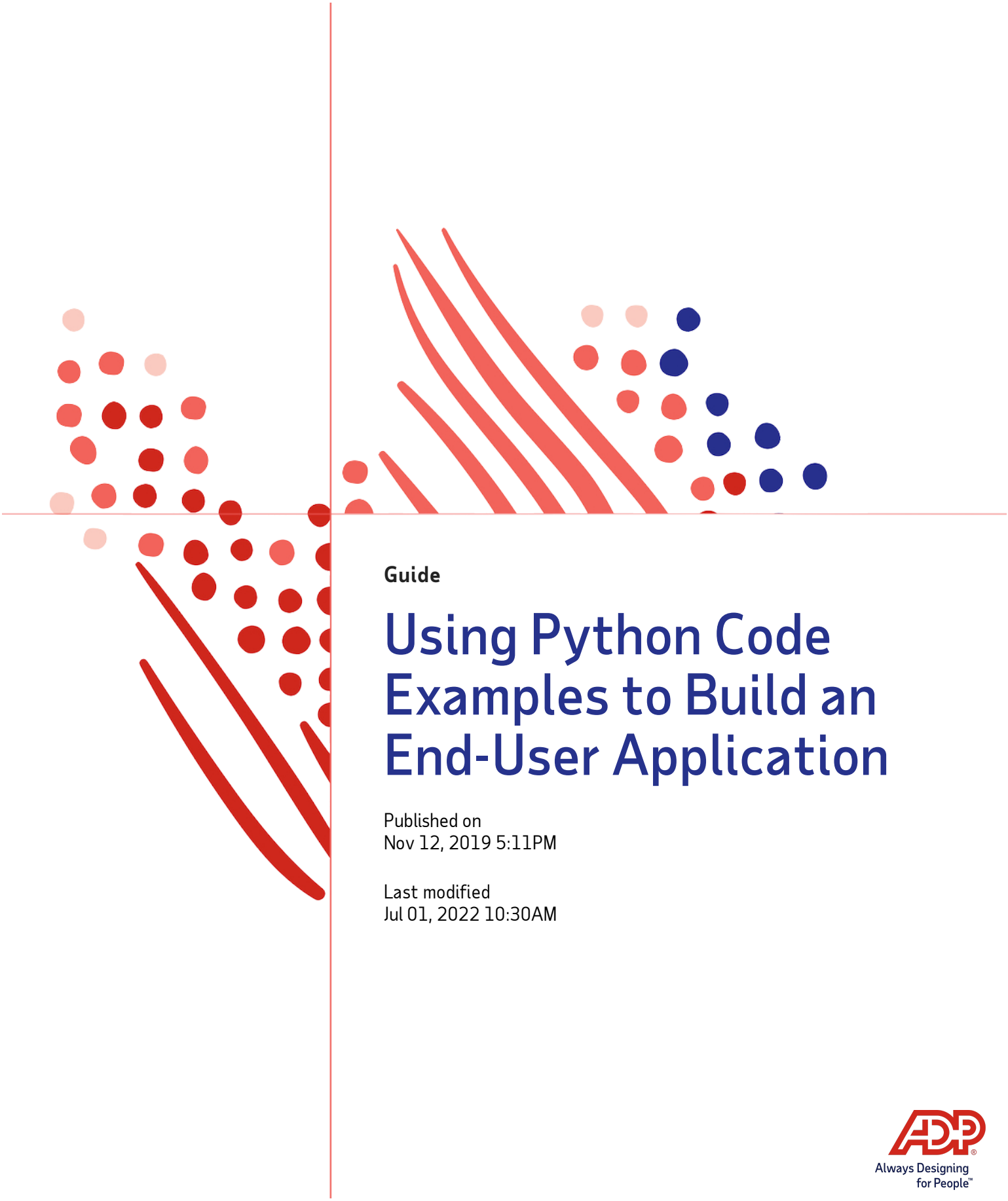


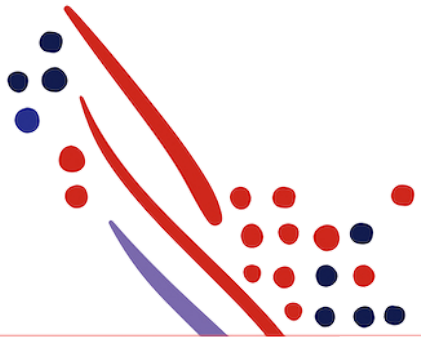
Guide

Using Python Code Examples to Build an End-User Application

Published on
Nov 12, 2019 5:11PM

Last modified
Jul 01, 2022 10:30AM

Always Designing for People™



ADP Copyright Information

ADP, the ADP logo, and Always Designing for People are trademarks of ADP, Inc.

