ModelArts

Gallery 用户指南

文档版本 01

发布日期 2024-10-26

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

目 录

1 AI Gallery（新版）

1.1 AI Gallery 使用流程

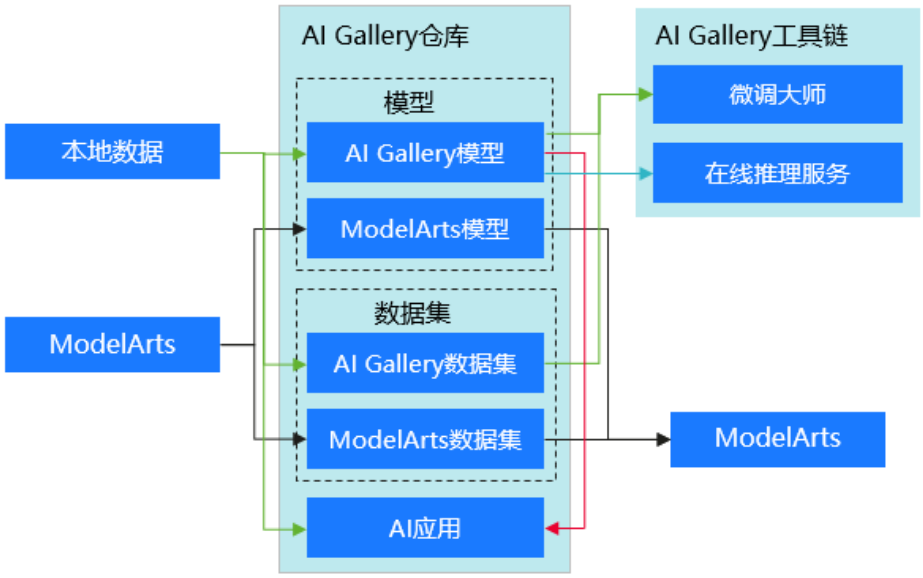
AI Gallery提供了模型、数据集、AI应用等AI数字资产的共享，为高校科研机构、AI应用开发商、解决方案集成商、企业级/个人开发者等群体，提供安全、开放的共享及交易环节，加速AI资产的开发与落地，保障AI开发生态链上各参与方高效地实现各自的商业价值。

使用流程

本节主要介绍在AI Gallery中管理资产的整体流程。

1. 在AI Gallery中，需要先将本地数据上传到AI Gallery仓库，创建AI Gallery模型、AI Gallery数据集、AI应用等资产，具体可参见**托管模型到AI Gallery**、**托管数据集到AI Gallery**、**发布本地AI应用到AI Gallery**。
2. 资产创建完成后，需要将资产进行发布操作，具体可参见**发布模型到AI Gallery**、**发布数据集到AI Gallery**。对于支持部署为AI应用的AI Gallery模型，可将此模型部署为AI应用，具体可参见**将AI Gallery中的模型部署为AI应用**。
3. 发布后的资产，可通过微调大师训练模型和在线推理服务部署模型，具体可参见**使用AI Gallery微调大师训练模型**、**使用AI Gallery在线推理服务部署模型**。

图 1-1 AI Gallery 使用流程



AI Gallery也支持管理从ModelArts中发布的模型和数据集等资产，具体可参见**发布数据集到AI Gallery**、**将Workflow工作流发布到AI Gallery**、**将ModelArts AI应用发布到AI Gallery**、**发布算法到AI Gallery**。

发布到AI Gallery中的资产，也支持在ModelArts中订阅使用，具体可参见**从AI Gallery订阅模型**、**从AI Gallery订阅Workflow工作流**。

1.2 发布和管理 AI Gallery 模型

1.2.1 构建模型

1.2.1.1 自定义模型规范

AI Gallery除了支持托管文本生成和文本问答任务类型的模型，还支持托管其他任务类型的模型，其他任务类型的模型被称为自定义模型。但是托管的自定义模型要满足规范才支持使用AI Gallery工具链服务（微调大师、在线推理服务）。

自定义模型的使用流程

步骤1 **托管模型到AI Gallery**。

- 模型基础设置里的“任务类型”选择除“文本问答”和“文本生成”之外的类型。
- 上传模型文件时需要确认待上传的文件是否满足自定义模型规范。如果模型要支持训练，则需要满足**自定义模型规范（训练）**；如果模型要支持推理，则需要满足**自定义模型规范（推理）**。

步骤2 **发布模型到AI Gallery**。

步骤3 **使用AI Gallery微调大师训练模型**或**使用AI Gallery在线推理服务部署模型**。

- 如果进行模型微调，则“训练任务类型”选择“自定义”。
- 如果部署为推理服务，则“推理任务类型”选择“自定义”

----结束

自定义模型规范（训练）

当托管自定义模型到AI Gallery时，如果模型要支持AI Gallery的模型微调，则需要在“模型文件”添加gallery_train文件夹，文件夹内容参考**表1-1**。

须知

- gallery_train文件夹必须是一级目录直接上传，否则会被判定不符合自定义模型规范，无法使用模型微调。
- 如果自定义模型的模型文件不符合gallery_train文件列表要求或文件内容为空，都将不能正常进行模型微调。

表 1-1 gallery_train 文件列表

文件类型	文件说明
“train.py”	必选文件，训练脚本文件，定义了自定义模型的训练处理方式。代码示例请参见**train.py示例**。 如果训练脚本里使用了其他脚本文件，则必须一起打包在gallery_train文件夹里上传，否则会导致微调失败。
“train_params.json”	必选文件，训练参数文件，定义了模型训练的必要参数，例如训练方式、超参信息。该参数会显示在微调工作流的“作业设置”页面的算法配置和超参数设置里面。代码示例请参见**train_params.json示例**。
“dataset_readme.md”	必选文件，数据集要求说明，定义了模型训练时对数据集的要求，会显示在微调工作流的“准备数据”页面。
“requirements.txt”	非必选文件，环境配置文件，定义了项目依赖的python包。AI Gallery提供了基础镜像的依赖环境，如果要添加自定义依赖项，可通过requirements.txt文件实现。基础镜像包含python、PyTorch、cuda（GPU）、CANN（NPU）。

自定义模型规范（推理）

当托管自定义模型到AI Gallery时，如果模型要支持AI Gallery的推理服务，则需要在“模型文件”添加gallery_inference文件夹，文件夹内容参考**表1-2**。

须知

- gallery_inference文件夹必须是一级目录直接上传，否则会被判定不符合自定义模型规范，无法使用模型微调。
- 如果自定义模型的模型文件不符合gallery_inference文件列表要求或文件内容为空，都将不能正常部署在线推理服务。

表 1-2 gallery_inference 文件列表

文件类型	文件说明
“inference.py”	必选文件，推理脚本文件，定义了自定义模型的推理处理方式，包含初始化推理（init）和输入输出（call函数）。代码示例请参见**inference.py示例**。 如果推理脚本里使用了其他脚本文件，则必须一起打包在gallery_inference文件夹里上传，否则会导致推理失败。
“requirements.txt”	非必选文件，环境配置文件，定义了项目依赖的python包。AI Gallery提供了基础镜像的依赖环境，如果要添加自定义依赖项，可通过requirements.txt文件实现。基础镜像包含python、PyTorch、cuda（GPU）、CANN（NPU）。

自定义模型使用的预置镜像

AI Gallery提供了PyTorch基础镜像，镜像里已经安装好了运行任务所需的软件，供自定义模型直接使用，快速进行训练、推理。预置镜像的版本信息请参见**表1-3**。

表 1-3 AI Gallery 预置镜像列表

引擎类型	资源类型	版本名称
PyTorch	NPU	pytorch_2.0.1-cann_6.3.2-py_3.9-euler_2.10.7-aarch64
	GPU	pytorch_2.0.0-cuda_11.7-py_3.9.11-ubuntu_20.04-x86_64

“train.py”示例

表 1-4 环境变量说明

变量名称	说明	示例
ENV_AG_MODEL_DIR	模型存放路径，AI Gallery的模型仓库地址，包含模型仓库的所有文件。	“/home/ma-user/.cache/gallery/model/ur12345--gpt2”

变量名称	说明	示例
ENV_AG_DATASET_DIR	数据集存放路径，AI Gallery的数据集仓库地址，包含数据集仓库的所有文件。	“/home/ma-user/.cache/gallery/dataset/ur12345--data_demo”
ENV_AG_USER_PARAMS	配置的训练超参json字符串。创建训练任务时在算法配置页面设置的超参，用json字符串表示。	{"per_device_eval_batch_size":"32","lr":"0.001","logging_steps":"24"}
ENV_AG_TRAIN_OUTPUT_DIR	训练产物文件存放路径。训练产物将被保存到该路径。训练任务结束后，由AI Gallery平台将该目录上传到新模型的仓库中。	“/home/ma-user/.cache/gallery/output”
ENV_AG_USER_METRICS_LOG_PATH	训练数据的日志文件存放路径。训练过程中的迭代次数、LOSS和吞吐数据按照“迭代次数|loss|吞吐”格式记录在日志中，AI Gallery通过环境变量找到日志，从中获取实际数据绘制成“吞吐”和“训练LOSS”曲线，呈现在训练的“指标效果”中。具体请参见**查看训练效果**。 **说明** 日志文件中的迭代次数、LOSS和吞吐数据必须按照“迭代次数|loss|吞吐”格式存放，否则AI Gallery会数据解析失败，导致“吞吐”和“训练LOSS”曲线异常。	“/var/logs/user_metrics.log”

```
import json
import os

from datasets import load_dataset
from transformers import AutoImageProcessor
from torchvision.transforms import RandomResizedCrop, Compose, Normalize, ToTensor,
RandomHorizontalFlip
import numpy as np
from transformers import AutoModelForImageClassification, TrainingArguments, Trainer
from transformers import DefaultDataCollator
from sklearn import metrics

# 环境变量
# 工作目录
ENV_AG_WORK_DIR = 'ENV_AG_WORK_DIR'
# 模型存放路径
ENV_AG_MODEL_DIR = 'ENV_AG_MODEL_DIR'
# 数据集存放路径
ENV_AG_DATASET_DIR = 'ENV_AG_DATASET_DIR'
# 配置的训练超参json字符串
ENV_AG_USER_PARAMS = 'ENV_AG_USER_PARAMS'
# 训练产物存放路径
ENV_AG_TRAIN_OUTPUT_DIR = 'ENV_AG_TRAIN_OUTPUT_DIR'

_transforms = None


def _multi_class_classification_metrics(pred):
    raw_predictions, labels = pred
```

```
    predictions = np.argmax(raw_predictions, axis=1)
    results = {
        "f1_macro": metrics.f1_score(labels, predictions, average="macro"),
        "f1_micro": metrics.f1_score(labels, predictions, average="micro"),
        "f1_weighted": metrics.f1_score(labels, predictions, average="weighted"),
        "precision_macro": metrics.precision_score(labels, predictions, average="macro"),
        "precision_micro": metrics.precision_score(labels, predictions, average="micro"),
        "precision_weighted": metrics.precision_score(labels, predictions, average="weighted"),
        "recall_macro": metrics.recall_score(labels, predictions, average="macro"),
        "recall_micro": metrics.recall_score(labels, predictions, average="micro"),
        "recall_weighted": metrics.recall_score(labels, predictions, average="weighted"),
        "accuracy": metrics.accuracy_score(labels, predictions),
    }
    return results


def parse_args():
    """ 从AIGallery环境变量中获取用户配置的超参json """
    return json.loads(os.getenv(ENV_AG_USER_PARAMS))


def _process_input_data(image_processor):
    # 加载数据集
    dataset_path = os.getenv(ENV_AG_DATASET_DIR)
    dataset = load_dataset("imagefolder", data_dir=dataset_path)

    # 数据增强
    normalize = Normalize(mean=image_processor.image_mean, std=image_processor.image_std)
    size = (image_processor.size["shortest_edge"] if "shortest_edge" in image_processor.size else (
        image_processor.size["height"], image_processor.size["width"]))
    global _transforms
    _transforms = Compose([RandomResizedCrop(size), RandomHorizontalFlip(), ToTensor(), normalize])
    ret = dataset.with_transform(_format_transforms)
    return ret


# 转换函数
def _format_transforms(examples):
    examples["pixel_values"] = [_transforms(img.convert("RGB")) for img in examples["image"]]
    del examples["image"]
    return examples


def train(user_args):
    print('Start to process dataset')
    model_path = os.getenv(ENV_AG_MODEL_DIR)
    image_processor = AutoImageProcessor.from_pretrained(model_path)

    dataset = _process_input_data(image_processor)
    print(f"Dataset: {dataset}")
    # label和id映射
    classes = dataset["train"].features["label"].names
    label2id = {c: i for i, c in enumerate(classes)}
    id2label = {i: c for i, c in enumerate(classes)}

    print('Start to load model')
    # 加载模型
    model = AutoModelForImageClassification.from_pretrained(
        model_path,
        num_labels=len(classes),
        id2label=id2label,
        label2id=label2id,
        ignore_mismatched_sizes=True
    )

    print('Start to set training args')
    # 训练参数
    training_args = TrainingArguments(
        output_dir=os.getenv(ENV_AG_TRAIN_OUTPUT_DIR),
```



```
        remove_unused_columns=False,
        evaluation_strategy="epoch",
        save_strategy=user_args['save_strategy'],
        learning_rate=float(user_args['lr']),
        save_total_limit=3,
        per_device_train_batch_size=32,
        gradient_accumulation_steps=1,
        per_device_eval_batch_size=int(user_args['per_device_eval_batch_size']),
        num_train_epochs=int(user_args['num_train_epochs']),
        warmup_ratio=float(user_args['warmup_ratio']),
        logging_steps=int(user_args['logging_steps']),
        load_best_model_at_end=True,
        metric_for_best_model="accuracy",
        push_to_hub=False,
    )

    print('Start to train')
    # 训练参数
    trainer = Trainer(
        model=model,
        args=training_args,
        data_collator=DefaultDataCollator(),
        train_dataset=dataset["train"],
        eval_dataset=dataset["test"],
        tokenizer=image_processor,
        compute_metrics=_multi_class_classification_metrics,
    )

    # 开始训练
    train_results = trainer.train()
    print('Start to save model')
    # 保存模型
    trainer.save_model()
    trainer.log_metrics("train", train_results.metrics)
    trainer.save_metrics("train", train_results.metrics)
    trainer.save_state()

    print('Start to evaluate')
    # 在验证集上做准确性评估
    eva_metrics = trainer.evaluate()
    trainer.log_metrics("eval", eva_metrics)
    trainer.save_metrics("eval", eva_metrics)

    print('All Done')


if __name__ == '__main__':
    args = parse_args()
    train(args)
```

“train_params.json”示例

表 1-5 training_methods 参数说明

参数名称	说明
name	自定义的训练方式。
hyperparameters	训练方式包含的超参。具体参数说明请参见**表1-6**。

表 1-6 hyperparameters 参数说明

参数名称	说明
name	超参的名称，只能包含英文、数字、下划线。
type	支持的超参类型，支持float、int、str或bool。
required	超参是否必选，支持true、false。必选不可删除，非必选可删除。
default	超参的默认值，如果无默认值，则填写空双引号。
help	超参的说明，不能超过20个字符。

```
{
  "training_methods": [
    {
      "name": "全参微调",
      "hyperparameters": [
        {
          "name": "lr",
          "type": "float",
          "required": true,
          "default": 0.001,
          "help": "学习率"
        },
        {
          "name": "per_device_eval_batch_size",
          "type": "int",
          "required": false,
          "default": 32,
          "help": "批大小"
        },
        {
          "name": "logging_steps",
          "type": "int",
          "required": false,
          "default": 24,
          "help": "每多少步记录一次步骤"
        },
        {
          "name": "save_strategy",
          "type": "str",
          "required": true,
          "default": "epoch",
          "help": "训练过程中保存checkpoint的策略"
        },
        {
          "name": "num_train_epochs",
          "type": "int",
          "required": true,
          "default": 20,
          "help": "训练的总epochs数"
        },
        {
          "name": "warmup_ratio",
          "type": "float",
          "required": true,
          "default": 0.1,
          "help": "用于指定线性热身占总训练步骤的比例"
        }
      ]
    }
```



```
    ]
}
```

“inference.py”示例

```
from typing import Dict, List, Any
from transformers import pipeline
import os


class EndpointHandler:
    def __init__(self, path=""):
        # Use a pipeline as a high-level helper
        self.pipe = pipeline("question-answering", model=path)

    def __call__(self, data: Dict[str, Any]) -> List[Dict[str, Any]]:
        """
         data args:
              inputs (:obj: `str`)
        Return:
              A :obj:`list` | `dict`: will be serialized and returned
        """
        # get inputs
        inputs = data.pop("inputs", data)
        question = inputs["question"]
        context = inputs["context"]
        resp = self.pipe(question=question, context=context)
        return resp
```

1.2.1.2 自定义镜像规范

AI Gallery支持托管自定义镜像，但是托管的自定义镜像要满足规范才支持使用AI Gallery工具链服务（微调大师、在线推理服务）。

自定义镜像的使用流程

步骤1 托管自定义镜像，操作步骤请参考**托管模型到AI Gallery**。

- 如果自定义镜像要支持训练，则需要满足**自定义镜像规范（训练）**。
- 如果自定义镜像要支持推理，则需要满足**自定义镜像规范（推理）**。

步骤2 上架自定义镜像，操作步骤请参考**发布模型到AI Gallery**。

步骤3 在AI Gallery进行自定义镜像训练或推理。**使用AI Gallery微调大师训练模型**或**使用AI Gallery在线推理服务部署模型**。

- 如果使用自定义镜像进行训练，操作步骤可以参考**使用AI Gallery微调大师训练模型**，其中“训练任务类型”默认选择“自定义”，且不支持修改。
- 如果使用自定义镜像进行部署推理服务，操作步骤可以参考**使用AI Gallery在线推理服务部署模型**，其中“推理任务类型”默认选择“自定义”，且不支持修改。

----结束

自定义镜像规范（训练）

当托管自定义镜像到AI Gallery时，如果镜像要支持AI Gallery的模型微调，则需要在“README.md”文件配置自定义镜像的训练参数，如下示例所示，参数说明请参见**表1-7**。

```
Framework:
  - mindspore
Minimum_hardware_requirement: 1-snt9b-16-cpu-24-96
```

```
Hardware: gpu
Language:
 - en
Train_image_url: swr.<swr-domain-name><namespace><repository>.myhuaweicloud.com
Train_command_path: /xxx/xxx/xxx.py
```

须知

Readme的文件必须按照YAML语法书写才能使配置生效。

表 1-7 自定义镜像的训练参数

参数名称	说明
Train_image_url	必填，训练镜像路径，输入镜像存放的SWR路径地址，例如“swr.<swr-domain-name><namespace><repository>.myhuaweicloud.com”（地址必须是swr开头、myhuaweicloud.com结尾）。仅当配置了该参数，AI Gallery才会使用自定义镜像进行训练，否则使用AI Gallery的预置镜像进行训练。
Train_command_path	必填，训练启动脚本，输入启动脚本地址，例如“/xxx/xxx/main.py”。仅支持shell脚本和python脚本。脚本示例可以参考**train.py示例**。如果是SWR容器内的地址，则填写绝对路径；如果是AI Gallery仓库内的地址，则填写相对路径。

同时，还需要在“模型文件”添加gallery_train文件夹，文件夹内容参考**表1-8**。

表 1-8 gallery_train 文件列表

文件类型	文件说明
“train_params.json”	必选文件，训练参数文件，定义了模型训练的必要参数，例如训练方式、超参信息。该参数会显示在微调工作流的“作业设置”页面的算法配置和超参数设置里面。代码示例请参见**train_params.json示例**。
“dataset_readme.md”	必选文件，数据集要求说明，定义了模型训练时对数据集的要求，会显示在微调工作流的“准备数据”页面。

自定义镜像规范（推理）

当托管自定义镜像到AI Gallery时，如果镜像要支持AI Gallery的推理服务，则需要在“README.md”文件配置自定义镜像的推理参数，如下示例所示，参数说明请参见**表1-9**。

```
Framework:
 - mindspore
Minimum_hardware_requirement: 1-snt9b-16-cpu-24-96
Hardware: gpu
Language:
 - en
```

```
Infer_image_url: swr.<swr-domain-name><namespace><repository>.myhuaweicloud.com
Infer_command_path: /xxx/xxx/xxx.py
Infer_port: 8081
```

须知

Readme的文件必须按照YAML语法书写才能使配置生效。

表 1-9 自定义镜像的推理参数

参数名称	说明
Infer_image_url	必填，推理镜像路径，输入镜像存放的SWR路径地址，例如“swr.<swr-domain-name><namespace><repository>.myhuaweicloud.com”（地址必须是swr开头、myhuaweicloud.com结尾）。仅当配置了该参数，AI Gallery才会使用自定义镜像进行推理，否则使用AI Gallery的预置镜像进行推理。
Infer_command_path	必填，推理启动脚本，输入启动脚本地址，例如“/xxx/xxx/main.py”。仅支持shell脚本和python脚本。如果是SWR容器内的地址，则填写绝对路径；如果是AI Gallery仓库内的地址，则填写相对路径。
Infer_port	选填，推理服务提供的端口，缺省值为8080。只支持部署HTTP服务。

自定义镜像可以通过是否上传自定义推理参数文件“gallery_inference/inference_params.json”决定镜像在部署推理服务时是否支持设置推理参数。

如果在自定义镜像的“模型文件”下上传了“gallery_inference/inference_params.json”文件，则在推理启动脚本中需要使用环境变量来指定“inference_params.json”中的参数，否则配置的参数将无法在推理过程中生效。

“inference_params.json”文件的参数请参见表1-10。该参数会显示在部署推理服务页面，在“高级设置”下会新增“参数设置”，基于配置的推理参数供模型使用者修改自定义镜像的部署参数。

表 1-10 自定义推理参数说明

参数名称	说明
name	参数名称，只能包含英文、数字、下划线。
type	参数类型，可选值：float、int、str、bool、enum。
default	参数默认值，如果是“none”则无默认值，否则需要填写。
help	参数描述，非必选。当用来替换placehoder时，超过20个字符则截断。

参数名称	说明
valid_range	参数属性值范围，当“type”是“enum”时表示的是枚举值，当“type”是“float”或“int”时表示的是属性值范围。

“inference_params.json”的代码示例如下所示。

```
{
   "name": "max_input_length",
   "type": "int",
   "default": 2048,
   "valid_range": [
      0,
      10000
   ]
}
```

对应的推理启动脚本“main.py”如下所示。

```
text-generation-launcher \
--model-id ${model-id} \
--max-input-length ${max_input_length} \
--max-total-tokens ${max-total-tokens} \
--max-batch-prefill-tokens 4096 \
--trust-remote-code \
--sharded false \
--num-shard 1 \
--max-waiting-tokens 1 \
--max-concurrent-requests 1000 \
--waiting-served-ratio 0.2 \
--hostname 0.0.0.0 \
--port 8085
```

“train.py”示例

表 1-11 环境变量说明

变量名称	说明	示例
ENV_AG_MODEL_DIR	模型存放路径，AI Gallery的模型仓库地址，包含模型仓库的所有文件。	“/home/ma-user/.cache/gallery/model/ur12345--gpt2”
ENV_AG_DATASET_DIR	数据集存放路径，AI Gallery的数据集仓库地址，包含数据集仓库的所有文件。	“/home/ma-user/.cache/gallery/dataset/ur12345--data_demo”
ENV_AG_USER_PARAMS	配置的训练超参json字符串。创建训练任务时在算法配置页面设置的超参，用json字符串表示。	{"per_device_eval_batch_size":"32","lr":"0.001","logging_steps":"24"}
ENV_AG_TRAIN_OUTPUT_DIR	训练产物文件存放路径。训练产物将被保存到该路径。训练任务结束后，由AI Gallery平台将该目录上传到新模型的仓库中。	“/home/ma-user/.cache/gallery/output”

