

iOS In-App Purchasing:

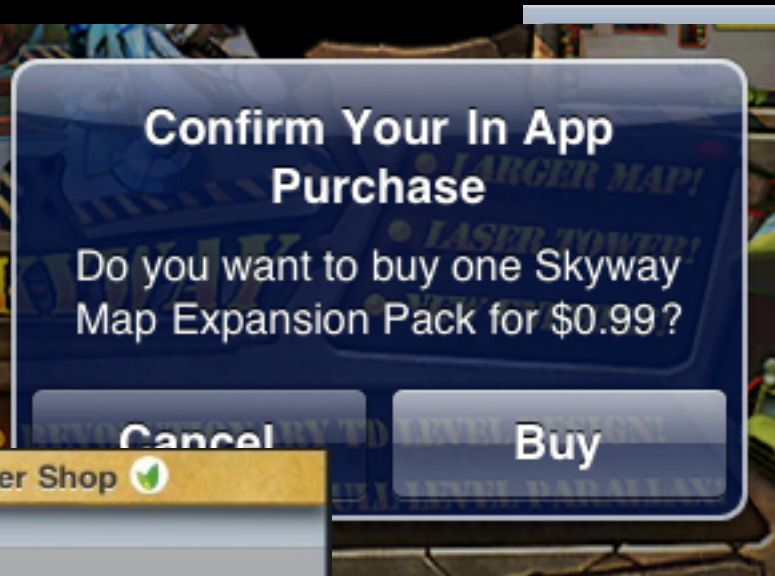


mobiledevnj

@techieGary

In App Purchase

- Premium Features
- New content:
 - data (subscriptions, books, currency, etc)
 - levels (games, challenges, etc)
 - properties (notable example: Paper App)



Flower Shop

Fertilizer

- Fertilizer (20 doses) \$0.99
- Fertilizer (70 doses) \$0.99
- Fertilizer (150 doses) \$0.99

Seeds

- Seeds of Gratitude \$0.99 **NEW**
- Seeds of Love \$0.99

Navigation: Garden, Bouquet, Seeds, Flower Shop

Content

5. Zombie Farm FREE

The Playforge, LLC
★★★★★ 35593 Ratings

6. Words With Friends \$2.99

Newtoy Inc.
★★★★★ 4633 Ratings

7. Smurfs' Village + FREE

Capcom Interactive, Inc.
★★★★★ 49391 Ratings

iPod 12:51 PM

Top In-App Purchases

1	Unlimited Turns for TripleT...	\$6.99
2	2,000 Coins	\$0.99
3	10,000 Coins	\$2.99
4	5,000 Coins	\$1.99

Navigation: Featured, Categories, Top 25, Search, Updates (4)

Types

- Non-Consumable (purchased once, restorable, like app store)
- Consumable (can be purchased multiple times)
- Free Subscription (recurring)
- Auto-Renewable subscription (opt-out to stop)
- Non-Renewable subscription (requires users to renew)

IAP Content: Embedded

- Easy to develop
- No Server Costs
- Fatter Apps (may not be downloadable w/o USB cable)

IAP Content: Server-Based

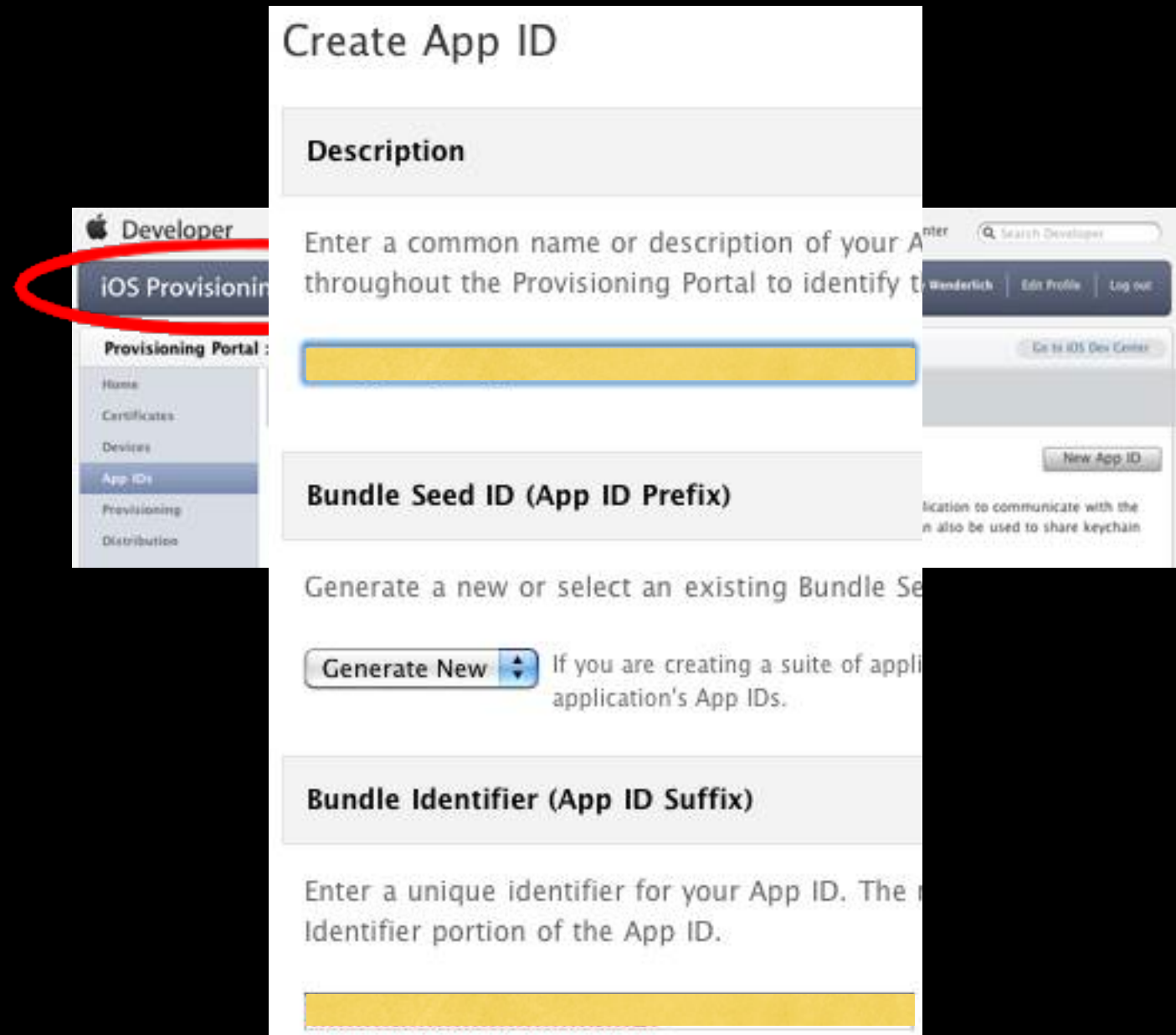
- Dynamic content
- Out of band verification (Apple WS)
- Server costs
- Added complexity (versioning, updates, etc)