Windows is a registered trademark of the Microsoft Corporation.

All other trademarks are the property of their respective owners.

Copyright © 2022 ADP, Inc. ADP Proprietary and Confidential - All Rights Reserved. These materials may not be reproduced in any format without the express written permission of ADP, Inc.

These materials may not be reproduced in any format without the express written permission of ADP, Inc. ADP provides this publication "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. ADP is not responsible for any technical inaccurancies or typographical errors which may be contained in this publication. Changes are periodically made to the information herein, and such changes will be incorporated in new editions of this publication. ADP may make improvements and/or changes in the product and/or the programes described in this publication.

Published on
Nov 12, 2019 5:11PM

Last modified
Jul 01, 2022 10:30AM

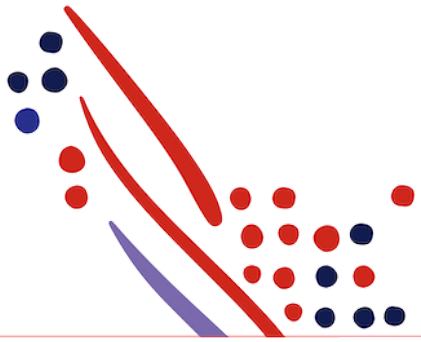


Table of Contents

Chapter 1

Step 1: Create a Connection Configuration - Python

Specify your connection configuration including the following: grant type, client id, client secret, client certificate and callback uri

Note

This code sample is intended to allow a developer to quickly download the library and test the sample app provided in our staging environment. Because of this, the URLs provided in this code sample are for our staging environment, and the clientID and clientSecret provided will only work in the staging environment.

When you are ready to build and test your app against production, you will need to use the clientID and clientSecret provided to your organization by ADP, and the production URLs beginning with: https://api.adp.com.

```
# Create the config dictionary variable
config = dict({})
config['clientID'] = '88a73992-07f2-4714-ab4b-de782acd9c4d'
config['clientSecret'] = 'a130adb7-aa51-49ac-9d02-0d4036b63541'
config['sslCertPath'] = 'certs/cert.pem'
config['sslKeyPath'] = 'certs/cert.key'
config['tokenServerURL'] = 'https://iat-api.adp.com/auth/oauth/v2/token'
config['disconnectURL'] = 'https://iat-accounts.adp.com/auth/oauth/v2/logout'
config['apiRequestURL'] = 'https://iat-api.adp.com'
config['grantType'] = 'client_credentials'

# Initialize and create the config object. Since we are using the authorization_code
# grantType we will receive a ClientCredentialsConfiguration object.
ClientCredentialsConfiguration = ConnectionConfiguration().init(config)
```

Chapter 2

Step 2: Create a Connection Object - Python

Create a connection based on your grant type using the connection factory

```
# Using the new configuration object create a connection
acConnection = ADPAPIConnectionFactory().createConnection(AuthorizationCodeConfiguration)
```

Chapter 3

Step 3: Connect to the ADP API Gateway - Python

Establish connection and retrieve your access token

```
#  For the Authorization Code type of application the user has to
#  authenticate and authorize themselves with ADP. So first obtain
#  the authorization URL to redirect the user

authURL = acConnection.getAuthorizationURL()

#  Respond to the user's browser http request with a 302 to the authURL obtained above.

#  After the user's authentication and authorization attempt they
#  will be redirected to the callback url of your application.
#  The callback route handler must extract the authorization code
#  from the url query parameters

parsed_url = urlparse(self.path)
query = parsed_url.query
query_components = parse_qs(query)
code = query_components['code'][0]

# Update the acConnection's configuration object with the code

acConnection.getConfig().setAuthorizationCode(code)

# Finally connect to ADP

acConnection.connect()
```

Chapter 4

Step 4: Access ADP APIs - Python

Use the connection from the previous step to work with API Product

```
if (acConnection.isConnectedIndicator()):
# Obtain a helper object for the user_info APIProduct
userInfoHelper = UserInfoHelper(ccConnection)

# Use the helper to get the userinfo JSON object.
# This interally calls the userinfo ADP API.
userinfo = userInfoHelper.getUserInfo()
for aKey in userinfo.keys():
print aKey + ': ' + userinfo[aKey]
```