

Use pipeline parameters

5 minutes

You can increase the flexibility of a pipeline by defining parameters.

Defining parameters for a pipeline

To define parameters for a pipeline, create a **PipelineParameter** object for each parameter, and specify each parameter in at least one step.

For example, you could use the following code to include a parameter for a regularization rate in the script used by an estimator:

Python

 Copy

```
from azureml.pipeline.core.graph import PipelineParameter

reg_param = PipelineParameter(name='reg_rate', default_value=0.01)

...

step2 = PythonScriptStep(name = 'train model',
                        source_directory = 'scripts',
                        script_name = 'data_prep.py',
                        compute_target = 'aml-cluster',
                        # Pass parameter as script argument
                        arguments=['--in_folder', prepped_data,
                                '--reg', reg_param],
                        inputs=[prepped_data])
```

Note

You must define parameters for a pipeline before publishing it.

Running a pipeline with a parameter

After you publish a parameterized pipeline, you can pass parameter values in the JSON payload for the REST interface:

Python

 Copy

```
response = requests.post(rest_endpoint,  
                          headers=auth_header,  
                          json={"ExperimentName": "run_training_pipeline",  
                              "ParameterAssignments": {"reg_rate": 0.1}})
```

Next unit: Schedule pipelines

[Continue >](#)