

## Step by step Lab Guide for Authy-Workshop-Python

### Add to authyonetouch.py

```
from authy.api import AuthyApiClient
authy_api_key = os.environ['AUTHY_API_KEY']
authy_api = AuthyApiClient(authy_api_key)
```

*the import adds this lab's Authy Python library, which was extended to include OneTouch support*

*Use the AUTHY\_API\_KEY environment variable to ensure your API key does not end up in a repository. Before you launch your python program, you can export AUTHY\_API\_KEY=<your production key>, or include it in your environment initialization.*

*The authy\_api line will create an AuthyApiClient with your key*

### Add to @app.route('/register', methods = ['POST']) prior to the redirect

```
user =
authy_api.users.create(regdata["email"],regdata["phone_number"],regdata["country_code
"])
if user.ok():
    print "Authy ID = %s " % user.id
    regdata["authyid"]=user.id
else:
    print user.errors()
```

*create a user variable to receive the results of Authy user registration. Send in email, phone number and country code as the only parameters, all required.*

*Check to ensure no errors, then print out the Authy ID (user.id) to your console, and store the value – in this lab, just assign to the regdata dictionary as regdata["authyid"]*

### Add to @app.route('/processtransaction', methods = ['POST']) prior to the redirect

```
message = {
    "message":"Transfer Money to" + transfer["email"],
    "details": {
        "From":regdata["email"],
        "To":transfer["email"],
        "Account Number":transfer["acct"],
        "US Dollar Amount":transfer["amt"]
    },
    "seconds_to_expire":"600"
}
print message
```

*create a dictionary for the message to be displayed to the user in their Authy OneTouch transaction. Include message, details (From/To/Acct#/US Dollar amt) and seconds\_to\_expire set to 600. Print the message to the console to validate.*

```
onetouch = authy_api.users.send_onetouch(regdata["authyid"], message)
print onetouch.uuid
transfer["transactionID"] = onetouch.uuid
```

*create a onetouch variable to receive the results of your Authy Onetouch transaction request. Send in your message created earlier.*

*Check to ensure no errors, then print out the OneTouch unique transaction ID (onetouch.uuid) to your console, and store the value – in this lab, just assign to the transfer dictionary as transfer["transactionID"]*

```
while True:
    onetouchStatus = authy_api.users.poll_onetouch(onetouch.uuid)
    print "Authy Onetouch Approval Status: %s " % onetouchStatus.status
    if onetouchStatus.status != "pending" :
        break
    time.sleep(1)
transfer["status"] = onetouchStatus.status
```

*Create a polling loop using while True:*

*Create a onetouchStatus variable, and call the (poll api) with onetouch.uuid (or transfer["transactionID"]). Print the onetouchStatus.status result to your console, and check to see if the status is anything but “pending”. If not pending, break the while loop, otherwise sleep for 1 second. After the while loop, set transfer[“status”] = onetouchStatus.status, and delete the earlier line transfer[“status”]=“Complete”*

*To save time, we’ve built in a transaction modal dialog box into transaction.html, which won’t show until you’ve put in the while loop delay.*