Using Java Functions in KNIX

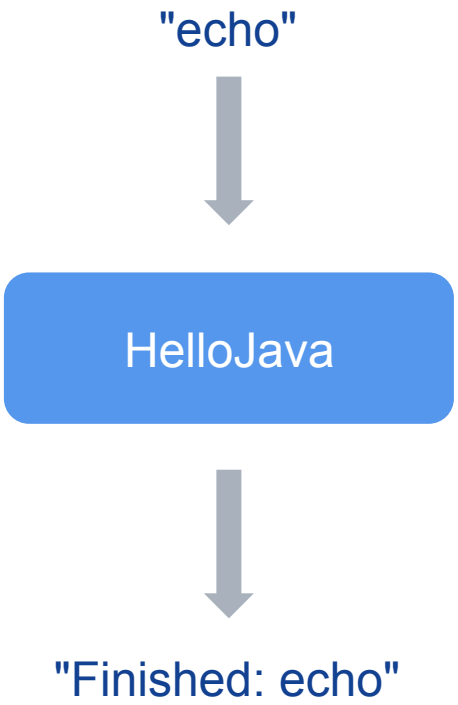
Istemi Ekin Akkus

KNIX Team

Agenda

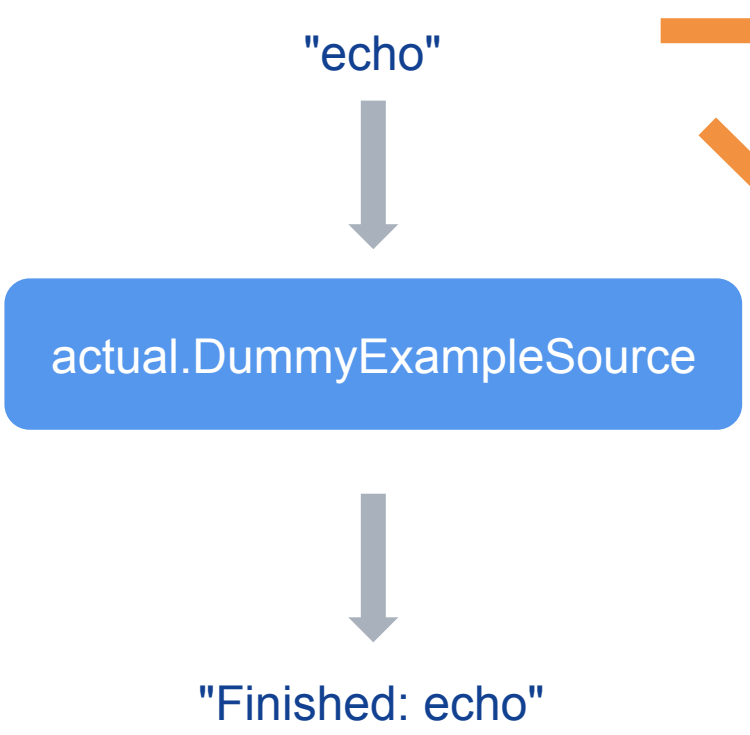
1. Hello Java: creating a single Java function
2. Uploading a jar package with multiple Java functions
3. Using external dependencies with Maven
4. Combining Java and Python functions in a workflow

Hello Java




```
import org.microfunctions.mfnapi.MicroFunctionsAPI;

public class HelloJava
{
    public Object handle(Object event, MicroFunctionsAPI context)
    {
        return "Finished: " + (String) event;
    }
}
```

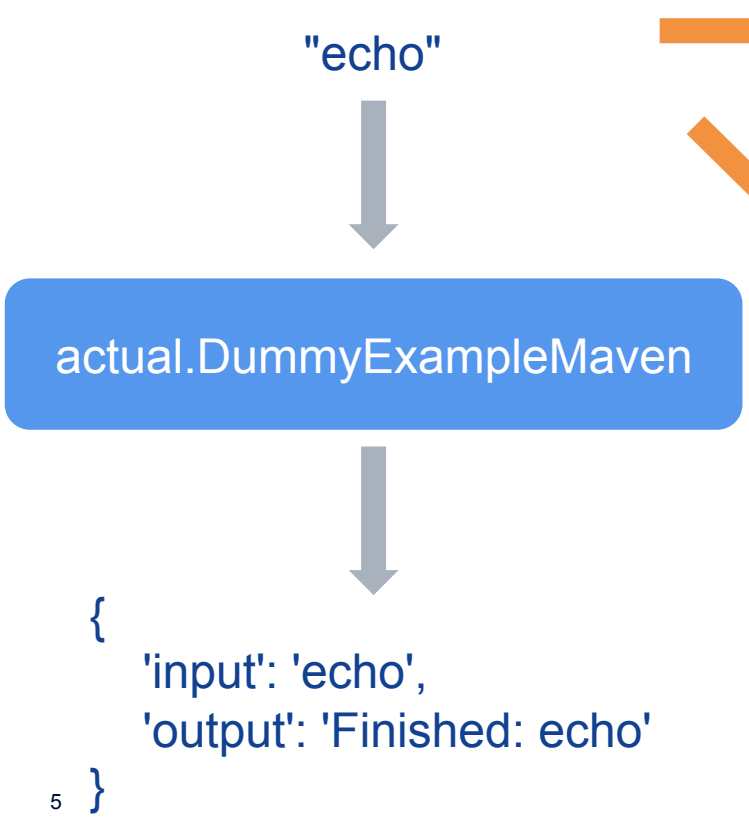
Uploading a jar package with source code



