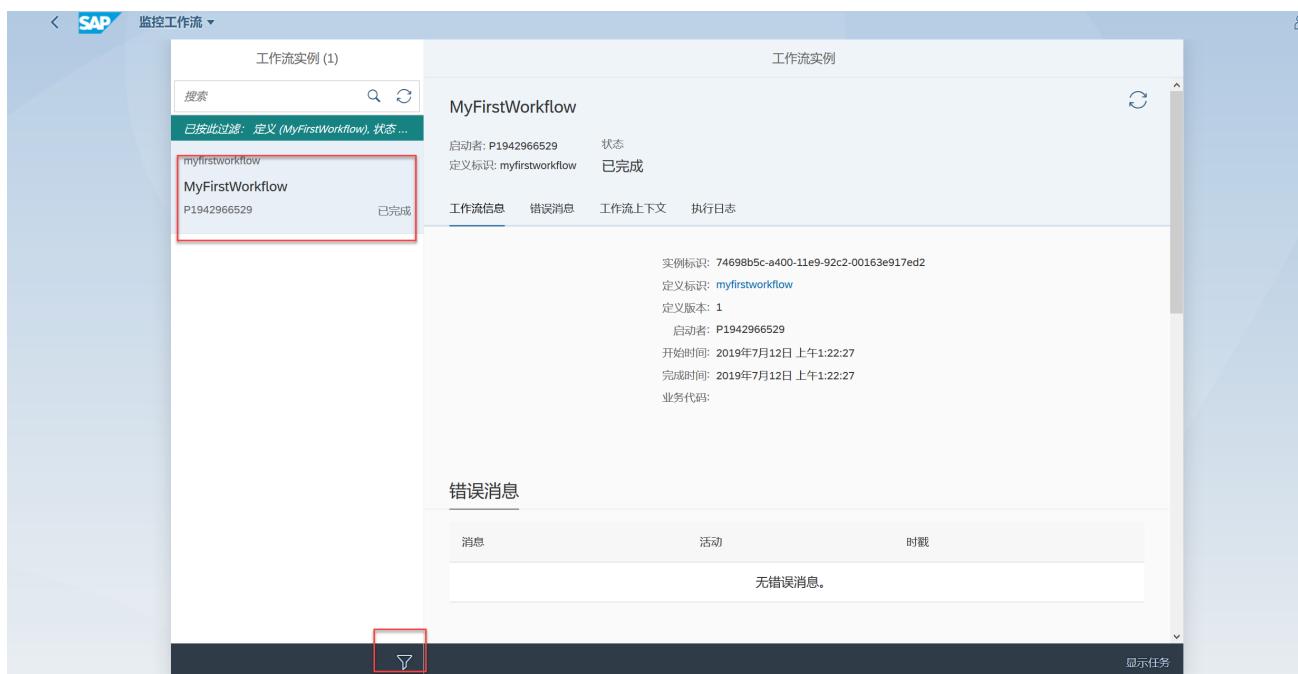
2. 打开工作流定义，点击启用新实例



使用默认的Sample数据启动实例

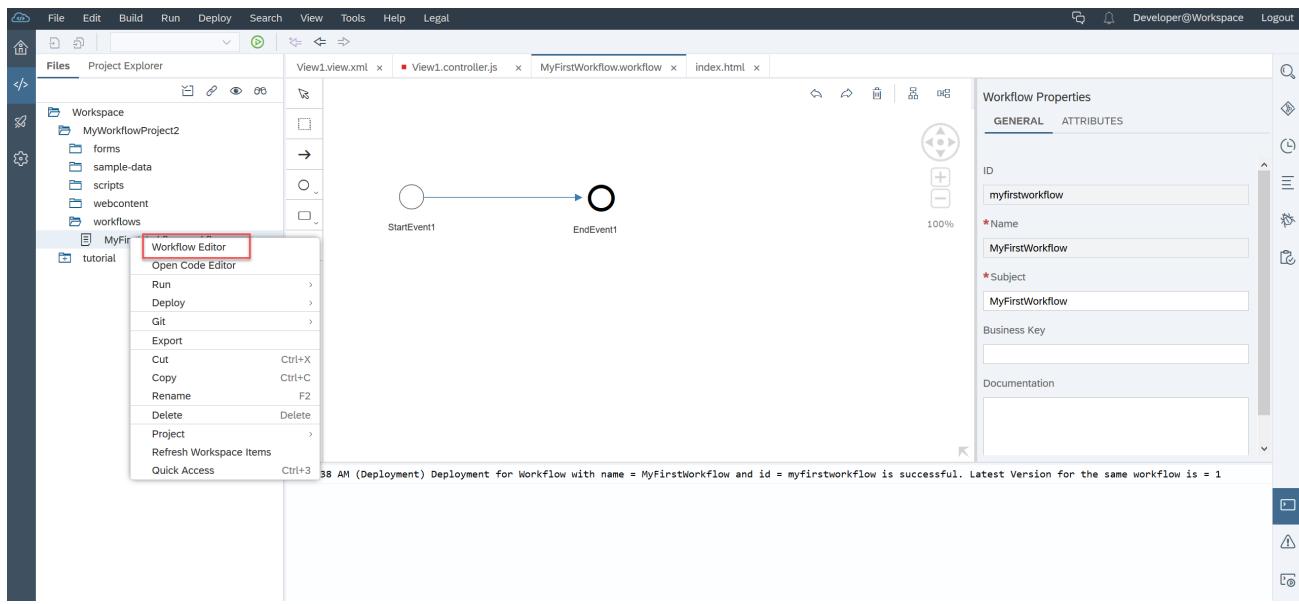


点击筛选按钮，选中查询完成状态的实例，可以查看到刚刚运行完成的实例



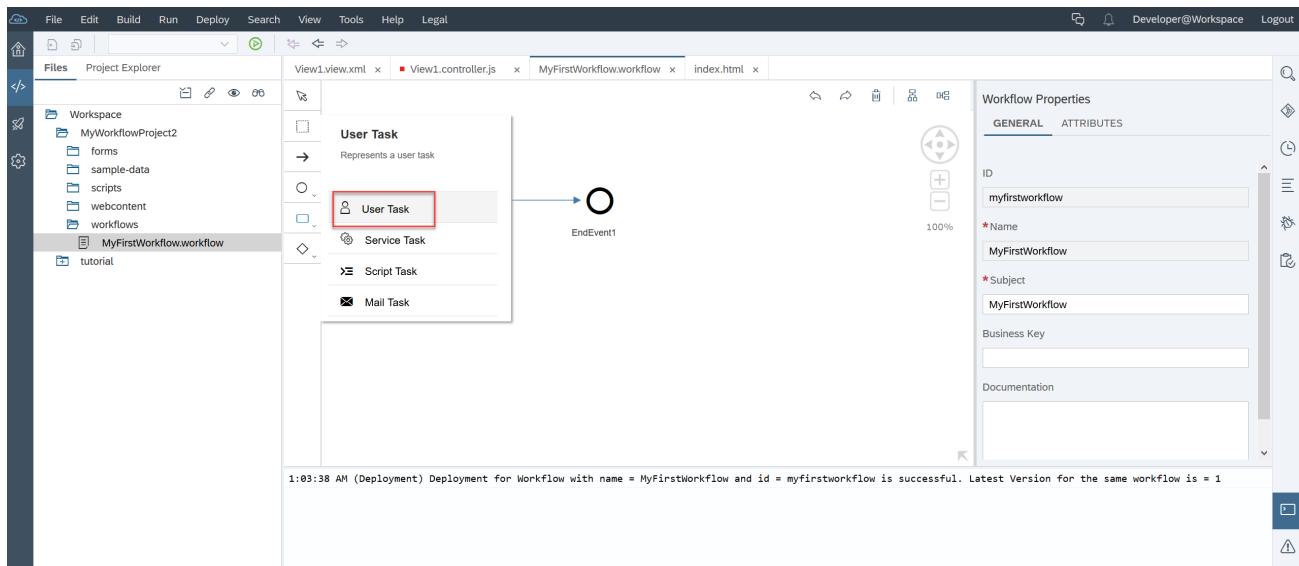
3. 使用工作流编辑器

右击工作流文件，选中 Workflow Editor

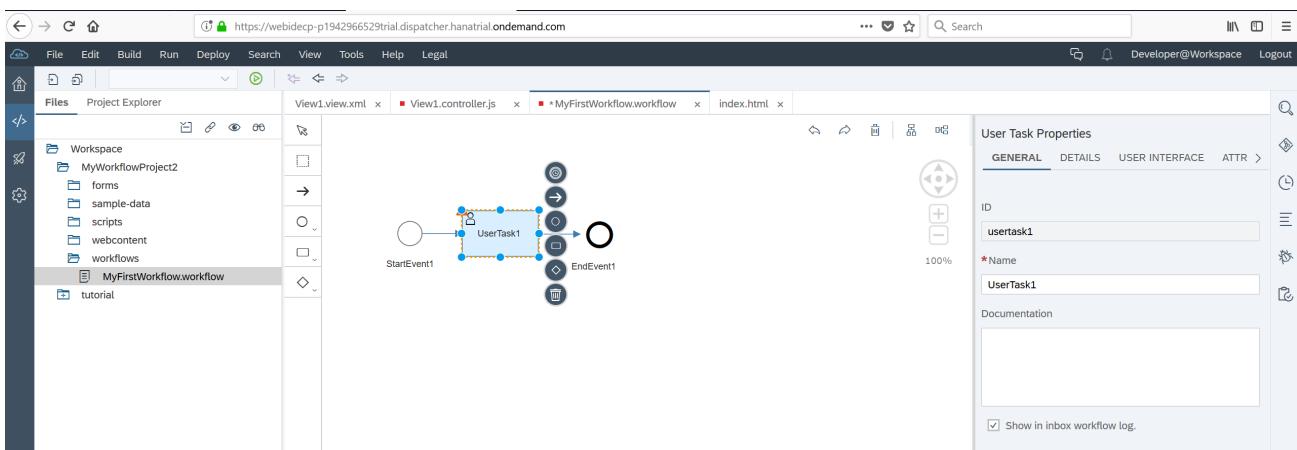


4. 添加 user task 到工作流中

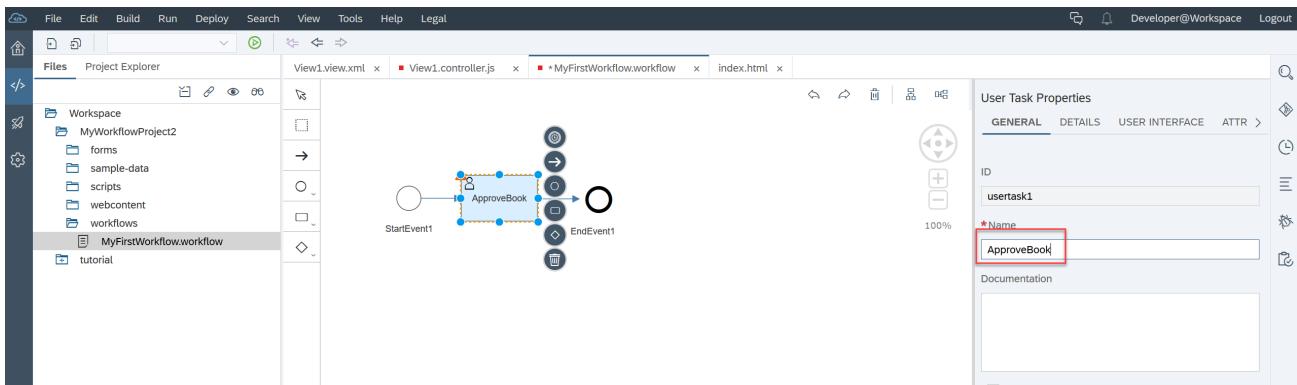
选中 User Task



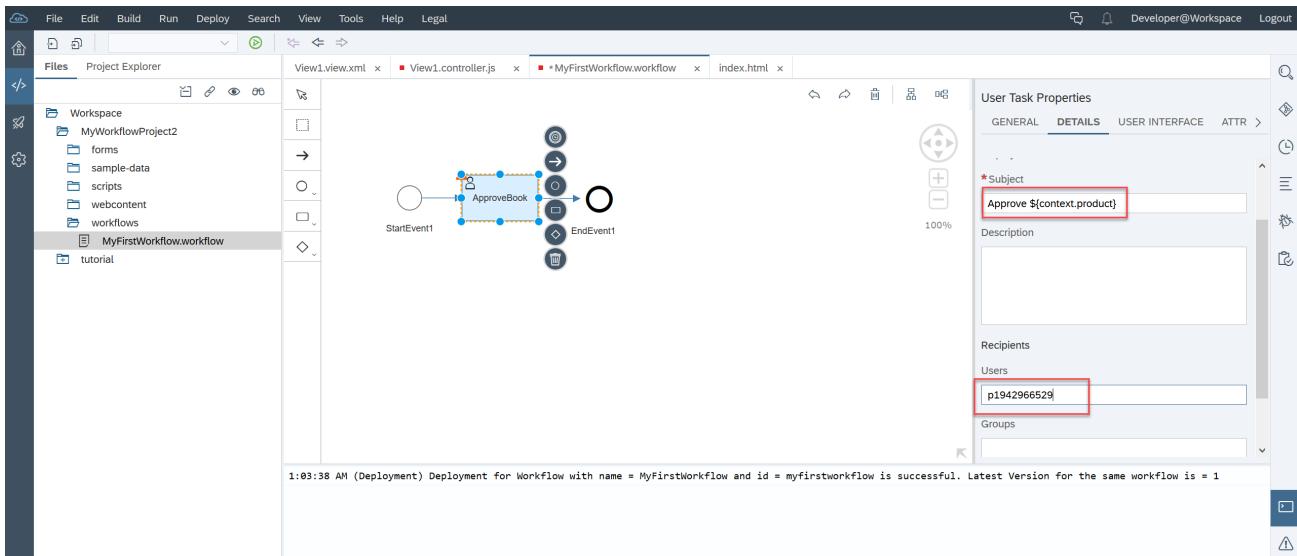
拖拽到开始与结束事件之间



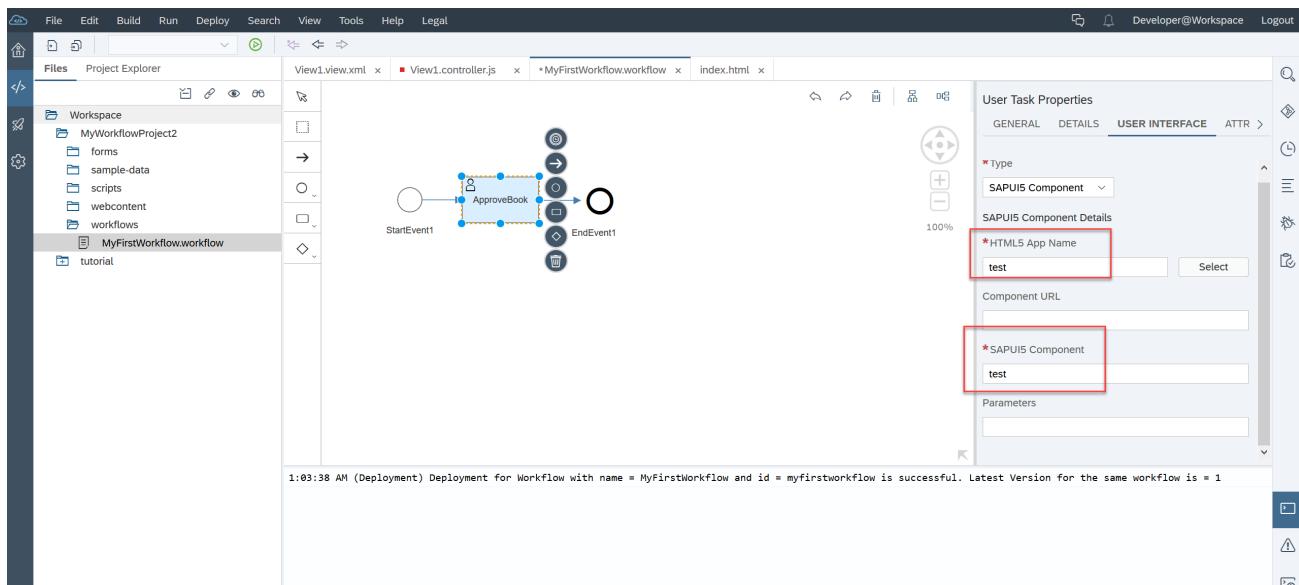
将名称改为 ApproveBook



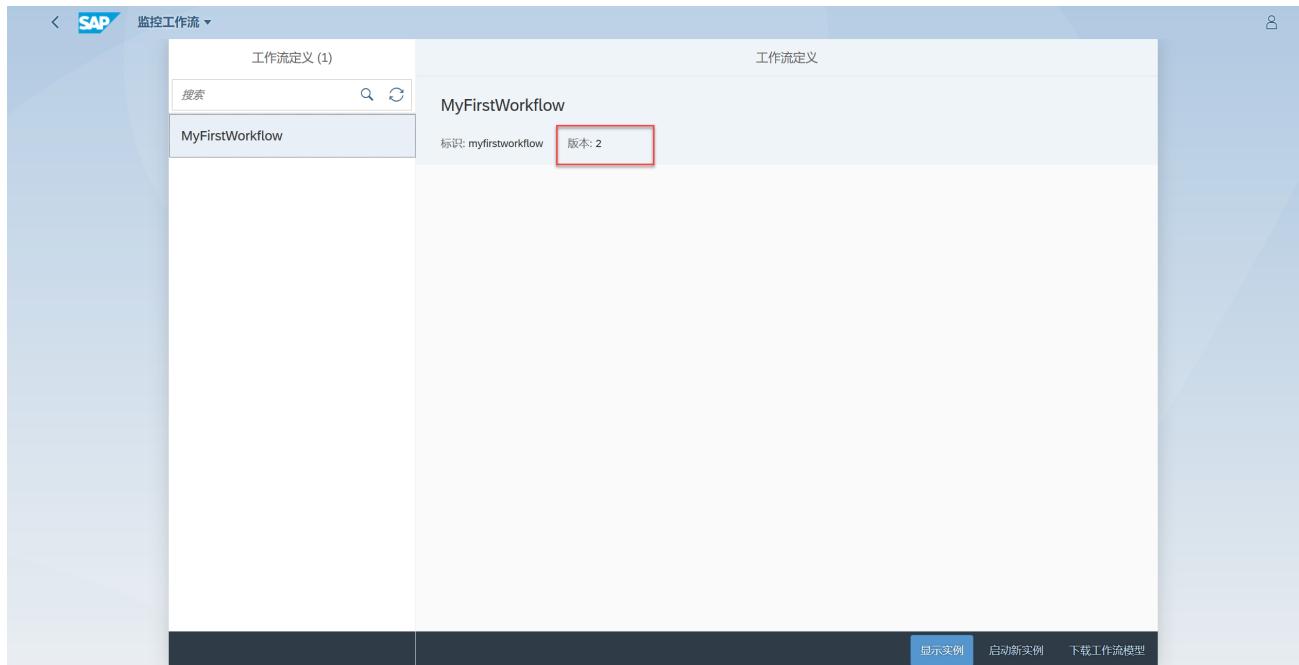
Detail页中，在Subject 中填入 Approve \${context.product}，
Recipients中输入你的ID（区分大小写）



User Interface页中，App Name 与 Component中输入test，这样在
Inbox中会出现未处理任务



保存更改，右击工作流文件 Deploy | Deploy to SAP Cloud Platform Workflow，新部署的工作流会有新的版本号



5. 新建工作流实例

启动新实例后，选择显示实例

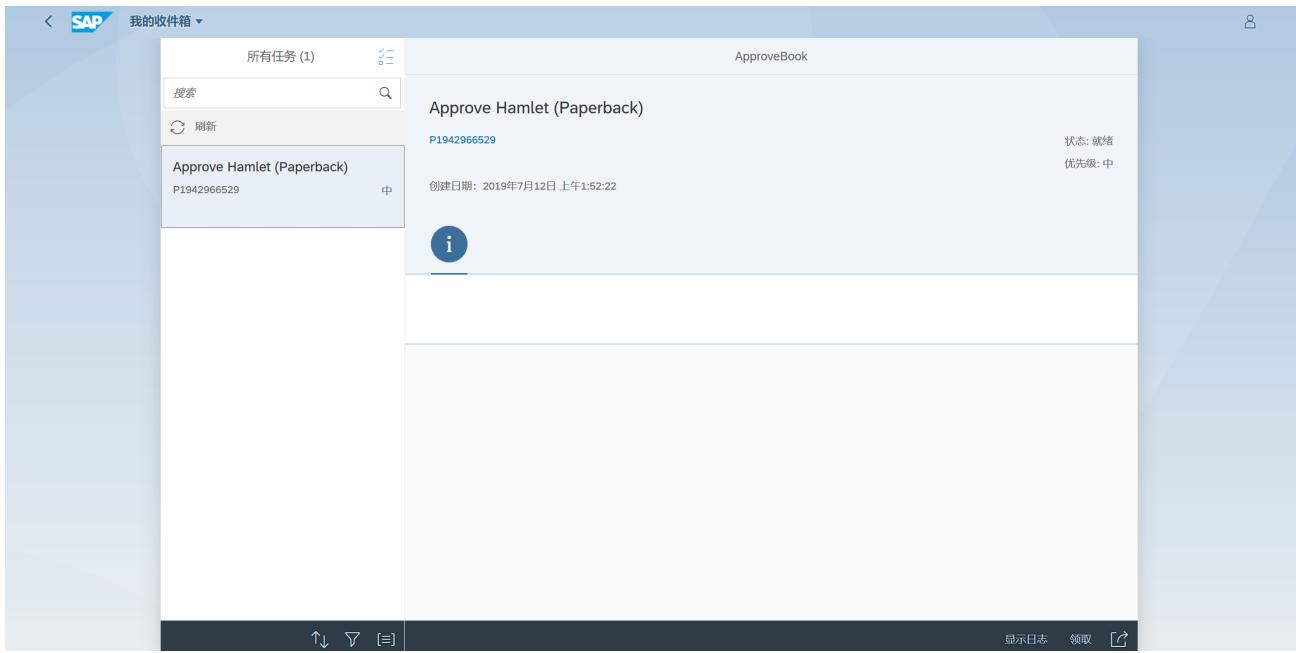
The screenshot shows the SAP Fiori Workflows monitoring interface. On the left, a sidebar titled '工作流实例 (2)' lists two workflow instances: 'myfirstworkflow' and 'MyFirstWorkflow'. The 'MyFirstWorkflow' entry is highlighted with a red box and has the status '正在运行' (Running). The main panel displays detailed information for the selected workflow instance. At the top, it shows the workflow name 'MyFirstWorkflow', the initiator 'P1942966529', and the status '正在运行' (Running) with another red box. Below this, tabs for '工作流信息' (Workflow Information), '错误消息' (Error Messages), '工作流上下文' (Workflow Context), and '执行日志' (Execution Log) are visible. The '工作流信息' tab is active, showing details like instance identifier, definition identifier, version, initiator, start time (2019年7月12日上午1:46:13), end time, and business key. The '错误消息' tab shows a table with columns '消息' (Message), '活动' (Activity), and '时截' (Timestamp), with a note '无错误消息。' (No error messages.). At the bottom right, buttons for '显示任务' (Show Tasks), '暂停' (Pause), and '终止' (Terminate) are present.

