

amazon dash

Replenishment

Grow your business with Dash Replenishment



開発者はDRSをどう活用するのか？

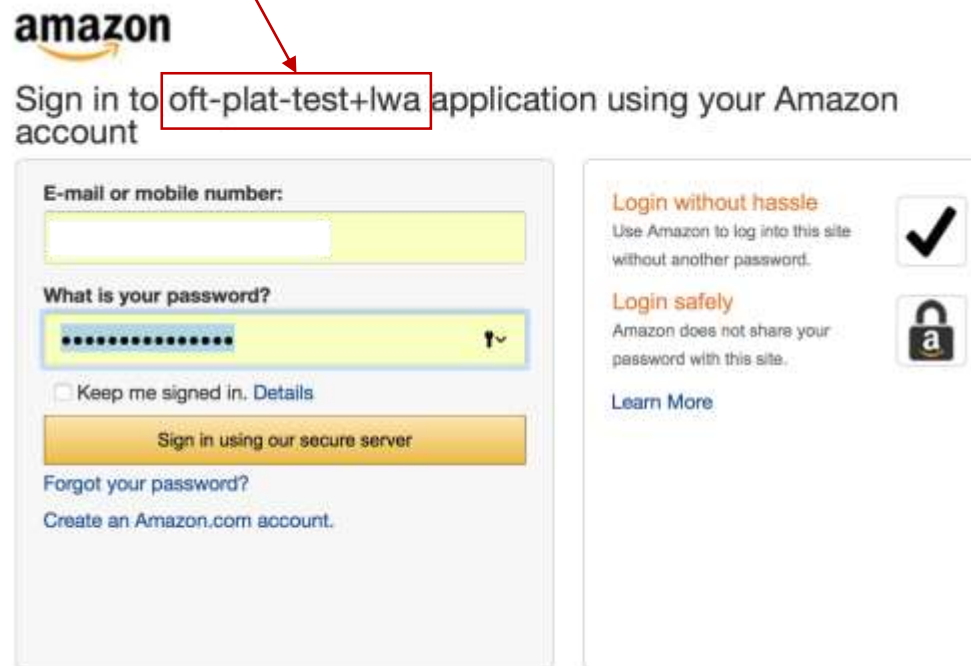
- DRSは3つの重要な技術で構成されている
 - Login With Amazon (LWA)
 - DRS Restful API エンドポイント
 - 非同期のSNS 通知サービス
- この会では、実装の観点からDRSがどのように動作するのか、いくつかポイントをご紹介します。

Login With Amazon (LWA)

- LWAは、既存のAmazonアカウントの仕組みで、開発者が容易に使えるようにします。
 - モバイルSDK (iOS & Android) やJAVAウェブSDKを使用してLWAを呼び出すことができます。
 - <https://developer.amazon.com/public/solutions/devices/dash-replenishment-service/>
 - LWAはOAuth2 スタンダードベースです。
 - Amazonアカウントでサインインし開発者用の申し込みに同意すると、LWAは認証後のフローをウェブページで提供します。
 - そのウェブページにて製品を選択します。
 - すべてのフローが完了すると、LWAはアクセスコードを返します。
 - APIエンドポイントを使用する為、access_token/refresh_token とアクセスコードを交換します。

