



JavaOne

Fun and Profit With the Google Checkout API in Java™ Technology

Patrick Chanezon, API Evangelist, Google

Inderjeet Singh, Software Engineer, Google

Ignacio Blanco, Software Engineer, Globant

<http://code.google.com/apis/checkout/>

TS-8849

Goals

Learn the Google Checkout API features

Learn how to Integrate Google Checkout with a Java™ technology-based Store reusing the sample Java code

Learn how to test your integration

Agenda

Google APIs

Google Checkout Service

Google Checkout API

Sample Java Code

Posting a Cart

Processing Notifications

Adding a Shipping Tracking Number to an Order

Integrating the Java Pet Store Demo

Testing Your Integration

Conclusion

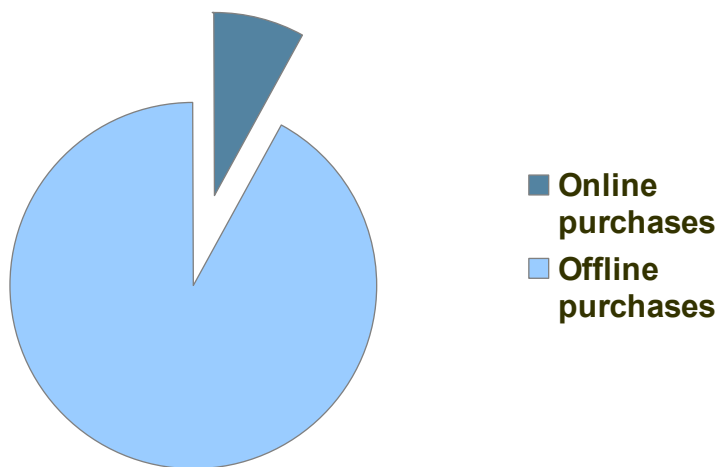
Google APIs

- Why are we doing it?
 - Our mission: “Organize the world’s information and make it universally accessible and useful”
 - We can’t organize it all ourselves
- 32 Google APIs on <http://code.google.com>...and counting!
 - Checkout API
 - Data API
 - Eight services
 - Maps API
 - AdWords API
 - Ajax Search API
 - Ajax Feed API
 - Desktop SDK
 - Enterprise APIs
 - Homepage API
 - YouTube API
 - Sitemaps
 - Talk (XMPP)
 - Toolbar API
 - Google Web Toolkit
 - ...

Why Google Checkout

Only 8% of all retail purchases are expected to occur online in 2006*

2006 Consumer Purchases by Channel



* 2006 State of Retailing Online, Shop.org/Forrester

Problem: Discovery, Checkout

- Difficult to find trustworthy sellers
- Long, repetitive, checkout processes
- 63% of shopping carts are abandoned after beginning checkout*

* Online Customer Experience Survey, Allurent, February 2006

Opportunity: Search, Convenience

- 37% of online purchases start with search*
- 25% of search queries result in a purchase directly related to the query**
- Fast checkout can improve conversion

* 2005 Online Holiday Mood Study, Shop.org/BizRate Research

** The Role of Search in Consumer Buying, comScore/Google



DEMO

Using Google Checkout



Product Benefits

① Search

② Find

③ Buy

For Buyers...

Convenience

- Google Checkout badge helps identify merchants to buy from
- One place to track purchases

Speed

- Single log-in for easy buying across the web

Confidence

- Credit card number concealment
- Fraud protection
- E-mail forwarding

For Merchants...