6. 打开 My Inbox 中的任务

在 Fiori 启动面板中，My Inbox 有收到的未处理任务

The screenshot shows the SAP Fiori Home page. A sidebar on the left lists three sections: '我的主页' (My Home), 'My Inbox' (highlighted with a red box), and 'Monitor Workflows'. The 'My Inbox' section contains the text 'All Tasks' and a blue download icon with the number '1' next to it, indicating one unread task. To the right, there are two other sections: 'Monitor Workflows Workflow Definitions' and 'Monitor Workflows Workflow Instances', each with a blue download icon.

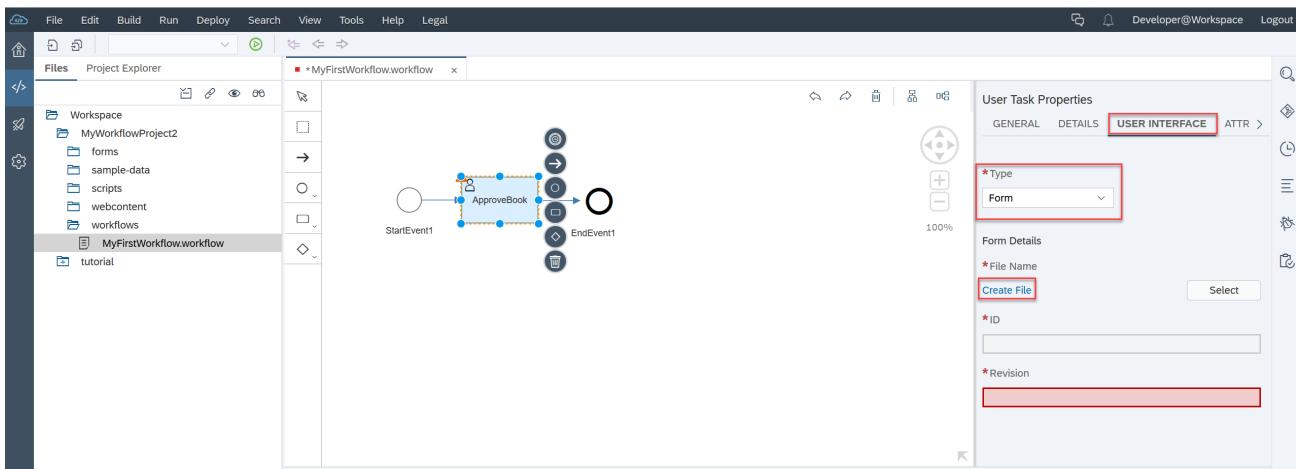
进入 My Inbox，可以查看任务



创建简单的审批界面，用于工作流表单

1. 为一个User Task创建表单

选中ApproveBook，在User Interface中，Type选Form，并点击 Create File



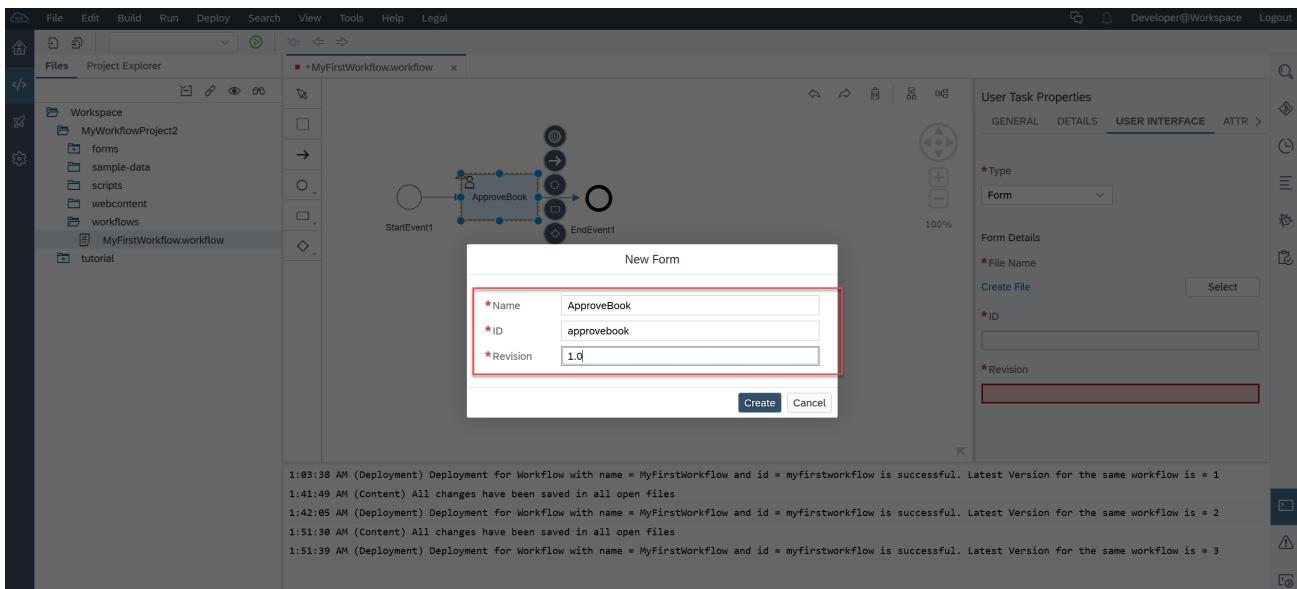
在New Form对话框中，输入

Name: ApproveBook

ID: approvebook

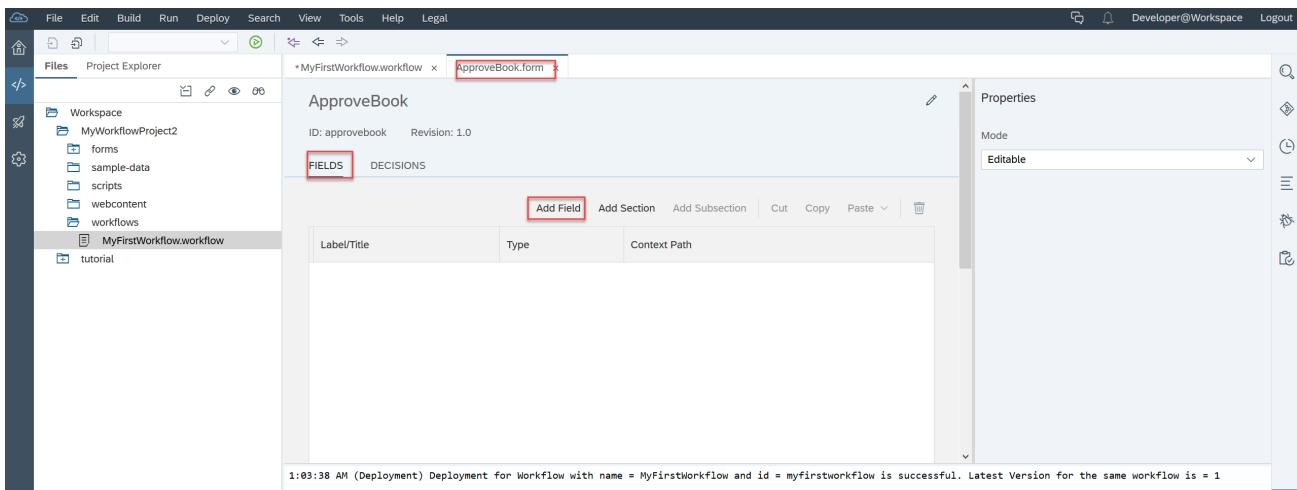
Revision: 1.0

点击 Create



3. 表单中加入字段

在FIELDS下，点击 Add Field



输入下列值

First row:

In column Label/Title, enter Title.

In column Type, select String.

In column Context Path, enter \${context.product}.

Second row:

In column Label/Title, enter Price.

In column Type, select Float.

In column Context Path, enter \${context.price}.

The screenshot shows the 'ApproveBook' form editor. On the left, the 'FIELDS' tab is selected, displaying two fields: 'Title' (String type) and 'Price' (Float type). The 'Price' field's context path is set to \${context.price}. To the right, the 'Properties' panel shows the field properties for 'Price': ID is 'price', Label is 'Price', Type is 'Float', Context Path is \${context.price}, Mode is 'Editable', and Placeholder is 'My Placeholder'. A red box highlights the 'Price' field and its context path entry.

4. 添加Approve与Reject按钮

在Decisions下，点击 Add

The screenshot shows the 'ApproveBook' form editor with the 'DECISIONS' tab selected. A red box highlights the 'Add' button in the toolbar above the decision table. The table has columns for Text, ID, and Type. A message at the bottom says 'To start, add a decision.'

输入下面的值

First row:

In column Text, enter Approve.

In column ID, make sure that approve is entered.

In column Type, select Positive.

Second row:

In column Text, enter Reject.

In column ID, make sure that reject is entered.

In column Type, select Negative.

The screenshot shows the SAP Cloud Platform Studio interface. In the center, there's a workspace with a project named 'MyWorkflowProject2'. An 'ApproveBook.form' file is open. On the right, there's a 'Properties' panel. In the main area, under the 'DECISIONS' tab, there's a table with two rows. The first row has 'Approve' in the 'Text' column, 'approve' in the 'ID' column, and 'Positive' in the 'Type' column. The second row has 'Reject' in the 'Text' column, 'reject' in the 'ID' column, and 'Negative' in the 'Type' column. A red box highlights the 'Type' column for the 'reject' row. Below the table, there's a log window showing deployment logs.

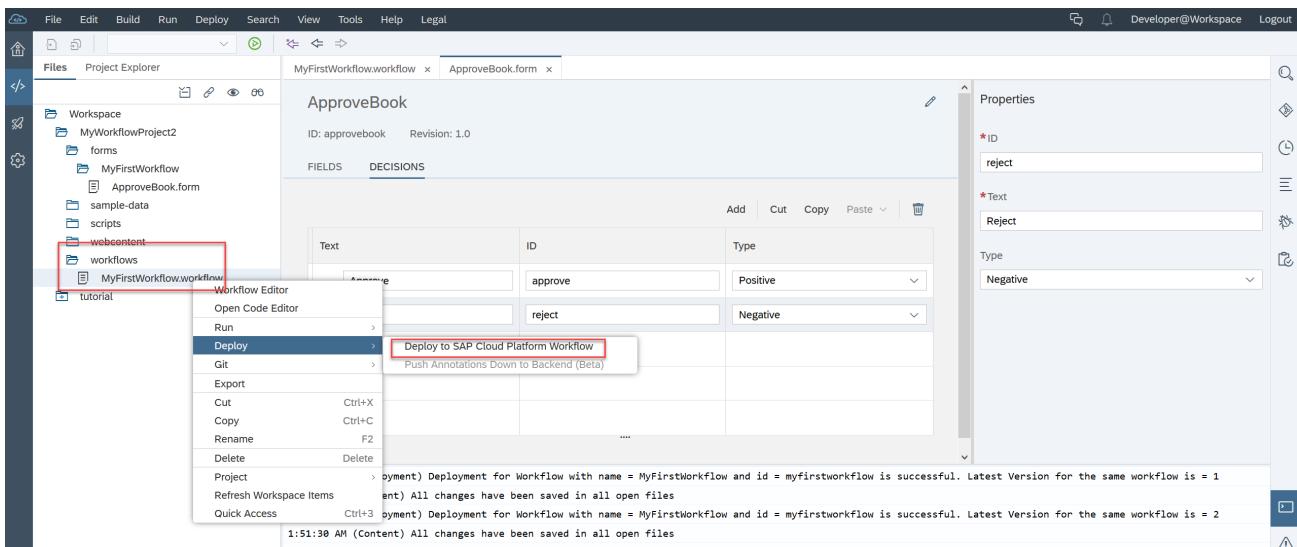
保存你的更改

5. 部署表单、工作流

选中工作流表单文件 Deploy | Deploy to SAP Cloud Platform Workflow.