变量名称	说明	示例
ENV_AG_USER_METRICS_LOG_PATH	训练数据的日志文件存放路径。训练过程中的迭代次数、LOSS和吞吐数据按照“迭代次数|loss|吞吐”格式记录在日志中，AI Gallery通过环境变量找到日志，从中获取实际数据绘制成“吞吐”和“训练LOSS”曲线，呈现在训练的“指标效果”中。具体请参见**查看训练效果**。 **说明** 日志文件中的迭代次数、LOSS和吞吐数据必须按照“迭代次数|loss|吞吐”格式存放，否则AI Gallery会数据解析失败，导致“吞吐”和“训练LOSS”曲线异常。	“/var/logs/user_metrics.log”

```
import json
import os

from datasets import load_dataset
from transformers import AutoImageProcessor
from torchvision.transforms import RandomResizedCrop, Compose, Normalize, ToTensor,
RandomHorizontalFlip
import numpy as np
from transformers import AutoModelForImageClassification, TrainingArguments, Trainer
from transformers import DefaultDataCollator
from sklearn import metrics

# 环境变量
# 工作目录
ENV_AG_WORK_DIR = 'ENV_AG_WORK_DIR'
# 模型存放路径
ENV_AG_MODEL_DIR = 'ENV_AG_MODEL_DIR'
# 数据集存放路径
ENV_AG_DATASET_DIR = 'ENV_AG_DATASET_DIR'
# 配置的训练超参json字符串
ENV_AG_USER_PARAMS = 'ENV_AG_USER_PARAMS'
# 训练产物存放路径
ENV_AG_TRAIN_OUTPUT_DIR = 'ENV_AG_TRAIN_OUTPUT_DIR'

_transforms = None


def _multi_class_classification_metrics(pred):
    raw_predictions, labels = pred
    predictions = np.argmax(raw_predictions, axis=1)
    results = {
        "f1_macro": metrics.f1_score(labels, predictions, average="macro"),
        "f1_micro": metrics.f1_score(labels, predictions, average="micro"),
        "f1_weighted": metrics.f1_score(labels, predictions, average="weighted"),
        "precision_macro": metrics.precision_score(labels, predictions, average="macro"),
        "precision_micro": metrics.precision_score(labels, predictions, average="micro"),
        "precision_weighted": metrics.precision_score(labels, predictions, average="weighted"),
        "recall_macro": metrics.recall_score(labels, predictions, average="macro"),
        "recall_micro": metrics.recall_score(labels, predictions, average="micro"),
        "recall_weighted": metrics.recall_score(labels, predictions, average="weighted"),
        "accuracy": metrics.accuracy_score(labels, predictions),
    }
    return results
```

```
def parse_args():
    """ 从AIGallery环境变量中获取用户配置的超参json """
    return json.loads(os.getenv(ENV_AG_USER_PARAMS))


def _process_input_data(image_processor):
    # 加载数据集
    dataset_path = os.getenv(ENV_AG_DATASET_DIR)
    dataset = load_dataset("imagefolder", data_dir=dataset_path)

    # 数据增强
    normalize = Normalize(mean=image_processor.image_mean, std=image_processor.image_std)
    size = (image_processor.size["shortest_edge"] if "shortest_edge" in image_processor.size else (
        image_processor.size["height"], image_processor.size["width"]))
    global _transforms
    _transforms = Compose([RandomResizedCrop(size), RandomHorizontalFlip(), ToTensor(), normalize])
    ret = dataset.with_transform(_format_transforms)
    return ret


# 转换函数
def _format_transforms(examples):
    examples["pixel_values"] = [_transforms(img.convert("RGB")) for img in examples["image"]]
    del examples["image"]
    return examples


def train(user_args):
    print('Start to process dataset')
    model_path = os.getenv(ENV_AG_MODEL_DIR)
    image_processor = AutoImageProcessor.from_pretrained(model_path)

    dataset = _process_input_data(image_processor)
    print(f"Dataset: {dataset}")
    # label和id映射
    classes = dataset["train"].features["label"].names
    label2id = {c: i for i, c in enumerate(classes)}
    id2label = {i: c for i, c in enumerate(classes)}

    print('Start to load model')
    # 加载模型
    model = AutoModelForImageClassification.from_pretrained(
        model_path,
        num_labels=len(classes),
        id2label=id2label,
        label2id=label2id,
        ignore_mismatched_sizes=True
    )

    print('Start to set training args')
    # 训练参数
    training_args = TrainingArguments(
        output_dir=os.getenv(ENV_AG_TRAIN_OUTPUT_DIR),
        remove_unused_columns=False,
        evaluation_strategy="epoch",
        save_strategy=user_args['save_strategy'],
        learning_rate=float(user_args['lr']),
        save_total_limit=3,
        per_device_train_batch_size=32,
        gradient_accumulation_steps=1,
        per_device_eval_batch_size=int(user_args['per_device_eval_batch_size']),
        num_train_epochs=int(user_args['num_train_epochs']),
        warmup_ratio=float(user_args['warmup_ratio']),
        logging_steps=int(user_args['logging_steps']),
        load_best_model_at_end=True,
        metric_for_best_model="accuracy",
        push_to_hub=False,
    )
```



```
    print('Start to train')
    # 训练参数
    trainer = Trainer(
        model=model,
        args=training_args,
        data_collator=DefaultDataCollator(),
        train_dataset=dataset["train"],
        eval_dataset=dataset["test"],
        tokenizer=image_processor,
        compute_metrics=_multi_class_classification_metrics,
    )

    # 开始训练
    train_results = trainer.train()
    print('Start to save model')
    # 保存模型
    trainer.save_model()
    trainer.log_metrics("train", train_results.metrics)
    trainer.save_metrics("train", train_results.metrics)
    trainer.save_state()

    print('Start to evaluate')
    # 在验证集上做准确性评估
    eva_metrics = trainer.evaluate()
    trainer.log_metrics("eval", eva_metrics)
    trainer.save_metrics("eval", eva_metrics)

    print('All Done')


if __name__ == '__main__':
    args = parse_args()
    train(args)
```

“train_params.json”示例

表 1-12 training_methods 参数说明

参数名称	说明
name	自定义的训练方式。
hyperparameters	训练方式包含的超参。具体参数说明请参见**表1-13**。

表 1-13 hyperparameters 参数说明

参数名称	说明
name	超参的名称，只能包含英文、数字、下划线。
type	支持的超参类型，支持float、int、str或bool。
required	超参是否必选，支持true、false。必选不可删除，非必选可删除。
default	超参的默认值，如果无默认值，则填写空双引号。
help	超参的说明，不能超过20个字符。


```
{
   "training_methods": [
      {
         "name": "全参微调",
         "hyperparameters": [
            {
               "name": "lr",
               "type": "float",
               "required": true,
               "default": 0.001,
               "help": "学习率"
            },
            {
               "name": "per_device_eval_batch_size",
               "type": "int",
               "required": false,
               "default": 32,
               "help": "批大小"
            },
            {
               "name": "logging_steps",
               "type": "int",
               "required": false,
               "default": 24,
               "help": "每多少步记录一次步骤"
            },
            {
               "name": "save_strategy",
               "type": "str",
               "required": true,
               "default": "epoch",
               "help": "训练过程中保存checkpoint的策略"
            },
            {
               "name": "num_train_epochs",
               "type": "int",
               "required": true,
               "default": 20,
               "help": "训练的总epochs数"
            },
            {
               "name": "warmup_ratio",
               "type": "float",
               "required": true,
               "default": 0.1,
               "help": "用于指定线性热身占总训练步骤的比例"
            }
         ]
      }
   ]
}
```

1.2.1.3 使用 AI Gallery SDK 构建自定义模型

AI Gallery的Transformers库支持部分开源的模型结构框架，并对昇腾系列显卡进行了训练/推理性能优化，可以做到开箱即用。如果你有自己从头进行预训练的模型，AI Gallery也支持使用SDK构建自定义模型接入AI Gallery。

Transformers 库介绍

AI Gallery使用的Transformers机器学习库是一个开源的基于Transformer模型结构提供的预训练语言库。Transformers库注重易用性，屏蔽了大量AI模型开发使用过程中的技术细节，并制定了统一合理的规范。使用者可以便捷地使用、下载模型。同时支持用户上传自己的预训练模型到在线模型资产仓库中，并发布上架给其他用户使用。AI Gallery在原有Transformers库的基础上，融入了对于昇腾硬件的适配与支持。对AI

有使用诉求的企业、NLP领域开发者，可以借助这个库，便捷地使用昇腾算力进行自然语言理解（NLU）和自然语言生成（NLG）任务的SOTA模型开发与应用。

支持的模型结构框架

AI Gallery的Transformers库支持的开源模型结构框架如表1-14所示。

表 1-14 支持的模型结构框架

模型结构	PyTorch	MindSpore	GPU	昇腾
Llama	支持	不支持	支持	支持
Bloom	支持	不支持	支持	不支持
Falcon	支持	不支持	支持	不支持
BERT	支持	不支持	支持	不支持
MPT	支持	不支持	支持	不支持
ChatGLM	支持	不支持	支持	支持

核心基础类介绍

使用AI Gallery SDK构建自定义模型，需要了解2个核心基础类“PretrainedModel”和“PretrainedConfig”之间的交互。

- “PretrainedConfig”：预训练模型的配置基类

  提供模型配置的通用属性和两个主要方法，用于序列化和反序列化配置文件。

  ```
  PretrainedConfig.from_pretrained(dir) # 从目录中加载序列化对象（本地或者是url），配置文件为dir/config.json
  PretrainedConfig.save_pretrained(dir) # 将配置实例序列化到dir/config.json
  ```

- “PretrainedModel”：预训练模型的基类

  包含一个配置实例“config”，提供两个主要方法，用来加载和保存预训练模型。

  ```
  # 1. 调用 init_weights() 来初始化所有模型权重
  # 2. 从目录中（本地或者是url）中导入序列化的模型
  # 3. 使用导入的模型权重覆盖所有初始化的权重
  # 4. 调用 PretrainedConfig.from_pretrained(dir)来将配置设置到self.config中
  PretrainedModel.from_pretrained(dir)

  # 将模型实例序列化到 dir/pytorch_model.bin 中
  PretrainedModel.save_pretrained(dir)

  # 给定input_ids，生成 output_ids，在循环中调用 PretrainedModel.forward() 来做前向推理
  PretrainedModel.generate()
  ```

操作步骤

本文使用**NewBert**模型介绍构建自定义模型的流程。

步骤1 安装AI Gallery SDK。

通过pip在本地或云上开发环境安装AI Gallery SDK（galleryformers）。

```
pip install galleryformers
```

说明

建议在虚拟环境（Python 3.8+）中安装AI Gallery SDK，以便管理不同的项目，避免依赖项之间产生兼容性问题。

步骤2 构建自定义模型。

1. 编写自定义配置类。

   模型的configuration包含了构建模型所需的所有信息的对象，需要尽可能完整。

```
from galleryformers import PretrainedConfig
from typing import List

class NewBertConfig(PretrainedConfig):
    model_type = "bert"

    def __init__(
        self,
        vocab_size=30522,
        hidden_size=768,
        num_hidden_layers=12,
        num_attention_heads=12,
        intermediate_size=3072,
        hidden_act="gelu",
        hidden_dropout_prob=0.1,
        attention_probs_dropout_prob=0.1,
        max_position_embeddings=512,
        type_vocab_size=2,
        initializer_range=0.02,
        layer_norm_eps=1e-12,
        pad_token_id=0,
        position_embedding_type="absolute",
        use_cache=True,
        classifier_dropout=None,
        **kwargs,
    ):
        super().__init__(pad_token_id=pad_token_id, **kwargs)

        self.vocab_size = vocab_size
        self.hidden_size = hidden_size
        self.num_hidden_layers = num_hidden_layers
        self.num_attention_heads = num_attention_heads
        self.hidden_act = hidden_act
        self.intermediate_size = intermediate_size
        self.hidden_dropout_prob = hidden_dropout_prob
        self.attention_probs_dropout_prob = attention_probs_dropout_prob
        self.max_position_embeddings = max_position_embeddings
        self.type_vocab_size = type_vocab_size
        self.initializer_range = initializer_range
        self.layer_norm_eps = layer_norm_eps
        self.position_embedding_type = position_embedding_type
        self.use_cache = use_cache
        self.classifier_dropout = classifier_dropout
```

   - 自定义配置类必须继承自“PretrainedConfig”。
   - 自定义配置类的“__init__”必须接受任何“kwargs”，这些“kwargs”需要传递给“__init__”。

2. 完成自定义配置类的编写后，可以使用该类创建配置实例。

```
newbert1_config = NewBertConfig(num_hidden_layers=6, num_attention_heads=10, use_cache=False)
newbert1_config.save_pretrained("mynewbert")
```

   这一步会在本地名为mynewbert的文件夹中保存一个名为config.json的文件。

   该配置实例同样可以通过调用from_pretrained方法加载。

```
newbert1_config.from_pretrained("mynewbert")
```


3. 编写完配置部分，开始编写自定义模型。

   下面展示了3种模型基类的代码示例，为了确保示例不过于复杂，本文对部分代码片段进行了省略展示。

   - 预训练模型基类NewBertPreTrainedModel

```
from galleryformers import PreTrainedModel
from .configuration_newbert import NewBertConfig


class NewBertPreTrainedModel(PreTrainedModel):
    config_class = NewBertConfig
    load_tf_weights = load_tf_weights_in_bert
    base_model_prefix = "bert"
    supports_gradient_checkpointing = True

    def _init_weights(self, module):
        """Initialize the weights"""
        if isinstance(module, nn.Linear):
            module.weight.data.normal_(mean=0.0, std=self.config.initializer_range)
            if module.bias is not None:
                module.bias.data.zero_()
        elif isinstance(module, nn.Embedding):
            module.weight.data.normal_(mean=0.0, std=self.config.initializer_range)
            if module.padding_idx is not None:
                module.weight.data[module.padding_idx].zero_()
        elif isinstance(module, nn.LayerNorm):
            module.bias.data.zero_()
            module.weight.data.fill_(1.0)
```

   - 基础模型类NewBertModel：该类继承自NewBertPreTrainedModel。

```
class NewBertModel(NewBertPreTrainedModel):

    def __init__(self, config, add_pooling_layer=True):
        super().__init__(config)
        self.config = config

        self.embeddings = BertEmbeddings(config)
        self.encoder = BertEncoder(config)

        self.pooler = BertPooler(config) if add_pooling_layer else None

        # Initialize weights and apply final processing
        self.post_init()

    def get_input_embeddings(self):
        return self.embeddings.word_embeddings

    def set_input_embeddings(self, value):
        self.embeddings.word_embeddings = value

    def _prune_heads(self, heads_to_prune):
        for layer, heads in heads_to_prune.items():
            self.encoder.layer[layer].attention.prune_heads(heads)


    def forward(
        self,
        input_ids: Optional[torch.Tensor] = None,
        attention_mask: Optional[torch.Tensor] = None,
        token_type_ids: Optional[torch.Tensor] = None,
        position_ids: Optional[torch.Tensor] = None,
        head_mask: Optional[torch.Tensor] = None,
        inputs_embeds: Optional[torch.Tensor] = None,
        encoder_hidden_states: Optional[torch.Tensor] = None,
        encoder_attention_mask: Optional[torch.Tensor] = None,
        past_key_values: Optional[List[torch.FloatTensor]] = None,
        use_cache: Optional[bool] = None,
        output_attentions: Optional[bool] = None,
        output_hidden_states: Optional[bool] = None,
```



```
        return_dict: Optional[bool] = None,
        ...)
```

所有的模型都需要通过“forward”方法来实现自己的推理逻辑，这个方法会在执行“model(input_ids)”的时候进行调用

- 模型基类NewBertForXXX：该类承自NewBertPreTrainedModel。

  该类可用于执行AI Gallery工具链服务，此处以文本问答(Question Answering)的任务类型为例：

```
class NewBertForQuestionAnswering(NewBertPreTrainedModel):
    def __init__(self, config):
        super().__init__(config)
        self.num_labels = config.num_labels

        self.bert = BertModel(config, add_pooling_layer=False)
        self.qa_outputs = nn.Linear(config.hidden_size, config.num_labels)

        # Initialize weights and apply final processing
        self.post_init()

    def forward(
        self,
        input_ids: Optional[torch.Tensor] = None,
        attention_mask: Optional[torch.Tensor] = None,
        token_type_ids: Optional[torch.Tensor] = None,
        position_ids: Optional[torch.Tensor] = None,
        head_mask: Optional[torch.Tensor] = None,
        inputs_embeds: Optional[torch.Tensor] = None,
        start_positions: Optional[torch.Tensor] = None,
        end_positions: Optional[torch.Tensor] = None,
        output_attentions: Optional[bool] = None,
        output_hidden_states: Optional[bool] = None,
        return_dict: Optional[bool] = None,
    )
        return_dict = return_dict if return_dict is not None else self.config.use_return_dict

        outputs = self.bert(
            input_ids,
            attention_mask=attention_mask,
            token_type_ids=token_type_ids,
            position_ids=position_ids,
            head_mask=head_mask,
            inputs_embeds=inputs_embeds,
            output_attentions=output_attentions,
            output_hidden_states=output_hidden_states,
            return_dict=return_dict,
        )

        sequence_output = outputs[0]

        logits = self.qa_outputs(sequence_output)
        start_logits, end_logits = logits.split(1, dim=-1)
        start_logits = start_logits.squeeze(-1).contiguous()
        end_logits = end_logits.squeeze(-1).contiguous()

        total_loss = None
        if start_positions is not None and end_positions is not None:
            # If we are on multi-GPU, split add a dimension
            if len(start_positions.size()) > 1:
                start_positions = start_positions.squeeze(-1)
            if len(end_positions.size()) > 1:
                end_positions = end_positions.squeeze(-1)
            # sometimes the start/end positions are outside our model inputs, we ignore these
terms
            ignored_index = start_logits.size(1)
            start_positions = start_positions.clamp(0, ignored_index)
            end_positions = end_positions.clamp(0, ignored_index)

            loss_fct = CrossEntropyLoss(ignore_index=ignored_index)
```



```
            start_loss = loss_fct(start_logits, start_positions)
            end_loss = loss_fct(end_logits, end_positions)
            total_loss = (start_loss + end_loss) / 2

        if not return_dict:
            output = (start_logits, end_logits) + outputs[2:]
            return ((total_loss,) + output) if total_loss is not None else output

        return QuestionAnsweringModelOutput(
            loss=total_loss,
            start_logits=start_logits,
            end_logits=end_logits,
            hidden_states=outputs.hidden_states,
            attentions=outputs.attentions,
        )
```

这个多头模型的“forward”函数会先调用“self.bert.forward()”，然后再调用“self.masked_lm_head.__call__()”方法来生成最终的结果。

4. 完成了自定义模型类的编写后，可以使用该类创建一个模型实例：

```
newbert = NewBertForQuestionAnswering(newbert1_config)
```

模型权重可以通过调用“.from_pretrained()”加载：

```
newbert.from_pretrained(pretrained_model_name_or_path="./您的权重文件本地存储路径/.")
```

----结束

后续操作

自定义模型文件构建完成后，可以参考**托管模型到AI Gallery**将模型文件托管至AI Gallery。建议托管的模型文件列表参见**表1-15**。

表 1-15 模型实例包含的文件

文件名称	描述
config.json	模型配置文件。
model.safetensors或pytorch_model.bin	预训练模型的权重文件。
tokenizer.json	（可选）预处理器的词表文件，用于初始化Tokenizer。
tokenizer_config.json	（可选）预处理器的配置文件。
modeling_xxx.py	（可选）自定义模型的代码文件，继承自PretrainedModel，包含实现自定义推理逻辑的代码。
configuration_xxx.py	（可选）自定义配置的代码文件，继承自PretrainedConfig，包含实现自定义配置的逻辑代码。

1.2.2 托管模型到 AI Gallery

AI Gallery上每个资产的文件都会存储在线上的AI Gallery存储库（简称AI Gallery仓库）里面。每一个模型实例视作一个资产仓库，模型实例与资产仓库之间是一一对应的关系。例如，模型名称为“Test”，则AI Gallery仓库有个名为“Test”的仓库，其中只存放Test模型实例的全部文件。

功能说明

- 支持本地文件托管至AI Gallery仓库且支持多个文件同时上传。
  单个仓库的容量上限为50GB。
- 支持管理托管的资产文件，例如在线预览、下载、删除文件。
  只支持预览大小不超过10MB、格式为文本类或图片类的文件。
- 支持编辑资产介绍。每个资产介绍可分为基础设置和使用描述。
  - 基础设置部分包含了该资产所有重要的结构化元数据信息。选择填入的信息将会变成该模型资产的标签，并且自动同步在模型描述部分，保存到“README.md”文件里。
  - 模型描述部分是一个可在线编辑、预览的Markdown文件，里面包含该模型的简介、能力描述、训练情况、引用等信息。编辑内容会自动保存在“README.md”文件里。

  更新后的“README.md”文件自动存放在数据集详情页的“文件版本”页签或者是模型详情页的“模型文件”页签。

创建模型资产

1. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
2. 单击左上方“创建资产”，选择“模型”。
3. 在“创建模型”弹窗中配置参数，单击“创建”。

表 1-16 创建模型

参数名称	说明
英文名称	必填项，模型的英文名称。 如果没有填写“中文名称”，则资产发布后，在模型页签上会显示该“英文名称”。
中文名称	模型的中文名称。 如果填写了“中文名称”，则资产发布后，在模型页签上会显示该“中文名称”。
许可证	模型资产遵循的使用协议，根据业务需求选择合适的许可证类型。
描述	填写资产简介，模型发布后将作为副标题显示在模型页签上，方便用户快速了解资产。 支持0~90个字符，请勿在描述中输入涉政、迷信、违禁等相关敏感词，否则发布审核无法通过。

创建完成后，跳转至模型详情页。

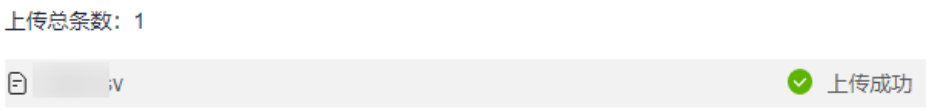
上传模型文件

1. 在模型详情页，选择“模型文件”页签。
2. 单击“添加文件”，进入上传文件页面，选择本地的数据文件单击“点击上传”或拖动文件，单击“确认上传”启动上传。

- 上传单个超过5GB的文件时，请使用Gallery CLI工具。CLI工具的获取和使用请参见**Gallery CLI配置工具指南**。
- 文件合集大小不超过50GB。
- 文件上传完成前，请不要刷新或关闭上传页面，防止意外终止上传任务，导致数据缺失。
- 当模型的“任务类型”是除“文本问答”和“文本生成”之外的类型（即自定义模型）时，上传的模型文件要满足**自定义模型规范**，否则该模型无法正常使用AI Gallery工具链服务（微调大师和在线推理服务）。
- 当托管的是自定义镜像时，上传的模型文件要满足**自定义镜像规范**，否则该镜像无法正常使用AI Gallery工具链服务（微调大师和在线推理服务）。

3. 当文件状态变成“上传成功”表示数据文件成功上传至AI Gallery仓库进行托管。单击“完成”返回模型文件页面。

图 1-2 上传成功



说明

文件上传过程中请耐心等待，不要关闭当前上传页面，关闭页面会中断上传进程。

1.2.3 发布模型到 AI Gallery

除了Gallery提供的已有资产外，还可以将个人创建的资产发布至Gallery货架上，供其他AI开发者使用，实现资产共享。

模型资产上架

1. 登录AI Gallery，选择右上角“我的Gallery”。
2. 在左侧“我的资产 > 模型”下，选择未发布的模型，单击模型名称，进入模型详情页。
3. 在模型详情页，单击右侧“发布”，在发布模型页面编辑发布信息后，单击“发布”。

表 1-17 发布模型的参数说明

参数名称	说明
中文名称	模型发布后显示的名称，在创建模型时设置的名称，此处不可编辑。
任务类型	选择合适的任务类型。
许可证	必填项，根据业务需求选择合适的许可证类型。

参数名称	说明
描述	必填项，填写资产简介，模型发布后将显示在模型页签上，方便用户快速了解资产。 支持1~90个字符，请勿在描述中输入涉政、迷信、违禁等相关敏感词，否则发布审核无法通过。
可见范围	● “所有用户可见”：表示公开资产，所有用户都可以查看该资产。 ● “指定用户可见”：输入账号名、账号ID或用户昵称搜索并选择用户，使其可见该资产。
可用范围	选择是否启用“申请用户可用”。 ● 勾选启用：当用户要使用该模型时需要提交申请，只有模型所有者同意申请后，才能使用或复制模型。 ● 不勾选不启用（默认值）：所有可见资产的用户都可以直接使用模型。

4. 发布后，资产会处于“审核中”，审核中的资产仅资产所有者可见。
   审核完成后，资产会变成“已发布”状态，并在模型列表可见。

1.2.4 管理 AI Gallery 模型

编辑模型介绍

说明

资产发布上架后，准确、完整的资产介绍有助于提升资产的排序位置和访问量，能更好的支撑用户使用该资产。

1. 在模型详情页，选择“模型介绍”页签，单击右侧“编辑介绍”。
2. 编辑模型基础设置和模型描述。

表 1-18 模型介绍的参数说明

参数名称		说明
基础设置	中文名称	显示模型的名称，不可编辑。
	许可证	模型遵循的使用许可协议，根据业务需求选择合适的许可证类型。
	语言	选择使用模型时支持的输入输出语言。
	框架	选择构建模型使用的AI开发框架。