More Leads

- Google Checkout badge helps attract more customers

More Conversions

- Streamlined checkout process increases conversions

Lower Costs

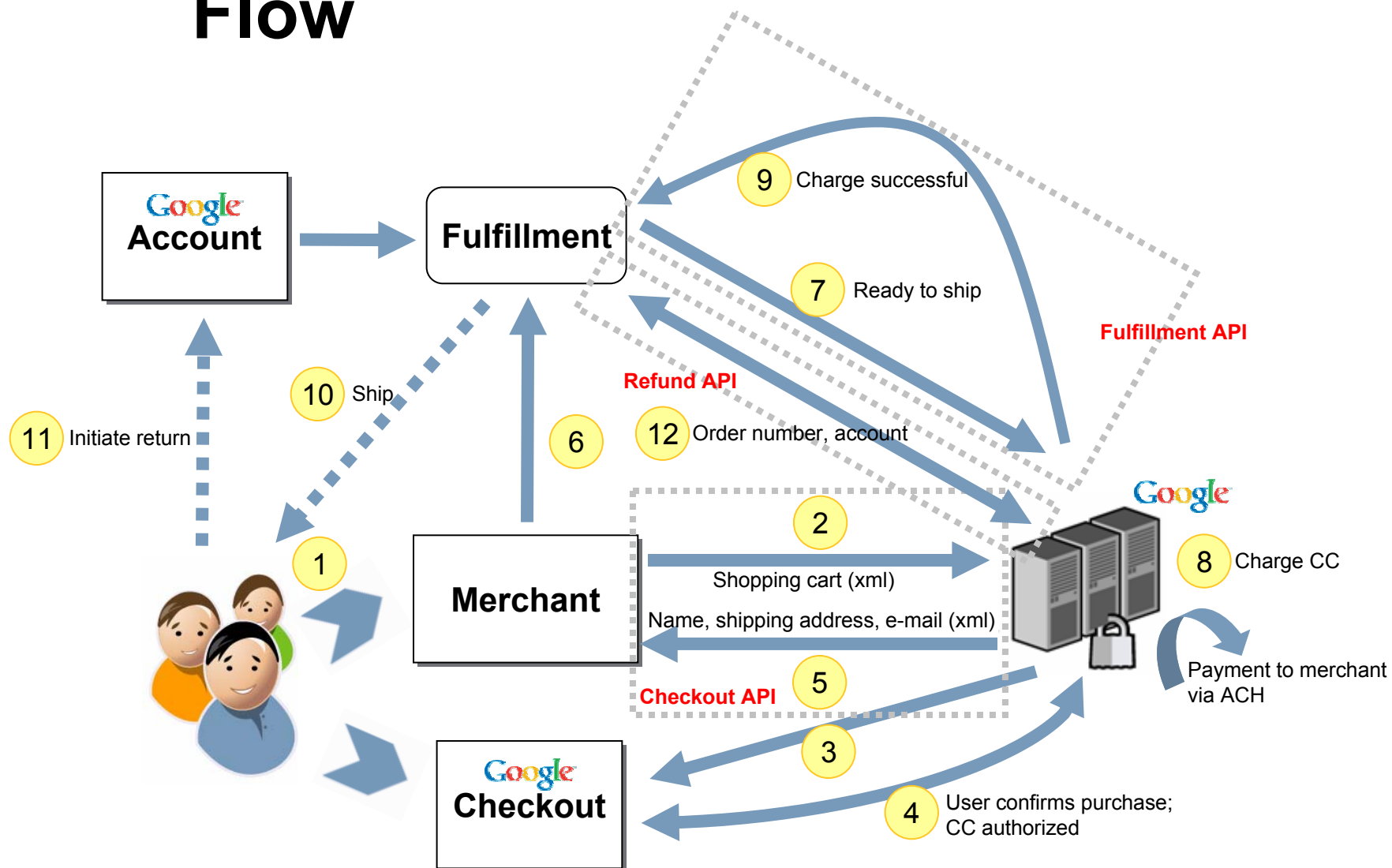
- Free transaction processing for sales up to 10X AdWords spend
- Low 2% + \$0.20 per transaction fee for all other sales
- Fraud protection