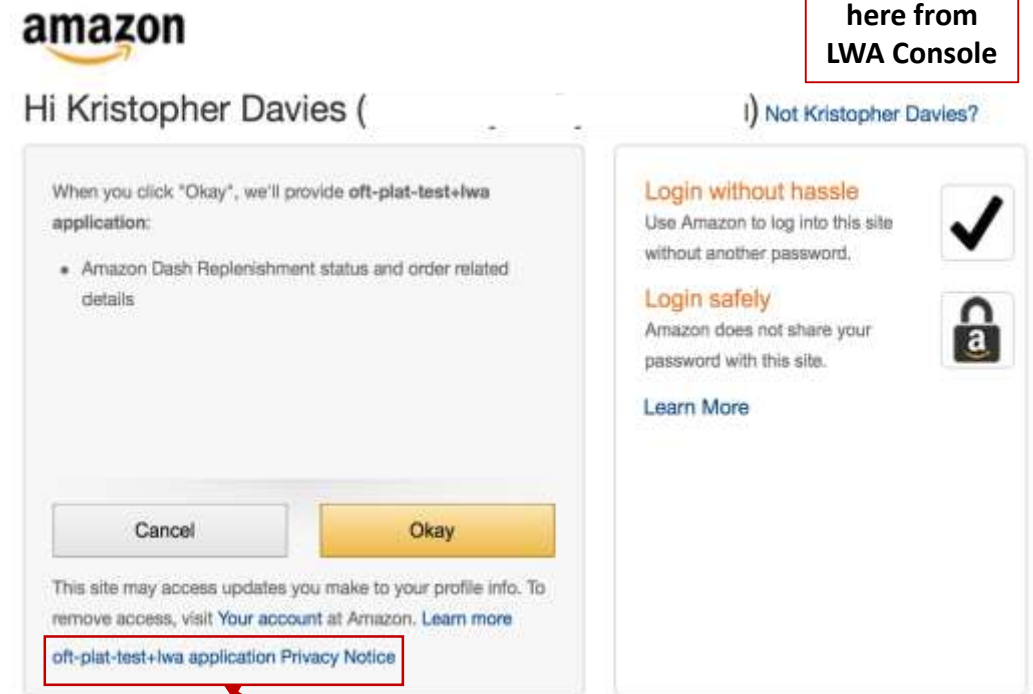
Login with Amazon Post Authorization Flow 1

Developer Configurable in LWA console



The image shows the Amazon login page. At the top left is the Amazon logo. Below it, the text "Sign in to **oft-plat-test+lwa** application using your Amazon account" is displayed. The "oft-plat-test+lwa" text is highlighted with a red box, and a red arrow points from the "Developer Configurable in LWA console" text above to it. The login form includes a field for "E-mail or mobile number:", a field for "What is your password?" with a visibility toggle, a checkbox for "Keep me signed in. Details", and a "Sign in using our secure server" button. Below the button are links for "Forgot your password?" and "Create an Amazon.com account.". To the right of the form are two security options: "Login without hassle" (checked) and "Login safely" (unchecked), each with a description and a "Learn More" link.