参数名称		说明
	任务类型	选择模型支持的任务类型，不同任务类型支持的AI Gallery工具链服务请参见**表1-19**。 • 文本问答：从给定文本中检索问题的答案，适用于从文档中搜索答案的场景。 • 文本生成：基于给定文本进行续写，生成新的文本。 • 其他类型：基于实际场景选择合适的任务类型。 **说明** 如果模型的“任务类型”是除“文本问答”和“文本生成”之外的类型，则被定义为自定义模型。自定义模型必须要满足**自定义模型规范**，才支持使用AI Gallery工具链服务。
	硬件资源	选择支持运行该模型的硬件类型。
	最低可运行规格	设置能够运行该模型的最低计算规格。在AI Gallery工具链服务中使用该模型时，只能选取等同或高于该规格的算力资源进行任务下发。
	是否支持分布式训练/推理	选择该模型资产是否支持在单机多卡的资源节点上进行并行训练或推理。
README.md	-	资产的README内容，支持添加资产的简介、使用场景、使用方法等信息。 当托管的是自定义镜像时，填写的内容要满足**自定义镜像规范**，否则该镜像无法正常使用AI Gallery工具链服务（微调大师和在线推理服务）。 **说明** 建议写清楚模型的使用方法，方便使用者更好的完成训练、推理任务。

表 1-19 任务类型支持的 AI Gallery 工具链服务

任务类型	微调大师	在线推理服务	AI应用
文本问答/文本生成	支持	支持	支持
其他类型	支持	支持	不支持

3. 编辑完成后，单击“确认”保存修改。

管理模型文件

- **预览文件**

  在模型详情页，选择“模型文件”页签。单击文件名称即可在线预览文件内容。

  说明

  仅支持预览大小不超过10MB、格式为文本类或图片类的文件。

- **下载文件**

  在模型详情页，选择“模型文件”页签。单击操作列的“下载”，即可下载文件到本地。

- **删除文件**

  在模型详情页，选择“模型文件”页签。单击操作列的“删除”，确认后即可将已经托管的文件从AI Gallery仓库中删除。

  说明

  文件删除后不可恢复，请谨慎操作。

管理模型可见范围

模型发布后，支持修改可见范围。

- “所有用户可见”：表示公开资产，所有用户都可以查看该资产。
- “指定用户可见”：输入账号名、账号ID或用户昵称搜索并选择用户，使其可见该资产。

管理模型可用范围

仅当发布模型时，“可用范围”启用“申请用户可用”时，才支持管理模型的可用范围。管理操作包含如何添加可使用资产的新用户、如何审批用户申请使用资产的请求。

- 添加可使用资产的新用户。

  模型发布成功后，如果模型所有者要新增可使用资产的新用户，则可以在模型详情页添加新用户。

  a. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
  b. 选择“我的资产 > 模型”，在“我创建的模型”页面找到待修改的“已发布”状态的模型，单击模型页签进入详情页。
  c. 在模型详情页，选择“设置”。
  d. 在“可用申请”处输入账号名、账号ID或用户昵称搜索并选择新用户，单击“添加新用户”完成用户添加。

     单击“查看使用用户”会跳转到“申请管理 > 资产申请审核”页面，可以查看当前支持使用该模型的用户列表。

- 管理用户可用资产的权限。

  模型发布成功后，模型所有者可以管理资产的用户申请 。

  a. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
  b. 选择“我的资产 > 模型”，在“我创建的模型”页面找到待修改的“已发布”状态的模型，单击模型页签进入详情页。

c. 在模型详情页，选择“设置”。

d. 在“可用申请”单击“查看使用用户”跳转到“申请管理 > 资产申请审核”页面，在页面进行用户权限处理。

- 撤销审批：单击用户操作列的“撤销”可以取消已审批通过或已拒绝的用户权限，用户的“审批状态”从“已审批”变成“未审批”，或者从“已拒绝”变成“未审批”。
- 同意用户使用该资产：单击用户操作列的“同意”可以通过用户的申请，用户的“审批状态”从“未审批”变成“已审批”。
- 拒绝用户使用该资产：单击用户操作列的“拒绝”并填写拒绝理由，单击确定可以拒绝用户的申请，用户的“审批状态”从“未审批”变成“已拒绝”。

下架模型

AI Gallery中已上架的资产支持下架操作。

1. 在AI Gallery首页，选择右上角“我的Gallery”。
2. 在“我的资产”下，查看已上架的资产。
3. 单击资产名称，进入资产详情页。
4. 在资产详情页，单击“下架”，在弹窗中单击“确定”。即可将资产下架。

删除模型

当资产不使用时，支持删除，释放AI Gallery仓库的存储空间。

1. 在资产详情页，选择“设置”页签。
2. 在“删除资产”处，单击“删除”按钮，确认后资产将被删除。

须知

- 删除操作不可撤销，执行此操作后该资产及相关文件将被永久删除，请谨慎操作。
- 已被订阅的资产不可用删除。

1.3 发布和管理 AI Gallery 数据集

1.3.1 托管数据集到 AI Gallery

AI Gallery上每个资产的文件都会存储在线上的AI Gallery存储库（简称AI Gallery仓库）里面。每一个数据集实例视作一个资产仓库，数据集实例与资产仓库之间是一一对应的关系。例如，模型名称为“Test”，则AI Gallery仓库有个名为“Test”的仓库，其中只存放Test模型实例的全部文件。

功能说明

- 支持本地文件托管至AI Gallery仓库且支持多个文件同时上传。

单个仓库的容量上限为50GB。

- 支持管理托管的资产文件，例如在线预览、下载、删除文件。

  只支持预览大小不超过10MB、格式为文本类或图片类的文件。
- 支持编辑资产介绍。每个资产介绍可分为基础设置和使用描述。
  - 基础设置部分包含了该资产所有重要的结构化元数据信息。选择填入的信息将会变成该模型资产的标签，并且自动同步在模型描述部分，保存到“README.md”文件里。
  - 模型描述部分是一个可在线编辑、预览的Markdown文件，里面包含该模型的简介、能力描述、训练情况、引用等信息。编辑内容会自动保存在“README.md”文件里。

  更新后的“README.md”文件自动存放在数据集详情页的“文件版本”页签或者是模型详情页的“模型文件”页签。

创建数据集资产

1. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
2. 单击左上方“创建资产”，选择“数据集”。
3. 在“创建数据集”弹窗中配置参数，单击“创建”。

表 1-20 创建数据集

参数名称	说明
英文名称	必填项，数据集的英文名称。 如果没有填写“中文名称”，则资产发布后，在数据集页签上会显示该“英文名称”。
中文名称	数据集的中文名称。 如果填写了“中文名称”，则资产发布后，在数据集页签上会显示该“中文名称”。
许可证	数据集资产遵循的使用协议，根据业务需求选择合适的许可证类型。
描述	填写资产简介，数据集发布后将作为副标题显示在数据集页签上，方便用户快速了解资产。 支持0~90个字符，请勿在描述中输入涉政、迷信、违禁等相关敏感词，否则发布审核无法通过。

创建完成后，跳转至数据集详情页。

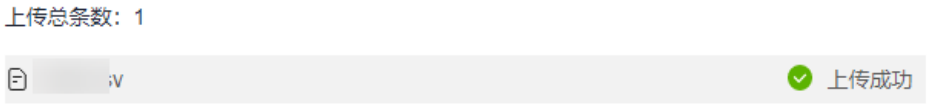
上传数据集文件

1. 在数据集详情页，选择“数据集文件”页签。
2. 单击“添加文件”，进入上传文件页面，选择本地的数据文件单击“点击上传”或拖动文件，单击“确认上传”启动上传。
   - 上传单个超过5GB的文件时，请使用Gallery CLI工具。CLI工具的获取和使用请参见**Gallery CLI配置工具指南**。

- 文件合集大小不超过50GB。
- 文件上传完成前，请不要刷新或关闭上传页面，防止意外终止上传任务，导致数据缺失。

3. 当文件状态变成“上传成功”表示数据文件成功上传至AI Gallery仓库进行托管。单击“完成”返回数据集文件页面。

图 1-3 上传成功



说明

文件上传过程中请耐心等待，不要关闭当前上传页面，关闭页面会中断上传进程。

1.3.2 发布数据集到 AI Gallery

除了Gallery提供的已有资产外，还可以将个人创建的资产发布至Gallery货架上，供其他AI开发者使用，实现资产共享。

数据集资产上架

1. 登录AI Gallery，选择右上角“我的Gallery”。
2. 在“我的资产 > 数据集”下，选择未发布的数据集，单击数据集名称，进入数据集详情页。
3. 在数据集详情页，单击右侧“发布”，在发布数据集页面编辑发布信息后，单击“发布”。

表 1-21 发布数据集的参数说明

参数名称	说明
中文名称	数据集发布后显示的名称，在创建数据集时设置的名称，此处不可编辑。
任务类型	选择合适的任务类型。
许可证	必填项，根据业务需求选择合适的许可证类型。
描述	必填项，填写资产简介，数据集发布后将显示在数据集页签上，方便用户快速了解资产。 支持1~90个字符，请勿在描述中输入涉政、迷信、违禁等相关敏感词，否则发布审核无法通过。
可见范围	• “所有用户可见”：表示公开资产，所有用户都可以查看该资产。 • “指定用户可见”：输入账号名、账号ID或用户昵称搜索并选择用户，使其可见该资产。

参数名称	说明
可用范围	选择是否启用“申请用户可用”。 ● 勾选启用：当用户要使用该数据集时需要提交申请，只有数据集所有者同意申请后，才能使用数据集。 ● 不勾选不启用（默认值）：所有可见资产的用户都可以直接使用数据集。

4. 发布后，资产会处于“审核中”，审核中的资产仅资产所有者可见。
   审核完成后，资产会变成“已发布”状态，并在数据集列表可见。

1.3.3 管理 AI Gallery 数据集

编辑数据集介绍

资产发布上架后，准确、完整的资产介绍有助于提升资产的排序位置和访问量，能更好的支撑用户使用该资产。

1. 在数据集详情页，选择“数据集介绍”页签，单击右侧“编辑介绍”。
2. 编辑数据集基础设置和数据集描述。

表 1-22 数据集介绍的参数说明

参数名称		说明
基础设置	中文名称	显示数据集的名称，不可编辑。
	许可证	数据集遵循的使用许可协议，根据业务需求选择合适的许可证类型。
	语言	选择使用数据集时支持的输入输出语言。
	任务类型	选择数据集支持用于什么类型的训练模型。
	运行平台	选择数据集额外支持的运行平台。 ● 设置运行平台后，当资产上架后，该资产支持通过订阅的方式同步到所选运行平台使用。 ● 设置运行平台后，单击“设置”，在弹窗中可以自定义设置运行平台的资产标签，且标签可以被一起同步至运行平台。
数据集描述	-	资产的README内容，支持添加资产的简介、使用场景、使用方法等信息。

3. 编辑完成后，单击“确认”保存修改。

管理数据集文件

- **预览文件**

  在数据集详情页，选择“数据集文件”页签。单击文件名称即可在线预览文件内容。

  说明

  仅支持预览大小不超过10MB、格式为文本类或图片类的文件。

- **下载文件**

  在数据集详情页，选择“数据集文件”页签。单击操作列的“下载”，选择保存路径单击“确认”，即可下载文件到本地。

- **删除文件**

  在数据集详情页，选择“数据集文件”页签。单击操作列的“删除”，确认后即可将已经托管的文件从AI Gallery仓库中删除。

  说明

  文件删除后不可恢复，请谨慎操作。

管理数据集可用范围

仅当发布数据集时，“可用范围”启用“申请用户可用”时，才支持管理数据集的可用范围。管理操作包含如何添加可使用资产的新用户、如何审批用户申请使用资产的请求。

- 添加可使用资产的新用户。

  数据集发布成功后，如果数据集所有者要新增可使用资产的新用户，则可以在数据集详情页添加新用户。

  a. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
  b. 选择“我的资产 > 数据集”，在“我创建的数据集”页面找到待修改的数据集，单击数据集页签进入详情页。
  c. 在数据集详情页，选择“设置”。
  d. 在“可用申请”处输入账号名、账号ID或用户昵称搜索并选择新用户，单击“添加新用户”完成用户添加。

     单击“查看使用用户”会跳转到“申请管理 > 资产申请审核”页面，可以查看当前支持使用该数据集的用户列表。

- 管理用户可用资产的权限。

  数据集发布成功后，数据集所有者可以管理资产的用户申请 。

  a. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
  b. 选择“我的资产 > 数据集”，在“我创建的数据集”页面找到待修改的数据集，单击数据集页签进入详情页。
  c. 在数据集详情页，选择“设置”。
  d. 在“可用申请”单击“查看使用用户”跳转到“申请管理 > 资产申请审核”页面，在页面进行用户权限处理。

     - 撤销审批：单击用户操作列的“撤销”可以取消已审批通过或已拒绝的用户权限，用户的“审批状态”从“已审批”变成“未审批”，或者从“已拒绝”变成“未审批”。

- 同意用户使用该资产：单击用户操作列的“同意”可以通过用户的申请，用户的“审批状态”从“未审批”变成“已审批”。
- 拒绝用户使用该资产：单击用户操作列的“拒绝”并填写拒绝理由，单击确定可以拒绝用户的申请，用户的“审批状态”从“未审批”变成“已拒绝”。

下架数据集

AI Gallery中已上架的资产支持下架操作。

1. 在AI Gallery首页，选择右上角“我的Gallery”。
2. 在“我的资产”下，查看已上架的资产。
3. 单击资产名称，进入资产详情页。
4. 在资产详情页，单击“下架”，在弹窗中单击“确定”。即可将资产下架。

删除数据集

当资产不使用时，支持删除，释放AI Gallery仓库的存储空间。

1. 在资产详情页，选择“设置”页签。
2. 在“删除资产”处，单击“删除”按钮，确认后资产将被删除。

须知

删除操作不可撤销，执行此操作后该资产及相关文件将被永久删除，请谨慎操作。

1.4 发布和管理 AI Gallery 项目

在AI Gallery中，您可以将个人开发的Notebook代码免费分享给他人使用。

前提条件

在ModelArts的Notebook或者CodeLab中已创建好ipynb文件，开发指导可参见**开发工具**。

发布 Notebook

1. 登录ModelArts管理控制台，选择“开发环境 > Notebook”。
2. 打开“运行中”的Notebook实例进入JupyterLab页面，在待分享的ipynb文件右侧，单击“创建分享”按钮，弹出“发布AI Gallery Notebook”页面。

图 1-4 单击“创建分享”

3. 在“发布AI Gallery Notebook”页面填写参数，单击“创建”将Notebook代码样例分享至AI Gallery。
   - 填写“发布标题”，标题长度为3~64个字符，不能包含字符“\ / : * ? " < > | ' &”。
   - 选择运行环境：CPU、GPU或ASCEND。
   - 勾选“我已阅读并同意《华为云AI Gallery数字内容发布协议》和《华为云AI Gallery服务协议》”。

   图 1-5 发布 AI Gallery Notebook

4. 界面提示成功创建分享后，返回至AI Gallery，进入示例的详情页面查看示例。
   a. 进入AI Gallery首页。选择“项目”，进入项目列表页面。
   b. 在搜索框中输入创建好的Notebook名称，单击页签进入详情页。

编辑资产详情

资产发布成功后，发布者可以进入详情页修改该资产的名称、描述，让资产更吸引人。也可以修改资产的可见性。

编辑Notebook介绍

1. 在Notebook详情页，单击“项目介绍”。
2. 在基础设置中设置“许可证”、“语言”、“框架”、“任务类型”和“硬件资源”等信息。
3. 单击“确定”。

编辑设置

- 基本设置
  a. 单击右侧的 ，可以更改Notebook名称和描述。
  b. 编辑完成之后单击“确定”。
- 关联资产
  在输入框中输入资产ID后，单击“关联”即可关联其他资产，更方便其他使用者进行查找。算法可以关联数据集资产。

a. 选择“关联资产”，在输入框中输入待关联资产的ID，单击“关联”。

b. 在弹出的“资产信息”页面，单击“确定”即可关联资产。

- 可见范围设置

  您可以选择更改您的资产可见性，可选择“公开”或“私密”（私密状态下，也可以选择“仅自己可见”或“指定成员可见”）。

  说明

  在编辑资产详情时，请勿输入涉政、迷信、违禁等相关敏感词汇。

删除项目

当资产不使用时，支持删除，释放AI Gallery仓库的存储空间。

1. 在资产详情页，选择“设置”页签。
2. 在“删除资产”处，单击“删除”按钮，确认后资产将被删除。

须知

- 删除操作不可撤销，执行此操作后该资产及相关文件将被永久删除，请谨慎操作。
- 已被订阅的资产不可用删除。

1.5 发布和管理 AI Gallery 镜像

1.5.1 托管镜像到 AI Gallery

创建镜像资产

1. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
2. 单击左上方“创建资产”，选择“镜像”。
3. 在“创建镜像”弹窗中配置参数，单击“创建”。

表 1-23 创建镜像

参数名称	说明
英文名称	必填项，镜像的英文名称。 如果没有填写“中文名称”，则资产发布后，在镜像页签上会显示该“英文名称”。
中文名称	镜像的中文名称。 如果填写了“中文名称”，则资产发布后，在镜像页签上会显示该“中文名称”。

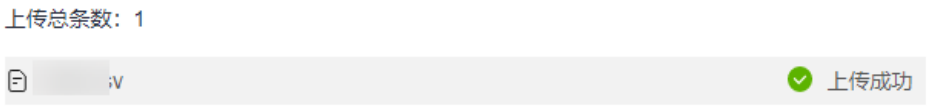
参数名称	说明
描述	填写资产简介，镜像发布后将作为副标题显示在镜像页签上，方便用户快速了解资产。 支持0~90个字符，请勿在描述中输入涉政、迷信、违禁等相关敏感词，否则发布审核无法通过。

创建完成后，跳转至镜像详情页。

上传镜像文件

1. 在镜像详情页，选择“镜像文件”页签。
2. 单击“添加文件”，进入上传文件页面，选择本地的数据文件单击“点击上传”或拖动文件，单击“确认上传”启动上传。
   - 上传单个超过5GB的文件时，请使用Gallery CLI工具。CLI工具的获取和使用请参见**Gallery CLI配置工具指南**。
   - 文件合集大小不超过50GB。
   - 文件上传完成前，请不要刷新或关闭上传页面，防止意外终止上传任务，导致数据缺失。
3. 当文件状态变成“上传成功”表示数据文件成功上传至AI Gallery仓库进行托管。单击“完成”返回镜像文件页面。

图 1-6 上传成功



说明

文件上传过程中请耐心等待，不要关闭当前上传页面，关闭页面会中断上传进程。

1.5.2 发布镜像到 AI Gallery

除了Gallery提供的已有资产外，还可以将个人创建的资产发布至Gallery货架上，供其他AI开发者使用，实现资产共享。

镜像资产上架

1. 登录AI Gallery，选择右上角“我的Gallery”。
2. 在“我的资产 > 镜像”下，选择未发布的镜像，单击镜像名称，进入镜像详情页。
3. 在镜像详情页，单击右侧“发布”，在发布镜像页面编辑发布信息后，单击“发布”。

表 1-24 发布镜像的参数说明

参数名称	说明
中文名称	镜像发布后显示的名称，在创建镜像时设置的名称，此处不可编辑。
描述	必填项，填写资产简介，镜像发布后将显示在镜像页签上，方便用户快速了解资产。 支持1~90个字符，请勿在描述中输入涉政、迷信、违禁等相关敏感词，否则发布审核无法通过。
可见范围	● “所有用户可见”：表示公开资产，所有用户都可以查看该资产。 ● “指定用户可见”：输入账号名、账号ID或用户昵称搜索并选择用户，使其可见该资产。
可用范围	选择是否启用“申请用户可用”。 ● 勾选启用：当用户要使用该镜像时需要提交申请，只有镜像所有者同意申请后，才能使用镜像。 ● 不勾选不启用（默认值）：所有可见资产的用户都可以直接使用镜像。

4. 发布后，资产会处于“审核中”，审核中的资产仅资产所有者可见。
   审核完成后，资产会变成“已发布”状态，并在镜像列表可见。

1.5.3 管理 AI Gallery 镜像

编辑镜像介绍

资产发布上架后，准确、完整的资产介绍有助于提升资产的排序位置和访问量，能更好的支撑用户使用该资产。

1. 在镜像详情页，选择“镜像介绍”页签，单击右侧“编辑介绍”。
2. 编辑镜像基础设置和镜像描述。

表 1-25 镜像介绍的参数说明

参数名称		说明
基础设置	中文名称	显示镜像的名称，不可编辑。
README.md	-	资产的README内容，支持添加资产的简介、使用场景、使用方法等信息。

3. 编辑完成后，单击“确认”保存修改。

管理镜像文件

- **预览文件**
  在镜像详情页，选择“镜像文件”页签。单击文件名称即可在线预览文件内容。

> **说明**
>
> 仅支持预览大小不超过10MB、格式为文本类或图片类的文件。

- **下载文件**

  在镜像详情页，选择“镜像文件”页签。单击操作列的“下载”，选择保存路径单击“确认”，即可下载文件到本地。

- **删除文件**

  在镜像详情页，选择“镜像文件”页签。单击操作列的“删除”，确认后即可将已经托管的文件从AI Gallery仓库中删除。

> **说明**
>
> 文件删除后不可恢复，请谨慎操作。

下架镜像

AI Gallery中已上架的资产支持下架操作。

1. 在AI Gallery首页，选择右上角“我的Gallery”。
2. 在“我的资产”下，查看已上架的资产。
3. 单击资产名称，进入资产详情页。
4. 在资产详情页，单击“下架”，在弹窗中单击“确定”。即可将资产下架。

删除镜像

当资产不使用时，支持删除，释放AI Gallery仓库的存储空间。

1. 在资产详情页，选择“设置”页签。
2. 在“删除资产”处，单击“删除”按钮，确认后资产将被删除。

须知

删除操作不可撤销，执行此操作后该资产及相关文件将被永久删除，请谨慎操作。

1.6 发布和管理 AI Gallery 中的 AI 应用

1.6.1 发布本地 AI 应用到 AI Gallery

场景描述

AI Gallery自定义AI应用能力为您提供了一个自由灵活的AI应用创建方式，您可以基于AI Gallery上提供的基础能力，发挥您的创造力，通过自定义代码的形式，自由地构建出您需要的AI应用形态。

准备 AI 应用运行文件“app.py”

AI应用运行文件“app.py”的代码示例如下。其中，加粗的代码为必须保留的内容。

```
import gradio as gr
import os
POD_IP = os.getenv('POD_IP')  // 获取容器IP
ROOT_PATH = os.getenv('ROOT_PATH')  //获取服务根路径
def greet(name):
    return "Hello " + name + "!"
with gr.Blocks() as demo:
    name = gr.Textbox(label="Name")
    output = gr.Textbox(label="Output Box")
    greet_btn = gr.Button("Greet")
    greet_btn.click(
        fn=greet,
        inputs=name,
        outputs=output,
        api_name="greet",
        queue=False)        // AI Gallery不支持应用将事件放入队列中，必须将queue设置为false。
demo.launch(server_name=POD_IP, root_path=ROOT_PATH)  //指定应用启动路径。
```

创建 AI 应用

1. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
2. 单击左上方“创建资产”，选择“AI应用”。
3. 在“创建AI应用”页面配置参数。

表 1-26 创建 AI 应用

参数	是否必填	说明
AI应用英文名称	是	自定义一个易于分辨的AI应用英文名称。 只能以数字、大小字母、下划线组成，且字符长度在3到90之间。
中文名称	是	自定义一个易于分辨的AI应用中文名称。 字符长度在1到30之间。
许可证	否	选择AI应用遵循的许可证。

参数	是否必填	说明
计算规格选择	是	按需选择计算规格。单击“选择”，在弹窗中选择资源规格并设置运行时长控制，单击“确定”。 • 在“所在区”选择计算规格所在的区域。默认显示全部区域的计算规格。 • 选择计算规格不可用的资源会置灰。右侧“配置信息”区域会显示计算规格的详细数据，AI Gallery会基于资产和资源情况分析该任务是否支持设置“商品数量”，用户可以基于业务需要选择任务所需的资源卡数。 • 在“运行时长控制”选择是否指定运行时长。 – 不限时长：不限制作业的运行时长，AI Gallery工具链服务部署完成后将一直处于“运行中”。 – 指定时长：设置作业运行几小时后停止，当AI Gallery工具链服务运行时长达到指定时长时，系统将会暂停作业。时长设置不能超过计算资源的剩余额度。 **说明** 如果选择付费资源，则请确认账号未欠费，且余额高于所选计算规格的收费标准，否则可能会导致AI Gallery工具链服务异常中断。AI Gallery的计算规格的计费说明请参见**计算规格说明**。
AI应用封面图	否	上传一张AI应用封面图，AI应用创建后，将作为AI应用页签的背景图展示在AI应用列表。建议使用16：9的图片，且大小不超过7MB。 如果未上传图片，AI Gallery会为AI应用自动生成封面。
应用描述	否	输入AI应用的功能介绍，AI应用创建后，将展示在AI应用页签上，方便其他用户了解与使用。 支持0~100个字符。