AdWords and Google Checkout

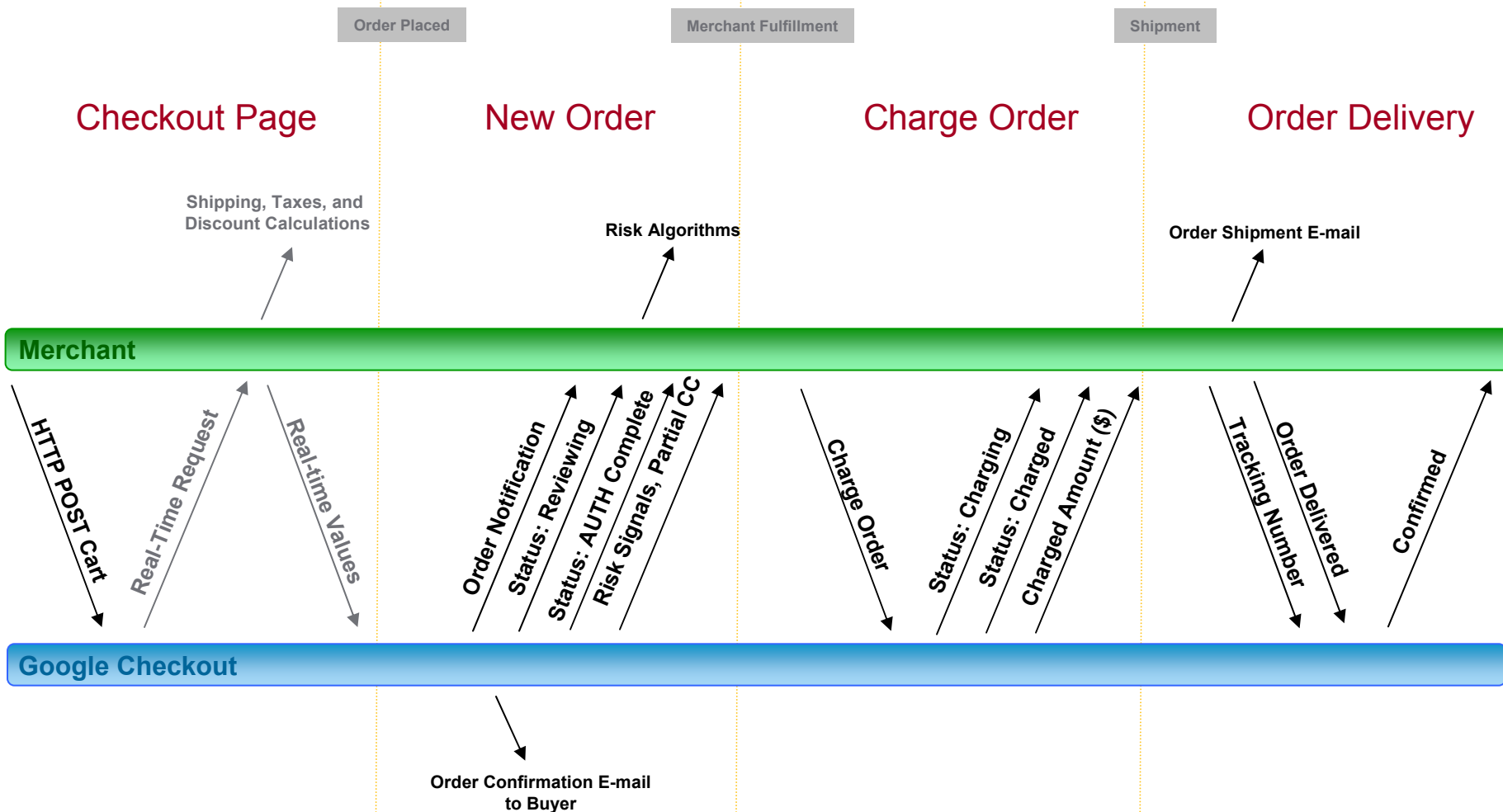
AdWords and Google Checkout work together to increase sales and lower costs



Google Checkout Transaction Flow



Order Flows: Typical



Google Checkout Integration Options

- Buy Now buttons
- E-commerce partners
- HTML (Name/Value pair) API
- Google Checkout XML API
- Systems Integrators specialized in Checkout
 - <http://checkout.google.com/seller/developers.html>
- For developers: bonuses for bringing Google Checkout to your merchants
 - https://services.google.com/inquiry/checkout_ecommerce
- Promotions
 - Free processing for merchants in 2007
 - Frequent buyer promotions

