

Google™ 



Introducing Google In-App Payments for the Web

Amit Fulay – Product Manager
Mikhail Seregine & Luke Bayes – Software Engineer
05.11.11



Session Details

- Feedback: <http://goo.gl/RtNlf>
- Hashtags: #io2011, #Commerce
- Available on YouTube in 24 hours



Next 45 minutes....



The Road to Moscone....

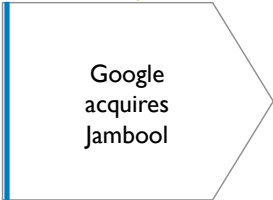
Social Gold Joins Google – A Letter From Our Founders



50 tweets
retweet



August 13, 2010



Projects >
In-App Payments
January 3, 2011
We are building an in-app payments solution for digital & virtual goods.
Team
Product: Amit Fulay, Benjamin Lee
Engineering: Mikhail Seregine, Dan Zink, Luke Bayes, Ali Mills, Ryan Kuykendall, Joel Leitch
UX: Matt Terich
Ops: Gina Del Vecchio, Kiran Babu
Marketing: Jessica Smallman

The web has evolved &
so has the way we make money....



eCommerce



Ads



INTERACTIVE ADVERTISING
MEDIA MORE ENGAGING

Interactive Advertising Bureau

REACH THE RIGHT AUDIENCE → CREATING A GREAT EXPERIENCE → MEASURE YOUR EFFECTIVENESS

HOW-TO's

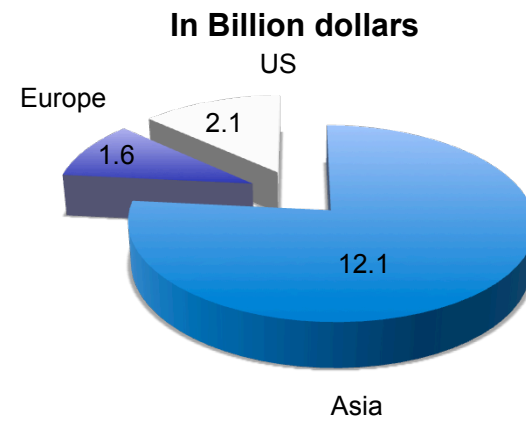
- DIGITAL VIDEO
- DISPLAY
- EMAIL
- GAMING
- LEAD GENERATION
- LOCAL
- MOBILE
- SEARCH

Interactive Advertising: At the center of every powerful campaign.

By using interactive marketers can target the audience that matters to their brand, allow their customers to experience the brand story and then track to and then measure the effectiveness of their marketing dollars.