Your Logo
here from
LWA Console

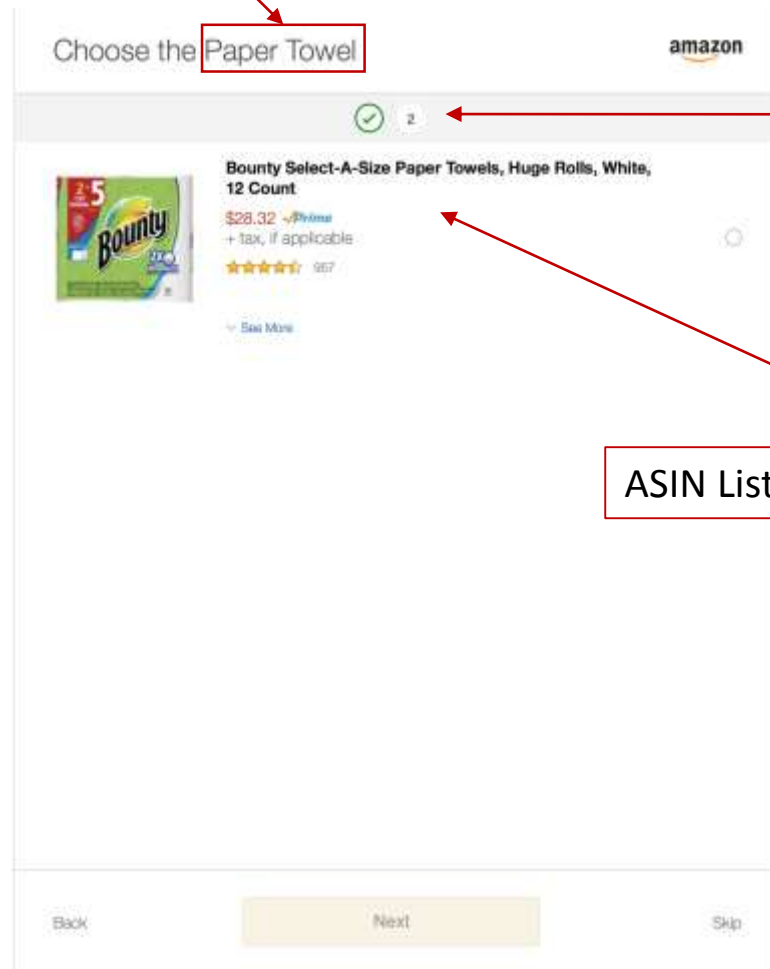


The image shows the Amazon post-authorization flow. At the top left is the Amazon logo. Below it, the text "Hi Kristopher Davies (i) Not Kristopher Davies?" is displayed. The main content area has a heading "When you click 'Okay', we'll provide oft-plat-test+lwa application:" followed by a list item "Amazon Dash Replenishment status and order related details". Below this are "Cancel" and "Okay" buttons. At the bottom, there is a privacy notice: "This site may access updates you make to your profile info. To remove access, visit Your account at Amazon. Learn more oft-plat-test+lwa application Privacy Notice". The "oft-plat-test+lwa application Privacy Notice" text is highlighted with a red box, and a red arrow points from the "Developer Configurable in LWA Console" text below to it. To the right of the main content area are the same "Login without hassle" and "Login safely" options as in the previous image.

Developer Configurable in LWA Console

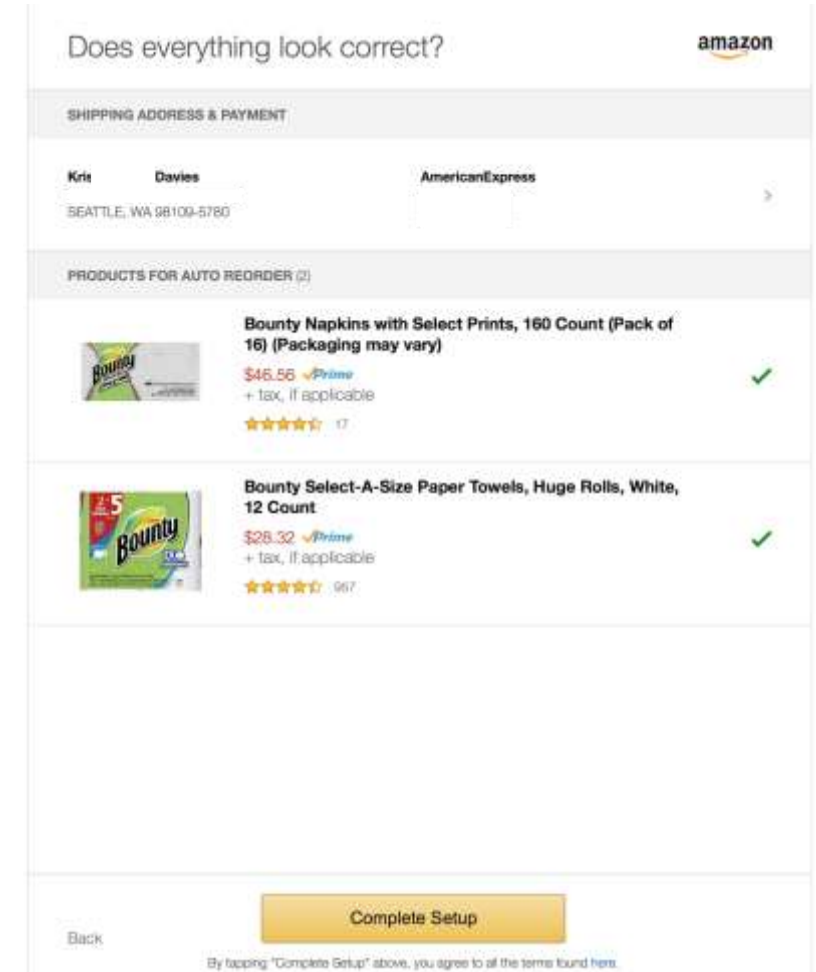
Login with Amazon Post Authorization Flow 2