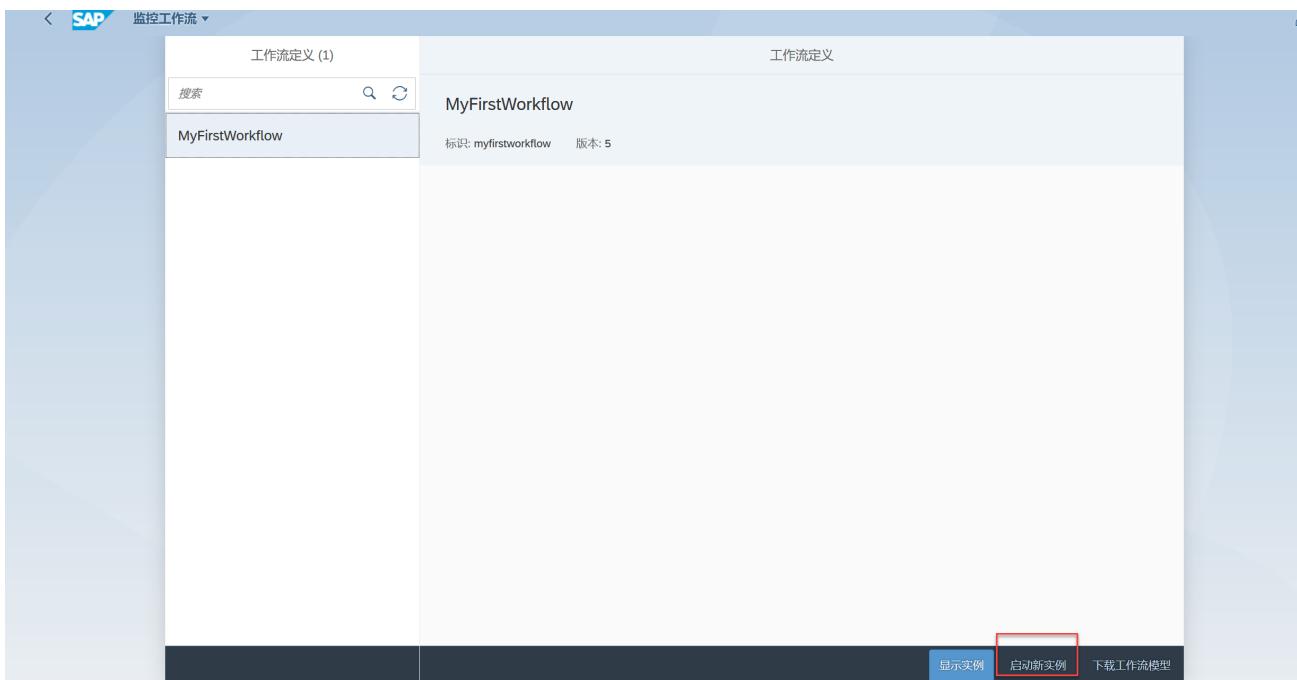
The screenshot shows the SAP Cloud Platform Studio interface. In the left sidebar, under 'MyWorkflowProject2', 'forms' is expanded, and 'ApproveBook.form' is selected. A context menu is open over this file, with 'Deploy' highlighted. A sub-menu for 'Deploy' is shown, with 'Deploy to SAP Cloud Platform Workflow' selected. A red box highlights this option. To the right, the 'ApproveBook' form editor and properties panel are visible. Below the editor, there's a log window showing deployment logs.

选中工作流文件 Deploy | Deploy to SAP Cloud Platform Workflow.

