

# UbiOps Cheat Sheet - Python

## Connection set-up

```
import ubiops
config = ubiops.Configuration(api_key={"Authorization": API_TOKEN})
client = ubiops.ApiClient(config)
api = ubiops.CoreApi(client)
```

## Deployment.py tricks

### Getting the file path to a file in your package from the deployment.py:

```
file_path = os.path.join(base_directory, "your-filename")
```

### Reading in an environment variable:

```
import os
my_var = os.environ["var_name"]
```

### Reading in input data of request:

```
def request(self, data):
    input_x = data['input_x'] # For structured data
    input_x = data # For plain data
    # ... rest of code
```

### Outputting a file:

```
return {
    "output_field_name": 'rel/path/to/file' #i.e. 'some_file.csv'
}
```

## Notebook tips

### For creating a deployment.py directly from your notebook cell, type this at the top of the cell:

```
%%writefile deployment_package_folder/deployment.py
```

### Loading contents of a file into a cell:

```
%load rel/path/to/file
```

### Zip and upload your package to UbiOps:

```
# Zip the deployment package
import shutil
shutil.make_archive('deployment_package',
'zip', '.', 'deployment_package')

# Upload the zipped deployment package
file_upload_result =api.revisions_file_upload
(
    project_name=PROJECT_NAME,
    deployment_name=DEPLOYMENT_NAME,
    version=DEPLOYMENT_VERSION,
    file='deployment_package.zip'
)
```

## Making requests with the Python Client Library

### A request to the default version of a deployment:

```
data = {'input_field_1': 123}
request_result = api.deployment_requests_create(
    project_name=PROJECT_NAME,
    deployment_name=DEPLOYMENT_NAME,
    data=data)
```

### A batch request to the default version of a deployment:

```
response = api.batch_deployment_version_requests_create(
    project_name=PROJECT_NAME,
    deployment_name=DEPLOYMENT_NAME,
    version=DEPLOYMENT_VERSION,
    data=data)
```

### A request to the default version of a pipeline

```
request_result = api.pipeline_requests_create(
    project_name=PROJECT_NAME,
    pipeline_name=PIPELINE_NAME,
    data=data)
```

### A batch request to the default version of a pipeline

```
response = api.batch_pipeline_version_requests_create(
    project_name=PROJECT_NAME,
    pipeline_name=DEPLOYMENT_NAME,
    version=DEPLOYMENT_VERSION,
    data=data)
```

### Making a request with a Blob:

```
img_path = "/path/to/image"
blob = api.blobs_create(project_name=PROJECT_NAME, file=img_path)
blob_data = {'image': blob.id}
# <Your regular request code with blob_data passed as data>
```

## Data type names in Client Library

int	array_int
string	array_string
double	array_double
bool	blob

## Handy resources

[Swagger \(API\)](#)

[Client library](#)

[Cookbook](#)

[CLI](#)

[Our GitHub repo](#)

[Documentation](#)

You can also test your deployments locally before pushing to UbiOps [with this script here](#).

