

# Joker.com FAQ

## Table of Content

<b>Python</b> .....	2
Intro .....	2
Login and list your domains .....	3
Login and list all A and CNAME records .....	6

# Python Intro

Python is easy to use with DMAPI. Our examples are tested with python 3 on Windows and Linux, but should work on all platforms.

Running the [Python example](#) will look like:

```
$ ./dmapi-example.py
Request-URL: https://dmapi.ote.joker.com/request/login
Login: Status-Code: 0

Request-URL: https://dmapi.ote.joker.com/request/query-domain-list
Domain List: Status-Code: 0

  domain: another-privacy-test.com
  expiration_date: 2018-06-30

  domain: another-privacy-test.net
  expiration_date: 2020-06-30

Request-URL: https://dmapi.ote.joker.com/request/logout
Logout: Status-Code: 0
```

Keywords: DMAPI, examples, code, python

Last update: 2021-07-05 09:39

# Python

## Login and list your domains

```
#!/usr/bin/env python
import requests

dmapiURL = 'https://dmapi.ote.joker.com'
dmapiUser = 'username'
dmapiPassword = 'password'

def main():
    loginResponse = login(dmapiUser, dmapiPassword)
    print("Login: Status-Code:", loginResponse.header['Status-Code'])
    if loginResponse.header['Status-Code'] != '0':
        print(loginResponse.header['Status-Text'])
        return

    sessionId = loginResponse.header['Auth-Sid'];
    print("")
    domainResponse = domainList(sessionId, 1, 5)
    print("Domain List: Status-Code:", domainResponse.header['Status-Code'])
    print("")
    domains = domainResponse.resultListWithNames()
    for domain in domains:
        for key, value in domain.items():
            print(" %s: %s" % (key, value))
        print("")
    logoutResponse = logout(sessionId)
    print("Logout: Status-Code:", logoutResponse.header['Status-Code'])
)

# implement dmapi commands as functions
def login(username, password):
    parameters = { 'username': username, 'password': password }
    message = sendCommand('login', parameters)
    return message;

def logout(sessionId):
    parameters = { 'auth-sid': sessionId }
    message = sendCommand('logout', parameters)
    return message;

def domainList(sessionId, list_from=1, list_to=""):
    parameters = { 'auth-sid': sessionId , 'from': list_from, 'to': list_to }
    message = sendCommand('query-domain-list', parameters)
    return message;
```

# Python

```
# general dmapi command call
def sendCommand(command,parameter={}):
    try:
        url = dmapiURL+'/request/'+command
        print("Request-URL: ", url)
        response = requests.get(url, params=parameter)
        # print URL with parameters for debugging purposes
        # print("Request-URL: ", response.url)
        if response.status_code != requests.codes.ok:
            raise CommandError("Command Failed! HTTP Status Code: %s"
% response.status_code)
        return DmapiResponse(response.text)
    except requests.ConnectionError as e:
        raise CommandError("Connection Error: %s" % str(e))
    except requests.HTTPError as e:
        raise CommandError("Http Error: %s" % str(e))
    except CommandError as e:
        raise
    except Exception as e:
        raise CommandError("Unexpected Error: %s" % str(e))

class DmapiResponse():
    def __init__(self,responseBody):
        parts = responseBody.split("\n\n",1)
        if len(parts)>0:
            self.header = self.__parseKeyValueList(parts[0])
        if len(parts)>1:
            self.body = parts[1]

    def __parseKeyValueList(self,text):
        lines = text.split("\n")
        keyValueList = {}
        for line in lines:
            keyValue = line.split(' ',1)
            key = keyValue[0].rstrip(':')
            value = keyValue[1]
            keyValueList[key] = value
        return keyValueList

    def __getSeparator(self):
        if self.header.get('Separator') == 'TAB':
            return "\t"
        else:
            return " "

    def resultList(self):
        lines = self.body.split("\n")
        resultList = []
```

# Python

```
separator = self.__getSeparator()
for line in lines:
    values = line.split(separator)
    resultList.append(line.split(separator))
return resultList

def resultListWithNames(self):
    columnNames = self.resultListColumns()
    resultList = []
    if len(columnNames) > 0:
        rawList = self.resultList()
        resultList = []
        for row in rawList:
            columns = {}
            for idx, column in enumerate(row):
                columns[columnNames[idx]] = column
            resultList.append(columns)
    return resultList

def resultListColumns(self):
    if 'Columns' in self.header:
        columnsText = self.header['Columns']
        columns = columnsText.split(',')
        return columns
    else:
        return []

def resultValues(self):
    return self.__parseKeyValueList(self.body)

class CommandError(Exception):
    def __init__(self, value):
        self.value = value
    def __str__(self):
        return repr(self.value)

# call main function
try:
    main()
except CommandError as e:
    print("Error:", str(e).strip("'"))
```

