

项目申请书

项目名称：为 DolphinScheduler 添加服务初始化的
工作流 demo

项目编号：222290459

项目导师：蔡顺峰

申请人：毛国清

日期：2022. 05. 16

邮箱：18552873120@163. com

目录

| | |
|---|----|
| 1. 项目背景 | 3 |
| 1.1 项目相关仓库 | 3 |
| 1.2 项目产出要求 | 3 |
| 1.3 项目描述 | 3 |
| 1.4 项目备注 | 3 |
| 1.5 项目理解 | 4 |
| 2. 技术方法及可行性 | 4 |
| 2.1 Java 相关 | 4 |
| 2.2 Shell 相关 | 4 |
| 2.3 可行性 | 5 |
| 3. 项目实施细节梳理 | 5 |
| 3.1 启动脚本添加 init 模式 | 5 |
| 1) 工作流的功能 | 6 |
| 2) 实现功能具体内容 | 6 |
| 3) 初始化工作流 | 7 |
| 3.2 添加工作流 demo 内容 | 8 |
| 3.3 前端的展示 | 9 |
| 4. 项目开发时间计划 | 10 |
| 4.1 项目研发第一阶段（07 月 01 日-08 月 31 日） | 10 |
| 4.2 项目研发第二阶段（09 月 01 日-09 月 30 日） | 11 |
| 4.3 期望 | 11 |

1. 项目背景

1.1 项目相关仓库

- <https://github.com/apache/dolphinscheduler>
- <https://github.com/apache/dolphinscheduler-website>

1.2 项目产出要求

预置工作流 demo 的 tool 程序

- 服务启动脚本添加 init 模式
- 工作流 demo 包括但不限于 shell 任务、逻辑组件任务 (switch、dependent、subprocess、condition)、参数传递等

1.3 项目描述

用户在启动 DolphinScheduler 服务后,可以使用 demo-tool 程序预置工作流 demo,包括但不限于简单的 shell 任务、逻辑组件任务 (switch、dependent、subprocess、condition)、参数传递等功能,引导用户更方便地使用 DolphinScheduler。

1.4 项目备注

通过项目学到如何熟悉开源社区、它是什么、它是如何工作的以及如何做出贡献;熟悉 apache/dolphinscheduler 项目,关于任务调度系统的知识,它的重点是什么,它是如何运行的;

1.5 项目理解

通过对系统项目的学习，系统中核心功能中 workflow 一系列的流程如下：



用户在启动 DolphinScheduler 服务后，在 workflow 流程最开始的阶段，需要创建新的 workflow。原本 workflow 是用户按照自己的需求，自行去创建并定义 workflow，这点需要用户对系统 workflow 功能熟悉的情况，因此项目希望可以通过预置 workflow demo 的 tool 程序，帮助用户去快速创建 workflow 并开启应用。

2. 技术方法及可行性

2.1 Java 相关

我日常在学校实验室中负责智能制造执行系统，主要功能是对生产资源、生产过程的管理，在其中负责服务器架构，后端开发。我参加过 Java 后端开发的暑期实习，有过一定的 Java 基础和框架实践开发经验。并且我曾尝试利用 DolphinScheduler 安排项目 workflow，就是上手 DolphinScheduler 实现在线调度，因此对 Java 和 DolphinScheduler 都有一定的了解和认识。

2.2 Shell 相关

实验室中的 MES 系统，我利用自动化脚本打包代码，前后端均以 Docker 作为容器，进行持续集成和持续部署，以实现构建部署的自动化，因此对 Shell 脚本和自动化部署一系列流程有着自己的理解。本次项目也是需要通过脚本实现自动化初始工作流 Demo 的功能，我可以借鉴之前对于自动化部署的经验，更好地运用在本次项目中。

2.3 可行性

对于该项目的开发，有不少开源项目给出相近的处理。我需要的是熟悉该项目的功能，将现有的技术开发应用到项目中去，要去分析开发过程并解决可能出现的问题。

参考：<https://github.com/jenkinsci/jenkins>

<https://gitee.com/openspug/spug>

3. 项目实施细节梳理

3.1 启动脚本添加 init 模式

项目的最开始就是要对整个项目的流程梳理分析。对于初始化 demo 功能的开发，本质是用户对工作流的创建。在开发的过程中，首先要的就是理清开发过程的整体逻辑，因此先是要熟悉工作流的功能，再明白所需要实现项目功能的具体内容，最后才是将初始化脚本的创建添加到系统代码中。