Slot Name Configurable in DRS Console



Slot Counter

ASIN List Configurable in DRS Console



DRS API エンドポイント

- DRDにある5つのhttps エンドポイント
 - Replenish – お客さまの商品を注文
 - SlotStatus - 使用状況をAmazonに通知 (「毎日」の設定が望ましい)
 - DeviceStatus – デバイスがオンラインである事をAmazonに通知 (「毎日」の設定が望ましい)
 - Deregistration – AmazonアカウントからDRSの登録を解除
 - SubscriptionInfo – お客さまの登録状況と詳細を問合せ

SNS 通知

- Amazon Simple Notification Service (SNS) は、開発者のアプリやサービスに非同期でアクションを通知します。
 - <https://aws.amazon.com/sns/>
- DRSの6つのSNS 通知
 - Order Placed – Replenishment API call was received/processed without error
 - Order Cancelled – Customer has cancelled an order received on replenishment endpoint
 - Device Registered – Device Registration Successful
 - Device Deregistered – Customer has deregistered the device from their account
 - Subscription Changed – Customer has modified their subscription
 - Item Shipped Notification – Item for a successful order has been shipped (Includes customer promise date)

Device/App CX ガイ ド ラ イ ン

- <https://developer.amazon.com/public/solutions/devices/dash-replenishment-service/docs/dash-customer-experience-guidelines>
- DRSを利用する為の様々な重要な事項が含まれているリンクです。

ティザーページの作り方

- あなたのデバイスのヒーローイメージ、再注文する製品の詳細をフォーラムに作成します。
- Amazonはティザーページを作成し、コンポジットIDをあなたにメールします。

お客さまはDRSをどう使うのか？

- お客さまは**DRS**が搭載されたデバイスを購入します。
- お客さまデバイスのネットワーク設定をします。
- お客さまは**3P**アプリを利用して製品選択の為**LWA**フローをたどります。
- お客さまはアプリ内で設定を完了します。
- お客さまはデバイスを使用します。
- デバイスが消費を確認し、**3P**サービスがその状態を利用します。
- デバイスや**3P**サービスが閾値に到達したことを感知し、**Amazon**へ注文します。

お客さまはどのようにDRSを使うのか？

- Amazonはお客さまにEメールを送り、注文されたことを通知します。
- お客さまは**24時間**の注文猶予があります。
 - 何もしない – 注文はそのままにし、商品がお客さまの登録した住所に発送されます。
 - 注文取消 – お客さまはAmazonのマイオーダーメニューで注文を取り消します。取り消した注文の情報はSNS通知でデバイスへ送られます。
- お客さまは製品を受け取り、商品を交換し使用状況を確認します。

Thank you!

何ができるか楽しみにしています！

- <https://www.hackster.io/contests/DRS>

Appendix – Sample iOS Code

For each API in the DRS service we have sample iOS code using the newest LWA 3.0.1 SDK available at <https://developer.amazon.com>

LWA Login Sample iOS Code

```
- (void)login {  
    AMZNAuthorizationManager *manager = [AMZNAuthorizationManager sharedManager];  
    AMZNAuthorizeRequest *request = [[AMZNAuthorizeRequest alloc] init];  
  
    request.interactiveStrategy = AMZNInteractiveStrategyAuto;  
  
    NSMutableArray *scopeObjectArray = [NSMutableArray array];  
    NSString *requestScopes = @"dash:replenish";  
    NSMutableDictionary *options = [NSMutableDictionary new];  
    options[@"serial"] = @"DEVICE_SERIAL_NUMBER";  
    options[@"device_model"] = @"DEVICE_MODEL_NAME";  
  
    id<AMZNScope> drsScope = [AMZNScopeFactory scopeWithName:requestScopes data:options];  
    [scopeObjectArray addObject:drsScope];  
  
    request.scopes = scopeObjectArray;  
    request.grantType = AMZNAuthorizationGrantTypeCode;  
    request.codeChallenge = @"CODE_CHALLENGE";  
    request.codeChallengeMethod = @"S256";  
    [manager authorize:request withHandler:[self requestLoginHandler]];  
}
```