This site provides a roadmap for how you can integrate

Virtual Goods



Source: eMarketer.com

eCommerce Wave

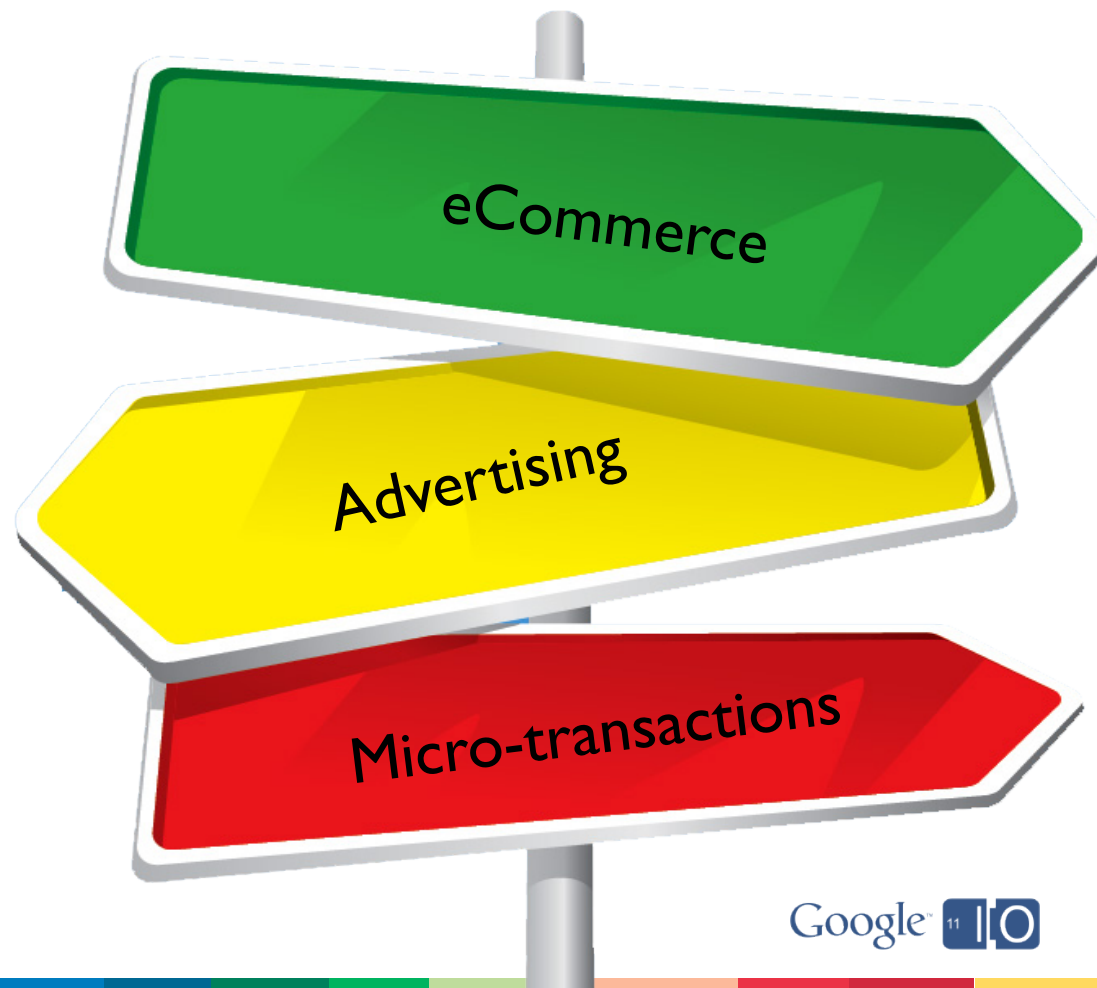
Models: B2B, B2C

Advertising Wave

Models: CPM, CPC

Micro-transactions Wave

Models: In-App, virtual goods



Connected Users



- 10 Billion Social Networking and Online World Accounts created worldwide
- 4.5 Billion accounts active