Keywords: DMAPI, examples, code, python

Last update: 2021-07-05 09:35

# Python

## Login and list all A and CNAME records

```
#!/usr/bin/env python
import requests

dmapiURL = 'https://dmapi.joker.com'
dmapiUser = 'username'
dmapiPassword = 'password'

def main():
    loginResponse = login(dmapiUser, dmapiPassword)
    #print("Login: Status-Code:", loginResponse.header['Status-Code'])
    if loginResponse.header['Status-Code'] != '0':
        print(loginResponse.header['Status-Text'])
        return

    sessionId = loginResponse.header['Auth-Sid'];
    #print("")
    dnsZoneListResponse = dnsZoneList(sessionId, "")
    #print("DNS Zone List: Status-Code:", dnsZoneListResponse.header['
Status-Code'])
    dnslist = dnsZoneListResponse.resultList()
    for row in dnslist:
        domain = row[0]
        expiration = row[1]
        #print(" domain: %s" % (domain))
        #print(" expiration: %s" % (expiration))
        #print("")
        dnsZoneResponse = dnsZoneGet(sessionId, domain)
        #print("DNS Zone GET for %s: Status-Code:" % (domain), dnsZone
Response.header['Status-Code'])
        zoneEntries = dnsZoneResponse.resultList()
        for entry in zoneEntries:
            #print(' '.join(entry))
            if len(entry)<5:
                continue
            eLabel = entry[0]
            eType = entry[1]
            ePriority = entry[2]
            eTarget = entry[3]
            eTTL = entry[4]
            if eType == 'A' or eType == 'CNAME':
                print(("%.%s\t%s" % (eLabel, domain, eTarget)).lstrip('
@.'))

    logoutResponse = logout(sessionId)
    #print "Logout: Status-Code:", logoutResponse.header['Status-Code'
]
]
```

# Python

```
# implement dmapi commands as functions
def login(username,password):
    parameters = { 'username': username, 'password': password }
    message = sendCommand('login', parameters)
    return message;

def logout(sessionId):
    parameters = { 'auth-sid': sessionId }
    message = sendCommand('logout', parameters)
    return message;

def domainList(sessionId, pattern="", list_from=1, list_to=""):
    parameters = { 'auth-sid': sessionId , 'from': list_from, 'to': list_to, 'pattern': pattern }
    message = sendCommand('query-domain-list', parameters)
    return message;

def dnsZoneList(sessionId, pattern="", list_from=1, list_to=""):
    parameters = { 'auth-sid': sessionId , 'from': list_from, 'to': list_to, 'pattern': pattern }
    message = sendCommand('dns-zone-list', parameters)
    return message;

def dnsZoneGet(sessionId, domain):
    parameters = { 'auth-sid': sessionId , 'domain': domain }
    message = sendCommand('dns-zone-get', parameters)
    return message;

# general dmapi command call
def sendCommand(command,parameter={}):
    try:
        url = dmapiURL+'/request/'+command
        #print("Request-URL: ", url)
        response = requests.get(url, params=parameter)
        # print URL with parameters for debugging purposes
        # print("Request-URL: ", response.url)
        if response.status_code != requests.codes.ok:
            raise CommandError("Command Failed! HTTP Status Code: %s"
% response.status_code)
        return DmapiResponse(response.text)
    except requests.ConnectionError as e:
        raise CommandError("Connection Error: %s" % str(e))
    except requests.HTTPError as e:
        raise CommandError("Http Error: %s" % str(e))
    except CommandError as e:
        raise
    except Exception as e:
```

# Python

```
raise CommandError("Unexpected Error: %s" % str(e))
```

```
class DmapiResponse():
    def __init__(self,responseBody):
        parts = responseBody.split("\n\n",1)
        if len(parts)>0:
            self.header = self.__parseKeyValueList(parts[0])
        if len(parts)>1:
            self.body = parts[1]

    def __parseKeyValueList(self,text):
        lines = text.split("\n")
        keyValueList = {}
        for line in lines:
            keyValue = line.split(' ',1)
            key = keyValue[0].rstrip(':')
            value = keyValue[1]
            keyValueList[key] = value
        return keyValueList

    def __getSeparator(self):
        if self.header.get('Separator') == 'TAB':
            return "\t"
        else:
            return " "

    def resultList(self):
        lines = self.body.split("\n")
        resultList = []
        separator = self.__getSeparator()
        for line in lines:
            values = line.split(separator)
            resultList.append(line.split(separator))
        return resultList

    def resultListWithNames(self):
        columnNames = self.resultListColumns()
        resultList = []
        if len(columnNames) > 0:
            rawList = self.resultList()
            resultList = []
            for row in rawList:
                columns = {}
                for idx, column in enumerate(row):
                    columns[columnNames[idx]] = column
                resultList.append(columns)
        return resultList

    def resultListColumns(self):
```



# Python

```
if 'Columns' in self.header:
    columnsText = self.header['Columns']
    columns = columnsText.split(',')
    return columns
else:
    return []

def resultValues(self):
    return self.__parseKeyValueList(self.body)

class CommandError(Exception):
    def __init__(self, value):
        self.value = value
    def __str__(self):
        return repr(self.value)

# call main function
try:
    main()
except CommandError as e:
    print("Error:", str(e).strip(""))
```

Keywords: DMAPI, examples, code, python

Last update: 2021-07-05 09:52

# Python