Login Sample iOS Code cont'd

```
- (AMZNAuthorizationRequestHandler)requestLoginHandler {
    AMZNAuthorizationRequestHandler requestHandler = ^(AMZNAuthorizeResult * result, BOOL userDidCancel,
    NSError * error) {
        if (error) {
            NSString *errorMessage = error.userInfo[@"AMZNLWLErrorNonLocalizedDescription"];
            //handle login error
        } else if (userDidCancel) {
            //handle user cancellation
        } else {
            // Authentication was successful. Obtain the user profile data.
            NSString *authCode = result.authorizationCode;
            NSString *clientId = result.clientId;
            NSString *redirectUri = result.redirectUri;
            //request the access and the refresh token
        }
    };
    return [requestHandler copy];
}
```

... continued next slide

Logout Sample iOS Code

```
- (void)logout {
    AMZNAuthorizationManager *manager = [AMZNAuthorizationManager sharedManager];
    [manager signOut:[self requestLogoutHandler]];
}

- (AMZNSignOutRequestHandler)requestLogoutHandler {
    AMZNSignOutRequestHandler requestHandler = ^(NSError * _Nullable error) {
        if(error) {
            NSString *errorMessage = error.userInfo[@"AMZNLWAEErrorNonLocalizedDescription"];
            //handle logout error
        } else {
            //handle successful logout. Delete the refresh and the access token.
        }
    };
    return [requestHandler copy];
}
```

Replenish Sample iOS Code

```
- (void)initiateReplenish {
    NSURL *url = @"https://dash-replenishment-service-na.amazon.com/replenish/SLOT_ID"; // your slot ID here
    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:url];
    [request addValue:@"com.amazon.dash.replenishment.DrsReplenishResult@1.0" forHTTPHeaderField:@"x-amzn-accept-type"];
    [request addValue:@"com.amazon.dash.replenishment.DrsReplenishInput@1.0" forHTTPHeaderField:@"x-amzn-type-version"];
    [request setHTTPMethod:@"POST"];

    //build session
    NSString *authorizationToken = [NSString stringWithFormat:@"%s", @"Bearer", @"ACCESS_TOKEN"]; //your access token
    NSURLSessionConfiguration *config = [NSURLSessionConfiguration defaultSessionConfiguration];
    config.HTTPAdditionalHeaders = @{@"Authorization": authorizationToken,};
    NSURLSession *session = [NSURLSession sessionWithConfiguration:config];

    NSURLSessionDataTask *task = [session dataTaskWithRequest:request completionHandler:[self handler]];
    [task resume];
    [session finishTasksAndInvalidate];
}

- (void (^)(NSData *, NSURLResponse *, NSError *))handler {
    __block id safeBlockSelf = self;
    return ^(NSData *data, NSURLResponse *response, NSError *error) {
        if (error) {
            //handle error
            [safeBlockSelf handleError];
        } else {
            //handle success
            [safeBlockSelf handleSuccess];
        }
    };
}
```

Slot Status Sample iOS Code

```
- (void)sendSlotStatus {
    NSURL *url = @"https://dash-replenishment-service-na.amazon.com/slotStatus/SLOT_ID"; // your slot ID here
    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:url];
    [request addValue:@"com.amazon.dash.replenishment.DrsSlotStatusResult@1.0" forHTTPHeaderField:@"x-amzn-accept-type"];
    [request addValue:@"com.amazon.dash.replenishment.DrsSlotStatusInput@1.0" forHTTPHeaderField:@"x-amzn-type-version"];
    [request addValue:@"application/json" forHTTPHeaderField:@"Content-Type"];
    [request setHTTPMethod:@"POST"];

    //build session
    NSString *authorizationToken = [NSString stringWithFormat:@"%s", @"Bearer", @"ACCESS_TOKEN"]; //your access token
    NSURLSessionConfiguration *config = [NSURLSessionConfiguration defaultSessionConfiguration];
    config.HTTPAdditionalHeaders = @{@"Authorization": authorizationToken,};
    NSURLSession *session = [NSURLSession sessionWithConfiguration:config];

    NSURLSessionDataTask *task = [session dataTaskWithRequest:request completionHandler:[self handler]];
    [task resume];
    [session finishTasksAndInvalidate];
}
```