4. 参数填写完成后，单击“创建”，确认订单信息无误后，单击“确定”跳转至AI应用详情页。
   当AI应用的状态变为“待启动”时，表示创建完成。

启动 AI 应用

1. 上传AI应用的运行文件“app.py”。在AI应用详情页，选择“应用文件”页签，单击“添加文件”，进入上传文件页面。
   运行文件的开发要求请参见**准备AI应用运行文件app.py**。
   – 上传单个超过5GB的文件时，请使用Gallery CLI工具。CLI工具的获取和使用请参见**Gallery CLI配置工具指南**。
   – 文件合集大小不超过50GB。
   – 文件上传完成前，请不要刷新或关闭上传页面，防止意外终止上传任务，导致数据缺失。
   – 如果上传的文件名称和已有文件重名，系统会自动用新文件内容覆盖已有文件内容。

2. 运行文件上传完成后，在AI应用详情页，选择“设置”页签，在“运行资源设置”处单击“启动”，完成订单信息确认后单击“确定”开始构建AI应用。

   当AI应用状态变为“运行中”时，表示启动成功。在AI应用详情页的“应用”页签，可以在线体验应用。

1.6.2 将 AI Gallery 中的模型部署为 AI 应用

AI Gallery支持将模型部署为AI应用，在线共享给其他用户使用。

前提条件

选择的模型必须是支持部署为AI应用的模型，否则模型详情页没有“部署 > AI应用”选项。

部署 AI 应用

1. 登录AI Gallery。
2. 单击“模型”进入模型列表。
3. 选择需要部署为AI应用的模型，单击模型名称进入模型详情页。
4. 在模型详情页，选择“部署 > AI应用”进入创建AI应用页面。
5. 在创建AI应用页面填写相关参数。

表 1-27 创建 AI 应用

参数	是否必填	说明
AI应用英文名称	是	自定义一个易于分辨的AI应用英文名称。 只能以数字、大小字母、下划线组成，且字符长度在3到90之间。
中文名称	否	自定义一个易于分辨的AI应用中文名称。 字符长度在1到30之间。
许可证	否	选择AI应用遵循的许可证。

参数	是否必填	说明
计算规格选择	是	按需选择计算规格。单击“选择”，在弹窗中选择资源规格并设置运行时长控制，单击“确定”。 • 在“所在区”选择计算规格所在的区域。默认显示全部区域的计算规格。 • 选择计算规格不可用的资源会置灰。右侧“配置信息”区域会显示计算规格的详细数据，AI Gallery会基于资产和资源情况分析该任务是否支持设置“商品数量”，用户可以基于业务需要选择任务所需的资源卡数。 • 在“运行时长控制”选择是否指定运行时长。 – 不限时长：不限制作业的运行时长，AI Gallery工具链服务部署完成后将一直处于“运行中”。 – 指定时长：设置作业运行几小时后停止，当AI Gallery工具链服务运行时长达到指定时长时，系统将会暂停作业。时长设置不能超过计算资源的剩余额度。 **说明** 如果选择付费资源，则请确认账号未欠费，且余额高于所选计算规格的收费标准，否则可能会导致AI Gallery工具链服务异常中断。AI Gallery的计算规格的计费说明请参见**计算规格说明**。
AI应用封面图	否	上传一张AI应用封面图，AI应用创建后，将作为AI应用页签的背景图展示在AI应用列表。建议使用16：9的图片，且大小不超过7MB。 如果未上传图片，AI Gallery会为AI应用自动生成封面。
应用描述	否	输入AI应用的功能介绍，AI应用创建后，将展示在AI应用页签上，方便其他用户了解与使用。 支持0~100个字符。

6. 参数填写完成后，单击“创建”，确认订单信息无误后，单击“确定”跳转至AI应用详情页。

   当资产状态变为“运行中”表示AI应用部署完成。在AI应用详情页的“应用”页签，可以在线体验应用。

1.6.3 管理 AI Gallery 中的 AI 应用

当AI应用创建完成后，支持修改内容，例如修改环境变量、可见范围。

约束限制

当AI应用的“可见范围”是“私密”时，才支持修改环境变量、可见范围或删除AI应用。

管理 AI 应用环境变量

AI应用支持增删改查环境变量，配置好的环境变量可以在运行文件中直接调用。

1. 在AI应用详情页，选择“设置”页签。
2. 在“环境变量管理”处，可以查看、新增、修改、删除环境变量。
   最多支持创建100个环境变量。变量名称不可重复，只能由下划线、字母与数字组成且不能以数字开头。
   - 查看环境变量的值：单击，可以查看当前环境变量的值。
   - 新增环境变量：单击“新增”，在编辑环境变量弹窗中配置“变量名称”和“变量值”，单击“确定”完成配置。
   - 修改环境变量：单击，在编辑环境变量弹窗中修改“变量名称”或“变量值”，单击“确定”完成配置。
   - 删除环境变量：单击，确认永久删除环境变量，单击“确定”完成删除。
3. 重启AI应用，使环境变量的新增、修改、删除生效。
   当AI应用的状态为“运行中”时，则在“运行资源设置”处，单击“重启”。
   当AI应用的状态为非“待启动”时，则环境变量的变更会随应用启动自动生效。

管理 AI 应用可见范围

创建AI应用时，默认“可见范围”是“私密”，且“仅自己可见”。创建完成后，支持修改可见范围。

- “公开”：表示公开资产，所有用户都可以查看该资产。
  当选择公开AI应用，系统会自动提交资产公开申请，审核通过之前资产还是私密状态，审核通过后就会变成公开状态。
- “私密”：表示仅部分用户可见。
  - “仅自己可见”：默认状态，表示仅AI应用创建者可见该资产。
  - “指定用户”：表示AI应用创建者和指定的用户可见该资产。
    当指定用户可见时，保存可见用户名单后即可生效。

删除 AI 应用

当AI应用不再使用时，支持删除，释放AI Gallery仓库的存储空间。

1. 在AI应用详情页，选择“设置”页签。
2. 确认AI应用状态是否为“运行中”。
   - 是，则在“运行资源设置”处，单击“暂停”，停止AI应用再执行下一步。
   - 否，则执行下一步。
3. 在“删除AI应用”处，单击“删除AI应用”按钮，确认后AI应用将被删除。

须知

删除操作不可撤销，执行此操作后该AI应用及相关文件将被永久删除，请谨慎操作。

1.7 使用 AI Gallery 微调大师训练模型

AI Gallery支持将模型进行微调，训练后得到更优模型。

场景描述

模型微调是深度学习中的一种重要技术，它是指在预训练好的模型基础上，通过调整部分参数，使其在特定任务上达到更好的性能。 在实际应用中，预训练模型是在大规模通用数据集上训练得到的，而在特定任务上，这些模型的参数可能并不都是最合适的，因此需要进行微调。

AI Gallery的模型微调，简单易用，用户只需要选择训练数据、创建微调任务，模型微调就会对数据进行训练，快速生成模型。

约束限制

- 如果模型的“任务类型”是“文本问答”或“文本生成”，则支持模型微调。如果模型的“任务类型”是除“文本问答”和“文本生成”之外的类型（即自定义模型），则模型文件必须满足**自定义模型规范（训练）**才支持模型自定义训练。
- 当使用自定义镜像进行模型微调时，要确认镜像是否满足**自定义镜像规范**，否则无法成功完成自定义训练。

进入模型微调

1. 登录AI Gallery。
2. 单击“模型”进入模型列表。
3. 选择需要进行微调训练的模型，单击模型名称进入模型详情页。
4. 在模型详情页，选择“训练 > 微调大师”进入微调工作流页面。

选择训练任务类型

选择模型微调的训练任务类型。

- 当模型的“任务类型”是“文本问答”或“文本生成”时，“训练任务类型”默认和模型“任务类型”一致。“训练任务类型”支持修改，如果模型文件满足**自定义模型规范（训练）**，则“训练任务类型”支持选择“自定义”。
- 当模型的“任务类型”是除“文本问答”和“文本生成”之外的类型（即自定义模型）时，则“训练任务类型”默认为“自定义”，支持修改为“文本问答”或“文本生成”。
- 当使用自定义镜像进行模型微调时，“训练任务类型”默认为“自定义”，且不支持修改。

准备数据

说明

- 本地上传数据需要确保数据已按照数据集要求完成编排。如果是自定义模型，此处的数据集要求即为模型文件“dataset_readme.md”里的内容。
- 单个文件最大5GB，所有文件总大小不超过50G。

1. 在微调工作流的“数据准备”环节选择数据集。
   - **从本地上传**
     i. 在“从本地上传”处，单击“点击上传”，选择本地编排好的训练数据。
     ii. 数据上传成功后，页面会有提示信息。

此时AI Gallery会自动新建一个数据集，单击提示信息处的“查看”可以进入数据集详情页，也可以在“我的Gallery > 数据集 > 我创建的数据集”进入数据集详情页查看。

- **从AI Gallery中选**
  i. 单击“从AI Gallery中选择”。
  ii. 在弹窗中，从“我创建的”或“我收藏的”数据集中选择所需要数据集。
  iii. 选择完成后，单击“确定”。

2. 数据准备完成后，单击“下一步”进入“作业设置”环节。

设置并启动作业

1. 在微调工作流的“作业设置”环节配置训练作业参数。
   a. 算法配置，会显示已选模型的信息，基于已选模型选择微调方式。
      - 当“训练任务类型”是“文本问答”或“文本生成”时，AI Gallery支持的微调方式是LoRA。
      - 当“训练任务类型”是“自定义”时，微调方式来自于模型文件“train_params.json”。

      说明

      低秩适应（LoRA）是一种重参数化方法，旨在减少具有低秩表示的可训练参数的数量。权重矩阵被分解为经过训练和更新的低秩矩阵。所有预训练的模型参数保持冻结。训练后，低秩矩阵被添加回原始权重。这使得存储和训练LoRA模型更加高效，因为参数明显减少。

   b. 超参数设置，基于训练作业配置超参。超参指的是模型训练时原始数据集中实际字段和算法需要字段之间的映射关系。
      - 当“训练任务类型”是“文本问答”或“文本生成”时，则常见的超参说明请参见**表1-28**。
      - 当“训练任务类型”是“自定义”时，超参信息来自于模型文件“train_params.json”。如果不使用可选超参，建议单击右侧的删除按钮，删除参数。

表 1-28 常见超参说明

参数名称	参数类型	说明
data_url	String	数据OBS存储路径。
train_url	String	微调产物输出OBS路径。
train_data_file	String	训练数据文件名。
test_data_file	String	测试数据文件名。
prompt_field	String	数据prompt列名。

参数名称	参数类型	说明
response_field	String	数据response列名。
history_field	String	数据history列名。
prefix	String	数据格式化时使用的前缀。
instruction_template	String	数据格式化时使用的指令模板。
response_template	String	数据格式化时使用的回答模板。
lora_alpha	int	Lora scaling的alpha参数。
lora_dropout	float	Lora dropout概率。
lora_rank	int	Lora attention维度。
per_device_train_batch_size	int	用于训练的每个GPU/TPU core/CPU的批处理大小。
gradient_accumulation_steps	int	梯度累计步数。
max_steps	int	训练最大步数，如果数据耗尽，训练将会在最大步数前停止。
save_steps	int	checkpoint保存步数。
logging_steps	int	日志输出步数。
learning_rate	float	初始学习率。
max_grad_norm	float	梯度裁剪最大范数。
warmup_ratio	float	热身步数比。
max_seq_length	int	数据最大序列长度。
finetuned_model	String	前序微调产物OBS路径。
bits	int	模型量化bit数，如4、8。
max_eval_samples	int	最大测试数据数。

c. 计算规格选择，按需选择计算规格。单击“选择”，在弹窗中选择资源规格，单击“确定”。

- 在“所在区”选择计算规格所在的区域。默认显示全部区域的计算规格。
- 选择计算规格不可用的资源会置灰。右侧“配置信息”区域会显示计算规格的详细数据，AI Gallery会基于资产和资源情况分析该任务是否支持设置“商品数量”，用户可以基于业务需要选择任务所需的资源卡数。

说明

如果选择付费资源，则请确认账号未欠费，且余额高于所选计算规格的收费标准，否则可能会导致AI Gallery工具链服务异常中断。AI Gallery的计算规格的计费说明请参见**计算规格说明**。

2. 作业参数配置完成后，单击“启动作业”。
3. 在“订单信息确认”页面，确认服务信息和费用，单击“确定”提交模型训练任务。

   单击“返回模型训练”跳转到微调大师页面，可以查看训练作业状态。当“状态”为“训练完成”时，表示微调任务完成。

   - 单击操作列的“查看模型”跳转到微调获得的新模型的详情页面。
   - 单击操作列的“任务详情”可以在弹窗中查看“训练信息”、“训练日志”和“指标效果”。
   - 单击操作列的“更多 > 删除任务”，可以删除微调任务，但是微调获得的新模型不会被删除。

查看训练效果

启动模型微调任务后，在微调大师列表单击操作列的“任务详情”，在弹窗中选择“指标效果”页签，可以查看训练效果。

表 1-29 训练效果的指标介绍

指标名称	指标说明
NPU/GPU利用率	在训练过程中，机器的NPU/GPU占用情况（横坐标时间，纵坐标占用率）。
显存利用率	在训练过程中，机器的显存占用情况（横坐标时间，纵坐标占用率）。
吞吐	在训练过程中，每卡处理tokens数量（tokens/s/p）。每种框架计算方式不一致，例如，ATB可通过“samples per second*seq_lenth/总卡数”得到tokens/s/p，输出给throughout字段，seq_lenth取值在训练脚本中可以查看。 单机8卡吞吐量一般为1650tokens/s/p，双机16卡吞吐量一般为1625tokens/s/p。 **说明** 自定义训练或自定义镜像训练，需要提前在训练启动脚本（例如“train.py”）中定义好迭代次数、LOSS和吞吐数据的存放位置，以及存放格式（必须是“迭代次数|loss|吞吐”），才能在此处正常查看吞吐和“训练LOSS”曲线。

指标名称	指标说明
训练LOSS	训练阶段的LOSS变化，模型在日志里用LOSS关键词记录数据，按照训练迭代周期记录LOSS值。

微调产物说明

模型微调完成后，会得到一个新模型，即微调产物。

在微调大师页面，单击操作列的“查看模型”跳转到微调获得的新模型的详情页面。选择“模型文件”页签可以查看微调产物。各文件说明请参见**表1-30**。

图 1-7 微调产物示例

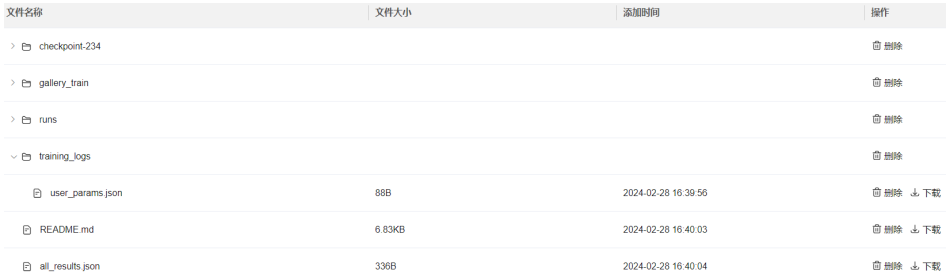


表 1-30 微调产物说明

文件名	文件说明
gallery_train文件夹	自定义模型的模型训练文件，仅当使用自定义模型微调时才会有这个微调产物，内容和预训练模型里的gallery_train文件一致。
training_logs/user_params.json	微调配置参数信息，AI Gallery会自动将微调设置的参数信息记录在此文件下。
“README.md”	模型的基础信息。内容和预训练模型里“模型文件”页签的“README.md”一致。
其他文件	当使用自定义模型微调时，可能还会有一些其他微调产物，这是由自定义模型的训练脚本文件train.py决定的，如果训练脚本定义了归档其他训练产物，就会在此处呈现。

1.8 使用 AI Gallery 在线推理服务部署模型

AI Gallery支持将训练的模型或创建的模型资产部署为在线推理服务，可供用户直接调用API完成推理业务。

约束限制

- 如果模型的“任务类型”是“文本问答”或“文本生成”，则支持在线推理。如果模型的“任务类型”是除“文本问答”和“文本生成”之外的类型（即自定义模型），则模型文件必须满足**自定义模型规范（推理）**才支持模型自定义推理。

- 当使用自定义镜像部署推理服务时，要确认镜像是否满足**自定义镜像规范**，否则无法成功完成推理服务的部署。

部署推理服务

1. 登录AI Gallery。
2. 单击“模型”进入模型列表。
3. 选择需要部署为推理服务的模型，单击模型名称进入模型详情页。
4. 在模型详情页，选择“部署 > 推理服务”进入部署推理服务页面。
5. 在部署推理服务页面完成参数配置。

表 1-31 部署推理服务

参数	子参数	说明
推理服务设置	服务名称	必填项，自定义一个在线推理服务的名称。 支持1~30个字符。
	安全认证	支持“公开”和“AppCode认证”。 • 公开：无需认证，API地址可被公开访问。 • AppCode认证：需使用有效的AppCode进行认证。AppCode使用API网关颁发的AppCode进行身份认证，调用者将AppCode放到请求头中进行身份认证，确保只有授权的调用者能够调用API接口。 AppCode的获取方法：单击AI Gallery页面右上角“我的Gallery”，在左侧菜单栏选择“鉴权管理”。在“鉴权管理”中单击“创建AppCode”，填写描述信息后，即可在列表中显示新增的AppCode。 **说明** 推理服务只能使用计算规格所在区域的AppCode进行认证鉴权。
	描述	输入在线服务的描述信息。 支持0~100个字符，请勿在描述中输入涉政、迷信、违禁等相关敏感词。
高级设置	推理任务类型	选择推理任务类型。 • 当模型的“任务类型”是“文本问答”或“文本生成”时，“推理任务类型”默认和模型“任务类型”一致。“推理任务类型”支持修改，如果模型文件满足**自定义模型规范（推理）**，则“推理任务类型”支持选择“自定义”。 • 当模型的“任务类型”是除“文本问答”和“文本生成”之外的类型（即自定义模型）时，则“推理任务类型”默认为“自定义”，支持修改为“文本问答”或“文本生成”。 • 当使用自定义镜像部署推理服务时，“推理任务类型”默认为“自定义”，且不支持修改。

参数	子参数	说明
	参数设置	当使用自定义镜像部署推理服务时，如果自定义镜像的“模型文件”中上传了“gallery_inference/inference_params.json”文件，则此处会显示inference_params文件里的参数配置项，支持修改自定义镜像的部署参数。
计算规格选择	-	按需选择计算规格。单击“选择”，在弹窗中选择资源规格并设置运行时长控制，单击“确定”。 ● 在“所在区”选择计算规格所在的区域。默认显示全部区域的计算规格。 ● 选择计算规格不可用的资源会置灰。右侧“配置信息”区域会显示计算规格的详细数据，AI Gallery会基于资产和资源情况分析该任务是否支持设置“商品数量”，用户可以基于业务需要选择任务所需的资源卡数。 ● 在“运行时长控制”选择是否指定运行时长。 – 不限时长：不限制作业的运行时长，AI Gallery工具链服务部署完成后将一直处于“运行中”。 – 指定时长：设置作业运行几小时后停止，当AI Gallery工具链服务运行时长达到指定时长时，系统将会暂停作业。时长设置不能超过计算资源的剩余额度。 **说明** 如果选择付费资源，则请确认账号未欠费，且余额高于所选计算规格的收费标准，否则可能会导致AI Gallery工具链服务异常中断。AI Gallery的计算规格的计费说明请参见**计算规格说明**。

6. 服务参数配置完成后，单击“启动部署”。
7. 在“订单信息确认”页面，确认服务信息和费用，单击“确定”跳转至在线推理服务列表页面。

   当“状态”变为“运行中”表示在线推理服务部署成功，可以进行服务预测。

推理服务预测

待在线推理服务状态变为“运行中”时，便可进行推理预测。

1. 在在线推理服务列表页面，选择服务“状态”为“运行中”的服务。
2. 单击操作列的“推理测试”，在测试页面根据任务类型以及页面提示完成对应的测试。

调用 API

待推理服务的状态变为“运行中”时，可单击操作列的“调用”，复制对应的接口代码，在本地环境或云端的开发环境中进行接口。

图 1-8 调用接口

说明

当部署推理服务的“安全认证”选择了“AppCode认证”，则需要将复制的接口代码中headers中的X-Apig-AppCode的参数值修改为真实的AppCode值。

Python示例代码如下：

```
import requests
API_URL = "https://xxxxxxx/v1/gallery/65f38c4a-bbd0-4d70-a724-5fccf573399a/"
headers = {
    "Content-Type": "application/json",
    "X-Apig-AppCode": "YOUR_AppCode"
}

def query(payload):
    response = requests.post(API_URL, headers=headers, json=payload)
    return response.json()

output = query({
    "inputs": "我是一名作家，喜欢写"
})
```

查看推理服务

在在线推理服务列表页面，单击服务操作列的“服务详情”（如果是“运行中”的推理服务，则需要单击操作列的“更多 > 服务详情”），可以在弹窗中查看推理服务的“服务信息”、“服务日志”和“指标效果”。

停止推理服务

当“运行中”的推理服务使用完成后，在在线推理服务列表页面，单击操作列的“更多 > 停止服务”即可停止推理服务，节约资源成本。

查看推理效果

当推理服务处于“运行中”时，在服务列表单击操作列的“更多 > 服务详情”，在弹窗中选择“指标效果”页签，可以查看推理效果。

支持设置时间区间，查看不同时间下的推理效果。

说明

仅当推理服务处于“运行中”，才支持查看监控指标。

表 1-32 推理效果的指标介绍

指标名称	指标说明
CPU使用率	在推理服务启动过程中，机器的CPU占用情况。
内存使用率	在推理服务启动过程中，机器的内存占用情况。
显卡使用率	在推理服务启动过程中，机器的NPU/GPU占用情况。
显存使用率	在推理服务启动过程中，机器的显存占用情况。

1.9 Gallery CLI 配置工具指南

1.9.1 安装 Gallery CLI 配置工具

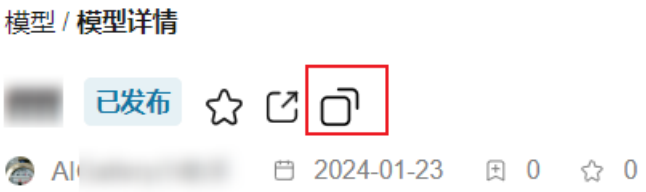
场景描述

- Gallery CLI配置工具支持将AI Gallery仓库的资产下载到云服务端，便于在云服务本地进行训练、部署推理。
- Gallery CLI配置工具支持将单个超过5GB的文件从本地上传至AI Gallery仓库中。

约束限制

- Gallery CLI配置工具下载文件时依赖集群的公网访问权限，所以在使用CLI时要求集群配置NAT网关，具体操作请参见**公网NAT网关**。
- 只有托管到AI Gallery仓库的资产才支持使用Gallery CLI配置工具下载文件，如果在资产详情页有“复制完整资产名称”按钮即表示该资产支持使用Gallery CLI配置工具下载，如**图1-9**所示。

图 1-9 复制完整资产名称



- “运行平台”设置为“Pangu Studio”的数据集，不支持使用CLI工具下载。

下载 Gallery CLI 配置工具包（本地）

如果是在本地服务器安装Gallery CLI配置工具，则参考本节将工具包下载至本地。

1. 下载AI Gallery CLI安装包、校验文件。
   - 单击**链接**，下载AI Gallery CLI安装包。
   - 单击**链接**，下载AI Gallery CLI校验文件。
2. 将AI Gallery CLI安装包及对应的校验文件放在同一目录下，执行如下命令使用OpenSSL工具进行校验工具包。
   ```
   openssl cms -verify -binary -in  gallery_cli-*-py3-none-any.whl.cms -inform DER -content gallery_cli-*-py3-none-any.whl -noverify > ./test
   ```
   出现如下信息则表示校验通过。
   ```
   Verification successful
   ```

下载 Gallery CLI 配置工具包（云服务器）

如果是在ModelArts Lite等云服务器安装Gallery CLI配置工具，则参考本节将工具包下载至云服务器。

1. 登录AI Gallery，单击右上角“我的Gallery”进入我的Gallery页面。
2. 左侧菜单栏选择“我的资源 > 云服务器”，单击专属资源池页签进入云服务详情页面。
3. 在节点页签，单击 ··· 选择“配置工具”，弹出该节点的配置工具页面。
4. 在配置工具页面，单击“下载”启动下载任务。当配置工具的状态记录中“工具状态”为“下载完成”时表示下载完成，工具包存放在“下载位置”的目录下。

   说明

   如果下载失败，单击“下载”可以重新下载。
5. 登录云服务器查看工具包是否下载成功。
   a. 在云服务详情页面，单击节点页签的 ··· 选择“前往控制台”跳转到云服务器控制台。
   b. 在云服务器控制台的节点基本信息页面，单击右上角“远程登录”选择登录方式远程登录云服务器节点。推荐使用CloudShell登录，直接页面单击“CloudShell登录”跳转到CloudShell页面，输入专属资源池信息登录服务器。具体步骤请参见**远程登录Linux弹性云服务器（CloudShell方式）**。
   c. 进入**4**获取的“下载位置”，执行如下命令使用OpenSSL工具进行校验工具包。
   ```
   openssl cms -verify -binary -in  gallery_cli-*-py3-none-any.whl.cms -inform DER -content gallery_cli-*-py3-none-any.whl -noverify > ./test
   ```