jar content: src/ (includes actual/ and dummy/)

```
package actual;

import org.microfunctions.mfnapi.MicroFunctionsAPI;

import dummy.SomeClass;

public class DummyExampleSource
{
    public Object handle(Object event, MicroFunctionsAPI context)
    {
        SomeClass sc = new SomeClass("myname", 10);
        sc.doSomething(context);

        return "Finished: " + (String) event;
    }
}
```

Using external dependencies with Maven

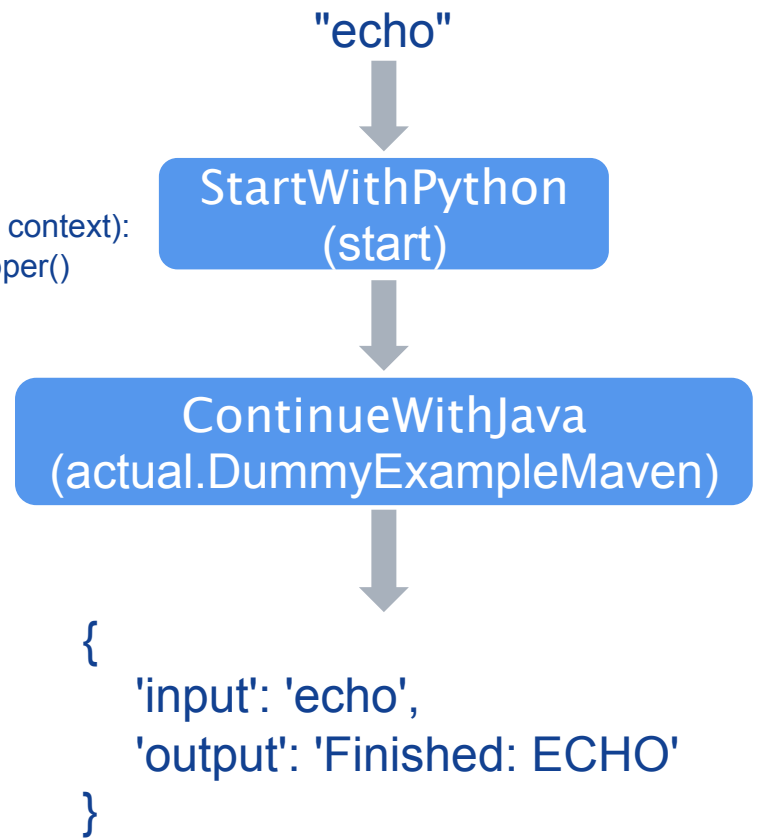


jar content: src/ + pom.xml

```
package actual;

import org.microfunctions.mfnapi.MicroFunctionsAPI;

import org.json.JSONObject;

public class DummyExampleMaven
{
    public Object handle(Object event, MicroFunctionsAPI context)
    {
        String eventstr = (String) event;

        JSONObject obj = new JSONObject();
        obj.put("input", eventstr);
        obj.put("output", "Finished: " + eventstr);

        return obj;
    }
}
```

Combining Java and Python functions in a workflow

```
#!/usr/bin/python

def handle(event, context):
    return event.upper()
```




```
{
  "Comment": "wf_python_java Workflow",
  "StartAt": "StartWithPython",
  "States": {
    "StartWithPython": {
      "Type": "Task",
      "Resource": "start",
      "Next": "ContinueWithJava"
    },
    "ContinueWithJava": {
      "Type": "Task",
      "Resource": "actual.DummyExampleMaven",
      "End": true
    }
  }
}
```

Thank you!

https://github.com/knix-microfunctions/knix/

https://knix.slack.com

Useful commands

- Creating a jar package
  - jar cf actual.DummyExampleSource.jar src/
- Creating a jar package with Maven dependencies
  - jar cf actual.DummyExampleSource.jar src/ pom.xml
- Creating a jar package only with classes
  - mkdir -p target/classes # the classes must be created in this folder
  - javac -cp /local/work/knix/JavaRequestHandler/mfnapi.jar src/main/java/actual/DummyExampleClass.java src/main/java/dummy/SomeClass.java -d target/classes
  - jar cf actual.DummyExampleClass.jar target/classes/