Users don't just buy goods



they buy experiences



Confluence of Trends



Micropayments + Connected Users + Experience-driven

=

Opportunity for Developers



Introducing



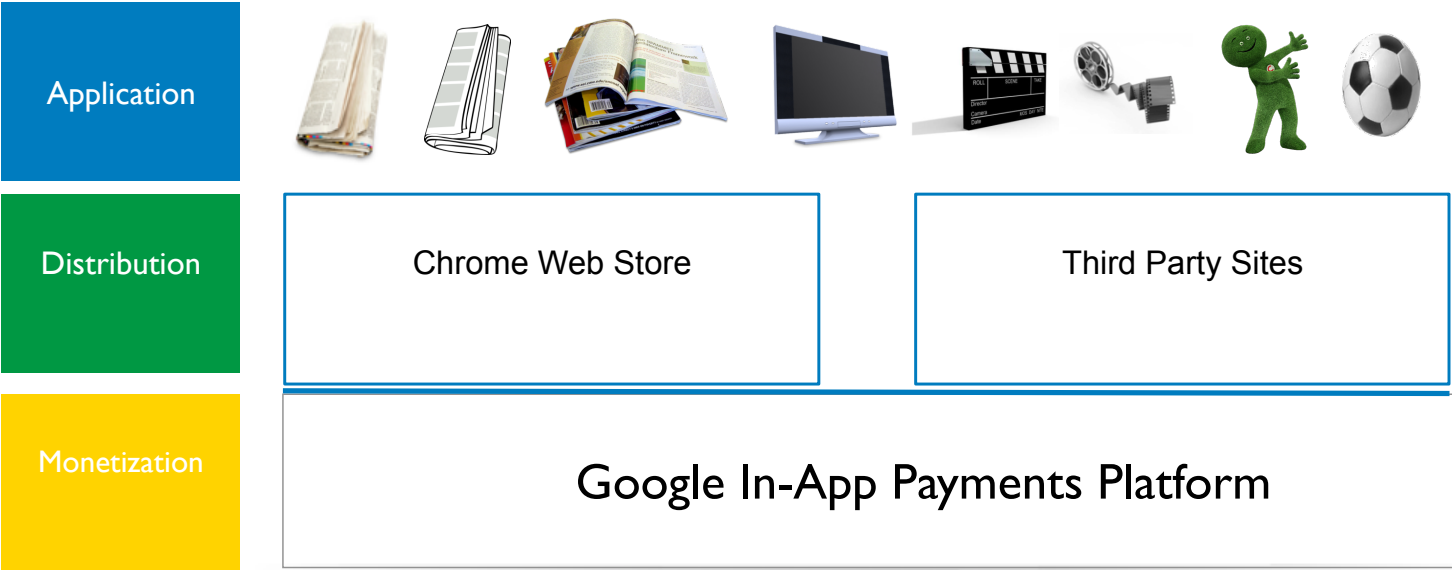
Google In-App Payments for the Web



Google I/O - Developer Launch

- Available today in sandbox for integration
- Flash API coming soon
- Going Live in Summer 2011

The WebApp Ecosystem



Friction-less UX
(in the app with minimal steps)

Simple API
(one call to pay, one call to collect)

Google Payments Infrastructure

Risk Management Analytics Compliance

Large User Base



What will it cost?



For some webapps, the cost of payments

30%



Wait...we charged less at Jambool

10%



But at Google we want to help you make money

5%



Pricing

- Flat 5% payments processing
- On Chrome Web Store & the entire web
- No fixed fee
- No licensing fee
- No minimum monthly fee
- Cost-effective for micro-transactions



Partner Demo

Jake McMahon

Director of Operations, Social Games



We make
life fun!