出现如下信息则表示校验通过。

```
Verification successful
```

安装 Gallery CLI 配置工具

当Gallery CLI配置工具包下载完成后，进入服务器安装工具。不管是ModelArts Lite云服务，还是本地Windows/Linux等服务器，安装操作都相同。

1. 登录服务器，激活python虚拟环境。

   ```
   conda activate [env_name] # 例如使用conda管理python环境（需要确认环境已安装Anaconda）
   ```

2. 在python环境中安装CLI工具。

   ```
   pip install ./gallery_cli-0.0.3-py3-none-any.whl
   ```

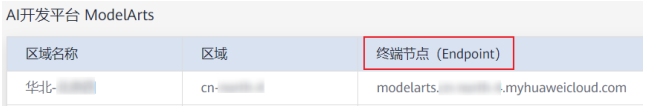
3. 配置CLI工具的环境信息。

   a. 在服务器的任意目录下（本文以“/gallerycli”为例）新建CLI配置文件“config.env”，包含如下配置信息。

   ```
   # IAM相关配置
   iam_url=https://iam.myhuaweicloud.com/v3/auth/tokens
   iam_project=cn-north-7
   iam_timeout=15
   # 账号密码，和AK/SK二选一
   iam_domain=xxx
   iam_user=xxx
   iam_password=xxx
   # AK/SK，和账号密码二选一
   iam_ak=xxx
   iam_sk=xxx

   # 托管仓库相关配置
   repo_url=https://{ModelArts-Endpoint}.myhuaweicloud.com

   # 系统相关配置
   cached_dir=/test

   # 加解密配置
   sdk_encrypt_implementation_func=/path/to/crypt.py.my_encrypt_func
   sdk_decrypt_implementation_func=/path/to/crypt.py.my_decrypt_func
   ```

表 1-33 配置项参数说明

参数名称	说明
iam_url	IAM地址，默认为“https://iam.myhuaweicloud.com/v3/auth/tokens”。
iam_project	服务器所在区域的项目名称，获取方式请参见**获取项目ID和名称**。如果是本地服务器则默认是北京四区域，此处填写“cn-north-4”。
iam_timeout	（可选）IAM访问超时时间，单位为秒，缺省值是5。当环境网络不稳定时，建议将该值改大。如果超过该时间IAM还没有响应，系统会返回超时错误码，便于定位链接故障。
iam_domain	用户的账号ID，获取方式请参见**获取账号名和账号ID**。
iam_user	IAM用户名，获取方式请参见**获取用户名和用户ID**。

参数名称	说明
iam_password	IAM用户密码，即账号的登录密码。
iam_ak	访问密钥AK，获取方式请参见**访问密钥**。
iam_sk	访问密钥SK，获取方式请参见**访问密钥**。
repo_url	AI Gallery仓库的地址，格式为“http://*{ModelArts-Endpoint}*.myhuaweicloud.com”，其中不同区域的Endpoint可以在**ModelArts地区和终端节点**获取。 
cached_dir	缓存目录，默认AI Gallery仓库的文件下载至该目录下。
sdk_encrypt_implementation_func	自定义加密函数，认证用的AK和SK硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件中密文存放，使用时解密，确保安全。
sdk_decrypt_implementation_func	自定义解密函数，认证用的AK和SK硬编码到代码中或者明文存储都有很大的安全风险，建议在配置文件中密文存放，使用时解密，确保安全。

须知

- 配置文件中，账号密码认证和AK/SK认证二选一即可。如果使用账号密码认证，则需要填写配置项“iam_domain”、“iam_user”和“iam_password”；如果使用AK/SK认证，则需要填写配置项“iam_ak”、“iam_sk”和加密配置。
- 华为账号只能使用AK/SK认证。如果要使用账号密码认证，且必须先创建一个IAM用户再获取IAM用户名和密码进行认证，操作指导请参见**创建IAM用户**。
- 配置项中的认证凭据信息不建议使用明文，可以通过下述方式扩展自定义的加解密组件。
  - 在module(yourmodule)自定义一个解（加）密方法，例如decrypt_func(cipher)，要求可以通过“from yourmodule import decrypt_func”的方式获取认证凭据信息。
  - 在配置文件中配置“sdk_decrypt_implementation_func=yourmodule.decrypt_func”指向自定义的解密方法的引用。程序加载时会通过import_lib加载认证凭据信息。
  - 配置文件中配置密文的格式“iam_ak={Crypto}cipher”，其中cipher会在配置项读取认证凭据信息时被解析传递进decrypt_func方法中，进行解密。
  - 其他类似自定义加密的方法，会在保存Token到本地时进行加密。

b. 配置CLI工具的环境变量，指定到上一步新建的配置文件。

```
export SDK_CONFIG_PATH=/gallerycli/config.env      # 填写正确的config.env路径
```

4. 配置完成后，执行如下命令查看CLI工具是否安装成功。

```
gallery-cli --help
```

如果安装成功会显示CLI中所有可用选项的列表，如下所示。

```
Usage: gallery-cli [OPTIONS] COMMAND [ARGS]...

╭─ Options ──────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ --install-completion          Install completion for the current shell.                                            │
│ --show-completion             Show completion for the current shell, to copy it or customize the installation.     │
│ --help                        Show this message and exit.                                                          │
╰────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Commands ─────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ download                      Download files from the AI Gallery                                                   │
│ login                         Log in using ak sk from huawei cloud iam                                             │
│ logout                        Log out                                                                              │
╰────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
```

说明

“--help”选项可以用于获取命令的更多详细信息，可以随时使用它来列出所有可用选项及其详细信息。例如，“gallery-cli download --help”可以获取使用CLI下载文件的更多帮助信息。

登录 Gallery CLI 配置工具

当Gallery CLI配置工具安装完成后，可以登录Gallery CLI上传或下载AI Gallery仓库的资产。

在服务器执行如下命令登录Gallery CLI配置工具。

```
gallery-cli login
```

显示如下信息表示登录成功。“/test”是自定义的服务器的缓存目录，token是系统自动生成的文件夹。

```
/test/token
```

登出 Gallery CLI 配置工具

上传或下载AI Gallery仓库的资产完成后，登出Gallery CLI清理缓存。

在服务器执行如下命令登出Gallery CLI配置工具。

```
gallery-cli logout
```

显示如下信息表示登出成功。系统自动清除缓存目录“/test”下的token文件夹。

```
Logout successful!
```

1.9.2 使用 Gallery CLI 配置工具下载文件

在服务器（ModelArts Lite云服务器或者是本地Windows/Linux等服务器）上登录Gallery CLI配置工具后，通过命令“gallery-cli download”可以从AI Gallery仓库下载资源。

命令说明

登录Gallery CLI配置工具后，使用命令“gallery-cli download --help”可以获取Gallery CLI配置工具下载文件的帮助信息。

```
gallery-cli download --help
```

获得命令“gallery-cli download”可用选项的完整列表如下所示。

```
Usage: gallery-cli download [OPTIONS] REPO_ID [FILENAMES]...

Download files from the AI Gallery

╭─ Arguments ──────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ *    repo_id        TEXT          ID of the repo to download from (e.g. `username/repo-name`). [required]          │
│      filenames      [FILENAMES]...  Files to download (e.g. `config.json`,`data/metadata.jsonl`).                   │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
╭─ Options ────────────────────────────────────────────────────────────────────────────────────────────────────────────╮
│ --include         TEXT  Glob patterns to match files to download.                                                    │
│ --exclude         TEXT  Glob patterns to exclude from files to download.                                             │
│ --local-dir       TEXT  Specified local dir to store model or dataset                                                │
│ --help                  Show this message and exit.                                                                  │
╰──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────╯
```

具体支持如下使用场景：

- **下载单个文件**
- **下载多个文件**
- **下载文件到指定路径**
- **下载单个AI Gallery仓库**

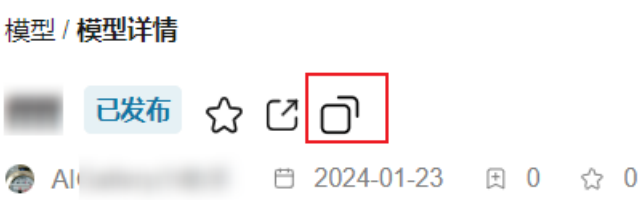
准备工作

获取“repo_id”和待下载的文件名。

- 获取“repo_id”

  在AI Gallery页面的资产详情页，单击 复制完整的资产名称，如**图1-10**所示，获取到的信息即为“repo_id”。例如，复制出的信息为“ur5468675/test_cli_model1”，则该资产的“repo_id”为“ur5468675/test_cli_model1”。

  图 1-10 复制完整资产名称

说明

如果资产详情页没有  按钮，则表示该资产不支持使用Gallery CLI配置工具下载文件。

- 获取待下载的文件名

  在AI Gallery页面的资产详情页，如果是模型资产，则选择“模型文件”页签，如果是数据集资产，则选择“文件版本”页签，获取“文件名称”。

下载单个文件

在服务器执行如下命令，可以从AI Gallery仓库下载单个文件到服务器的缓存目录下。

```
gallery-cli download {repo_id} {文件名}
```

如下所示，表示下载文件“config.json”到服务器的缓存目录“/test”下，当回显“100%”时表示下载完成。

```
gallery-cli download ur5468675/test_cli_model1 config.json

Download config.json: 100%|
##################################################################################
##| 665/665 [00:00<?, ?B/s]

/test/ur5468675--test_cli_model1/config.json
```

下载多个文件

- **根据文件名下载文件**

  在服务器执行如下命令，将待下载的文件名枚举出来即可从AI Gallery仓库依次下载多个文件到云服务器的缓存目录下。

  ```
  gallery-cli download {repo_id} {文件名} {文件名}
  ```

  其中，“repo_id”如何获取，文件名如何获取。

  如下所示，表示下载文件“config.json”和“merges.txt”到服务器的缓存目录“/test”下，当回显“100%”时表示下载完成。

  ```
  gallery-cli download ur5468675/test_cli_model1 config.json merges.txt

  Download 1/2 config.json: 100%|
  ##################################################################|
  665/665 [00:00<?, ?B/s]
  Download 2/2 merges.txt: 100%|
  ##################################################################| 456k/
  456k [00:00<00:00, 2.13MB/s]

  /test/ur5468675--test_cli_model1
  ```

- **通过include下载文件**

  在服务器执行如下命令，通过“--include”从AI Gallery仓库依次下载包含某种格式的文件到云服务器的缓存目录下。

  ```
  gallery-cli download {repo_id} --include "*.json"
  ```

  如下所示，表示下载所有“.json”格式的文件到服务器的缓存目录“/test”下，当回显“100%”时表示下载完成。

  ```
  gallery-cli download ur5468675/test_cli_model1 --include "*.json"

  Download 1/4 config.json: 100%|
  ##################################################################| 665/665
  [00:00<?, ?B/s]
  Download 2/4 generation_config.json: 100%|
  ```



```
##################################################| 124/124 [00:00<?, ?B/s]
Download 3/4 tokenizer.json: 100%|
##################################################################| 1.36M/1.36M
[00:00<00:00, 5.54MB/s]
Download 4/4 vocab.json: 100%|
##################################################################| 1.04M/
1.04M [00:00<00:00, 4.10MB/s]

/test/ur5468675--test_cli_model1
```

- **通过exclude下载文件**

  在服务器执行如下命令，通过“--exclude”从AI Gallery仓库依次下载除某种格式之外的其他格式的文件到服务器的缓存目录下。

  ```
  gallery-cli download {repo_id} --exclude "*.json"
  ```

  如下所示，表示下载除“.json”格式之外的其他格式的文件到服务器的缓存目录“/test”下，当回显“100%”时表示下载完成。

  ```
  gallery-cli download ur5468675/test_cli_model1 --exclude "*.json"

  Download 1/2 merges.txt: 100%|
  ##################################################################| 456k/
  456k [00:00<00:00, 2.78MB/s]
  Download 2/2 model.safetensors: 100%|
  ##################################################################| 548M/548M
  [00:13<00:00, 41.8MB/s]

  /test/ur5468675--test_cli_model1
  ```

下载文件到指定路径

在服务器执行如下命令，可以从AI Gallery仓库下载单个文件到服务器的指定路径下。

```
gallery-cli download {repo_id} {文件名} --local-dir={存放路径}
```

如下所示，表示下载文件“config.json”到服务器的“/tmp”路径下，当回显“100%”时表示下载完成。

```
gallery-cli download ur5468675/test_cli_model1 config.json --local-dir=/tmp

Download config.json: 100%|
##################################################################| 665/665 [00:00<?, ?
B/s]

/tmp/config.json
```

下载单个 AI Gallery 仓库

在服务器执行如下命令，可以将AI Gallery仓库的所有文件下载到服务器的缓存目录下。

```
gallery-cli download {repo_id}
```

如下所示，表示下载AI Gallery仓库“test_cli_model1”到服务器的缓存目录“/test”下，当回显“100%”时表示下载完成。

```
gallery-cli download ur5468675/test_cli_model1

Download 1/6 config.json: 100%|
##################################################################| 665/665
[00:00<?, ?B/s]
Download 2/6 generation_config.json: 100%|
##################################################| 124/124 [00:00<?, ?B/s]
Download 3/6 merges.txt: 100%|
##################################################################| 456k/456k
[00:00<00:00, 2.69MB/s]
```



```
Download 4/6 model.safetensors: 100%|
##########################################################| 548M/548M
[00:17<00:00, 32.0MB/s]
Download 5/6 tokenizer.json: 100%|
#############################################################| 1.36M/1.36M
[00:00<00:00, 4.88MB/s]
Download 6/6 vocab.json: 100%|
#################################################################| 1.04M/1.04M
[00:00<00:00, 4.43MB/s]

/test/ur5468675--test_cli_model1
```

1.9.3 使用 Gallery CLI 配置工具上传文件

在服务器（ModelArts Lite云服务器或者是本地Windows/Linux等服务器）上登录Gallery CLI配置工具后，通过命令“gallery-cli upload”可以往AI Gallery仓库上传资产。

命令说明

登录Gallery CLI配置工具后，使用命令“gallery-cli upload --help”可以获取Gallery CLI配置工具上传文件的帮助信息。

```
gallery-cli upload --help
```

获得命令“gallery-cli upload”可用选项的完整列表如下所示。

```
 Usage: gallery-cli upload [OPTIONS] REPO_ID [LOCAL_PATH] [PATH_IN_REPO]

 Upload File

╭─ Arguments ─────────────────────────────────────────────────────────────
──────────────────│ *    repo_id        TEXT          ID of the repo to upload to (e.g. `username/repo-
name`) [required]                                          ││     local_path       [LOCAL_PATH]    Directory
upload to repo [default: ./]                                                       ││
path_in_repo      [PATH_IN_REPO]  The repo path you want to upload (e.g. `dir1/
dir2`)                                                   │ ╰───────────────────────────────
───────────────────────────────────────────────────────────╯
╭─Options──────────────────────────────────────────────────────────────────
──────╮ ││ --include        TEXT  Glob patterns to match files to
download.                                                            ││ --exclude        TEXT  Glob
patterns to exclude from files to download.                                                         │
│ --help                 Show this message and
exit.                                                                       │ ╰──────────────
────────────────────────────────────────────────────────────────────────────╯
```

具体支持如下使用场景：

- 上传单个文件
- 上传多个文件
- 上传单个文件到指定仓库目录
- 上传整个文件夹

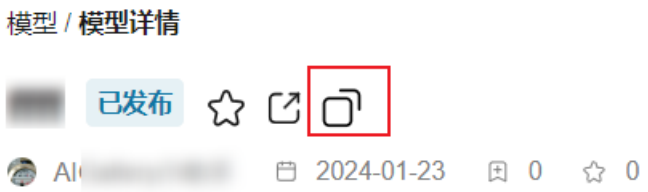
准备工作

获取“repo_id”和待上传的文件名。

- 获取“repo_id”

在AI Gallery页面的资产详情页，单击 复制完整的资产名称，如图1-10所示，获取到的信息即为“repo_id”。例如，复制出的信息为“ur5468675/test_cli_model1”，则该资产的“repo_id”为“ur5468675/test_cli_model1”。

图 1-11 复制完整资产名称



- 获取待上传的文件名

  获取待上传的文件在服务器的绝对路径。

上传单个文件

在服务器执行如下命令，可以将服务器上的文件上传到AI Gallery仓库里面。

```
gallery-cli upload {repo_id} {文件名}
```

如下所示，表示将服务器上的文件“D:\workplace\models\llama-7b\config.json”上传到AI Gallery仓库“test-cli-upload”的根目录下，当回显“100%”时表示上传完成。

```
gallery-cli upload ur5468675/test-cli-upload D:|workplace|models|llama-7b|config.json

Upload File Progress: 100%|
##################################################################################
####| 1/1 [00:00<00:00,  1.77it/s]
```

上传多个文件

在服务器执行如下命令，可以通过“--include”或“--exclude”将服务器上的某种格式的文件依次上传到AI Gallery仓库里面。

```
gallery-cli upload {repo_id} {文件目录} --include=*.json or --exclude=*.json
gallery-cli upload {repo_id} {文件目录} --exclude=*.json
```

如下所示，表示将服务器上文件目录下所有的json文件上传到AI Gallery仓库“test-cli-upload”的根目录下，当回显“100%”时表示上传完成。

```
gallery-cli upload ur5468675/test-cli-upload D:|workplace|models|llama-7b| --include=*.json

Upload File Progress: 100%|
##################################################################################
##############| 7/7 [00:03<00:00,  1.78it/s]
```

如下所示，表示将服务器上文件目录下面所有的非safetensors结尾的文件上传到AI Gallery仓库“test-cli-upload”的根目录下，当回显“100%”时表示上传完成。

```
gallery-cli upload ur5468675/test-cli-upload D:|workplace|models|llama-7b| --exclude=*.safetensors

Upload File Progress: 100%|
##################################################################################
##############| 9/9 [00:05<00:00,  1.60it/s]
```

上传单个文件到指定仓库目录

在服务器执行如下命令，可以将服务器上的文件上传到AI Gallery仓库的某个目录下面。

```
gallery-cli upload {repo_id} {文件名} {仓库目录}
```

如下所示，表示将服务器上的文件“D:\workplace\models\llama-7b\config.json”上传到AI Gallery仓库“test-cli-upload”的“model/config”目录下，当回显“100%”时表示上传完成。

```
gallery-cli upload ur5468675/test-cli-upload D:|workplace|models|llama-7b|config.json model/config

Upload File Progress: 100%|
##################################################################################
####| 1/1 [00:00<00:00,  1.77it/s]
```

上传整个文件夹

在服务器执行如下命令，可以将服务器上的文件夹上传到AI Gallery仓库里面。

```
gallery-cli upload {repo_id} {文件目录}
```

如下所示，表示将服务器上的文件夹“llama-7b”及其里面的所有文件上传到AI Gallery仓库“test-cli-upload”的仓库的根目录下，当回显“100%”时表示上传完成。

```
gallery-cli upload ur5468675/test-cli-upload D:\workplace\models\llama-7b\

Upload File Progress: 100%|
##################################################################################
##############| 7/7 [00:03<00:00,  1.78it/s]
```

1.10 计算规格说明

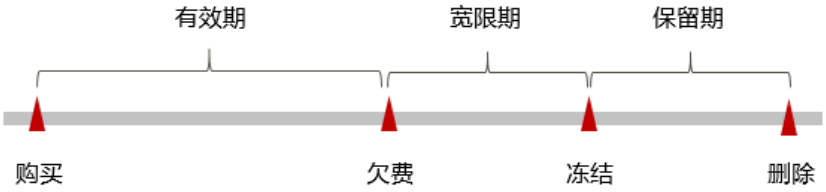
AI Gallery提供了多种计算规格供用户按需选用。只要用户的账号费用充足，就可以持续使用资源，详细计费说明请参见**计费说明**。

计费说明

AI Gallery的计费规则如**表1-34**所示。

表 1-34 计费说明

规则	说明
话单上报规则	仅当AI Gallery工具链服务创建成功且实际开始运行时，才会上报话单并开始计费，其他状态不上报就不计费，各个服务开始计费的状态如下。 • 微调大师：“训练中” • AI应用：“运行中” • 在线推理服务：“运行中”

规则	说明
计费规则	资源整点扣费，按需计费。 计费的最小单位为秒，话单上报后的每一小时对用户账号进行一次扣费。如果使用过程中暂停、终止了消耗资源的AI Gallery工具链服务，即服务不处于计费的状态中，则系统不会立即扣费，依然等到满1小时后再进行扣费，且基于当前1小时内的实际使用时长进行扣费。
实际计费规则	资源按时价扣费，真正计费的价格以实际账单为准。查看账单请参见**账单介绍**。 用户在创建AI Gallery工具链服务选择付费资源时，可以查看到付费资源的单价，在使用过程中，该资源可能由于平台的折扣优惠变化导致单价发生变化，而云服务是先使用后通过话单进行记录，计费会存在延时，因此，实际价格和折扣优惠可能与当前询价会不完全相同，请以真正计费的价格和优惠为准。
欠费说明	当用户账号余额不足造成扣费失败时，账号将变成欠费状态。 欠费后，按需资源不会立即停止服务，资源进入宽限期。如果在宽限期内仍未支付欠款，那么付费资源（如计算规格、OBS桶）、等都将被冻结，资源进入保留期。保留期的资源不支持任何操作。如果用户在宽限期内充值，则华为云会自动扣取欠费金额（含宽限期内产生的费用） 保留期到期时仍未支付欠款（含宽限期内产生的费用），则付费资源将释放，数据无法恢复。 宽限期和保留期的详细规则请参见**宽限期保留期**。 

2 AI Gallery（旧版）

2.1 AI Gallery 简介

AI Gallery算法、镜像、模型、Workflow等AI数字资产的共享，为高校科研机构、AI应用开发商、解决方案集成商、企业级/个人开发者等群体，提供安全、开放的共享及交易环节，加速AI资产的开发与落地，保障AI开发生态链上各参与方高效地实现各自的商业价值。

资产集市介绍

AI Gallery中，“资产集市”支持Notebook代码样例、数据集、算法、镜像、模型、Workflow等AI资产的共享。

- “资产集市 > Notebook”：共享了Notebook代码样例。

  AI Gallery的Notebook模块为开发者提供免费分享和灵活使用Notebook代码样例的功能。您可以将优秀的Notebook代码样例发布在AI Gallery社区，供其他开发者学习使用；也可以在AI Gallery上查看其他人共享的Notebook案例的详细描述、代码信息等，通过“Run in ModelArts”将Notebook案例在ModelArts控制台快速打开、运行以及进行二次开发等操作。

- “资产集市 > 数据集”：共享了数据集。

  AI Gallery的数据模块支持数据集的共享和下载。在AI Gallery的“数据”中，可以查找并下载满足业务需要的数据集。也可以将自己本地的数据集发布至AI Gallery中，共享给其他用户使用。

- “资产集市 > 算法”：共享了算法。

  AI Gallery的算法模块支持算法的共享和订阅。在AI Gallery的“算法”中，可以查找您想要的算法，订阅满足业务需要的资产，最后推送至ModelArts控制台使用。也可以将个人开发的算法分享发布至AI Gallery中，共享给其他用户使用。

- “资产集市 > 镜像”：共享了ModelArts镜像。

  AI Gallery的ModelArts镜像支持发布和使用共享的镜像。在AI Gallery的“镜像”中，可以查找您想要的ModelArts镜像，最后在ModelArts管理控制台使用。也可以将个人开发的ModelArts镜像分享发布至AI Gallery中，共享给其他用户使用。

- “资产集市 > 模型”：共享了ModelArts模型和HiLens技能。

  AI Gallery的模型模块包括ModelArts模型和HiLens技能，支持发布和订阅共享的模型。在AI Gallery的“模型”中，可以查找您想要的ModelArts模型或HiLens技

能，订阅满足业务需要的资产，最后推送至ModelArts或HiLens等管理控制台使用。也可以将个人开发的ModelArts模型或HiLens技能分享发布至AI Gallery中，共享给其他用户使用。其中，HiLens技能为HiLens服务的技能市场功能，详细指导请参见《**HiLens用户指南**》。

- “资产集市 > Workflow”：共享了Workflow。

  AI Gallery的Workflow模块支持Workflow的共享和订阅。在AI Gallery的Workflow中，可以查找您想要的Workflow，订阅满足业务需要的资产，最后推送至ModelArts控制台使用。也可以将个人开发的Workflow分享发布至AI Gallery中，共享给其他用户使用。

AI 说介绍

AI Gallery的AI说模块为开发者提供自由分享各类AI领域内知识和经验的平台。开发者既可以发布个人技术文章，也可以阅读和学习他人分享的技术文章。

案例库介绍

AI Gallery的案例库是面向场景化交付的AI资产的组合和使用案例。案例中沉淀了基于业务场景的AI知识、经验和部分通用的业务逻辑，能够为某些具体的业务场景提供AI环节的解决方案。