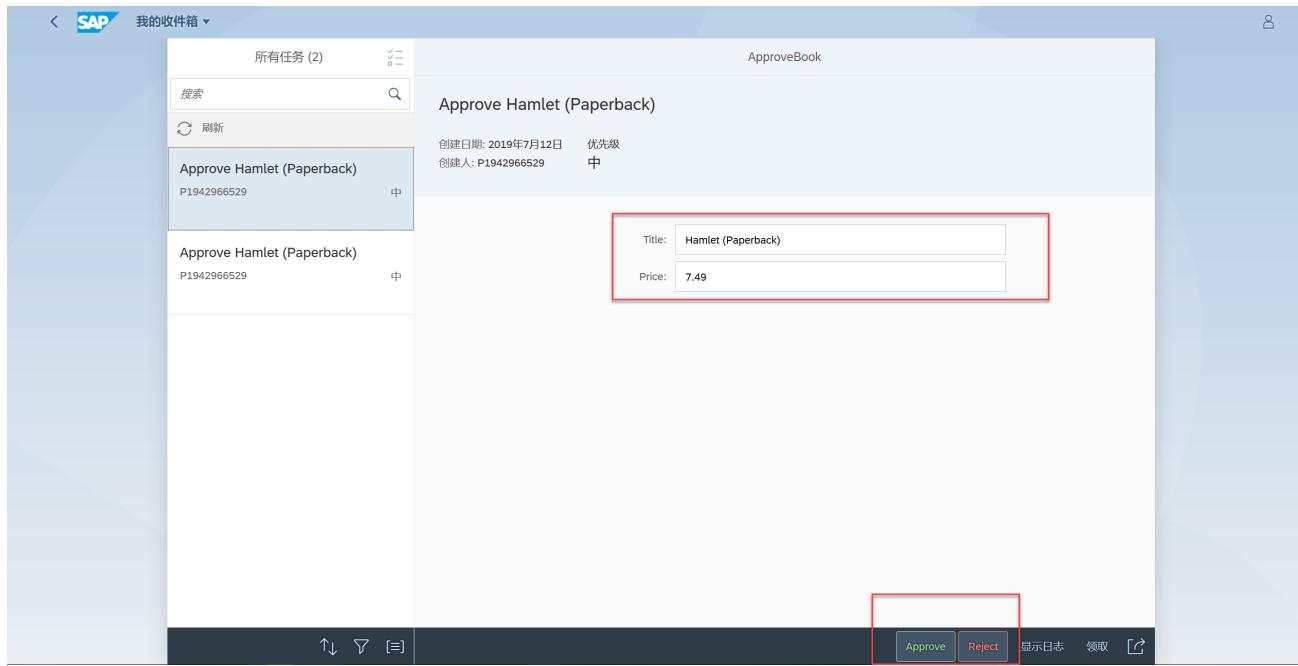


6. 创建新的工作流实例

进入 Monitor Workflow - Workflow Definitions，启动新实例



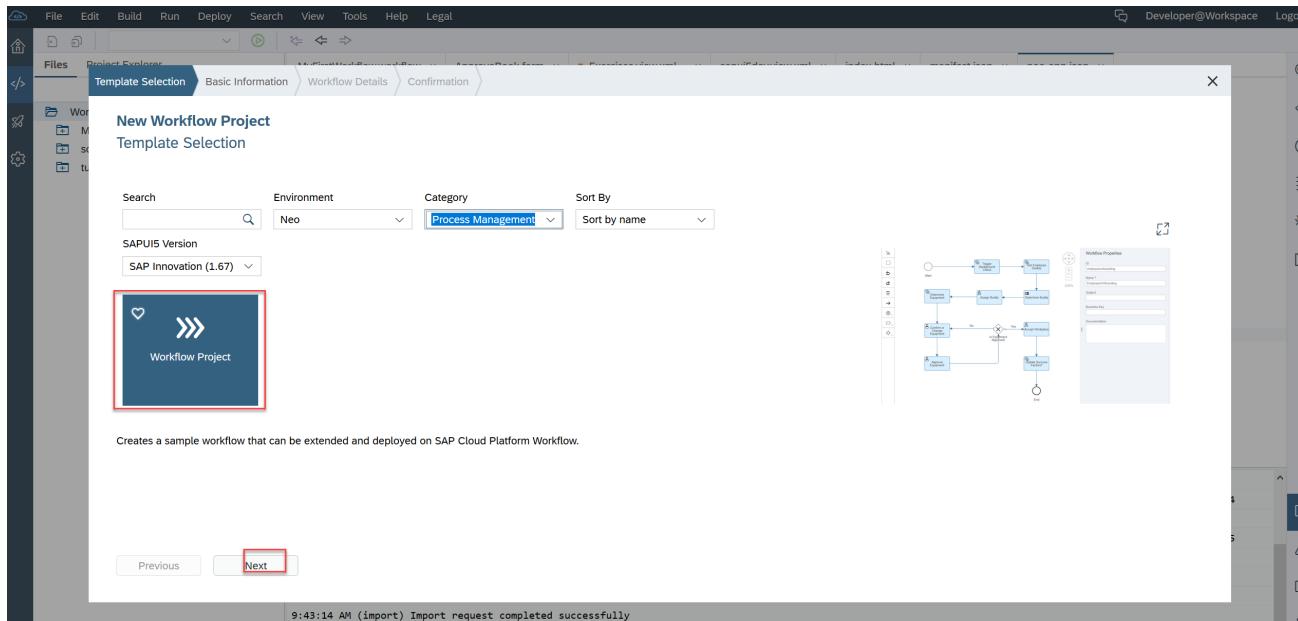
进入My Inbox, 打开任务



工作流集成外部服务

前提：已经创建好Northwind OData Destination

1. 创建新的工作流项目

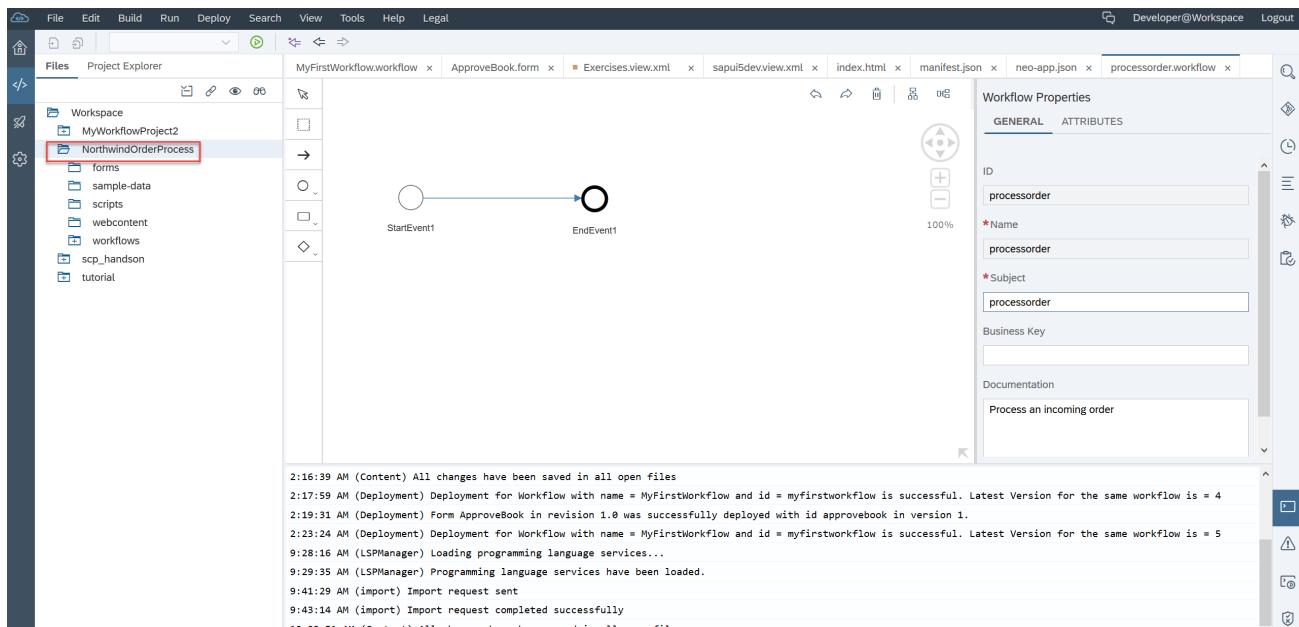


输入工作流的基本信息

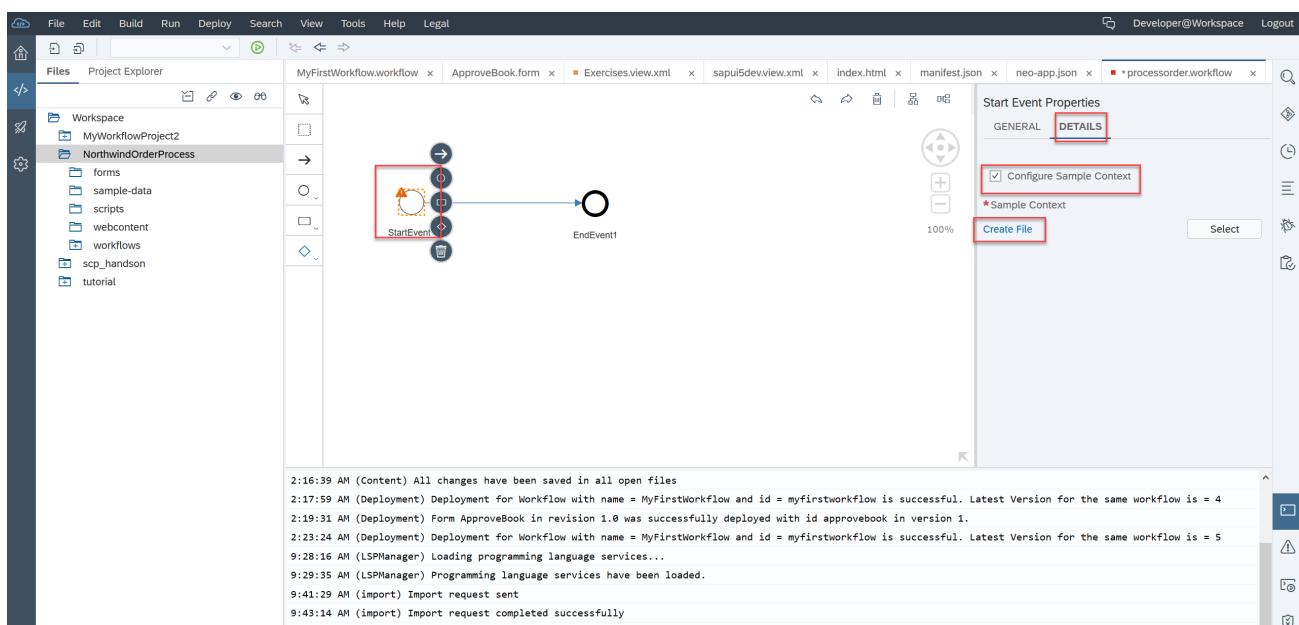
Project Name NorthwindOrderProcess

Name processorder

Description Process an incoming order



选中Start Event，在Details页下，勾选Configure Sample Context，点击Create File



输入文件名sample.json，生成相应的文件

The screenshot shows the Neo4j Studio interface. The Project Explorer on the left lists a workspace named 'MyWorkflowProject2' containing a 'NorthwindOrderProcess' folder. Inside this folder are 'forms', 'sample-data', and 'processororder' sub-folders. The 'sample-data' folder contains a file named 'sample.json', which is highlighted with a red box. The code editor on the right displays the contents of 'sample.json':

```
1 * {  
2   "product": "Hamlet (Paperback)",  
3   "instock": true,  
4   "inventory": 20000,  
5   "price": 7.49,  
6   "publishingDate": "1600-04-23T18:25:43.511Z",  
7   "author": { "name": "William Shakespeare" },  
8   "publishers": [ "simon & brown", "sparknotes", "Dover Publications" ]  
9 }
```

Below the code editor, the terminal window shows deployment logs:

```
2:16:39 AM (Content) All changes have been saved in all open files  
2:17:59 AM (Deployment) Deployment for Workflow with name = MyFirstWorkflow and id = myfirstworkflow is successful. Latest Version for the same workflow is = 4  
2:19:31 AM (Deployment) Form ApproveBook in revision 1.0 was successfully deployed with id approvebook in version 1.  
2:23:24 AM (Deployment) Deployment for Workflow with name = MyFirstWorkflow and id = myfirstworkflow is successful. Latest Version for the same workflow is = 5  
9:28:16 AM (LSPManager) Loading programming language services...  
9:29:35 AM (LSPManager) Programming language services have been loaded.  
9:41:29 AM (import) Import request sent  
9:43:14 AM (import) Import request completed successfully  
10:02:51 AM (Content) All changes have been saved in all open files
```

参照下面的代码，替换文件内容，并部署工作流

```
{  
  "ProductID": 1  
}
```

The screenshot shows the Neo4j Studio interface. The Project Explorer on the left lists a workspace named 'MyWorkflowProject2' containing a 'NorthwindOrderProcess' folder. Inside this folder are 'forms', 'sample-data', and 'processororder' sub-folders. The 'sample-data' folder contains a file named 'sample.json', which is highlighted with a red box. The code editor on the right displays the contents of 'sample.json':

```
1 * {  
2   "ProductID": 1  
3 }  
4
```

Below the code editor, the terminal window shows deployment logs:

```
2:16:39 AM (Content) All changes have been saved in all open files  
2:17:59 AM (Deployment) Deployment for Workflow with name = MyFirstWorkflow and id = myfirstworkflow is successful. Latest Version for the same workflow is = 4  
2:19:31 AM (Deployment) Form ApproveBook in revision 1.0 was successfully deployed with id approvebook in version 1.  
2:23:24 AM (Deployment) Deployment for Workflow with name = MyFirstWorkflow and id = myfirstworkflow is successful. Latest Version for the same workflow is = 5  
9:28:16 AM (LSPManager) Loading programming language services...  
9:29:35 AM (LSPManager) Programming language services have been loaded.  
9:41:29 AM (import) Import request sent  
9:43:14 AM (import) Import request completed successfully  
10:02:51 AM (Content) All changes have been saved in all open files
```

2. 初始化工作流实例

进入Monitor Workflows - Workflow Definitions，选中processororder，点击启动新实例

工作流定义 (2)

MyFirstWorkflow

processororder

标识: processororder 版本: 1

显示实例 启动新实例 下载工作流模型

JSON内容发生了变化，点击启动新实例

工作流定义 (2)

MyFirstWorkflow

processororder

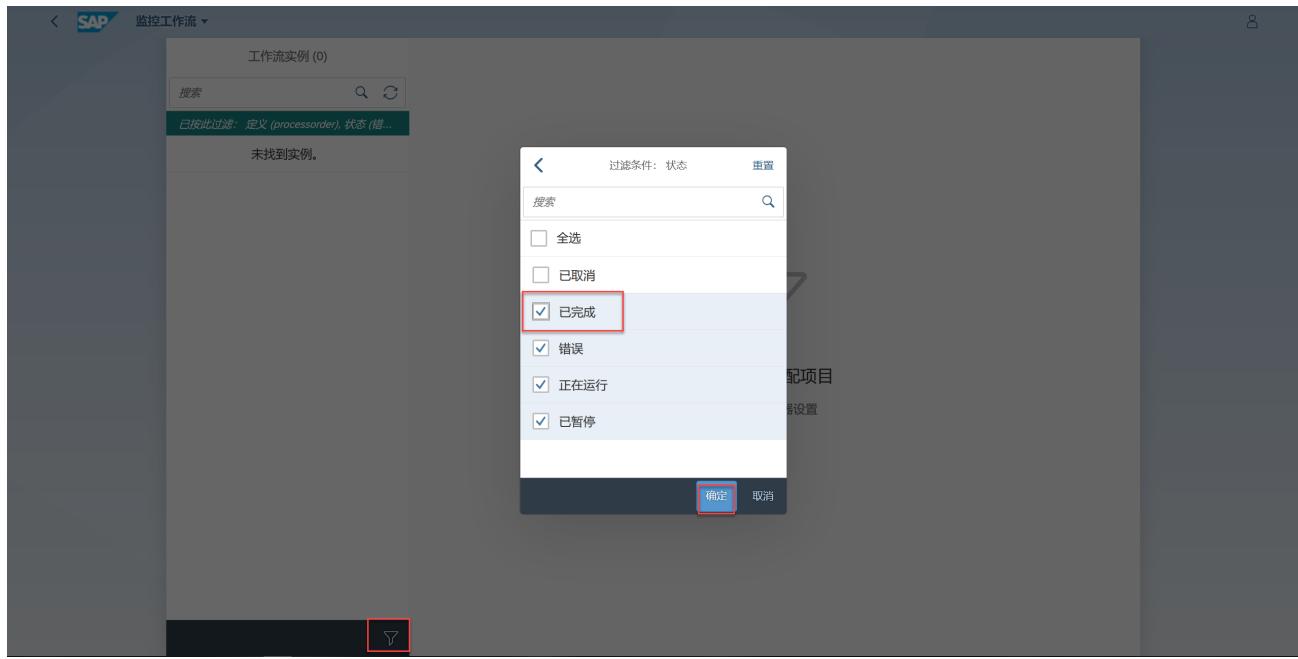
启动新实例

输入 JSON 上下文，通过此上下文启动新实例:

```
{ "ProductID": 1 }
```

启动新实例 取消

进入显示实例，勾选过滤条件：已完成，点击确定

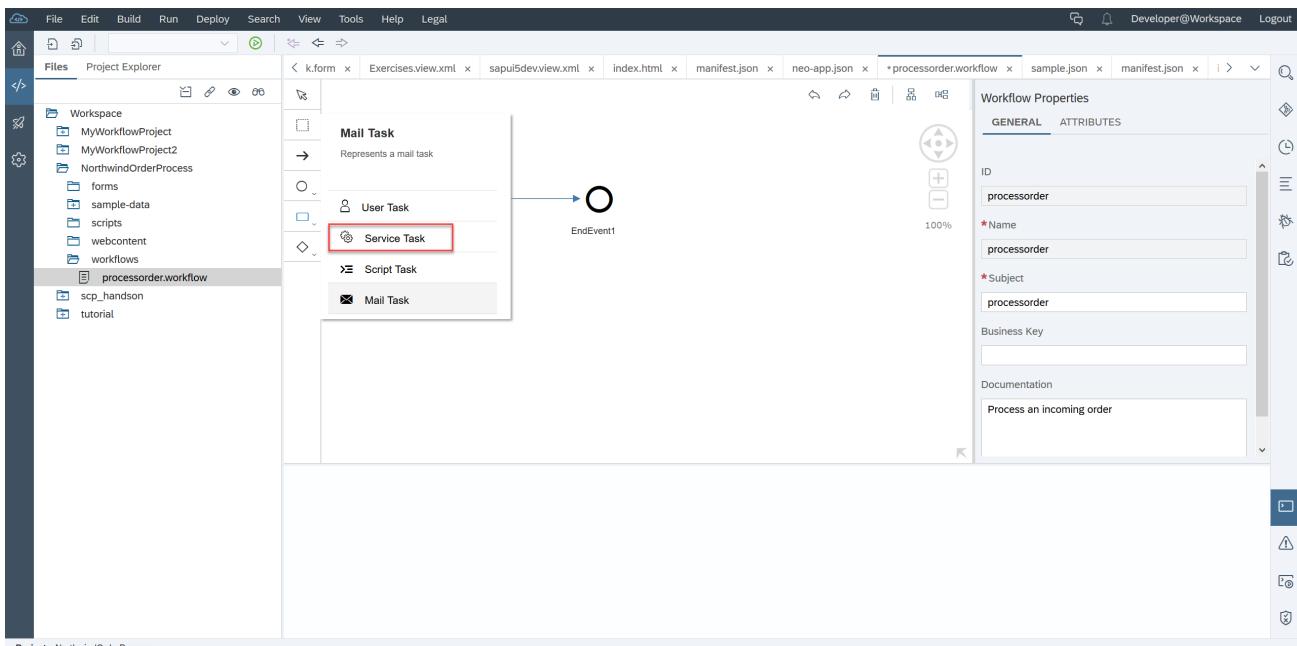


任务显示已完成，说明工作流正常工作

A screenshot of the SAP Monitoring Workflows interface showing a detailed view of a workflow instance. The left sidebar shows '工作流实例 (1)' (Workflow Instances (1)) with one entry: 'processorder' (启动者: P1942966529, 定义标识: processorder, 状态: 已完成). The main panel is titled 'processorder' and displays its details: 实例标识: 0420d333-a45b-11e9-92c2-00163e917ed2, 定义标识: processorder, 定义版本: 1, 启动者: P1942966529, 开始时间: 2019年7月12日下午12:10:42, 完成时间: 2019年7月12日下午12:10:43, 业务代码: . Below this, there are tabs for '工作流信息' (Workflow Information), '错误消息' (Error Messages), and '工作流上下文' (Workflow Context). The '错误消息' tab is active and shows a table with columns '消息' (Message), '活动' (Activity), and '时戳' (Timestamp). A note at the bottom says '无错误消息。' (No error messages.)