How it works?

- Get product list
- Request product list from iTunesConnect using product list identifiers
- Display list to User - User Select Option
- Create/Submit Payment Request
 - Take Action based on response (async)

SKProduct

- Every product has an identifier
- Need icon, screenshot



The image shows a screenshot of the Apple Developer portal's 'Create App ID' form. On the left, a sidebar menu is visible with the 'iOS Provisioning' option highlighted by a red circle. The main form area is titled 'Create App ID' and contains several sections:


- Description:** A text input field with the placeholder text 'Enter a common name or description of your App ID. This name will be used throughout the Provisioning Portal to identify the App ID.'
- Bundle Seed ID (App ID Prefix):** A section with the instruction 'Generate a new or select an existing Bundle Seed ID'. It includes a 'Generate New' button and a dropdown menu. A note states: 'If you are creating a suite of applications, you can use the same Bundle Seed ID for all application's App IDs.'
- Bundle Identifier (App ID Suffix):** A section with the instruction 'Enter a unique identifier for your App ID. The Bundle Identifier portion of the App ID.' followed by a text input field.

Night Before Poet — In-App Purchases

Pricing and Availability

Enter the pricing and availability details for this In-App Purchase below.

Cleared for Sale **Yes** ☒ **No** ☐

Price Tier Select 



[View Pricing Matrix](#)

Click Add Language to get started.



Get list of product
IDs from Server

Return
Product List

Send Product IDs
to App Store for
Description and
Prices

Return
Descriptions and
Prices to App

SKProductRequest



Display "Store"



Sends Payment
Request



App Store sends
Response
(Purchase, Fail,
Restore)



Take action on
Response



Store txn id and
receipt info

SKPaymentTransaction

**SKPayment
SKProductQueue**



Out of band
Payment Validation
(WebService)

App Store
validates
Transaction

Developer server
delivers the
requested content



Free + IAP

- In-App Purchase 2011 \$2.1 Billion
- Wider distribution
- Lower ratings
- Can only test on Device

Follow:
@MobileDevNJ
@TechieGary

// Payment Notifications

```
[[SKPaymentQueue defaultQueue] <your  
processing class here>];
```

// Delegates required

```
<SKProductsRequestDelegate,  
SKPaymentTransactionObserver>
```

```
// SKProductRequest
```

```
- (void)requestProducts {
```

```
    self.request = [[[SKProductsRequest alloc]  
initWithProductIdentifiers:_productIdentifiers]  
autorelease];
```

```
_request.delegate = self;  
[_request start];
```

```
}
```



```
// SKPayment added to SKPaymentQueue
```

```
- (void)buyProductIdentifier:(NSString  
*)productIdentifier {
```

```
    NSLog(@"Buying %@...", productIdentifier);
```

```
    SKPayment *payment = [SKPayment  
paymentWithProductIdentifier:productIdentifier]
```

```
[[SKPaymentQueue defaultQueue]  
addPayment:payment];  
}
```

// SKPaymentQueue Delegate

```
- (void)paymentQueue:(SKPaymentQueue *)queue
updatedTransactions:(NSArray *)transactions
{
    for (SKPaymentTransaction *transaction in transactions)
    {
        switch (transaction.transactionState)
        {
            case SKPaymentTransactionStatePurchased:
                [self completeTransaction:transaction];
                break;
            case SKPaymentTransactionStateFailed:
                [self failedTransaction:transaction];
                break;
            case SKPaymentTransactionStateRestored:
                [self restoreTransaction:transaction];
            default:
                ;
        }
    }
}
```

- Introductions
- What u are working on!