说明

AI案例的发布功能即将上线，当前只支持订阅使用。

生态合作介绍

AI Gallery的生态合作模块展示了伙伴赋能培训，该模块旨在与合作伙伴一起构建合作共赢的AI生态体系。

AI Gallery 使用限制

- 目前自动学习产生的模型暂不支持发布到AI Gallery。
- 订阅或购买主要是获取AI资产的使用配额和使用权，支持在配额定义的约束下，有限地使用AI资产。
- 使用AI资产时，可能需要消耗硬件资源，硬件资源费用将根据实际使用情况，由华为云ModelArts等管理控制台向使用方收取。
- 已发布的AI资产，如果不需要在资产列表中展示该资产，可以将资产下架。下架后，已发布资产仅发布者可见。已经被订阅的资产，即便资产下架后，基于配额资源的约束，仍然可有效使用该资产，不会因为该资产的下架而产生使用问题。

2.2 免费资产和商用资产

AI Gallery既有免费分享的AI资产，也有商业售卖的AI资产。

- **免费资产**无需支付费用，只需要支付在使用过程中消耗的硬件资源，硬件资源费用将根据实际使用情况由华为云ModelArts等管理控制台向使用方收取。

  当前支持免费分享和订阅的资产类型有：Notebook代码样例、数据集、算法、模型、镜像。

- **商用资产**由华为云云商店提供卖家发布和买家购买相关功能，AI Gallery仅提供列表展示。购买商业售卖的AI资产，本质上是购买算法、模型等AI资产的使用配额，在配额定义的约束下，有限地使用算法、模型等。

  卖家发布AI类资产操作请参考**发布AI资产类商品操作指导**。买家购买相关功能请参考**商品购买**。

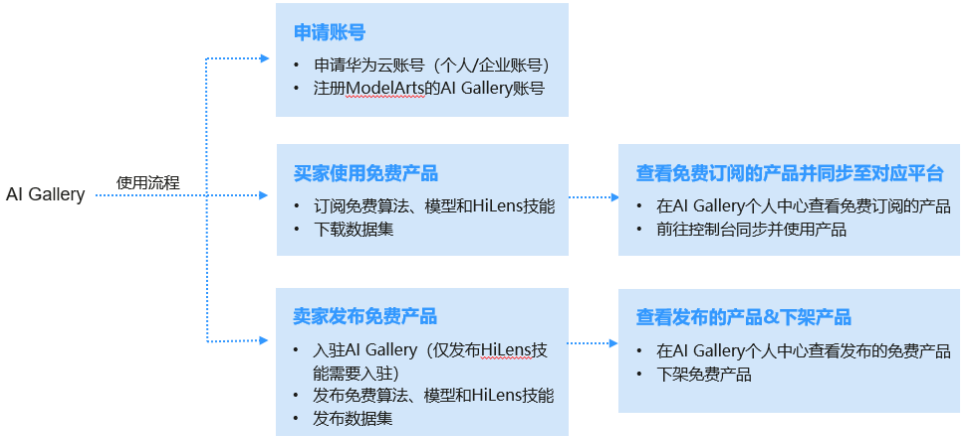
  当前支持商业售卖的资产类型：算法、模型

免费资产使用事项

- 订阅和发布免费资产需要您按照指导**注册华为帐号并开通华为云**；发布HiLens技能除了需要注册华为账号并开通华为云还需要**入驻AI Gallery**。
- 发布的免费资产将展示在AI Gallery的公共页签以及“我的Gallery > 我的资产”的各个模块的“我的发布”中。
- 已经订阅的免费资产将展示在AI Gallery的“我的Gallery > 我的资产”的各个模块的“我的订阅”或“我的下载”中。

免费资产在ModelArts的AI Gallery中的使用流程如下：

图 2-1 AI Gallery 中免费资产的使用流程

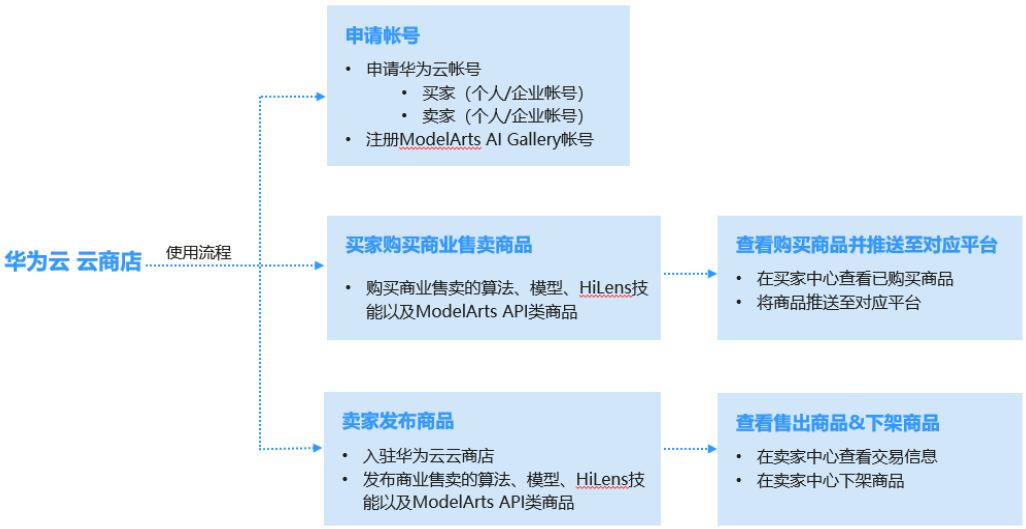


商业售卖商品使用事项

- 售出和购买商品需要您按照指导**注册华为帐号并开通华为云**。如果您是卖家则需要完成实名认证并进行**华为云云商店入驻**。
- 如果您是卖家，您可以在云商店查看自己售卖的商品是否上架成功。在云商店右上方单击“卖家中心 > 进入卖家中心”，选择“商品管理 > 我的商品”查看在售卖的商品。
- 如果您是买家，您可以在云商店搜索、购买商品，已经购买的商品将展示在“云商店 > 买家中心 > 已购买的服务”页面。
- 在AI Gallery内订阅的免费商品只展示在AI Gallery的“我的订阅”中，不会展示在AI云商店的“买家中心”中。
- 云商店当前付费商品默认发布后是隐藏商品，在Gallery首页将不可见，只有在云商店卖家中心改变商品为非隐藏，Gallery首页付费资产列表才对该商品可见。

更多关于商业售卖商品的使用指导请参见**《云商店用户指南》**，商业售卖商品在华为云云商店的使用流程如下：

图 2-2 华为云云商店商业商品



2.3 入驻 AI Gallery

如果需要在AI Gallery中发布HiLens、报名实践活动或发布AI说，则需要先完成入驻AI Gallery。

1. 如果没有入驻过AI Gallery，在报名实践活动或发布AI说时，将跳转至“欢迎入驻AI Gallery”页面。
2. 在“欢迎入驻AI Gallery”页面，填写“昵称”和“邮箱”，并根据提示获取验证码。阅读并同意《华为云AI Gallery数字内容发布协议》和《华为云AI Gallery服务协议》后，单击“确定”完成入驻。

图 2-3 入驻 AI Gallery

3. 注册完成后，您可以在AI Gallery中报名实践活动或发布技术文章（AI说）。

2.4 我的 Gallery 介绍

“我的Gallery”可以查看各类AI资产的发布订阅情况和个人资料等。

在“AI Gallery”页面中，单击右上角“我的Gallery > 我的主页”进入个人中心页面。

图 2-4 进入我的 Gallery

表 2-1 我的 Gallery 列表介绍

模块列表	功能介绍
我的主页	展示个人的成长值数据。 成长值可以通过“签到”和发布资产获取，每天只能签到一次。 **说明** 成长值相关数据和功能当前是Beta版本，在正式版本发布前可能会发生变化。
我的资产 > 算法	展示个人发布和订阅的算法列表。 ● “我的发布”：可以查看个人发布的算法信息，如浏览量、收藏量、订阅量等。通过右侧的“上架”、“下架”或“删除”可以管理已发布的算法。资产下架后，已订阅该资产的用户可继续正常使用，其他用户将无法查看和订阅该资产。下架后的资产可以重新上架。资产未被订阅时可以删除资产。 ● “我的订阅”：可以查看个人订阅的算法信息，如发布者、应用控制台、剩余配额等。通过右侧的“取消订阅”或“找回订阅”可以管理已订阅的算法。取消订阅后，ModelArts管理控制台算法管理模块-我的订阅列表中将不再展示该算法。已取消订阅的算法可以找回订阅，并在原配额约束下可以继续使用该算法。
我的资产 > 模型	展示个人发布和订阅的模型列表，包括ModelArts模型和HiLens技能。 ● “我的发布”：可以查看个人发布的模型信息，如浏览量、收藏量、订阅量等。通过右侧的“上架”、“下架”或“删除”可以管理已发布的模型。资产下架后，已订阅该资产的用户可继续正常使用，其他用户将无法查看和订阅该资产。下架后的资产可以重新上架。资产未被订阅时可以删除资产。 ● “我的订阅”：可以查看个人订阅的模型信息，如发布者、应用控制台、剩余配额等。通过右侧的“取消订阅”或“找回订阅”可以管理已订阅的ModelArts模型。取消订阅后，在ModelArts管理控制台“模型管理 > 模型 > 我的订阅”列表中，将不再展示该模型。已取消订阅的模型可以找回订阅，并在原配额约束下可以继续使用该模型。
我的资产 > 数据	展示个人发布和下载的数据集列表。 ● “我的发布”：可以查看个人发布的数据集信息，如文件大小、文件数量等。通过右侧的“重试”或“删除”可以管理已发布的数据集。 ● “我的下载”：可以查看个人下载的数据集信息。单击下拉三角，可以查看数据集ID、下载方式、目标区域等信息。
我的资产 > Notebook	展示个人发布的Notebook实例列表。 ● “我的发布”：可以查看实例浏览量、收藏量、订阅量等信息。通过右侧的“重试”或“删除”可以管理已发布的Notebook。 ● “我的运行”：可以查看个人运行的Notebook记录。

模块列表	功能介绍
我的资产 > Workflow	展示个人发布和订阅的Workflow列表。 ● **“我的发布”**：可以查看个人发布的Workflow信息，浏览量、收藏量、订阅量等。通过右侧的“上架”，“下架”或“删除”可以管理已发布的Workflow。 ● **“我的订阅”**：可以查看个人订阅的Workflow信息。通过右侧“取消订阅”或“找回订阅”可以管理已订阅的资产。
我的案例	展示个人发布的资产案例和已订阅的资产案例。 ● “我的发布”：可以查看个人发布的案例信息。 ● “我的订阅”：可以查看个人订阅的案例信息。
我的AI说	展示个人发布的技术文章列表，可以查看文章浏览量、收藏量、订阅量等信息。通过右侧的“删除”可以管理已发布的技术文章。
我的实践	展示个人报名参加的实践活动列表。
合作伙伴	展示伙伴的申请信息以及伙伴在Gallery中的申请状态。
解决方案	展示伙伴发布的解决方案列表。
我的需求	展示个人发布的需求列表。
我的导出	展示个人导出的资产列表。只有以管理员账号登录才会显示此模块。
我的资料	查看个人基本信息，包括“账号”、“头像”、“昵称”、“邮箱”、“简介”等信息。 ● 单击“编辑资料”，可以编辑“昵称”和“简介”。 ● 单击“更换头像”，可以自定义替换头像。

2.5 订阅使用

2.5.1 查找和收藏资产

AI Gallery共享了算法、Notebook代码样例、数据集、镜像、模型、Workflow等多种AI资产，为了方便快速搜索相关资产，提供了多种快速搜索方式以及收藏功能，提升资产的查找效率。

搜索资产

在各类资产模块页面，通过如下几种搜索方式可以提高资产的查找效率，快速找到适合的算法、模型、数据集、镜像、Workflow等资产。

图 2-5 搜索资产

表 2-2 快速搜索方式

区域	类型	搜索方式	支持的AI资产
1	搜索华为云官方资产	在页面单击“官方”，筛选出所有的华为云官方资产，该类资产均可**免费**使用。	Notebook、算法、模型
2	搜索精选商品	在页面单击“精选”，筛选出所有被标记为精选的资产。	Notebook、数据、算法、模型、Workflow
3	按标签搜索	在页面单击“所有标签”，选择标签，单击“确定”，筛选出相关资产。	Notebook、数据、算法、镜像、模型、Workflow
4	按排序方式搜索	在页面的排序列表选择排序方式，调整资产排序方式快速查找所需资产。	Notebook、数据、算法、镜像、模型、Workflow
5	搜索商用资产	在页面单击“商用”，筛选出所有的**商业**售卖资产。	算法、模型

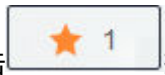
收藏免费资产

当搜索到感兴趣的免费资产时，可以收藏该资产，方便后续在“我的收藏”快速查找。商用资产如需收藏请前往云商店。

1. 单击目标资产，进入资产详情页面。
2. 在资产详情页面，单击 

按钮收藏资产。
   收藏成功后，在各个模块的“我的收藏”页签可以快速查看收藏的资产。

图 2-6 查看收藏的资产

3. （可选）如果需要取消收藏，再次单击 

按钮即可。

2.5.2 订阅免费算法

在AI Gallery中，您可以查找并订阅免费满足业务需要的算法，直接用于创建训练作业。

说明

AI Gallery中分享的算法支持免费订阅，但在使用过程中如果消耗了硬件资源进行部署，管理控制台将根据实际使用情况收取硬件资源的费用。

前提条件

注册并登录华为云，且创建好OBS桶用于存储数据和模型。

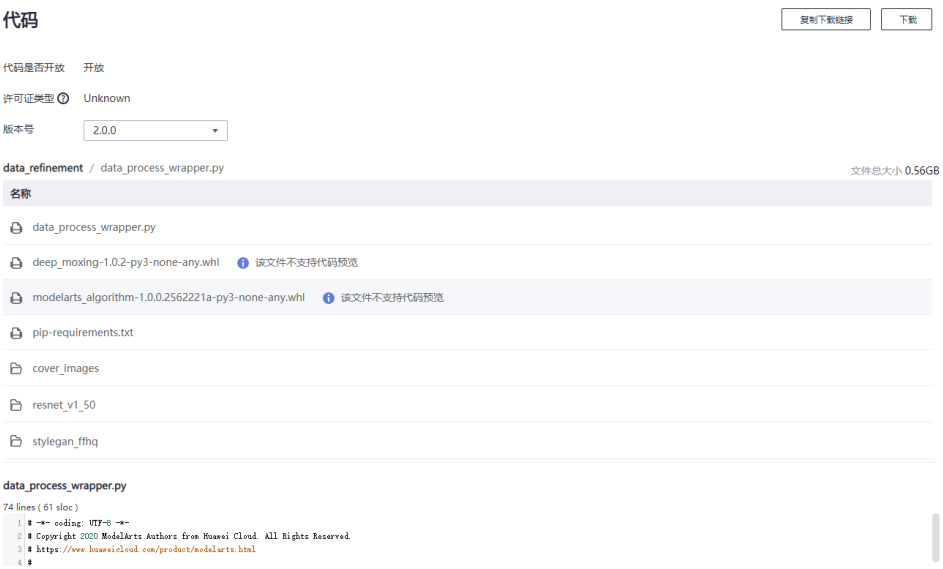
订阅算法

1. 登录“AI Gallery”。
2. 选择“资产集市 > 算法”，进入算法页面，该页面展示了所有共享的算法。
3. 搜索业务所需的算法，请参见**查找资产**。
4. 单击目标算法进入详情页面。
   - 在详情页面您可以查看算法的“描述”、“交付、”“限制”、“版本”、“关联资产”和“评论”等信息。
   - 为方便您的使用，在订阅算法时，建议您查看算法详情页“版本”页签中关于算法对应版本的“使用约束”，准备对应的数据和资源规格后进行使用。
   - 对于开放代码的算法，您也可以在详情页面预览或者下载对应代码。

     在“代码”页签，单击右侧的“下载”将完整代码下载到本地，您也可以单击下方列表中的文件名称进行预览。

     目前如下后缀结尾的文件类型支持代码预览：txt、py、h、xml、html、c、properties、yml、cmake、sh、css、js、cpp、json、md、sql、bat、conf

图 2-7 下载预览代码



5. 在详情页面单击“订阅”，根据算法是否具有使用约束进行不同操作：
   - 如果订阅是具有使用约束的算法，则弹出“使用约束”页面，查看并确认后单击“继续订阅”即可成功订阅。
   - 如果订阅是没有使用约束的算法，则直接成功订阅。

   说明

   如果订阅的是非华为云官方资产，则会弹出“温馨提示”页面，勾选并阅读《数据安全与隐私风险承担条款》和《华为云AI Gallery服务协议》后，单击“继续订阅”才能继续进行算法订阅。

   算法被订阅后，详情页的“订阅”按钮显示为“已订阅”，订阅成功的资产也会展示在“我的Gallery > 我的资产 > 算法 > 我的订阅”。

使用算法

1. 订阅成功的算法可在ModelArts管理控制台使用，如创建训练作业等。

   方式一：从算法详情页进入管理控制台

   a. 在算法详情页单击“前往控制台”。
   b. 在弹出的“选择云服务区域”页面选择ModelArts所在的云服务区域，单击“确定”跳转至ModelArts控制台的“算法管理 > 我的订阅”页面。

   方式二：从“我的Gallery”进入管理控制台

   a. 在AI Gallery，单击右上角“我的Gallery > 我的资产 > 算法”，进入“我的算法”页面。
   b. 选择“我的订阅”页签，进入个人订阅的算法列表。
   c. 在算法列表选择需要使用的算法，单击“应用控制台”列的“ModelArts”。
   d. 在弹出的“选择云服务区域”页面选择ModelArts所在的云服务区域，单击“确定”跳转至ModelArts控制台的“算法管理 > 我的订阅”页面。
2. 在“算法管理 > 我的订阅”页面，选择并展开订阅的目标算法。在版本列表中，单击“创建训练作业”跳转至创建训练作业页面。

取消或找回订阅的算法

当不需要使用AI Gallery中订阅的算法时，可以取消订阅该算法。取消订阅后，ModelArts管理控制台“算法管理 > 我的订阅”列表中将不再展示该算法；当需要再次使用该算法时，可以找回订阅，ModelArts管理控制台“算法管理 > 我的订阅”列表中也会再次展示该算法。

1. 在AI Gallery，单击右上角“我的Gallery > 我的资产 > 算法”，进入“我的算法”页面。
2. 选择“我的订阅”页签，进入个人订阅的算法列表。
   - **取消订阅**：仅已订阅的资产支持取消。

     单击目标资产右侧的“取消订阅”，在弹框中确认资产信息，单击“确定”取消订阅。
   - **找回订阅**：仅订阅后被取消订阅的资产支持找回。

     单击标资产右侧的“找回订阅”完成找回。

图 2-8 取消或找回订阅

2.5.3 订阅免费模型

在AI Gallery中，您可以查找并订阅免费的模型，包括ModelArts模型和HiLens技能。订阅成功的模型可以直接用于ModelArts模型部署和HiLens技能安装。

说明

AI Gallery中分享的模型支持免费订阅，但在使用过程中如果消耗了硬件资源进行部署，管理控制台将根据实际使用情况收取硬件资源的费用。

前提条件

注册并登录华为云，且创建好OBS桶用于存储数据和模型。

如果是订阅使用HiLens技能，则需要获取相关服务权限，详细操作请参见**准备工作（华为HiLens）**。

订阅免费模型

1. 登录“AI Gallery”。
2. 选择“资产集市 > 模型”，进入模型页面，该页面展示了所有共享的模型，包括ModelArts模型和HiLens技能。
3. 搜索业务所需的免费模型，请参见**查找资产**。
4. 单击目标模型进入详情页面。

   在详情页面您可以查看模型的“描述”、“交付”、“限制”、“版本”和“评论”等信息。
5. 在详情页面单击“订阅”。

   说明

   如果订阅的是非华为云官方资产，则会弹出“温馨提示”页面，勾选并阅读《数据安全与隐私风险承担条款》和《华为云AI Gallery服务协议》后，单击“继续订阅”才能继续进行模型订阅。

   模型被订阅后，详情页的“订阅”按钮显示为“已订阅”，订阅成功的资产也会展示在“我的Gallery > 我的资产 > 模型 > 我的订阅”。

使用免费模型

订阅成功的模型可在ModelArts或HiLens等管理控制台使用，支持模型部署或安装等。

1. 将订阅成功的模型推送至应用控制台。

   方式一：从模型详情页进入管理控制台

   在模型详情页单击“前往控制台”。

- 如果订阅的是ModelArts模型，在弹出的“选择云服务区域”页面选择ModelArts所在的云服务区域，单击“确定”跳转至ModelArts控制台的“AI应用管理 > AI应用 > 我的订阅”页面。

  模型对应版本列表的状态显示为“就绪”表示可以使用。

图 2-9 推送免费模型

- 如果订阅的是HiLens技能，在弹出的“选择云服务区域”页面选择HiLens所在的云服务区域，单击“确定”跳转至HiLens控制台的“产品订购 > 订单管理 > AI Gallery”页面。该HiLens技能自动同步至HiLens。

方式二：从“我的Gallery”进入管理控制台

a. 在AI Gallery，单击右上角“我的Gallery > 我的资产 > 模型”，进入“我的模型”页面。
b. 选择“我的订阅”页签，进入个人订阅的模型列表。
c. 在模型列表选择需要推送的模型，单击“应用控制台”列的服务名称将模型推送至不同应用控制台。

图 2-10 选择应用控制台

- 如果订阅的是ModelArts模型，在弹出的“选择云服务区域”页面选择ModelArts所在的云服务区域，单击“确定”跳转至ModelArts控制台的“AI应用管理 > AI应用 > 我的订阅”页面。

  模型对应版本列表的状态显示为“就绪”表示可以使用。

图 2-11 推送免费模型

- 如果订阅的是HiLens技能，在弹出的“选择云服务区域”页面选择HiLens所在的云服务区域，单击“确定”跳转至HiLens控制台的“产品订购 > 订单管理 > AI Gallery”页面。该HiLens技能自动同步至HiLens。

2. 在应用控制台使用订阅的模型。
   - 使用订阅的ModelArts模型：

     在“AI应用管理 > AI应用 > 我的订阅”页面，选择并展开订阅的目标模型。在版本列表单击“部署”，可以将订阅的ModelArts模型部署为“在线服务”、“批量服务”或“边缘服务”，详细操作步骤请参见**部署服务**。
   - 使用订阅的HiLens技能：

     在“产品订购 > 订单管理 > AI Gallery”页面，单击技能名称左侧，在技能的版本信息单击“安装”即可安装技能至设备上使用，详情请参见**安装技能**。

图 2-12 安装技能-40

取消或找回订阅的免费模型

当不需要使用AI Gallery中订阅的模型时，可以取消订阅该模型。取消订阅后，ModelArts管理控制台“AI应用管理 > AI应用 > 我的订阅”列表中将不再展示该模型；当需要再次使用该模型时，可以找回订阅，ModelArts管理控制台“AI应用管理 > AI应用 > 我的订阅”列表中也会再次展示该模型。

说明

HiLens技能不支持取消订阅。

1. 在AI Gallery，单击右上角“我的Gallery > 我的资产 > 模型”，进入“我的模型”页面。
2. 选择“我的订阅”页签，进入个人订阅的模型列表。
   - **取消订阅**：仅已订阅的资产支持取消。
     单击目标资产右侧的“取消订阅”，在弹框中确认资产信息，单击“确定”取消订阅。
   - **找回订阅**：仅订阅后被取消订阅的资产支持找回。
     单击标资产右侧的“找回订阅”完成找回。

2.5.4 下载数据

在AI Gallery中，您可以下载满足业务需要的数据集。

前提条件

注册并登录华为云，且创建好OBS桶用于存储数据。

下载数据集

1. 登录“AI Gallery”。
2. 选择“资产集市 > 数据集”，进入数据页面，该页面展示了所有共享的数据集。
3. 搜索业务所需的数据集，请参见查找和收藏资产。
4. 单击目标数据集进入详情页面。
   在详情页面可以查看数据集的“描述”、“预览”、“限制”、“版本”和“评论”等信息。
5. 在详情页面单击“下载”。弹出“选择云服务区域”，选择区域后单击“确定”进入下载详情页面。根据数据集下载至OBS还是ModelArts数据集列表，填写不同配置信息：

   说明

   ModelArts数据管理模块在重构升级中，对未使用过数据管理的用户不可见。建议新用户选择将数据集下载至OBS使用。

   - **将数据集下载至OBS**
     - “下载方式”选择“对象存储服务（OBS）”。
     - “目标区域”选择您需要将该数据集下载到的区域位置，如“华北-北京四”。
     - “目标位置”选择OBS桶路径，桶内如有同名的文件或文件夹，将被新下载的文件或文件夹覆盖。