PopCap makes games that
everyone loves to play, anywhere
with anyone

Enduring Franchises



BEJEWELLED™

PLANTS
vs.
ZOMBIES™

ZUMA™

BOOK
WORM™

MYSTERY
P.I.™

Peggle™

PopCap's Reach



60 +

Million MAU



Bejeweled



2 +

Billion games played each month



PopCap In-App Purchasing



3 +

Million transactions a month

75¢



PopCap In-App Purchasing



75% +

Repeat purchasers

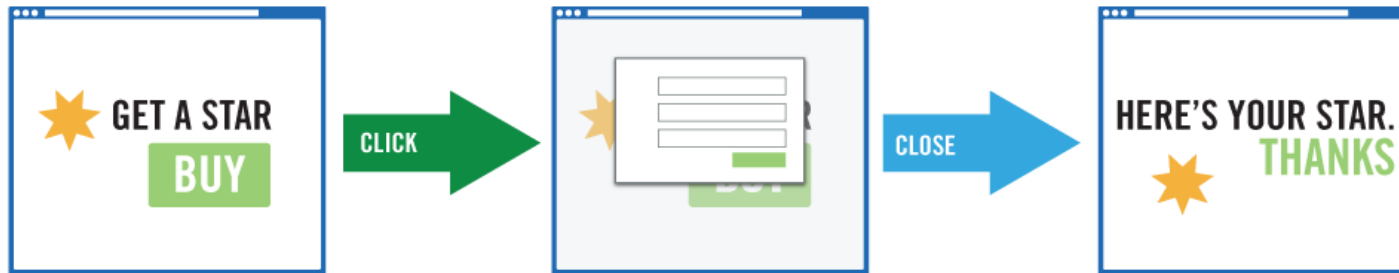




Integration in Action



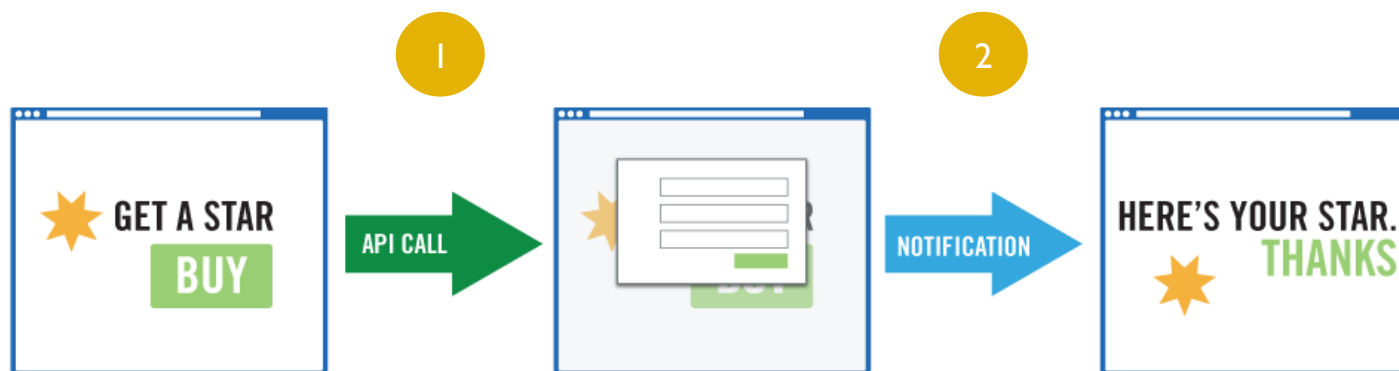
Click, pay, use



Monetize in two calls

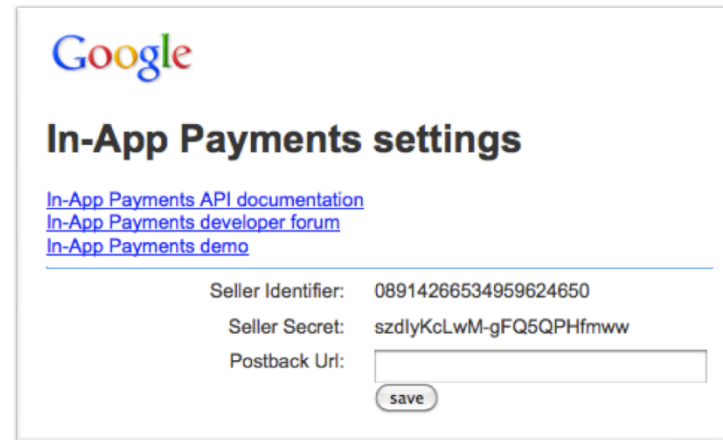
- 1 Open the payment screen
- 2 Accept the notification

Click, open payments screen, accept notification



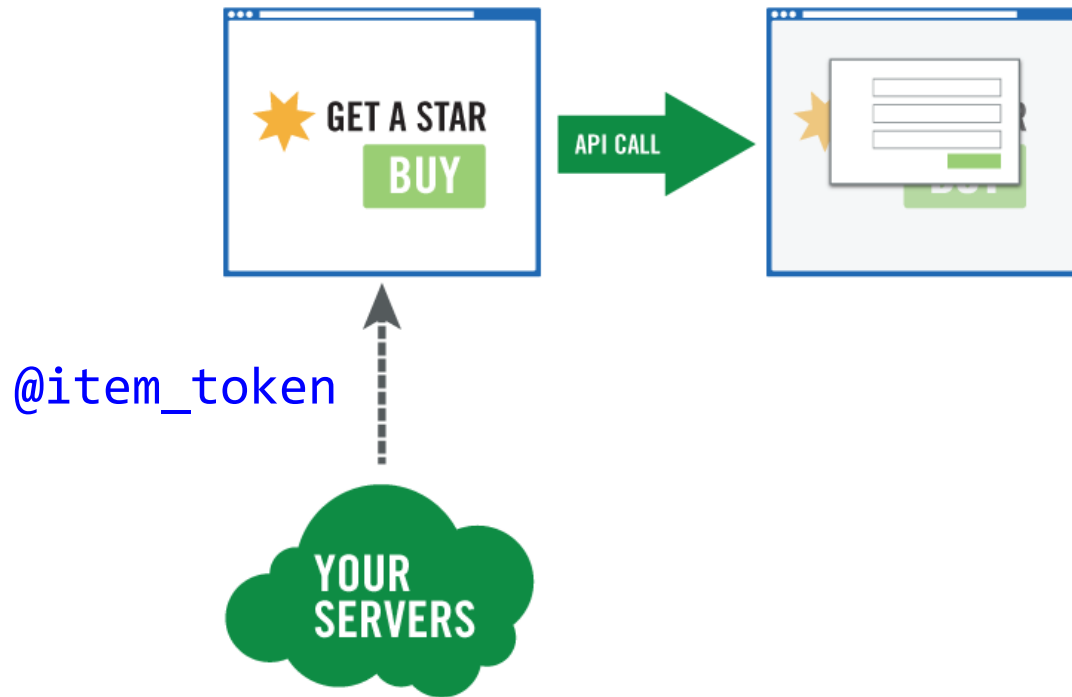
Before Integrating

- Sign up as a seller
- Get **seller id**
- Get **secret key**
- Configure postback URL