Google Checkout API

- Two levels of integration
 - Level 1: post shopping carts (taxes, shipping, coupons)
 - Level 2: merchant calculations, order management, two-way API
- REST API
 - XML over HTTP
 - XML schema
 - Synchronous/asynchronous (notifications)
 - HMAC-SHA1 for signature, basic auth over SSL
 - Samples and libraries in many environments
 - PHP, Java environment, Windows Classic ASP (COM), .NET
 - OsCommerce, Zen Cart

Google Checkout API Level 1

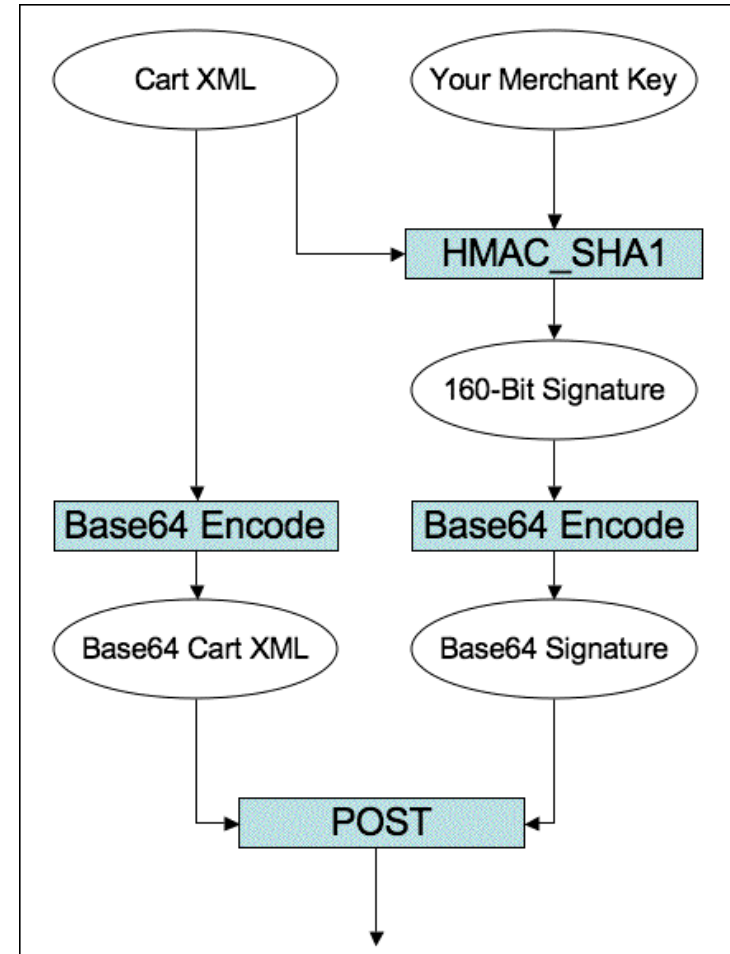
- Static Checkout buttons
- Form with hidden parameters



```
<form method="POST"
action="https://sandbox.google.com/cws/v2/Merchant/123456789
0/checkoutForm" accept-charset="utf-8">
    <input type="hidden" name="item_name_1" value="Peanut
Butter"/>
    <input type="hidden" name="item_description_1"
value="Chunky peanut butter."/>
    <input type="hidden" name="item_quantity_1" value="1"/>
    <input type="hidden" name="item_price_1" value="3.99"/>
    <input type="image" ...
src="http://sandbox.google.com/buttons/checkout.gif..."/>
</form>
```

Google Checkout API Level 1

- Post Cart XML + Signature



Level 2 Integration

- Level 1: Merchants use the UI to manage orders
- Level 2: Lets you integrate your order management system with Google
 - Google calls an endpoint on the merchant side
 - SSLv3 + basic auth
- Merchant calculations API
 - Merchants can apply their own business logic to compute values
 - Shipping, taxes, coupons
 - 3s to answer, else Google uses default values
- Notification API
 - New order, risk information, order state change, amount
 - If fails, retry policy up to 30 days
- Order processing API
 - Change order state