3. 通过service task调用外部服务

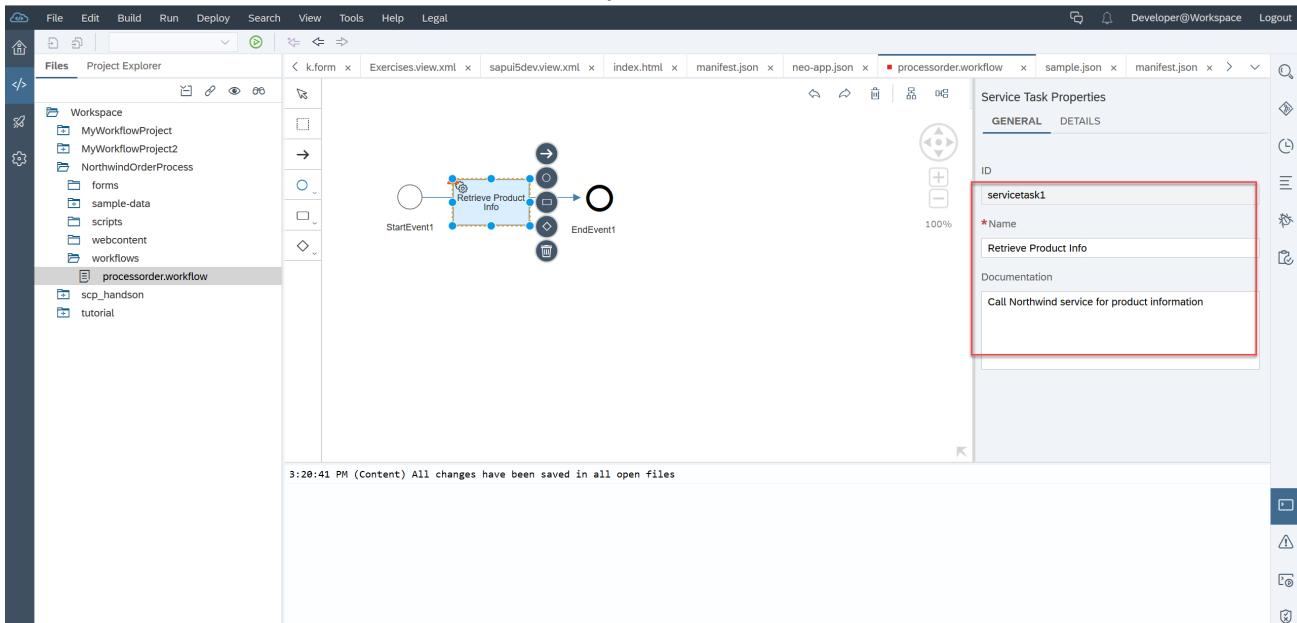
编辑 processorder.workflow文件，添加Service Task



在GENERAL页下输入下列信息

Name Retrieve Product Info

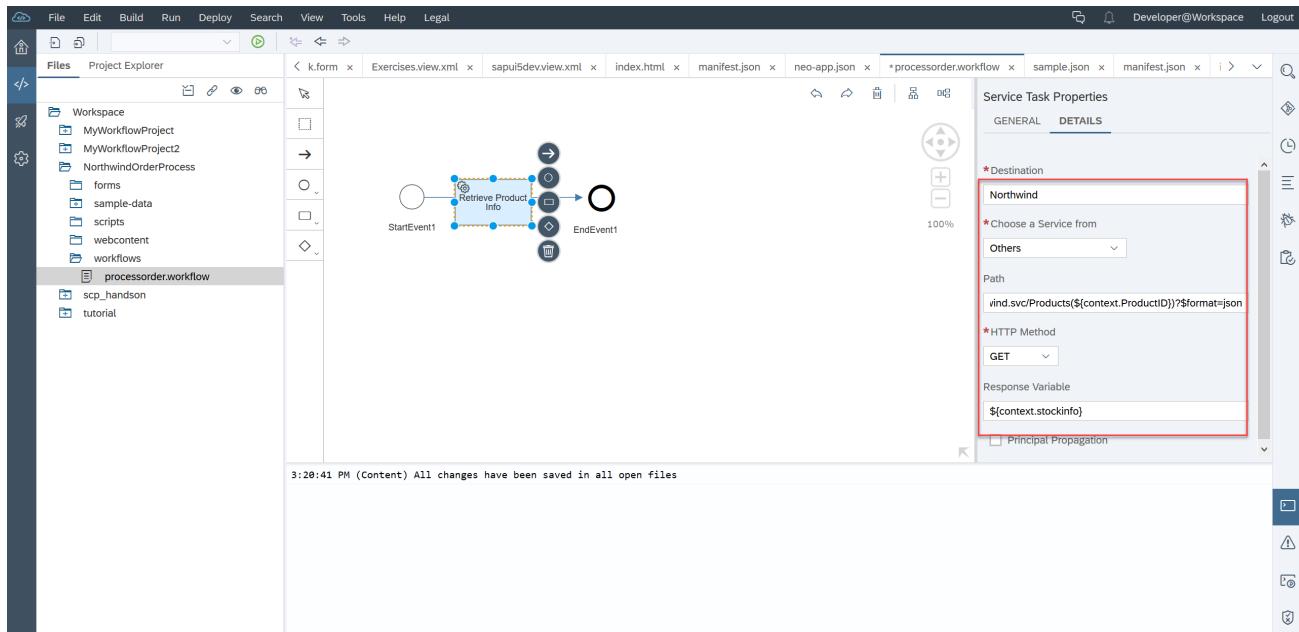
Documentation Call Northwind service for product information



在DETAILS页下输入下列信息



Destination	Northwind
Path	/V3/Northwind/Northwind.svc/Products(\${context.ProductID})?\$format=json
HTTP Method	GET
Response Variable	\${context.stockinfo}



启动新实例，检查工作流上下文

```

1 * {
2     "stockinfo": {
3         "odata.metadata": "https://services.odata.org/V3/northwind/Northwind.svc/$metadata#Products@Element",
4         "ProductID": 1,
5         "ProductName": "Chai",
6         "SupplierID": 1,
7         "CategoryID": 1,
8         "QuantityPerUnit": "10 boxes x 20 bags",
9         "UnitPrice": "18.0000",
10        "UnitsInStock": 39,
11        "UnitsOnOrder": 0,
12        "ReorderLevel": 10,
13        "Discontinued": false
14    },
15    "ProductID": 1
16 }

```

添加User Task与Generic UI到工作流中

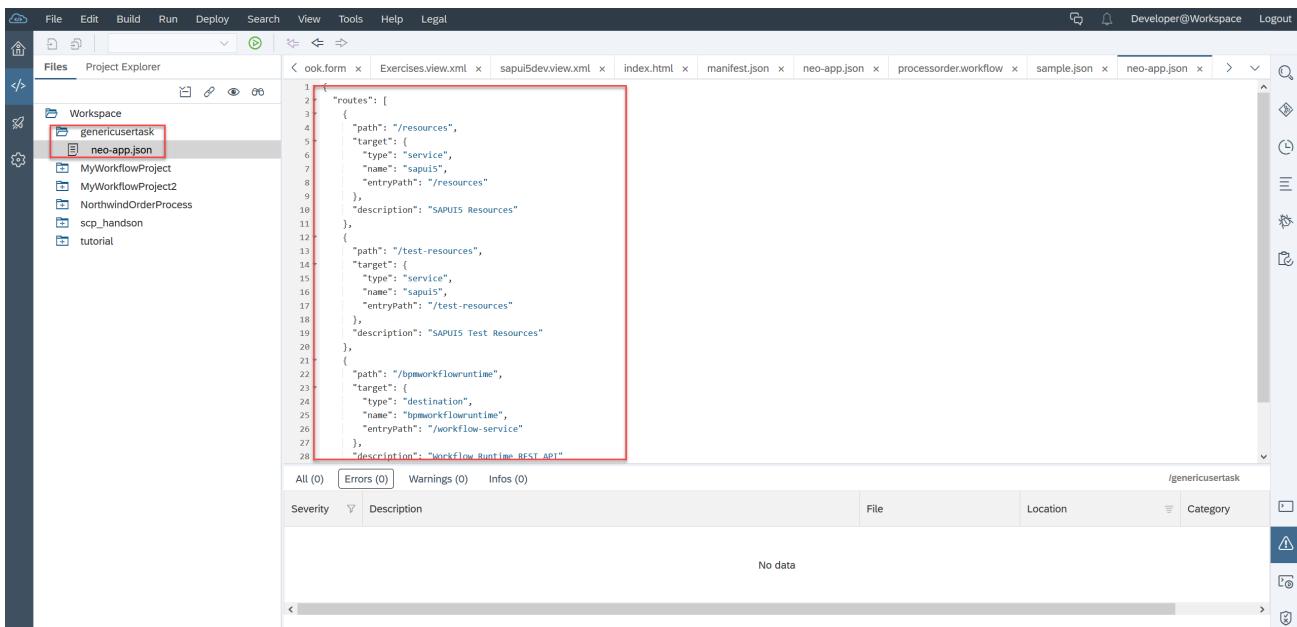
1. 创建文件夹genericusertask，并添加文件neo-app.json，输入下面的代码

```

{
  "routes": [
    ...
  ]
}

```

```
{  
    "path": "/resources",  
    "target": {  
        "type": "service",  
        "name": "sapui5",  
        "entryPath": "/resources"  
    },  
    "description": "SAPUI5 Resources"  
},  
{  
    "path": "/test-resources",  
    "target": {  
        "type": "service",  
        "name": "sapui5",  
        "entryPath": "/test-resources"  
    },  
    "description": "SAPUI5 Test Resources"  
},  
{  
    "path": "/bpmworkflowruntime",  
    "target": {  
        "type": "destination",  
        "name": "bpmworkflowruntime",  
        "entryPath": "/workflow-service"  
    },  
    "description": "Workflow Runtime REST API"  
}  
]  
}
```



在 genericusertask/webapp/下创建Component.js，输入下面代码

```

sap.ui.define([
    "sap/ui/core/UIComponent",
    "sap/ui/Device",
    "sap/ui/model/json/JSONModel"
], function(UIComponent, Device, JSONModel) {
    "use strict";

    return UIComponent.extend("tutorial.genericusertask.Component", {

        metadata: {
            manifest: "json"
        },

        init: function() {
            UIComponent.prototype.init.apply(this, arguments);
            this.getRouter().initialize();

            this.setModel(new JSONModel(Device).setDefaultBindingMode(
"OneWay"), "device");
            this.setModel(new JSONModel({taskDescription:""}), "app");

            // get task data
            var startupParameters = this.getComponentData().startupPar