The screenshot shows the Google In-App Payments settings page. At the top is the Google logo. Below it is the title "In-App Payments settings". There are three links: "In-App Payments API documentation", "In-App Payments developer forum", and "In-App Payments demo". A horizontal line separates the links from the settings fields. The fields are: "Seller Identifier: 08914266534959624650", "Seller Secret: szdlyKcLwM-gFQ5QPHfmww", and "Postback Url: [input field]". Below the input field is a "save" button.

- 1 Open the payment screen



1 Open the payment screen: Server signs parameters

Define purchase parameters:

```
item = {  
  "itemName"           : "Gold Star",  
  "itemDescription"    : "A shining badge of distinction",  
  "itemPrice"          : "4.99",  
  "isoCurrencyCode"    : "USD",  
  
  "iss"                : "seller_id",  
  "aud"                : "goog.payments.inapp.buyItem",  
  "iat"                : Time.now.to_i.to_s  
}
```

And sign them:

```
@item_token = JWT.encode(item, secret_key)
```

I Open the payment screen: Client calls the API

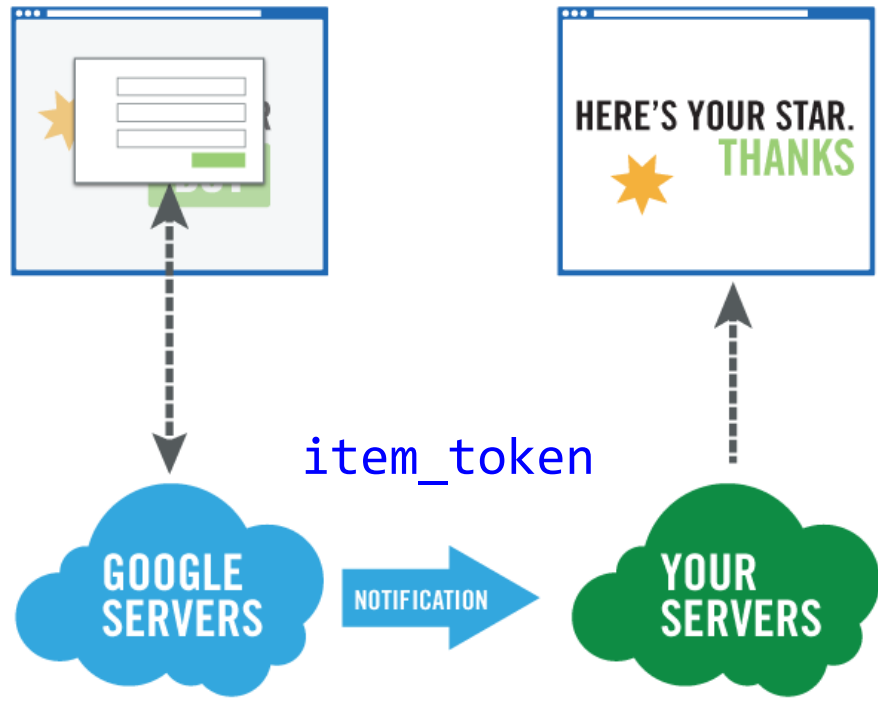
Add library to the page:

```
<script type="text/javascript" src="http://www.google.com/jsapi"></script>
<script type="text/javascript">
  google.load('payments', '0.1', {'packages': ['sandbox_config']});
</script>
```

Add a click handler, passing the JWT to buyItem:

```
<div class="my_buy_item"
  onClick="goog.payments.inapp.buyItem({'jwt': '<%= @item_token %>'});">
Buy a star</div>
```