Order with Merchant Calculation Callback



1 On Merchant site, buyer creates cart, clicks Checkout button

Google

Merchant

Order States

Financial State Fulfillment State

Time

3 Buyer auto-signed-in, signs into existing account, or creates account with Google

Shopping Cart

2 Browser sends shopping cart, shipping methods, and backup tax and shipping amounts

4 Sends shipping addresses

Merchant Calculation Callback

Merchant Calculation Results

5 Calculates shipping and/or tax for all shipping addresses

6 Buyer can now:

- Select different shipping address (Google auto-updates page with stored values)
- Enter new address (repeat steps 4-6 with new address added)
- Enter coupon or gift certificate information (repeat steps 4-6 with coupon/gift certificate added)

7 Buyer clicks Place Order button

8a New order created

New Order Notification

Google conducts internal risk assessment and credit card authorization

8b Credit Card authorized, order now Chargeable

State Change Notification (chargeable)

8c Sends risk assessment results

Risk Information Notification

9 Merchant prepares items for shipping

Charge Order Request

11a Order is now Charging

State Change Notification (charging)

Google captures payment, and changes order state to Charged

11b Order is now Charged

State Change Notification (charged)

11c Sends amount of charge

Charge Amount Notification

10 When order is ready, Merchant requests payment

12 Merchant ships order

Add Tracking Request

13a Adds tracking information

Deliver Order Request

13b Tells Google to move order to Delivered state

13c Order changed to Delivered

State Change Notification (delivered)

REVIEWING

NEW

CHARGEABLE

CHARGING

CHARGED

DELIVERED

Google Checkout API in Java Platform

Sample code to demonstrate all the API calls

- Open Source project
 - <http://code.google.com/p/google-checkout-java-sample-code/>
 - Apache 2.0 license
- Designed to work for Java v.1.4 and above
 - org.w3c.dom for XML
 - No generics
- Java 2 Platform, Enterprise Edition (J2EE™ platform) sample
 - War sample: in memory notifications, logged to a file
 - Ear sample: Java Message Service (JMS) queue for notifications
- Easy to integrate in your Java application
 - Let's hook up the legendary Java Pet Store demo!

Learning From Our Mistakes

Minimizing dependencies is good, simplicity rules

- Initially used Java Architecture for XML Binding (JAXB)
 - Needed to maintain two projects
 - Java Web Services Developer Pack (Java WSDP) 1.6 for Java v.1.4 users
 - Java WSDP 2.0 for Java v.1.5 users
 - Plus JAXB software version conflicts for users
- Redesign 1
 - No generics, no JAXB software
 - Define interfaces for business objects, fully pluggable
- Redesign 2: Remove excessive pluggability
 - Business objects as classes
 - Sample application to customize: pure servlet or MDB
- Redesign 3: SDK for binary reuse

Google Checkout Sample Java Application

Console to experiment with the API

- Console view to make calls
 - UI to build your XML messages
 - Place calls
 - Watch notifications
- Very easy way to learn the API

Google Checkout Java Technology-Based Console



```
new CheckoutJavaSample();
```

[Shopping Cart Builder](#)
[Orders](#)
[Charge Order](#)
[Refund Order](#)
[Cancel Order](#)
[Authorize Order](#)
[Process Order](#)
[Add Merchant Order Number](#)
[Deliver Order](#)
[Add Tracking Data](#)
[Send Buyer Message](#)
[Archive Order](#)
[Unarchive Order](#)

Useful Links

[Java Sample Code Project Home](#)
[Checkout API Documentation](#)
[Typical Order Flow](#)
[Other Order Flows](#)

Shopping Cart Builder

MerchantId: 970855470745544

Build the Cart

Add Item	Name: iPod Nano	Description: Super small MP3 player.	Price: 100	Quantity: 1
Add Tax	State: CA	Amount: 0.08	Shipping Taxed: <input checked="" type="checkbox" value="true"/>	
Add Ship	Name: Fed Ex	State: CA	Amount: 5	

