**Surround Examples**

from surround import Stage, SurroundData, Surround

import logging

class HelloStage(Stage):

def operate(self, data, config):

data.text = "hello"

class BasicData(SurroundData):

text = None

if \_\_name\_\_ == "\_\_main\_\_":

logging.basicConfig(level=logging.INFO)

surround = Surround([HelloStage()])

output = surround.process(BasicData())

print(output.text)

This example shows how to print the text "hello" to the screen using surround. The process consists of defining the operation in the *operate()* method of *HelloStage*. The object being processed is an instance of *BasicData*, which inherits from *SurroundData*. The surround object is initialised with only one stage as surround = *Surround([HelloStage()]*, and the line output = *surround.process(BasicData())* calls the *operate()* method of *HelloStage*, using a new instance of *BasicData* as a parameter. Finally, the content of the *BasicData* object is printed to screen.

|  |
| --- |
| import logging |
|  | import os |
|  | from surround import Stage, SurroundData, Surround |
|  | from file\_system\_runner import FileSystemRunner |
|  |  |
|  | class WriteHello(Stage): |
|  | def dump\_output(self, surround\_data, config): |
|  | dir\_path = os.path.dirname(os.path.realpath(\_\_file\_\_)) |
|  | text\_file = open(dir\_path + "/stages/WriteHello/Output.txt", "w") |
|  | text\_file.write(surround\_data.text) |
|  | text\_file.close() |
|  |  |
|  | def operate(self, surround\_data, config): |
|  | surround\_data.text = "Hello" |
|  |  |
|  | class WriteWorld(Stage): |
|  | def dump\_output(self, surround\_data, config): |
|  | dir\_path = os.path.dirname(os.path.realpath(\_\_file\_\_)) |
|  | text\_file = open(dir\_path + "/stages/WriteWorld/Output.txt", "w") |
|  | text\_file.write(surround\_data.text) |
|  | text\_file.close() |
|  |  |
|  | def operate(self, surround\_data, config): |
|  | surround\_data.text = "World" |
|  |  |
|  | class BasicData(SurroundData): |
|  | text = None |
|  |  |
|  | if \_\_name\_\_ == "\_\_main\_\_": |
|  | logging.basicConfig(level=logging.INFO) |
|  |  |
|  | surround = Surround([WriteHello(), WriteWorld()]) |
|  | adapter = FileSystemRunner(surround, |
|  | description="A sample project to show dump intermediate ouput", |
|  | config\_file="Path to configuration file") |
|  | adapter.start() |
|  | surround.process(BasicData()) |

This example shows the use of the dump\_output() method of the Stage class in surround by writing the outputs of two stages (WriteHello and WriteWorld) in their respective output files.

* The surround object configuration is set in the FileSystemRunner class constructor, instantiated as adapter in *dump\_output.py*.
* Through surround.process(), stages WriteHello and Write�World are executed, setting the text value of BasicData to "Hello" and "World" in each stage's operate() method.
* Since enable\_stage\_output\_dump is set to True in config.yaml, surround automatically calls the dump\_output()method of each stage after operate().
* A folder stages/<StageName> is created for each stage, each with an Output.txt file, containing the value of BasicData.text.