2 Accept the notification



2

Accept the notification: Server

Handle notification from Google:

```
def payment_postback
  @item_token = params['jwt']
  item = JWT.decode(@item_token, secret_key)
  give_item_to_user(item)
  render :status => 200, :text => item.orderId

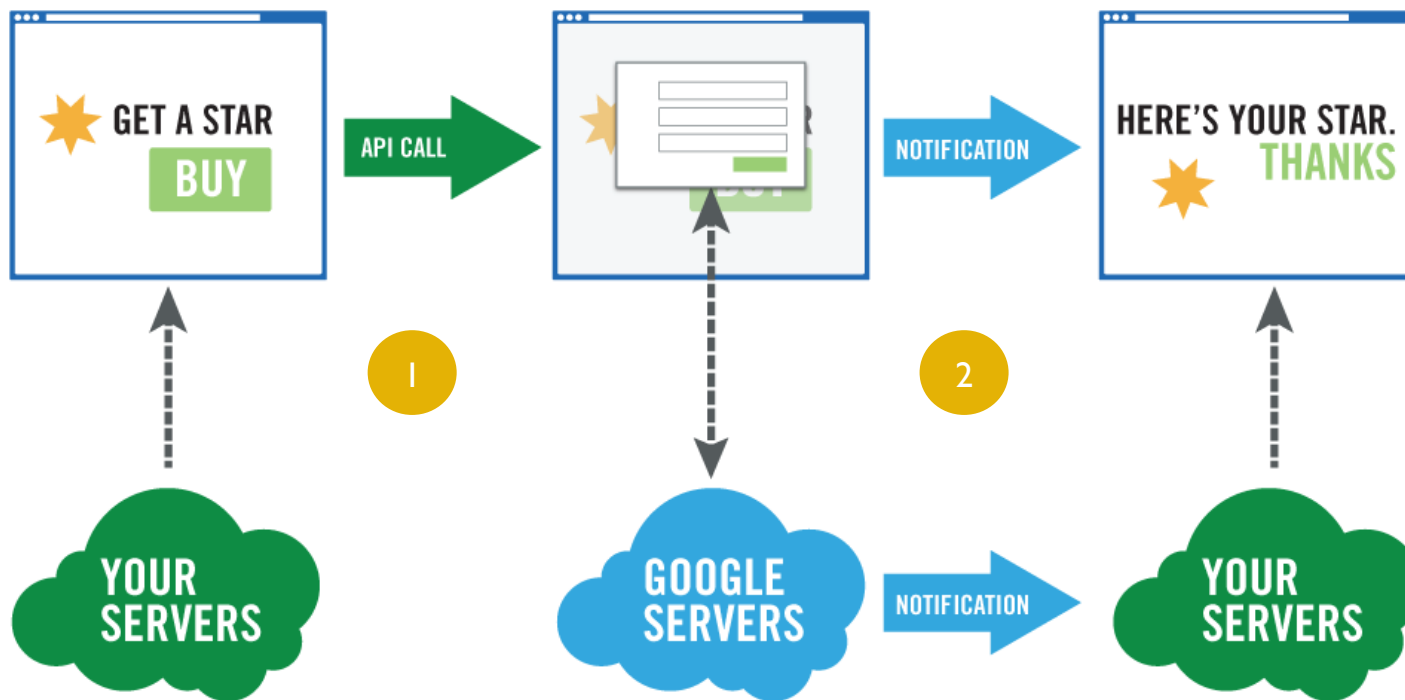
rescue JWT::DecodeError => e
  message = "Invalid token: " + @item_token
  render :status => 400, :text => message
end
```

2

Accept the notification: instant delivery

- Postbacks and polling
- Client callbacks
 - success and failure handlers
 - not authoritative: check with your server

The full picture



I Open the payments screen: Flash client setup

Compile the library into your SWF:

```
mxmclc --library-path=lib/GoogleInApp.swc GoldStars.as
```

Get the JWT for an item into your SWF:

```
<script type="text/javascript">  
  var so = new SWFObject("GoldStars.swf",  
    "a", "400", "200", "8", "#336699");  
  so.addParam("item_token", "<%= @item_token %>");  
  so.write("gold_stars");  
</script>
```

1 Open the payment screen: Server signs parameters

Define purchase parameters:

```
item = {
  "itemName"           : "Gold Star",
  "itemDescription"    : "A shining badge of distinction",
  "itemPrice"          : "4.99",
  "isoCurrencyCode"    : "USD",

  "iss"                : "seller_id",
  "aud"                : "goog.payments.inapp.buyItem",
  "iat"                : Time.now.to_i.to_s
}
```