1) 工作流的功能

对整个系统进一步的阅读，尤其是把要进行开发的页面功能进行深一步的理解后实践运行。对于原来项目中工作流的创建，在创建完项目后点击项目，显示创建工作流如下图：



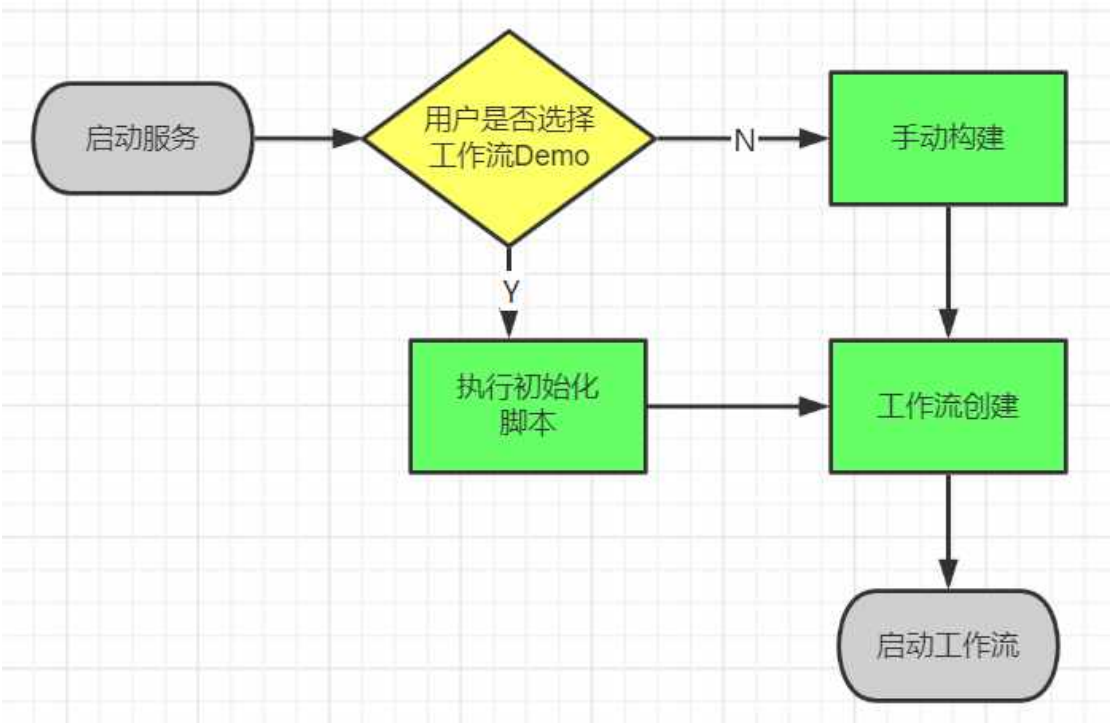
点击你所要添加的工作任务，将其拖拽到右侧，会自动跳出参数设置的窗口，其目的是对任务进行设置，参数填写完毕基本一个任务设置成功，然后就是再进行添加其他工作流任务，这个就是需要实现 demo 的基础功能。



2) 项目功能具体内容

基于用户的使用视角，我在设计这个初始化工作流 demo 的想法是在用户启动项目服务之后，用户可以自行去选择是否使用该工作流 demo 功能，如果不使用就是按原本系统构建，使用就

是基于原本系统接口，使用脚本自动化构建，其大致实现的流程如下图：



3) 初始化工作流

其中是否选择脚本来初始化工作流的需要发生在启动服务之后，并且脚本只是为了自动化手动构建工作流创建的过程，初始化脚本负责的就是调用工作流所包含的一系列的任務。因此在启动脚本的时候需要借鉴原本系统中工作流的一系列的接口，其中工作流流程定义创建的 API 如下：

| | | |
|----------|-------------------------|----|
| 流程定义相关操作 | | 文档 |
| GET | queryListPaging | |
| POST | createProcessDefinition | |
| GET | queryAllByProjectCode | |
| POST | batchCopyByCodes | |
| POST | batchDeleteByCodes | |
| POST | batchExportByCodes | |
| POST | batchMoveByCodes | |

调试

createProcessDefinition

接口地址 /dolphinscheduler/projects/{projectCode}/process-definition

请求方式 POST

consumes ["application/json"]

produces ["*/*"]