Request XML

```
<?xml version="1.0" encoding="UTF-8"?>
<checkout-shopping-cart xmlns="http://checkout.google.com/schema/2">
  <shopping-cart>
    <items>
      <item>
        <item-name>iPod Nano</item-name>
        <item-description>Super small MP3 player.</item-description>
        <unit-price currency="USD">100.0</unit-price>
        <quantity>1</quantity>
      </item>
    </items>
  </shopping-cart>
  <checkout-flow-support>
    <merchant-checkout-flow-support>
      <tax-tables>
        <default-tax-table>
          <tax-rules>
            <default-tax-rule>
              <shipping-taxed>true</shipping-taxed>
              <rate>0.08</rate>
            </tax-rule>
          </tax-rules>
        </default-tax-table>
      </tax-tables>
    </merchant-checkout-flow-support>
  </checkout-flow-support>
</checkout-shopping-cart>
```



Fast checkout through Google

[What is Google Checkout?](#)

Response XML

[Follow redirect \(in a new window\)](#)

Checkout Sample Java Technology Structure

Use: XML Marshalling/Unmarshalling and utilities library, no need to customize

- ▶  `JavaCheckoutRefImpl [trunk/ref-impl/JavaCheckoutRefImpl]`

Customize: Notification and Merchant Calculation processor examples

- ▶  `JavaCheckoutExampleCommon [trunk/examples/JavaCheckoutExampleCommon]`

Web application example

- ▶  `JavaCheckoutWebExampleWar [trunk/examples/JavaCheckoutWebExampleWar]`

Optional: EJB and EAR examples if you want to use MDB to process notifications

- ▶  `JavaCheckoutJ2eeExampleEjb [trunk/examples/JavaCheckoutJ2eeExampleEjb]`
- ▶  `JavaCheckoutJ2eeExampleEar [trunk/examples/JavaCheckoutJ2eeExampleEar]`

Checkout Java Library Structure (1/2)