And sign them:

```
@item_token = JWT.encode(item, secret_key)
```

1 Open the payment screen: Flash client calls the API

Call buyItem to launch the payment experience:


```
buyButton.addEventListener(MouseEvent.CLICK,  
    function(event:Event):void {  
        var item_token:String =  
            stage.loaderInfo.parameters.item_token;  
        var service:InAppApi = new InAppApi();  
        service.buyItem({'jwt': item_token});  
    }  
);
```

JSON Web Tokens

Coming from OAuth 2 to an API near you

- No HTML escaping problems
- No "signature base string" errors
- Create and verify JWTs with standard libraries
- Standard claims, also validated by libraries:
 - Audience ("aud")
 - Issuer ("iss")
 - Issued at ("iat")
 - Expiration ("exp")
- Available in Ruby, PHP, Python, Java and more...

API demo



In-App Payments demo

[In-App Payments API documentation](#)
[In-App Payments developer forum](#)
[In-App Payments settings](#)

This demo will bring up a payment screen. Completing a purchase won't work because you can't buy from yourself.

Field name	Value
Audience (aud)	<input type="text" value="goog.payments.inapp.buyItem"/>
Seller ID / Issuer (iss)	<input type="text" value="08914266534959624650"/>
Issued at (iat)	<input type="text" value="1304917667"/>
Expiration (exp)	<input type="text" value="1305004067"/>
Item name	<input type="text" value="Gold Star"/>
Item description	<input type="text" value="A shining badge of distinction"/>
Item price	<input type="text" value="3.99"/>
ISO currency code	<input type="text" value="USD"/>
Seller data	<input type="text" value="some opaque data"/>

JavaScript sample:

```
runDemoButton.onClick = function() {
  goog.payments.inapp.buyItem({
    parameters: {},
    jwt: "eyJhbGciOiJIUzI1NiIsImtpZCI6IjEifQ.eyJpc" +
      "3MiOiIwODkxNDI2NjUzNDk1OTYyNDY1MCI6ImF1Z" +
      "CI6Imdvb2cucGF5bWVudHMuaW5hcHAuYnV5SXRlb" +
      "SI6ImlhdCI6MTMwNDkxNzY2NywiZXhwIjoxMzA1M" +
      "DA0MDY3LzJpc29dXjYzW5jeUNvZGU0iJVVU0Qil" +
      "CJpdGVtUHJpY2UiOjMuOTksInNlbGxlcRhdGEiO" +
      "iJzb21lIG9wYXN1ZSBkYXRhIiwiaXN1ZS5hbWU" +
      "iJHb2xkIFN0YXIIiLCJpdGVtRGVzY3JpcHRpb24" +
      "iJBIHNoaW5pbmcgYmFkZ2Ugb2YgZGlzdGluY3R" +
      "24ifQ.88nv6AJpI73GKXXM75cxashwJQzDf9f7n_" +
      "su68lcHsc",
    success: function() { window.alert('success')},
    failure: function() { window.alert('failure')}
  })
}
```

1 2 Tying the calls together

- The sellerData parameter gets passed from buy button to server postback
- Encrypt sellerData to protect user information
- Example: identify *users*

```
sellerData = encrypt("user_id: 12345")
```

- Example: tag purchases with *context*

```
sellerData = encrypt("sale: HappyHour, referrer: Ad5")
```

Recap

Simple

1

API

2

UX

3




Pricing

http://goo.gl/5percentfees

Google In-App Payments

It's all in the app.

Get started

-  **Flexible monetization for the web**
Collect payments for digital and virtual goods right within a game, app, news or media site.
-  **Quick conversions in the app**
Buyers pay in a few short steps without ever leaving the app.
-  **Easy integration with one API**
There is just one API call to send and one to receive the information.

Get started!

Sign up or Sign In

Next Steps

1. Integrate

- Sign up as a seller
<http://sandbox.google.com/checkout/inapppayments>
- Documentation:
<http://sites.google.com/site/inapppaymentsapi>

2. Send us feedback

- In-App Payments Forum:
<http://groups.google.com/group/in-app-payments>

3. Get your apps ready to launch this summer

One more thing...

Collect your special In-App Session T-shirts on the way out



Google™ 