```

```
ameters;

    var taskModel = startupParameters.taskModel;
    var taskData = taskModel.getData();
    var taskId = taskData.InstanceID;

    // initialize model
    var contextModel = new JSONModel("/bpmworkflowruntime/res
t/v1/task-instances/" + taskId + "/context")
        .attachRequestCompleted(function(oEvent) {
            // Create an array of property/value pairs for gen
            eric display in the UI
            var oModel = oEvent.getSource(),
                mControl = oModel.getProperty("/genericusertask/control"),
                mSource = oModel.getProperty(mControl.source),
                aFormData = mControl.properties.map(function(s
Property) {
                    return {
                        property: sProperty,
                        value: mSource[sProperty]
                    };
                });
            oModel.setProperty("/genericusertask/formItems", a
FormData);
        });
    this.setModel(contextModel);

    // Ensure we have access to the Inbox API before continuin
g
    // (we don't except when running within the My Inbox conte
xt, ie
    // when running "for real", rather than in test mode).
    if (startupParameters.inboxAPI) {

        // get the task description
        var appModel = this.getModel("app");
        startupParameters.inboxAPI.getDescription("NA", taskI
d)
            .done(function(data){
                appModel.setProperty("/taskDescription", data.
```

```
Description);
        })
        .fail(function(errorText){
            jQuery.sap.require("sap.m.MessageBox");
            sap.m.MessageBox.error(errorText, { title: "Error"});
        });
    }

    //add actions
    startupParameters.inboxAPI.addAction({
        type: "Accept",
        label: "Complete"
    }, function() {
        this._completeTask(taskId, true);
    }, this);

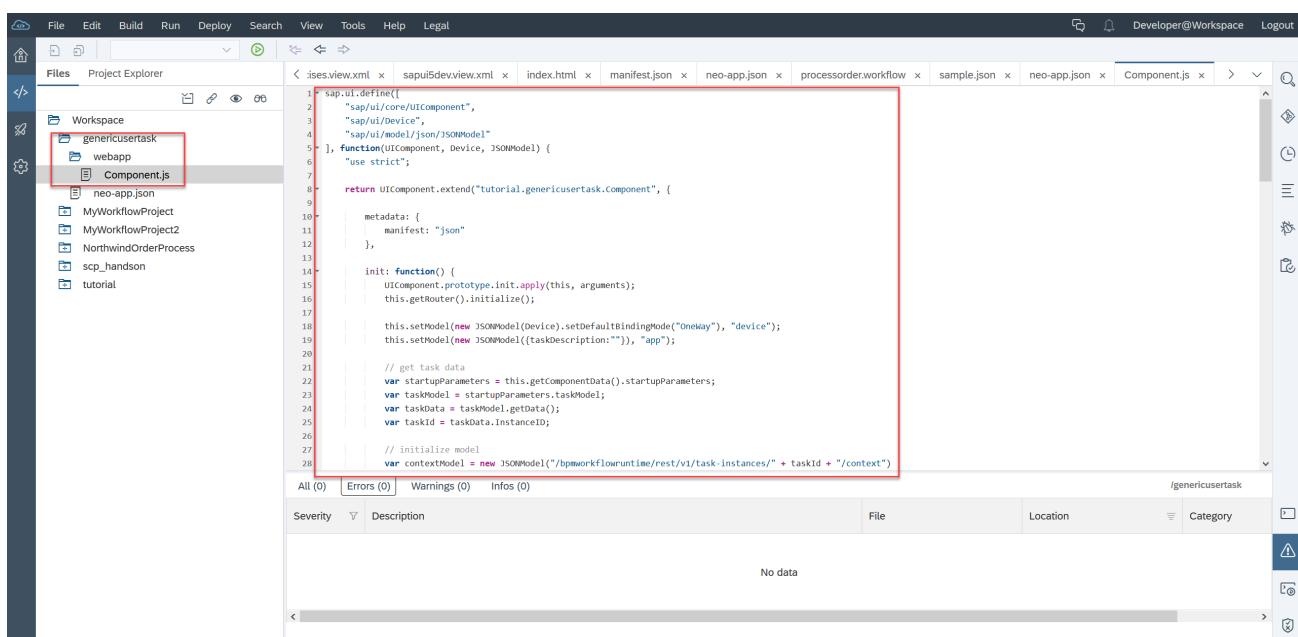
}

},
// Taken mostly straight out of the "Book Approval" tutorial for now
_completeTask: function(taskId) {
    var token = this._fetchToken();
    $.ajax({
        url: "/bpmworkflowruntime/rest/v1/task-instances/" + taskId,
        method: "PATCH",
        contentType: "application/json",
        async: false,
        data: JSON.stringify({
            status: "COMPLETED",
            context: this.getModel().getData()
        }),
        headers: {
            "X-CSRF-Token": token
        }
    });
    this._refreshTask(taskId);
},
```

```

    _fetchToken: function() {
        var token;
        $.ajax({
            url: "/bpmworkflowruntime/rest/v1/xsrf-token",
            method: "GET",
            async: false,
            headers: {
                "X-CSRF-Token": "Fetch"
            },
            success: function(result, xhr, data) {
                token = data.getResponseHeader("X-CSRF-Token");
            }
        });
        return token;
    },

    _refreshTask: function(taskId) {
        this.getComponentData().startupParameters.inboxAPI.updateT
ask("NA", taskId);
    }
});
```



在webapp中创建视图Main.view.xml，输入下面代码

```
<mvc:View
    displayBlock="true"
    xmlns:mvc="sap.ui.core.mvc"
    xmlns="sap.m">
    <App id="idAppControl">
        <pages>
            <Page
                showHeader="false">
                <content>
                    <ObjectHeader
                        icon="sap-icon://person-placeholder" />
                    <Text
                        text="{app>/taskDescription}" />
                    <List
                        items="{/genericusertask/formItems}">
                        <items>
                            <DisplayListItem label="{property}" value=
                                "{value}" />
                        </items>
                    </List>
                </content>
            </Page>
        </pages>
    </App>
</mvc:View>
```

The screenshot shows the SAP Studio interface. In the Project Explorer on the left, there is a folder named 'Component.js' which contains 'Main.view.xml'. This file is currently selected and highlighted with a red box. The main area displays the XML code for 'Main.view.xml'. Below the code editor, there is a status bar with tabs for 'All (1)', 'Errors (0)', 'Warnings (0)', and 'Infos (1)'. A message 'No data' is displayed in the status bar. On the right side, there is a 'Problems' view showing a single warning icon.

```
<i5devview.xml> index.html manifest.json neo-app.json processororder.workflow sample.json neo-app.json Component.js Main.view.xml mar > <Main.view.xml>
<i5devview.xml> index.html manifest.json neo-app.json processororder.workflow sample.json neo-app.json Component.js Main.view.xml mar >
<i5devview.xml>
1 <i5devview>
2   displayBlock="true"
3   xmlns:i5devview="sap.ui.core.mvc"
4   xmlns="sap.m"
5   <App id="idAppControl">
6     <pages>
7       <page>
8         showHeader="false"
9         <content>
10           <objectHeader>
11             icon="sap-icon://person-placeholder"
12             <text>
13               text="{app>/taskDescription}"
14             </text>
15             <list items="{/genericusertask/formItems}">
16               <item>
17                 <displayListItem label="{property}" value="{value}" />
18               </item>
19             </list>
20           </content>
21         </page>
22       </pages>
23     </App>
24   </i5devview>
```

在webapp中添加manifest.json文件，输入下面代码

The screenshot shows a code editor with the 'manifest.json' file open. The file contains the following JSON code:

```
{
  "_version": "1.8.0",
  "sap.app": {
    "id": "tutorial.genericusertask",
    "type": "application",
    "applicationVersion": {
      "version": "1.0.0"
    },
    "title": "{{appTitle}}",
    "description": "{{appDescription}}",
    "sourceTemplate": {
      "id": "ui5template.basicSAPUI5ApplicationProject",
      "version": "1.40.12"
    }
  },
  "sap.ui": {
    "technology": "UI5",
    "icons": {
      "icon": "",
      "favIcon": "",
      "phone": "",
      "phone@2": "",
      "tablet": ""
    }
  }
}
```

```
        "tablet@2": "",
    },
    "deviceTypes": {
        "desktop": true,
        "tablet": true,
        "phone": true
    },
    "supportedThemes": [
        "sap_hcb",
        "sap_belize"
    ]
},
"sap.ui5": {
    "rootView": {
        "viewName": "tutorial.genericusertask.Main",
        "type": "XML"
    },
    "dependencies": {
        "minUI5Version": "1.30.0",
        "libs": {
            "sap.ui.layout": {},
            "sap.ui.core": {},
            "sap.m": {}
        }
    },
    "contentDensities": {
        "compact": true,
        "cozy": true
    },
    "routing": {
        "config": {
            "routerClass": "sap.m.routing.Router",
            "viewType": "XML",
            "async": true,
            "viewPath": "tutorial.genericusertask",
            "controlAggregation": "pages",
            "controlId": "idAppControl"
        },
        "routes": [
            {

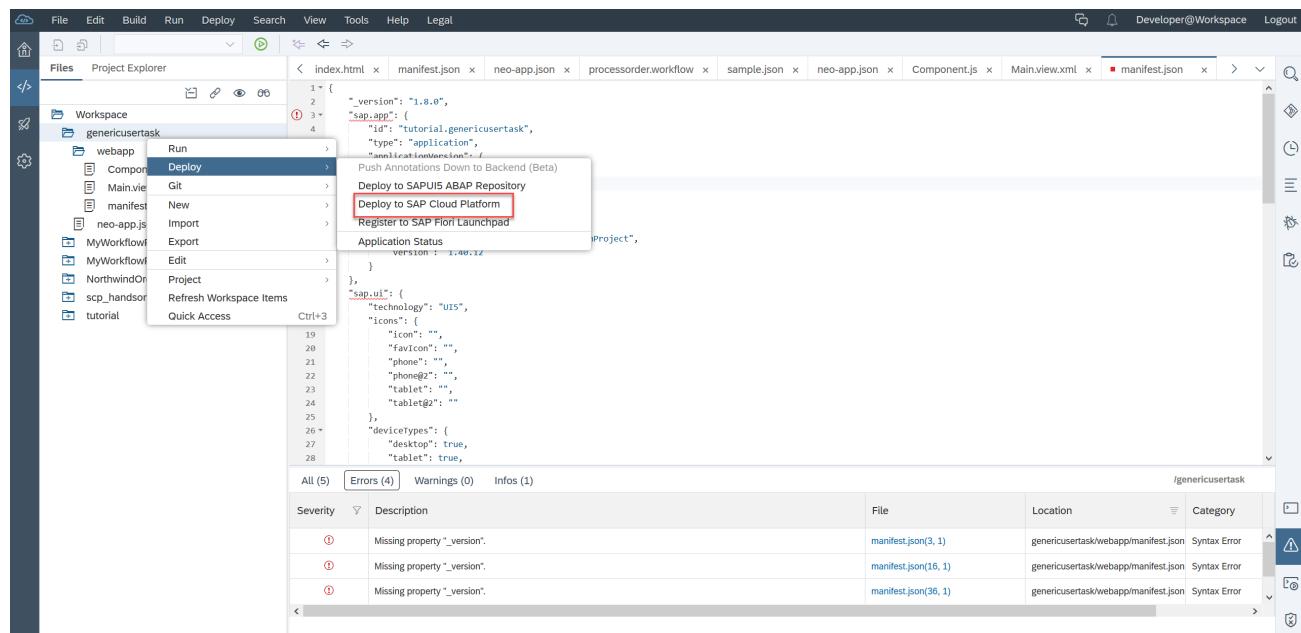
```

```

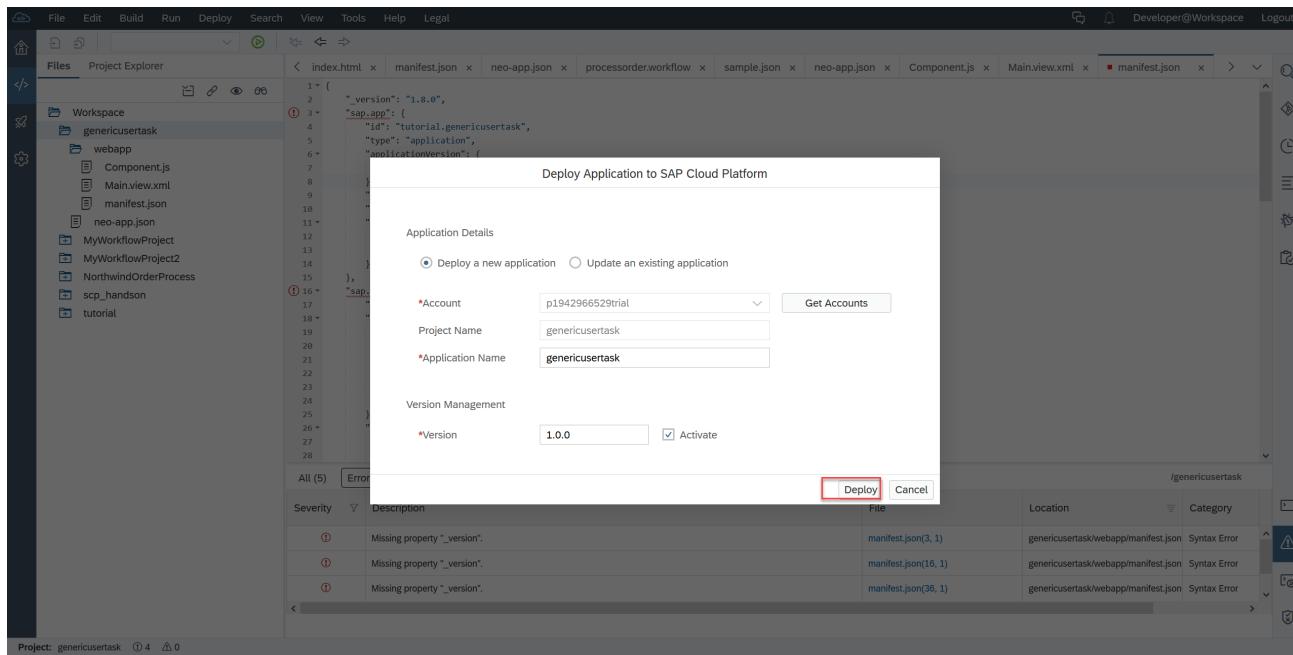
        "name": "RouteView1",
        "pattern": "RouteView1",
        "target": [
            "TargetView1"
        ]
    },
],
"targets": {
    "TargetView1": {
        "viewType": "XML",
        "transition": "slide",
        "clearAggregation": true,
        "viewName": "View1"
    }
},
},
"sap.platform.hcp": {
    "uri": "webapp",
    "_version": "1.1.0"
}
}
}