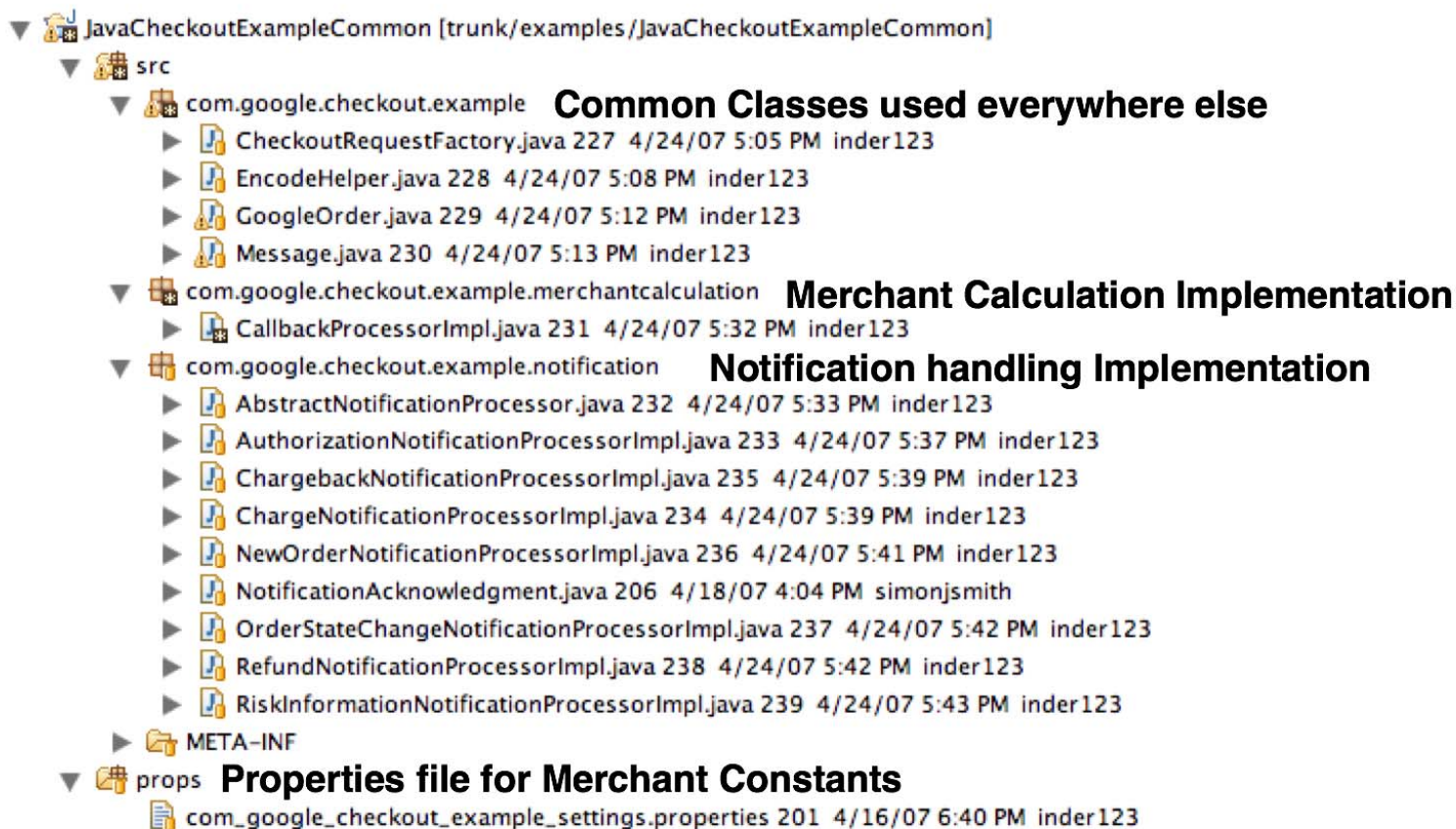
- ▼  JavaCheckoutRefImpl [trunk/ref-impl/JavaCheckoutRefImpl]
 - ▼  src
 - ▼  com.google.checkout **Common Classes used everywhere else**
 - ▶  AbstractCheckoutRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  CheckoutException.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  CheckoutResponse.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  EnvironmentType.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  MerchantConstants.java 205 4/18/07 3:07 PM simonjsmith
 - ▼  com.google.checkout.checkout **Shopping Cart Classes**
 - ▶  CheckoutShoppingCartRequest.java 207 4/18/07 4:28 PM simonjsmith
 - ▶  RoundingMode.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  RoundingRule.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  ShippingRestrictions.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  TaxArea.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  UrlParameter.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  UrlParameterType.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  USArea.java 205 4/18/07 3:07 PM simonjsmith
 - ▼  com.google.checkout.merchantcalculation **Merchant Calculation Interface**
 - ▶  CallbackProcessor.java 132 3/25/07 10:51 AM simonjsmith
 - ▼  com.google.checkout.notification **Notification handling Interfaces**
 - ▶  AuthorizationNotificationProcessor.java 132 3/25/07 10:51 AM simonjsmith
 - ▶  ChargebackNotificationProcessor.java 132 3/25/07 10:51 AM simonjsmith
 - ▶  ChargeNotificationProcessor.java 132 3/25/07 10:51 AM simonjsmith
 - ▶  NewOrderNotificationProcessor.java 132 3/25/07 10:51 AM simonjsmith
 - ▶  OrderStateChangeNotificationProcessor.java 132 3/25/07 10:51 AM simonjsmith
 - ▶  RefundNotificationProcessor.java 132 3/25/07 10:51 AM simonjsmith
 - ▶  RiskInformationNotificationProcessor.java 132 3/25/07 10:51 AM simonjsmith

Checkout Java Library Structure (2/2)

- ▼  **com.google.checkout.orderprocessing** **Order Management Classes**
 - ▶  AddMerchantOrderNumberRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  AddTrackingDataRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  ArchiveOrderRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  AuthorizeOrderRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  CancelOrderRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  ChargeOrderRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  DeliverOrderRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  ProcessOrderRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  RefundOrderRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  SendBuyerMessageRequest.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  UnarchiveOrderRequest.java 205 4/18/07 3:07 PM simonjsmith
- ▼  **com.google.checkout.util** **Utility Classes: XML, encoding**
 - ▶  Base64Coder.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  Constants.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  EncodeHelper.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  StringTuple.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  StringUtil.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  Tuple.java 205 4/18/07 3:07 PM simonjsmith
 - ▶  Utils.java 205 4/18/07 3:07 PM simonjsmith