...continued next slide

Slot Status Sample iOS Code Cont'd

```
- (void (^)(NSData *, NSURLResponse *, NSError *))handler {
    __block id safeBlockSelf = self;
    return ^(NSData *data, NSURLResponse *response, NSError *error) {
        if (error) {
            //handle error
            [safeBlockSelf handleError];
        } else {
            //handle success
            [safeBlockSelf handleSuccess];
        }
    };
}
```

Device Status Sample iOS Code

```
- (void)sendDeviceStatus {
    NSURL *url = @"https://dash-replenishment-service-na.amazon.com/deviceStatus";
    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:url];
    [request addValue:@"com.amazon.dash.replenishment.DrsDeviceStatusResult@1.0" forHTTPHeaderField:@"x-amzn-accept-type"]
    [request addValue:@"com.amazon.dash.replenishment.DrsDeviceStatusInput@1.0" forHTTPHeaderField:@"x-amzn-type-version"];
    [request addValue:@"application/json" forHTTPHeaderField:@"Content-Type"];
    [request setHTTPBody:[self getDeviceStatusData]];
    [request setHTTPMethod:@"POST"];

    //build session
    NSString *authorizationToken = [NSString stringWithFormat:@"%s%s", @"Bearer", @"ACCESS_TOKEN"]; //your access token
    NSURLSessionConfiguration *config = [NSURLSessionConfiguration defaultSessionConfiguration];
    config.HTTPAdditionalHeaders = @{@"Authorization": authorizationToken,};
    NSURLSession *session = [NSURLSession sessionWithConfiguration:config];

    NSURLSessionDataTask *task = [session dataTaskWithRequest:request completionHandler:[self handler]];
    [task resume];
    [session finishTasksAndInvalidate];
}
...continued next slide
```

Device Status Sample iOS Code Cont'd

```
/*
 * return The device status data.
 */
- (NSData *)getDeviceStatusData {
    return [SAMPLE_DEVICE_STATUS_DATA dataUsingEncoding:NSUTF8StringEncoding]; //for example: @"
    {"mostRecentlyActiveDate\":"2015-06-01\"}"
}

- (void (^)(NSData *, NSURLResponse *, NSError *))handler {
    __block id safeBlockSelf = self;
    return ^(NSData *data, NSURLResponse *response, NSError *error) {
        if (error) {
            //handle error
            [safeBlockSelf handleError];
        } else {
            //handle success
            [safeBlockSelf handleSuccess];
        }
    };
}
```

Deregister Sample iOS Code

```
- (void)deregisterDevice {  
    NSURL *url = @"https://dash-replenishment-service-na.amazon.com/registration";  
    NSMutableURLRequest *request = [[NSMutableURLRequest alloc] initWithURL:url];  
    [request addValue:@"com.amazon.dash.replenishment.DrsDeregisterResult@1.0" forHTTPHeaderField:@"x-amzn-accept-type"];  
    [request addValue:@"com.amazon.dash.replenishment.DrsDeregisterInput@2.0" forHTTPHeaderField:@"x-amzn-type-version"];  
    [request addValue:@"application/json" forHTTPHeaderField:@"Content-Type"];  
    [request setHTTPMethod:@"DELETE"];  
  
    //build session  
    NSString *authorizationToken = [NSString stringWithFormat:@"%s%s", @"Bearer", @"ACCESS_TOKEN"]; //your access token  
    NSURLSessionConfiguration *config = [NSURLSessionConfiguration defaultSessionConfiguration];  
    config.HTTPAdditionalHeaders = @{@"Authorization": authorizationToken,};  
    NSURLSession *session = [NSURLSession sessionWithConfiguration:config];  
  
    NSURLSessionDataTask *task = [session dataTaskWithRequest:request completionHandler:[self handler]];  
    [task resume];  
    [session finishTasksAndInvalidate];  
}
```

...continued next slide

Deregister Sample iOS Code Cont'd

```
- (void (^)(NSData *, NSURLResponse *, NSError *))handler {
    __block id safeBlockSelf = self;
    return ^(NSData *data, NSURLResponse *response, NSError *error) {
        if (error) {
            //handle error
            [safeBlockSelf handleError];
        } else {
            //handle success
            [safeBlockSelf handleSuccess];
        }
    };
}
```