图 2-13 下载数据集（至 OBS）

- **将数据集下载至ModelArts**
  - “下载方式”：选择“ModelArts数据集”。
  - “目标区域”：选择您需要将该数据集下载到的区域位置，如“华北-北京四”。
  - “数据类型”：选择需要处理的文件类型。数据类型更多信息请参考**数据集的类型**。
  - “数据集输出位置”：数据集输出位置的OBS路径，此位置会存放输出的标注信息等文件，此位置不能和OBS数据源中的文件路径相同或为其子目录。
  - “数据集输入位置”：AI Gallery的数据集下载到OBS的路径，此位置会作为数据集的数据存储路径，数据集输入位置不能和输出位置相同。
  - “名称”默认生成“data-xxxx”形式的数据集名称，该数据集将同步在ModelArts数据集列表中。
  - “描述”可以添加对于该数据集的相关描述。

图 2-14 下载数据集（至 ModelArts）

6. 单击“确定”，跳转至“我的数据 > 我的下载”页面。

   说明

   下载的数据集在AI Gallery“我的数据 > 我的下载”不会立即显示，需要刷新该页面才能看到新下载的数据集。

在 Notebook 中使用数据集

1. 登录“AI Gallery”。
2. 选择“资产集市 > 数据集”，进入数据页面，该页面展示了所有共享的数据集。
3. 搜索业务所需的数据集，请参见**查找和收藏资产**。
4. 单击目标数据集进入详情页面。

   在详情页面查看数据集的“描述”、“版本”和“限制”等信息。
5. 在详情页面单击“Run in ModelArts”，跳转到ModelArts控制台并自动创建Notebook，进入Notebook实例的JupyterLab页面。

   参考**使用JupyterLab**在JupyterLab页面进行开发调试。

2.5.5 使用 Notebook 代码样例

在AI Gallery中，您可以查找并直接打开使用Notebook实例。

前提条件

注册并登录华为云，详细操作请参见**准备工作**。

打开 Notebook 实例

1. 登录“AI Gallery”。
2. 选择“资产集市 > Notebook”，进入Notebook页面，该页面展示了所有共享的Notebook实例。
3. 搜索业务所需的Notebook实例，请参见**查找和收藏资产**。
4. 单击目标Notebook实例进入详情页面。

   在详情页面可以查看Notebook实例的“描述”、“限制”和“版本”等信息。
5. 在详情页面单击“Run in ModelArts”，跳转到ModelArts控制台并直接进入Notebook实例的JupyterLab页面。

   参考**使用JupyterLab**在JupyterLab页面进行开发调试。

2.5.6 使用镜像

在AI Gallery中，您可以查找共享的镜像并用于AI开发。

使用镜像

1. 登录“AI Gallery”。
2. 选择“资产集市 > 镜像”，进入镜像页面，该页面展示了所有共享的镜像。
3. 搜索业务所需的镜像，请参见**查找和收藏资产**。
4. 单击目标镜像进入详情页面。

   在详情页面您可以查看镜像的AI引擎框架、使用芯片、镜像URL、包含的依赖项等信息。
5. 复制镜像URL，可以在ModelArts控制台“镜像管理”注册并使用该镜像。

2.5.7 使用 AI 案例

在AI Gallery中，您可以根据您的业务场景和诉求，查找并订阅相应的场景化AI案例。订阅后可以一键运行案例。

说明

AI Gallery中分享的案例支持免费订阅，但在使用过程中如果消耗了硬件资源进行部署，管理控制台将根据实际使用情况收取硬件资源的费用。

前提条件

注册并登录华为云，且创建好OBS桶用于存储数据和模型。

订阅并使用 AI 案例

1. 登录“AI Gallery”。
2. 选择“案例库”，在下拉框中单击“案例库 >”，进入AI案例库首页，该页面展示了所有共享的案例。
3. 根据业务场景搜索所需的免费案例，单击案例进入详情页面。

   在详情页面您可以查看案例的“使用说明”、“关联资产”、“输出样例”、“体验Demo”和“评论”等信息。

> **说明**
>
> 部分案例可能发布者未提供“关联资产”、“输出样例”或“体验Demo”。

4. 在详情页面单击“订阅”。

   案例被订阅后，详情页的“订阅”按钮显示为“已订阅”，订阅成功的资产也会展示在“我的Gallery > 我的案例 > 我的订阅”。

5. 订阅成功后，单击“Run in ModelArts”跳转到管理控制台使用案例。

2.5.8 订阅 Workflow

在AI Gallery中，您可以查找并订阅免费的Workflow。订阅成功的Workflow通过AI Gallery导入后可以直接在ModelArts控制台使用。

> **说明**
>
> AI Gallery中分享的Workflow支持免费订阅，但在使用过程中如果消耗了硬件资源进行部署，管理控制台将根据实际使用情况收取硬件资源的费用。

前提条件

注册并登录华为云，且创建好OBS桶用于存储算法和Workflow。

订阅免费 Workflow

1. 登录“AI Gallery”。
2. 选择“资产集市 > MLOps > Workflow”，进入Workflow页面，该页面展示了所有共享的Workflow。
3. 搜索业务所需的免费Workflow，请参见**查找和收藏资产**。
4. 单击目标Workflow进入详情页面。

   在详情页面您可以查看Workflow的“描述”、“交付”、“版本”、“限制”和“评论”等信息。

5. 在详情页面单击“订阅”。

   > **说明**
   >
   > 如果订阅的是非华为云官方资产，则会弹出“温馨提示”页面，勾选并阅读《数据安全与隐私风险承担条款》和《华为云AI Gallery服务协议》后，单击“继续订阅”才能继续进行模型订阅。

   Workflow被订阅后，详情页的“订阅”按钮显示为“已订阅”，订阅成功的资产也会展示在“我的Gallery > 我的资产 > Workflow > 我的订阅”。

使用免费 Workflow

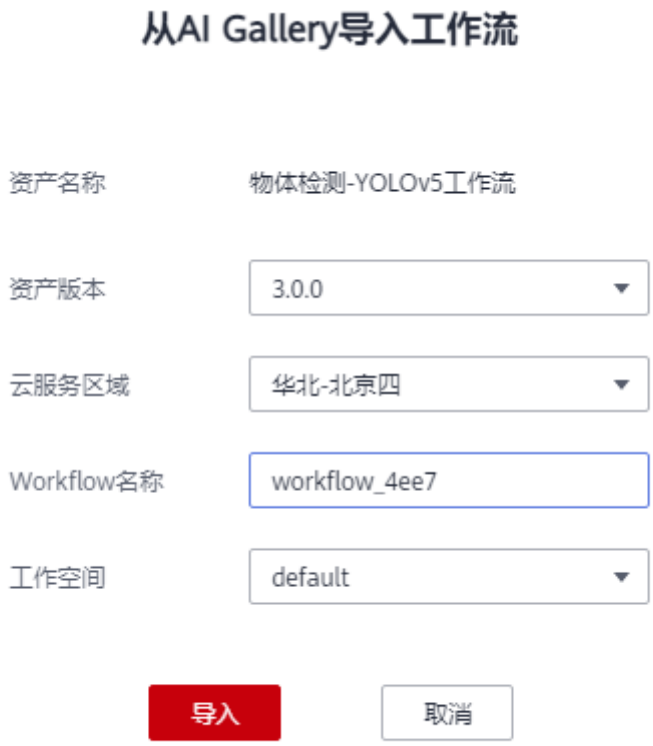
订阅成功的Workflow可在ModelArts管理控制台使用，支持导入工作流。

1. 将订阅成功的Workflow导入至ModelArts控制台。

   方式一：从Workflow详情页进入ModelArts控制台

   在Workflow详情页单击“运行”，在弹出来的对话框中选择、填写**图2-15**所示信息，单击“导入”跳转至ModelArts控制台的Workflow的详情页。

图 2-15 导入免费 Workflow



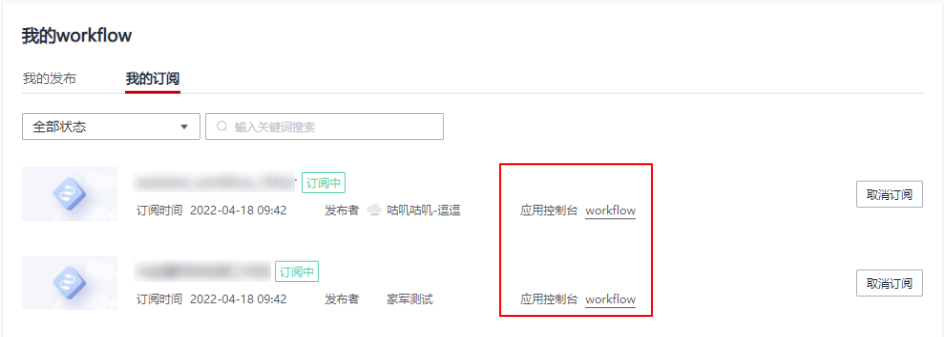
方式二：从“我的Gallery”进入ModelArts控制台

a. 在AI Gallery，单击右上角“我的Gallery > 我的资产 > Workflow”，进入“我的Workflow”页面。

b. 单击“我的订阅”，进入个人订阅的Workflow列表。

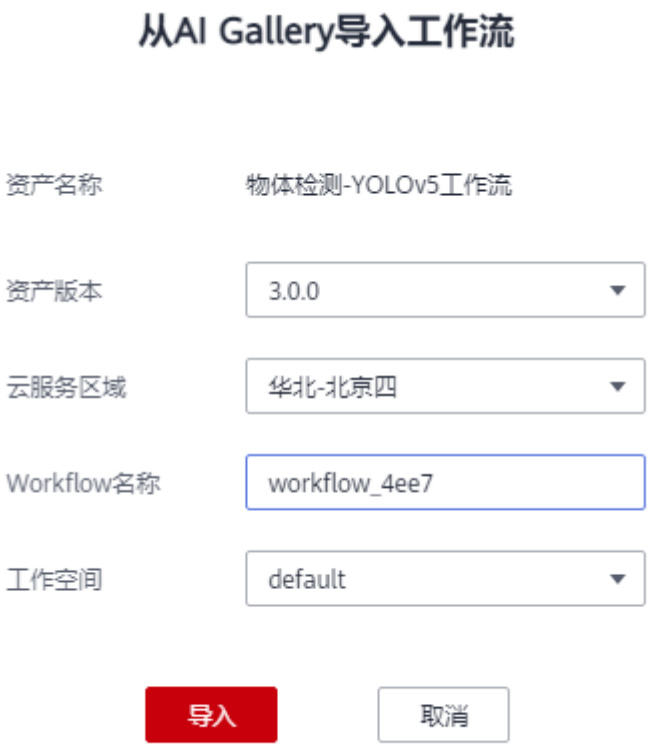
c. 在“我的订阅”列表，选择需要导入的Workflow，单击“应用控制台”旁的“Workflow”。

图 2-16 选择应用控制台



d. 在弹出来的对话框中选择、填写图2-17所示信息，单击“导入”跳转至ModelArts控制台的Workflow的详情页。

图 2-17 导入免费 Workflow



2. 在ModelArts控制台使用从Gallery导入的Workflow。

   在ModelArts控制台左侧导航栏，单击Workflow(Beta)。在Workflow列表中，找到从Gallery导入的Workflow，单击“配置”进入到该Workflow。

取消或找回已订阅的 Workflow

当不需要使用AI Gallery中订阅的Workflow时，可以取消订阅该Workflow。当需要再次使用该Workflow时，可以通过“找回订阅”恢复已取消的订阅。

2.6 发布分享

2.6.1 发布免费算法

在AI Gallery中，您可以将个人开发的算法免费分享给他人使用。

前提条件

- 在ModelArts的算法管理中已准备好待发布的算法。创建算法的相关操作请参见**创建算法**。

  说明

  创建算法时，算法代码存储的OBS桶内不能存在文件和文件夹重名的情况，这样算法可能会发布失败。如果算法发布成功，则**代码开放**会失败。

发布算法

1. 进入AI Gallery首页，选择“资产集市 > 算法”，进入算法页面。
2. 单击“发布”，弹出“选择云服务区域”，选择区域后单击“确定”跳转到“发布资产到AI Gallery”页面。
3. 在发布资产页面，填写相关信息，发布资产。
   - 如果是发布新资产。
     i. “发布方式”选择“创建新资产”。
     ii. 填写“资产标题”。即在AI Gallery显示的资产名称。
     iii. “来源”默认为“ModelArts”。
     iv. 选择“ModelArts区域”。设置可以使用该资产的ModelArts区域，以控制台实际可选值为准。
     v. 单击“算法名称”右侧的“选择”，从ModelArts算法管理中选择待发布的算法，单击“确认”。
     vi. 填写“资产版本”。版本号格式为“x.x.x”。
     vii. 设置“谁可以看”。
     设置资产的公开权限。可选值有：
     - “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。
     - “指定用户”：表示仅特定用户可以查看及使用该资产。
     - “仅自己可见”：表示只有当前账号可以查看并使用该资产。

     viii. 设置“时长限制”。
     设置订阅者可以免费使用资产的时长，默认关闭，即无限期使用。如果打开时长限制，除了设置资产免费使用的时长，还可以设置到期后是否续订。
   - 如果是更新已发布资产的版本。
     i. “发布方式”选择“添加资产版本”。
     ii. 在“资产标题”下拉框中选择已有资产名称。
     iii. “来源”默认为“ModelArts”。
     iv. 选择“ModelArts区域”。
     v. 单击“算法名称”右侧的“选择”，从ModelArts算法管理中选择需要添加版本号的算法，单击“确认”。
     vi. 在“资产版本”填写新的版本号。
4. 阅读并同意《华为云AI Gallery数字内容发布协议》和《华为云AI Gallery服务协议》。
5. 单击“发布”。

说明

发布使用容器镜像导入的资产时，后台会进行资产安全扫描，如果扫描发现资产有问题，则资产发布失败并邮件通知发布者。

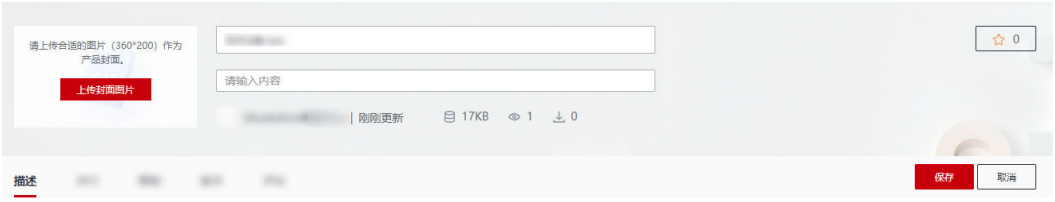
编辑资产详情

资产发布成功后，发布者可以进入详情页修改该资产的标题、封面图、描述等，让资产更吸引人。

修改封面图和二级标题

1. 在发布的资产详情页面，单击右侧的“编辑”，选择上传新的封面图，为资产编辑独特的主副标题。
2. 编辑完成之后单击“保存”。封面图和二级标题内容自动同步，您可以直接在资产详情页查看修改结果。

图 2-18 修改封面图和二级标题



编辑标签

1. 单击标签右侧的出现标签编辑框，在下拉框中勾选该资产对应的标签。
2. 单击编辑框右侧的对勾完成编辑。
   保存成功的标签信息会在资产搜索页成为过滤分类条件。

图 2-19 添加标签

编辑描述

1. 单击右侧的“编辑”，在编辑框中输入资产的描述内容，包含但不局限于背景、简介、使用方法、约束条件等。支持发布者以Markdown形式自由编辑。
2. 编辑完成之后单击“保存”。

编辑限制

支持修改资产的公开权限和时长限制。

1. 选择“限制”页签，单击右上方的“编辑”进入编辑模式：
   - 在“谁可以看”右侧的下拉框中选择公开权限。
     - “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。
     - “指定用户”：表示仅特定用户可以查看及使用该资产。
     - “仅自己可见”：表示只有当前账号可以查看并使用该资产。

     说明

     公开权限只支持权限的扩大，权限从小到大为“仅自己可见<指定用户<公开”。
   - “时长限制”可以选择“不启用”或“启用”。当启用时，可以设置资产的免费使用时长，以及到期后是否续订。
2. 单击“保存”，完成修改。

图 2-20 编辑限制

编辑版本

1. 选择“版本”页签，单击右上方的“编辑”。
2. 在此页面可以修改版本说明或者单击对应版本“操作”列的“下线”，下架不需要的资产版本。下线操作仅对已上架成功且存在多个可用版本的资产有效。
3. 在版本框右侧单击“添加版本”，弹出“选择云服务区域”，选择区域后单击“确定”跳转到“发布资产到AI Gallery”页面，参考**更新已发布资产的版本**添加资产版本。
4. 编辑完成后，单击右上方的“保存”完成修改。

图 2-21 编辑算法的版本

关联资产

算法可以关联数据集资产。当算法关联了数据集时，数据集页面也显示关联了算法。

1. 选择“关联资产”页签，单击右上方的“编辑”，在搜索框中输入待关联资产的ID，单击“关联”。
2. 在弹出的“资产信息”页面，单击“确定”即可关联资产。

对已经关联的资产，单击“取消关联”即可取消资产的关联。

编辑论文

1. 选择“论文”页签，单击右上方的“编辑”，在编辑框内填写论文名称和其对应的访问链接，订阅者可以直接单击该链接查看论文详细内容。
2. 编辑完成后单击“保存”完成修改。

编辑代码

1. 选择“代码”页签，单击右上方的“编辑”，可以选择“代码是否开放”。

   说明

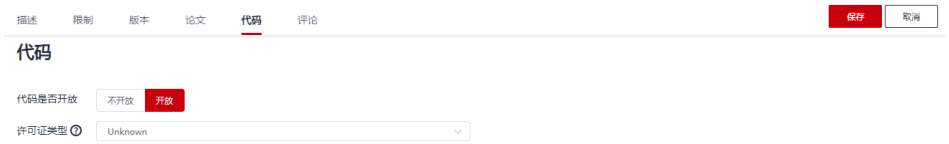
   订阅期满之前，下架代码不开放的算法不影响已订阅用户的使用。再次发布该算法代码开放后，主页列表不展示已经下架的算法，但用户可以在“我的Gallery > 我的资产 > 算法 > 我的订阅”页面单击该算法名称查看预览代码。

2. 如果开放代码可以选择修改“许可证类型”。

   单击许可证类型后面的感叹号可以了解许可证详情。

3. 编辑完成后单击“保存”完成修改。

图 2-22 编辑代码



发表评论

1. 请确保开启了邮箱通知。

   在“AI Gallery”页面中，单击右上角“我的Gallery > 我的资料”进入我的资料页面，查看“开启邮箱通知”开关，默认是打开的。如果未打开请开启。

2. 选择“评论”页签在输入框中输入评论内容，单击“发表评论”，即可成功发布评论。资产发布者可收到评论的通知，资产评论者也会收到评论回复的通知，所有用户均可查看资产评论并回复评论，对评论点赞等。

下架算法

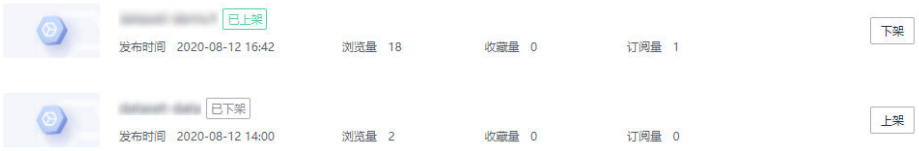
当您需要在AI Gallery下架共享的资产时，可以执行如下操作：

1. 在“AI Gallery”页面，选择“我的Gallery > 我的资产 > 算法”，进入“我的算法”页面。
2. 在“我的算法 > 我的发布”页面，单击目标资产右侧的“下架”，在弹框中确认资产信息，单击“确定”完成下架。

   说明

   资产下架后，已订阅该资产的用户在时长限制期内可继续正常使用，其他用户将无法查看和订阅该资产。

图 2-23 下架资产



资产下架成功后，操作列的“下架”会变成“上架”，您可以通过单击“上架”将下架的资产重新共享到AI Gallery中。

2.6.2 发布免费模型

在AI Gallery中，您可以个人开发的模型免费分享给他人使用，包括ModelArts模型和HiLens技能。

前提条件

- 如果是发布ModelArts模型，已经在ModelArts的“AI应用管理”中准备好待发布的模型。在“AI应用管理”界面创建或发布模型的相关操作请参见**管理AI应用简介**。使用容器镜像导入的模型和其他训练产生的模型都支持发布至AI Gallery。
- 如果是发布HiLens技能，已经在HiLens技能管理中准备好待发布的技能。发布技能的相关操作请参见**发布技能**。

发布免费模型

1. 进入AI Gallery首页，选择“资产集市 > 模型”，进入模型页面。
2. 单击“发布”，弹出“选择云服务区域”，选择区域后单击“确定”跳转到“发布资产到AI Gallery”页面。

   发布ModelArts模型

   - 如果是发布新资产。
     i. “发布方式”选择“创建新资产”。
     ii. 填写“资产标题”。即在AI Gallery显示的资产名称。
     iii. “来源”选择“ModelArts”。
     iv. 设置“ModelArts区域”。
     设置可以使用该资产的ModelArts区域，以控制台实际可选值为准。
     v. 选择“AI应用名称”。
     从ModelArts的AI应用管理中选择待发布的模型。支持将使用容器镜像导入的模型和其他训练产生的模型发布至AI Gallery。
     vi. 填写“资产版本”。版本号格式为“x.x.x”。
     vii. 设置“谁可以看”。
     设置资产的公开权限。可选值有：
     “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。
     “指定用户”：表示仅特定用户可以查看及使用该资产。
     “仅自己可见”：表示只有当前账号可以查看并使用该资产。
     viii. “时长限制”。

设置订阅者可以免费使用资产的时长，默认关闭，即无限期使用。如果打开时长限制，除了设置资产免费使用的时长，还可以设置到期后是否续订。

- 如果是更新已发布资产的版本。
  i. “发布方式”选择“添加资产版本”。
  ii. 在“资产标题”下拉框中选择已有资产名称。支持搜索资产名称。
  iii. 设置“ModelArts区域”。
  设置可以使用该资产的ModelArts区域，以控制台实际可选值为准。
  iv. 选择“AI应用名称”。
  从ModelArts的AI应用管理中选择待发布的模型。
  支持将使用容器镜像导入的模型和其他训练产生的模型发布至AI Gallery。
  v. 在“资产版本”填写新的版本号。

发布HiLens技能

表 2-3 发布 HiLens 技能的参数说明

参数	说明
资产分类	选择“模型”。
发布方式	发布方式选择“创建新资产”。
资产标题	在AI Gallery显示的资产名称，建议按照您的实现目的设置。
来源	选择“HiLens”。
HiLens区域	设置可以使用该资产的HiLens区域，以控制台实际可选值为准。
技能名称	从HiLens技能管理中选择待分享的技能。
谁可以看	设置资产的公开权限。可选值有： ● “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。 ● “指定用户”：表示仅特定用户可以查看及使用该资产。 ● “仅自己可见”：表示只有当前账号可以查看并使用该资产。
路数限制	设置订阅者可以免费使用HiLens技能的路数，默认关闭，即无限制使用。如果打开路数限制，除了设置资产免费使用的路数，还可以设置到期后是否续订。

3. 阅读并同意《华为云AI Gallery数字内容发布协议》和《华为云AI Gallery服务协议》。
4. 单击“发布”。

说明

发布使用容器镜像导入的资产时，后台会进行资产安全扫描，如果扫描发现资产有问题，则资产发布失败并邮件通知发布者。

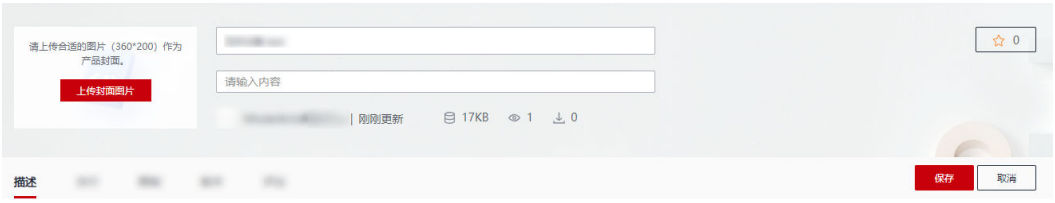
编辑资产详情

资产发布成功后，发布者可以进入详情页修改该资产的标题、封面图、描述等，让资产更吸引人。

修改封面图和二级标题

1. 在发布的资产详情页面，单击右侧的“编辑”，选择上传新的封面图，为资产编辑独特的主副标题。
2. 编辑完成之后单击“保存”。封面图和二级标题内容自动同步，您可以直接在资产详情页查看修改结果。

图 2-24 修改封面图和二级标题



编辑标签

1. 单击标签右侧的出现标签编辑框，在下拉框中勾选该资产对应的标签。
2. 单击编辑框右侧的对勾完成编辑。
   保存成功的标签信息会在资产搜索页成为过滤分类条件。