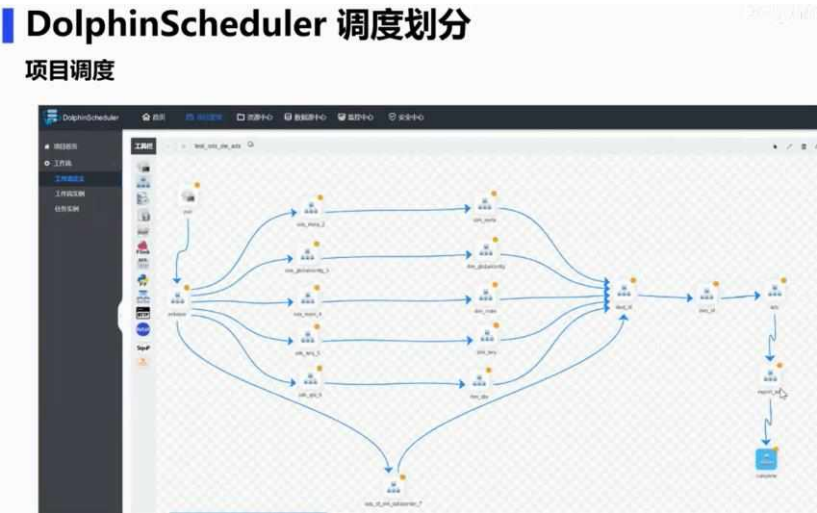
接口描述 创建流程定义

请求参数

深入这个接口去看，发现里面主要就是对 workflow 初始状态内容的数据操作，因此在可以选择使用这个接口，创建用户所需要的工作流任务，再调用这些接口来实现添加工作流具体内容从而构建初始化模式。这里参考开源项目 Jenkins 中，其对于项目的自动化处理功能也是代码接口嵌套自动化脚本进行开发，对本次项目开发有借鉴作用。

3.2 添加工作流 demo 内容

基本开发逻辑弄清楚之后，剩下的是针对 DolphinScheduler 项目工作流 demo 内容具体开发，项目要求的是包括但不限于 shell 任务、逻辑组件任务（switch、dependent、subprocess、condition）、参数传递。为了让系统的用户能够完整体验到工作流中的全部功能，对于原本系统工作流中所有类型的任务进行添加，这里需要和导师讨论并拟定一套开发者用户常用的基础工作流流程，例如下图的样式，其目的还是为了确保自动化 demo 创建完成的工作流效果和手动创建效果相同且更加高效。



用户如果想要快速创建工作流，就可以在这个最基础的工作流 demo 上修改工作流任务的参数或者修改具体任务之间调度的安排，这样用户可以更方便地使用 DolphinScheduler。因为需要基于用户角度进行考虑，所以在添加工作流 demo 内容上尽量先对用户的使用进行多次调研，对于用户常用功能进行添加，去掉不必要的功能。

3.3 前端的展示

因为在项目设计中还需要照顾到使用者的简单方便操作，所以具体的前端设计需要后续开发中进一步细化，从添加工作流 demo 内容到初始化样式等等，并确保所增加的功能是简洁易可用。比如在实现初始化工作流 demo 功能时，在和项目导师讨论所要达到的操作效果时，希望前端开发要能够实现一键创建工作流 demo 的操作。通过找到工作流定义页面，其前端的路径在 `dolphinscheduler-ui/src/views/projects/workflow`。



在这里我参考之前自动化部署的学习，发现开源项目 spug 对于操作自动化脚本的前端开发也是值得学习，本次项目也可以

显示常规创建和自定义创建工作流，常规创建就是下一步直接进入工作流 demo 模式。这个环节主要确保前端能接收数据并显示初始化工作流的全部操作，一些页面渲染的细节等待后续和导师的讨论。

4. 项目开发时间计划

因为我在暑假期间没有其他任务，我可以全职工作（不限时间，每周至少 40 小时）。项目的导师可以帮助我解决一些谷歌解决不了的问题。我会和导师讨论我自己无法克服的困难，并且每周通过电子邮件向导师发送一周总结，告知我迄今为止的工作进展以及我本周的一些疑问。

4.1 项目研发第一阶段（07 月 01 日-08 月 31 日）

由于存在开发新的功能，拟定编码周期为 2 个月（8 周），我计划使用 1 周的掌握系统和 6 周的功能开发。剩下的时间用于开发新的用例和时间用于意外延迟，所以前 7 周的工作量是：

- 读懂系统前后端的开发逻辑，进行开发文档的记录总结
- 完成服务启动脚本添加 init 模式，实现初始化工作流的逻辑操作
- 对用户使用 DolphinScheduler 系统常用功能进行调研
- 添加工作流 demo 内容，初始化脚本结合系统代码开发

- 前端完成开发，实现对 workflow 初始化全部功能
- 实现开发代码在测试环境上跑通并进行初步测试
- 完成 DolphinScheduler 仓库的 pr

4.2 项目研发第二阶段（09 月 01 日-09 月 30 日）

- 对系统的展示进一步的修改并美化样式
- 解决在中期验收阶段中发现的问题
- 对第一阶段完成的内容进行更详细的测试
- 对第一阶段的完成内容进行总结，并输出相关文档内容
- 思考可以改进或者补充的地方
- 如有必要，我也会在工作中征求用户对教程的建议：

<https://dolphinscheduler.apache.org/zh-cn/community/development/subscribe.html>

4.3 期望

学校老师目前的方向是生产智能调度系统，而开源的项目 DolphinScheduler 是一个可视化 DAG 工作流任务调度系统。两者都是调度问题，有一定相同点，我可以在项目开发中不断深入学习与探索。另外自己课题对于调度系统的研究，能给开源社区带来不少新的想法和实践应用。