```

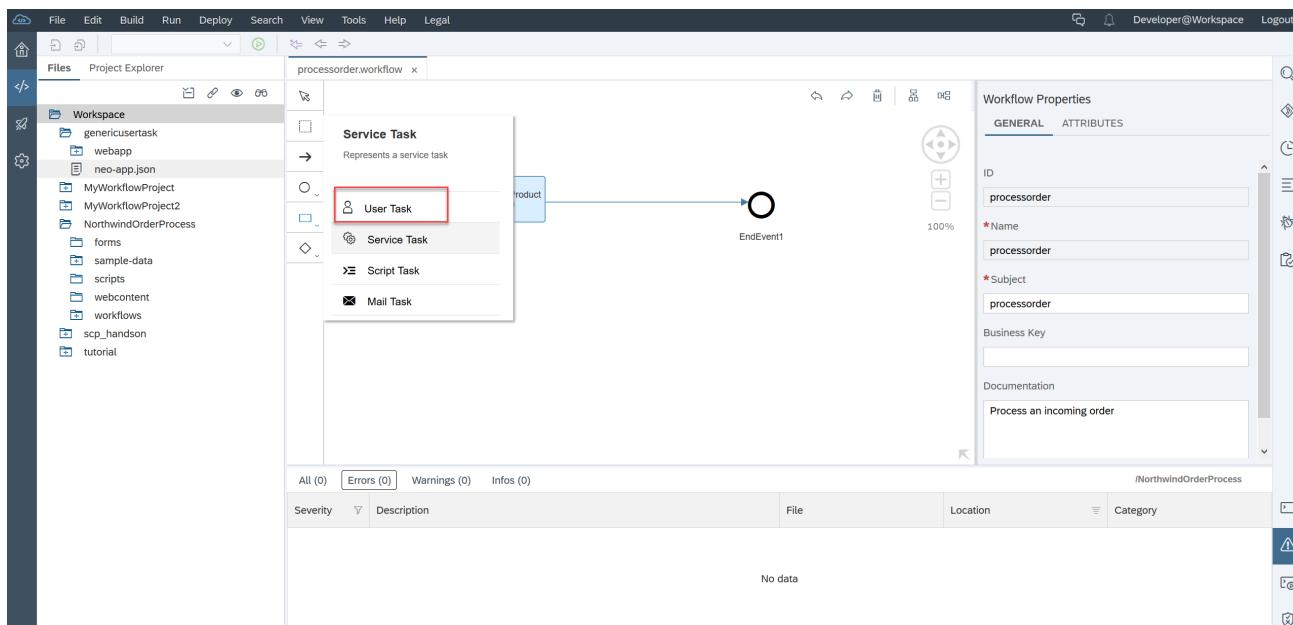
部署到云平台



点击Deploy



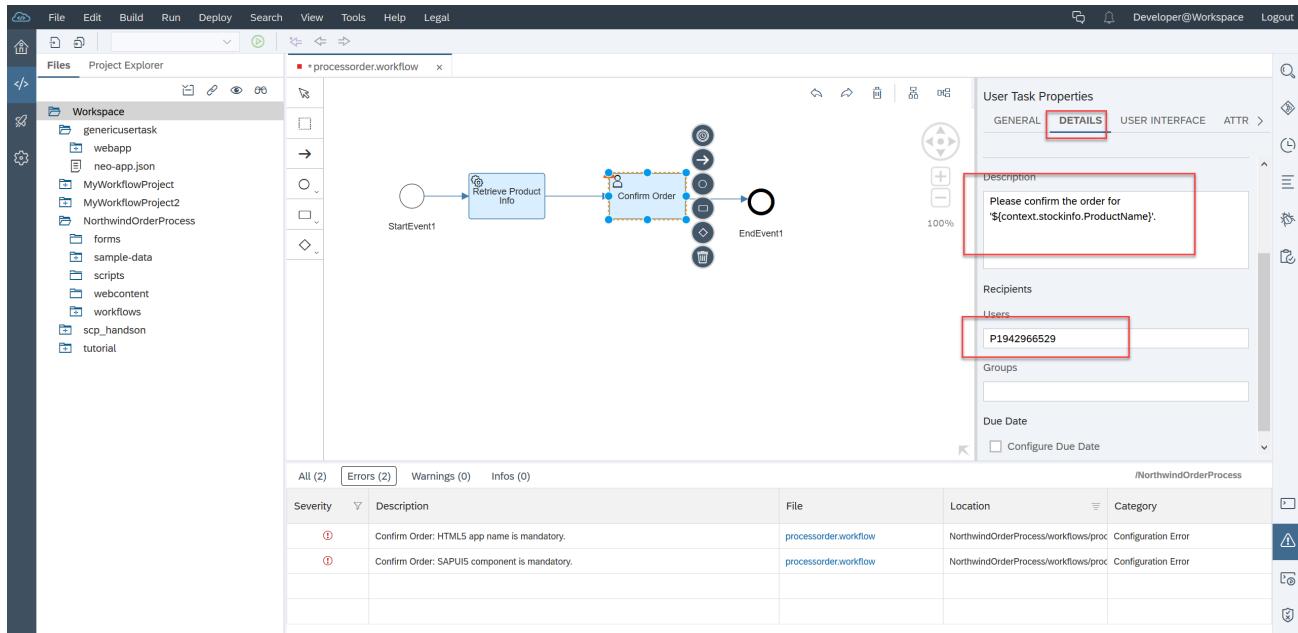
添加新的user task



在Detail中输入下面的信息



Subject Confirm order for \${context.stockinfo.ProductName}
Description Please confirm the order for '\${context.stockinfo.ProductName}'.
Users (Your trial user name, e.g. P999999 - use upper case for the initial letter)

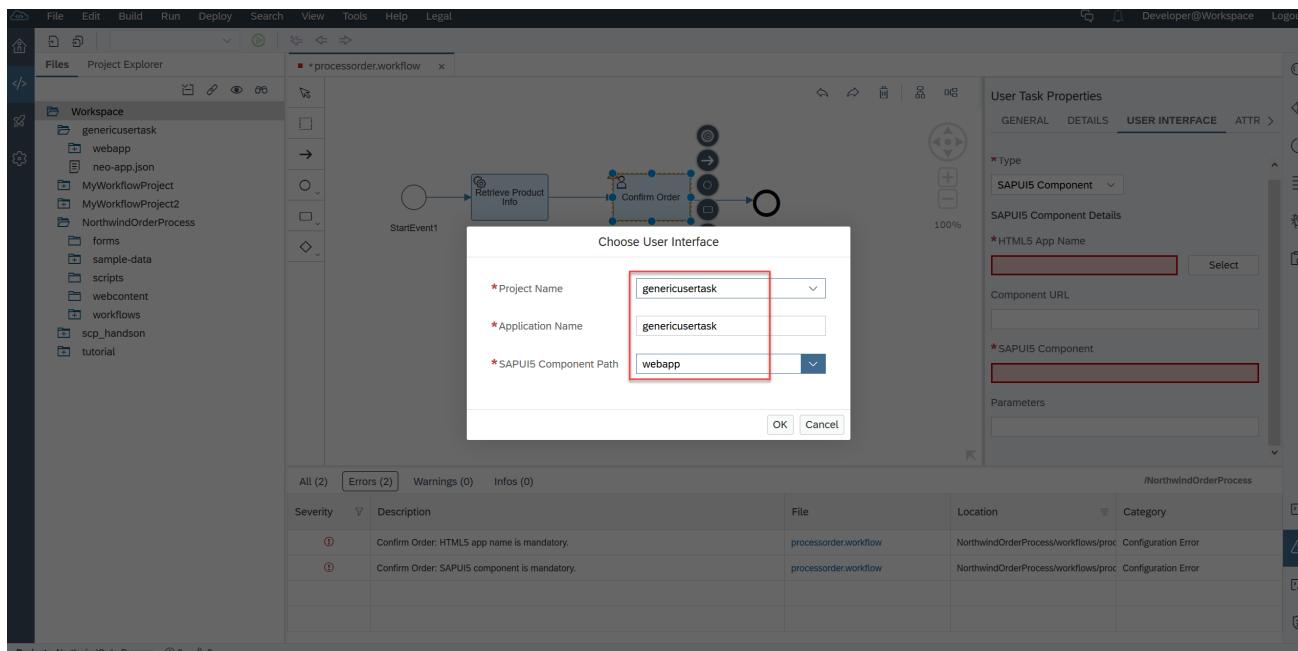


在USER INTERFACE 下输入如下信息

HTML5 App Name genericusertask

Component URL webapp

SAPUI5 Component tutorial.genericusertask



启动新实例



My Inbox ▾

All Items (1)



Confirm Order

Search



Confirm order for Chai

Medium



Please confirm the order for 'Chai'.

ProductID

1

ProductName

Chai

UnitsInStock

39

UnitsOnOrder

0

Complete Show Log Claim [