图 2-25 添加标签

编辑描述

1. 单击右侧的“编辑”，在编辑框中输入资产的描述内容，包含但不局限于背景、简介、使用方法、约束条件等。支持发布者以Markdown形式自由编辑。
2. 编辑完成之后单击“保存”。

编辑限制

支持修改资产的公开权限和时长限制或路数限制。

1. 选择“限制”页签，单击右上方的“编辑”进入编辑模式：
   - 在“谁可以看”右侧的下拉框中选择公开权限。
     - “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。
     - “指定用户”：表示仅特定用户可以查看及使用该资产。
     - “仅自己可见”：表示只有当前账号可以查看并使用该资产。

     说明

     公开权限只支持权限的扩大，权限从小到大为“仅自己可见<指定用户<公开”。所以如果一开始创建的是公开模型，将不支持修改“谁可以看”。
   - “时长限制”（ModelArts模型）或“路数限制”（HiLens技能）可以选择“不启用”或“启用”。当启用时，可以设置资产的免费使用时长或路数，以及到期后是否续订。
2. 单击“保存”，完成修改。

编辑版本

1. 选择“版本”页签，单击右上方的“编辑”。
2. 在此页面可以修改版本说明或者单击对应版本“操作”列的“下线”，下架不需要的资产版本。下线操作仅对已上架成功且存在多个可用版本的资产有效。
3. 添加模型版本：在版本框右侧单击“添加版本”，弹出“选择云服务区域”，选择区域后单击“确定”跳转到“发布资产到AI Gallery”页面，参考**更新已发布资产的版本**添加新版本。

   添加HiLens版本：在版本框右侧单击“添加版本”弹出“创建新版本”页面，选择需要用于新版本的HiLens技能，添加“版本说明”，编辑完成之后单击“确定”版本列表新增版本。

   说明

   添加HiLens版本时，先在HiLens平台修改HiLens的技能版本，然后在AI Gallery中添加版本。
4. 编辑完成后，单击右上方的“保存”完成修改。

图 2-26 编辑模型的版本

发表评论

1. 请确保开启了邮箱通知。

   在“AI Gallery”页面中，单击右上角“我的Gallery > 我的资料”进入我的资料页面，查看“开启邮箱通知”开关，默认是打开的。如果未打开请开启。
2. 选择“评论”页签在输入框中输入评论内容，单击“发表评论”，即可成功发布评论。资产发布者可收到评论的通知，资产评论者也会收到评论回复的通知，所有用户均可查看资产评论并回复评论，对评论点赞等。

下架免费模型

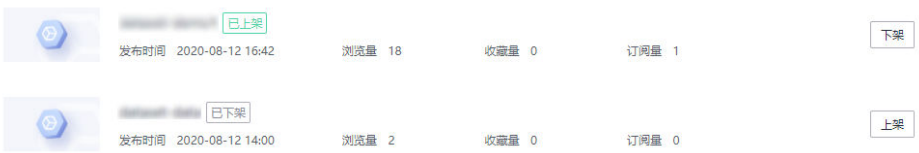
当您需要在AI Gallery下架共享的资产时，可以执行如下操作：

1. 在“AI Gallery”页面，选择“我的Gallery > 我的资产 > 模型”，进入“我的模型”页面。
2. 在“我的模型 > 我的发布”页面，单击目标资产右侧的“下架”，在弹框中确认资产信息，单击“确定”完成下架。

   说明

   资产下架后，已订阅该资产的用户在时长限制期内可继续正常使用，其他用户将无法查看和订阅该资产。

图 2-27 下架资产



资产下架成功后，操作列的“下架”会变成“上架”，您可以通过单击“上架”将下架的资产重新共享到AI Gallery中。

2.6.3 发布数据

在AI Gallery中，您可以将个人数据集分享给他人使用。

说明

ModelArts数据管理模块在重构升级中，对未使用过数据管理的用户不可见。建议新用户选择发布OBS或本地的数据集。

前提条件

- 本地或对象存储服务（OBS）中已准备好待发布的数据集，或ModelArts的数据集列表存在待发布的数据集。

发布数据集

1. 进入AI Gallery首页，选择“资产集市 > 数据集”，进入数据页面。
2. 单击“发布”弹出“选择云服务区域”，选择区域后单击“确定”进入发布数据集页面，填写相关信息。
   - 如果选择ModelArts已有的数据集发布，则参见表2-4配置数据集信息。

图 2-28 发布数据集（ModelArts）

表 2-4 参数说明（ModelArts）

参数	说明
资产标题	在AI Gallery显示的资产名称，建议按照您的目的设置。

参数	说明
来源	选择“ModelArts”。 单个数据集最多支持20000个文件，总大小不超过30G。
ModelArts区域	选择数据集所在的区域，以控制台实际可选值为准。
选择数据集	从下拉列表中选择当前区域中需要发布的目标数据集。
选择版本	选择目标数据集需要发布的版本。
许可证类型	根据业务需求和数据集类型选择合适的许可证类型。 单击许可证类型后面的 可以查看许可证详情。
谁可以看	设置此数据集的公开权限。可选值有： ● “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。 ● “指定用户”：表示仅特定用户可以查看及使用该资产。 ● “仅自己可见”：表示只有当前账号可以查看并使用该资产。

说明

发布来源为“ModelArts”的数据集，发布后在AI Gallery“我的数据 > 我的发布”不会立即显示，需要刷新该页面才能看到新发布的数据集。

- 如果选择对象存储服务（OBS）中已有的数据集发布，则参见表2-5配置数据集信息。

图 2-29 发布数据集（OBS）

表 2-5 参数说明（OBS）

参数	说明
资产标题	在AI Gallery显示的资产名称，建议按照您的目的设置。
来源	选择“对象存储服务（OBS）”。 单个数据集最多支持20000个文件，总大小不超过30G。
OBS区域	选择数据所在OBS桶的存储区域，以控制台实际可选值为准。
存储位置	选择待发布数据集所在对象存储服务（OBS）的路径。
数据类型	至少选择一个数据集类型的标签。 可选标签：图片、音频、视频、文本、表格、其他
许可证类型	根据业务需求和数据集类型选择合适的许可证类型。 单击许可证类型后面的 可以查看许可证详情。

参数	说明
谁可以看	设置此数据集的公开权限。可选值有： • “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。 • “指定用户”：表示仅特定用户可以查看及使用该资产。 • “仅自己可见”：表示只有当前账号可以查看并使用该资产。

- 如果选择本地的数据集发布，则参见表2-6配置数据集信息。

表 2-6 参数说明（本地上传）

参数	说明
资产标题	在AI Gallery显示的资产名称，建议按照您的目的设置。
来源	选择“本地上传”。 单次最多支持100个文件同时上传，总大小不超过5GB。
上传数据	从本地文件中选择需要发布的数据集。
数据类型	至少选择一个数据集类型的标签。 可选标签：图片、音频、视频、文本、表格、其他
许可证类型	根据业务需求和数据集类型选择合适的许可证类型。 单击许可证类型后面的 可以查看许可证详情。

参数	说明
谁可以看	设置此数据集的公开权限。可选值有： • “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。 • “指定用户”：表示仅特定用户可以查看及使用该资产。 • “仅自己可见”：表示只有当前账号可以查看并使用该资产。

3. 阅读并同意《华为云AI Gallery数字内容发布协议》和《华为云AI Gallery服务协议》。
4. 单击“发布”。

说明

发布数据时，数据集文件所在的OBS文件夹不能增加或删除文件，否则会引起发布前后文件数量或大小不一致，从而导致发布失败。

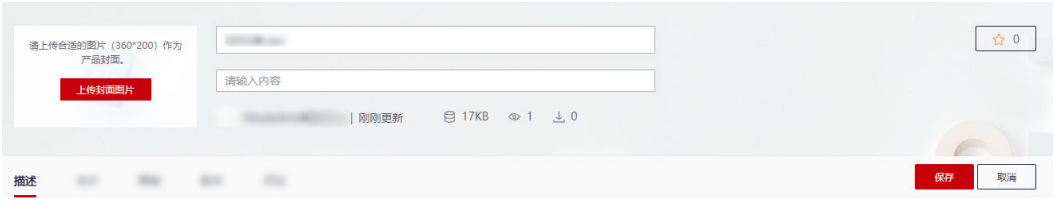
编辑资产详情

数据集发布成功后，发布者可以进入数据集的详情页修改该数据集“描述”、“版本”和“限制”等信息。

修改封面图和二级标题

1. 在发布的资产详情页面，单击右侧的“编辑”，选择上传新的封面图，为资产编辑独特的主副标题。
2. 编辑完成之后单击“保存”。封面图和二级标题内容自动同步，您可以直接在资产详情页查看修改结果。

图 2-30 修改封面图和二级标题



编辑许可证类型

1. 在发布的资产详情页面，单击右侧的“编辑”。
2. 在许可证类型右侧的下拉框中选择需要更新的许可证，单击“保存”完成修改。
   单击许可证类型后面的感叹号可以了解许可证详情。

编辑标签

1. 单击标签右侧的 出现标签编辑框。
2. 在下拉框中勾选该资产对应的标签，单击编辑框右侧的对勾完成编辑。
   标签信息会在资产主页成为过滤分类条件，让用户更容易找到您的资产。

图 2-31 添加标签

编辑描述

1. 单击右侧的“编辑”，在编辑框中输入资产的描述内容，包含但不局限于背景、简介、使用方法、约束条件等。支持发布者以Markdown形式自由编辑。
2. 编辑完成之后单击“保存”。

预览

预览可以查看数据集文件夹下所有文件，单击某个文件，可以查看文件内容。预览功能支持查看的文件类型请以界面显示为准。

编辑版本

1. 选择“版本”页签，单击右上方的“编辑”进入编辑模式。
2. 单击“版本说明”列的，添加版本说明，单击 完成添加。
   编辑数据集的版本信息便于区分数据集信息。

编辑限制

1. 选择“限制”页签，单击右上方的“编辑”进入编辑模式。
2. 在“谁可以看”右侧的下拉框中选择公开权限，单击“保存”完成修改。
   - “公开”：表示所有使用AI Gallery的用户都可以查看且使用该资产。
   - “指定用户”：表示仅特定用户可以查看及使用该资产。
   - “仅自己可见”：表示只有当前账号可以查看并使用该资产。
3. 单击“保存”，完成修改。

关联资产

数据集可以关联Notebook和算法。当数据集关联了Notebook或算法时，Notebook或算法页面也显示关联了数据集。

1. 选择“关联资产”页签，单击右上方的“编辑”，在搜索框中输入待关联资产的ID，单击“关联”。
2. 在弹出的“资产信息”页面，单击“确定”即可关联资产。

对已经关联的资产，单击“取消关联”即可取消资产的关联。

发表评论

1. 请确保开启了邮箱通知。

   在“AI Gallery”页面中，单击右上角“我的Gallery > 我的资料”进入我的资料页面，查看“开启邮箱通知”开关，默认是打开的。如果未打开请开启。

2. 选择“评论”页签在输入框中输入评论内容，单击“发表评论”，即可成功发布评论。资产发布者可收到评论的通知，资产评论者也会收到评论回复的通知，所有用户均可查看资产评论并回复评论，对评论点赞等。

重试发布数据集

如果数据集发布异常，您可以重试发布。

1. 在AI Gallery页面的右上角选择“我的Gallery > 我的资产 > 数据”，进入“我的数据”。
2. 在“我的发布”页签，查看发布异常的数据集。

图 2-32 查看发布异常的数据集

3. 根据异常状态的错误提示修改源数据后，单击目标数据集右侧的“重试”重新发布数据集。

删除发布的数据集

当您需要删除发布在AI Gallery中的数据集时，可以执行如下步骤进行删除。

1. 在AI Gallery页面的右上角选择“我的Gallery > 我的资产 > 数据”，进入“我的数据”。
2. 在“我的发布”页签，单击目标数据集右侧的“删除”，在弹窗中确认删除。

   说明

   由于数据集是下载至OBS使用的，所以删除已发布的数据集对使用者无影响。

2.6.4 发布 Notebook

在AI Gallery中，您可以将个人开发的Notebook代码免费分享给他人使用。

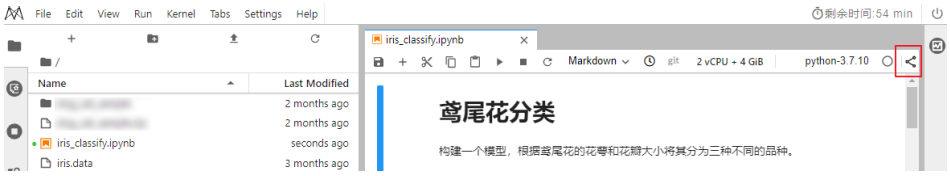
前提条件

- 在ModelArts的Notebook或者CodeLab中已创建好ipynb文件，开发指导可参见开发工具。

发布 Notebook

1. 登录ModelArts管理控制台。
2. 进入JupyterLab页面，在待分享的ipynb文件右侧，单击“创建分享”按钮，弹出“发布AI Gallery Notebook”页面。

图 2-33 单击“创建分享”

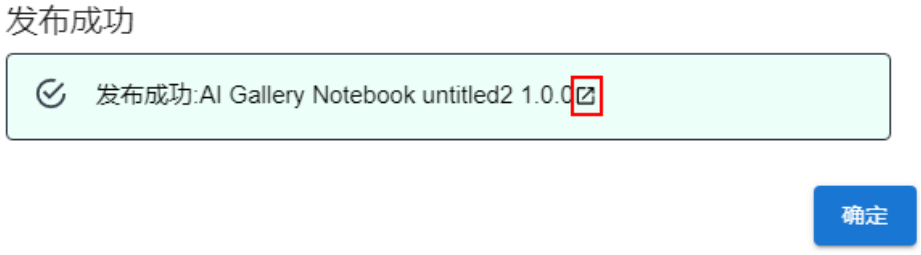


3. 在“发布AI Gallery Notebook”页面填写参数，单击“创建”将Notebook代码样例分享至AI Gallery。
   - 填写“发布标题”，标题长度为3~64个字符，不能包含以下字符“\ / : * ? " < > | ' &”。
   - 勾选“我已阅读并同意《华为云AI Gallery数字内容发布协议》和《华为云AI Gallery服务协议》”。
   - 选择运行环境：CPU、GPU或ASCEND。

图 2-34 发布 AI Gallery Notebook

4. 界面提示成功创建分享后，单击“ ”跳转至AI Gallery，进入Notebook代码样例的详情页面。

图 2-35 跳转至 AI Gallery



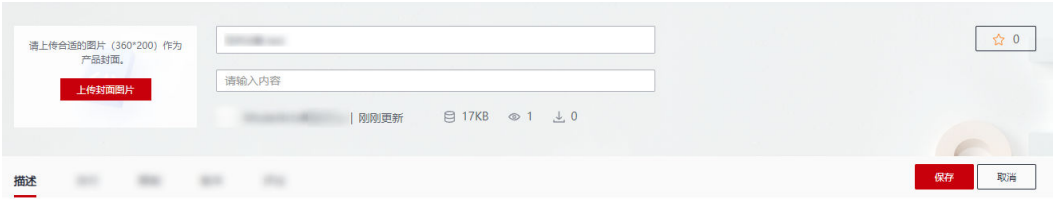
编辑资产详情

资产发布成功后，发布者可以进入详情页修改该资产的标题、封面图等，让资产更吸引人。其中，资产的公开权限和版本信息暂不支持修改。

修改封面图和二级标题

1. 在发布的资产详情页面，单击右侧的“编辑”，选择上传新的封面图，为资产编辑独特的主副标题。
2. 编辑完成之后单击“保存”，封面图和二级标题内容自动同步，您可以直接在资产详情页查看修改结果。

图 2-36 修改封面图和二级标题



编辑标签

1. 单击标签右侧的出现标签编辑框，在下拉框中勾选该资产对应的标签。
2. 单击编辑框右侧的对勾完成编辑。
   保存成功的标签信息会在资产搜索页成为过滤分类条件。

图 2-37 添加标签

关联资产

Notebook可以关联数据集资产。当Notebook关联了数据集时，数据集页面也显示关联了Notebook。

1. 选择“关联资产”页签，单击右上方的“编辑”，在搜索框中输入待关联资产的ID，单击“关联”。
2. 在弹出的“资产信息”页面，单击“确定”即可关联资产。

对已经关联的资产，单击“取消关联”即可取消资产的关联。

发表评论

1. 请确保开启了邮箱通知。

   在“AI Gallery”页面中，单击右上角“我的Gallery > 我的资料”进入我的资料页面，查看“开启邮箱通知”开关，默认是打开的。如果未打开请开启。
2. 选择“评论”页签在输入框中输入评论内容，单击“发表评论”，即可成功发布评论。资产发布者可收到评论的通知，资产评论者也会收到评论回复的通知，所有用户均可查看资产评论并回复评论，对评论点赞等。

下架 Notebook

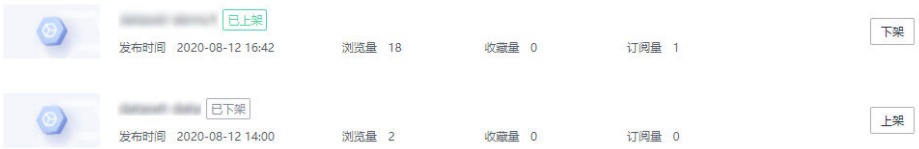
当您需要在AI Gallery下架共享的资产时，可以执行如下操作：

1. 在“AI Gallery”页面，选择“我的Gallery > 我的资产 > Notebook”，进入“我的Notebook”。
2. 在“我的Notebook > 我的发布”页面，单击目标资产右侧的“下架”，在弹框中确认资产信息，单击“确定”完成下架。

   说明

   资产下架后，已订阅该资产的用户可继续正常使用，其他用户将无法查看和订阅该资产。

   图 2-38 下架资产

3. 资产下架成功后，操作列的“下架”会变成“上架”，您可以通过单击“上架”将下架的资产重新共享到AI Gallery中。

2.7 参加活动

2.7.1 报名实践活动（实践）

在AI Gallery中，可以报名参加正在进行中的实践活动。

查找实践活动

1. 进入AI Gallery首页，单击“实践”，在下拉框中单击“实践 >”，进入实践首页。
2. 在实践页面，有“进行中”、“即将开始”和“已结束”三种状态的实践活动筛选方式。

图 2-39 查找实践活动

单击右上方的“我的实践”可以跳转到个人中心（“我的Gallery > 我的实践”），查看个人已参加的实践活动列表。

报名实践活动

1. 进入AI Gallery首页，单击“实践”，在下拉框中单击“实践 >”，进入实践首页。
2. 在实践列表选择您感兴趣的实践活动。
3. 报名实践活动：
   - 方式一：单击实践活动简介下的“立即报名”，进入邀请函页面，根据提示填写个人信息，单击“报名”。

     图 2-40 活动邀请函

   - 方式二：单击实践活动标题进入活动详情页面，在详情页面单击“立即报名”，进入邀请函页面报名。

     在详情页面可以查看Notebook实例的“描述”和“评论”信息。

2.7.2 发布技术文章（AI 说）

AI Gallery中的“AI说”，是一个AI开发人员的交流园地。在这里可以阅读其他用户分享的技术文章，并参与评论。也可以发布分享个人技术文章。

前提条件

已入驻AI Gallery。

发布技术文章

1. 进入AI Gallery首页，单击“AI说”，在下拉框中单击“AI说 >”，进入AI说首页。
2. 在“AI说”页面，单击右侧“说一说”进入发布页面。

3. 在“AI说”发布页面，填写相关信息。

图 2-41 发布技术文章

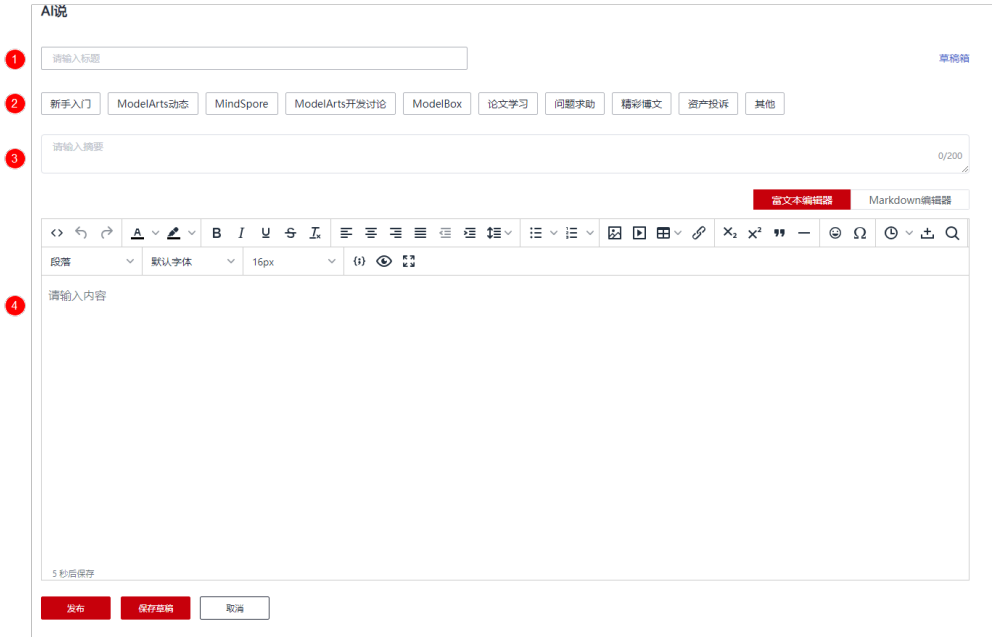


表 2-7 填写说明

区域	填写说明
1	输入技术文章的标题。
2	选择技术文章所属分类。
3	输入摘要信息。
4	编辑技术文章的内容。右侧可以选择使用“富文本编辑器”或“markdown”方式编辑内容，也可上传附件，支持rar，zip，doc，docx，xls，xlsx，ppt，pptx，pdf，txt格式的附件，单个附件大小不超过20M，最多可传5个附件。

说明

- 编辑AI说时，30秒钟后自动保存草稿，用户可单击“草稿箱”查看。
- 草稿箱可支持保存草稿记录共三条，存满后请手动删除；建议您及时发布重要内容，以免重要内容无法保存。

4. 单击“发布”，跳转至技术文章详情页面。

发表评论

当AI说有问题求助的时候，可以在AI说发表评论求助。

1. 请确保开启了邮箱通知。

   在“AI Gallery”页面中，单击右上角“我的Gallery > 我的资料”进入我的资料页面，查看“开启邮箱通知”开关，默认是打开的。如果未打开请开启。

2. 选择“评论”页签在输入框中输入评论内容，单击“发表评论”，即可成功发布评论。AI说发布者可收到评论的通知，AI说评论者也会收到评论回复的通知，所有用户均可查看资产评论并回复评论，对评论点赞等。

删除发布的技术文章

当您需要删除已发布在“AI说”的技术文章时，可以执行如下步骤：

1. 在AI Gallery页面的右上角单击“我的Gallery > 我的AI说”。
2. 在“我的发布”页签下查看发布的所有文章。
3. 单击目标文章右侧的“删除”，在弹窗中确认删除。

2.8 合作伙伴

2.8.1 注册伙伴

仅当暂未注册伙伴的用户可以注册伙伴。

1. 在“AI Gallery”页面中，单击右上角“我的Gallery > 我的主页”进入个人中心页面。
2. 左侧菜单栏选择“解决方案”进入解决方案列表页，单击右上方“发布”进入合作伙伴申请页面。

   说明

   如果已经是伙伴用户，则会进入发布解决方案页面。
3. 根据界面提示，填写注册成为合作伙伴需要提供的信息。
4. 单击“提交”，AI Gallery的运营人员将会审核您的申请，后续您可以在“我的Gallery > 合作伙伴”里查看审核进展以及审核结果。

图 2-42 查看审核进度

2.8.2 发布解决方案

如果你已经注册成为了AI Gallery平台上的合作伙伴，AI Gallery支持发布共享你的解决方案。

1. 在“AI Gallery”页面中，单击右上角“我的Gallery > 我的主页”进入个人中心页面。
2. 左侧菜单栏选择“解决方案”进入解决方案列表页，单击右上方的“发布”，进入发布解决方案页面。
3. 根据界面提示填写解决方案的相关信息，单击下方的“提交”。

   在解决方案列表页可以查看发布的方案信息。

2.9 需求广场

2.9.1 发布需求

如果你已经注册成为了AI Gallery平台上的合作伙伴，你可以在AI Gallery上发布你的需求。

1. 在“AI Gallery”页面中，单击右上角“我的Gallery > 我的主页”进入个人中心页面。
2. 左侧菜单栏选择“我的需求”进入我的需求列表页，单击右上方的“发布”，进入发布需求页面。
3. 填入需求的相关信息。
4. 单击“提交”，AI Gallery的运营人员将会审核您的所发布的需求，后续您可以在“我的Gallery > 我的需求”里看到审核进展以及审核结果。