Checkout Java Technology Example Structure

- Customize this project
- Implement your business logic, replace the factory



Checkout Java Web Application Structure

▼ JavaCheckoutWebExampleWar [trunk/examples/JavaCheckoutWebExampleWar]

▼ src

▼ com.google.checkout.webappexample **Notification and Merchant Calculation Servlets**

▶ JmsNotificationServlet.java 243 4/24/07 6:39 PM inder123

▶ MerchantCalculationServlet.java 242 4/24/07 6:36 PM inder123

▶ NotificationServlet.java 243 4/24/07 6:39 PM inder123

**Dispatch processing
to example project handlers**

▼ WebContent

Sample Console UI implemented as JSPs

▶ META-INF

▶ WEB-INF

addmerchantordernumber.jsp 226 4/24/07 5:01 PM inder123

addtrackingdata.jsp 226 4/24/07 5:01 PM inder123

archiveorder.jsp 226 4/24/07 5:01 PM inder123

authorizeorder.jsp 226 4/24/07 5:01 PM inder123

cancelorder.jsp 226 4/24/07 5:01 PM inder123

chargeorder.jsp 226 4/24/07 5:01 PM inder123

deliverorder.jsp 226 4/24/07 5:01 PM inder123

index.jsp 226 4/24/07 5:01 PM inder123

left_bottom.jsp 226 4/24/07 5:01 PM inder123

left_top.jsp 226 4/24/07 5:01 PM inder123

order_detail.jsp 226 4/24/07 5:01 PM inder123

orders.jsp 226 4/24/07 5:01 PM inder123

processorder.jsp 226 4/24/07 5:01 PM inder123

refundorder.jsp 226 4/24/07 5:01 PM inder123

sendbuyermessage.jsp 226 4/24/07 5:01 PM inder123

shopping_cart.jsp 226 4/24/07 5:01 PM inder123

unarchiveorder.jsp 226 4/24/07 5:01 PM inder123

**Lets you play with various aspects of the API
create carts, examine XML messages**

Posting a Cart in Java Platform (1/2)

```
// Set up Merchant Constants (usually from a properties file).
MerchantConstants mc = new MerchantConstants("248088419036991",
    "Q2x_yWSVUHokvFwx1MqVjp", "0", "US",
    "https://sandbox.google.com/checkout/cws/v2/Merchant/checkout",
    "https://sandbox.google.com/checkout/cws/v2/Merchant/merchantCheckout",
    "https://sandbox.google.com/checkout/cws/v2/Merchant/request");

// Create a shopping cart
CheckoutShoppingCartRequest cart = new CheckoutShoppingCartRequest(mc);

// Add items: 2 Biscuits
cart.addItem("Biscuit", "Yummy biscuit", new Float(9.99).floatValue(),
2);

// Add shipping: a shipping method restricted to California.
ShippingRestrictions sr = new ShippingRestrictions();
sr.addAllowedStateCode("CA");
cart.addFlatRateShippingMethod("USPS Ground", new Float(3.99), sr);
```

Posting a Cart in Java Platform (2/2)

```
// Add taxes: here 8.3% in California, shipping is taxed.
```

```
TaxArea ta = new TaxArea();
```

```
ta.addStateCode("CA");
```

```
cart.addDefaultTaxRule(0.08375, true, ta);
```

```
// Send the cart
```

```
CheckoutResponse response = cart.send();
```

```
if (response.isValidRequest()) {
```

```
    System.out.println(response.getXmlPretty());
```

```
    // Redirect the user to response.getRedirectUrl().
```

```
} else {
```

```
    System.out.println(response.getErrorMessage());
```

```
}
```

- Google Checkout Button posts to that servlet
- If the Cart post is successful redirect the user to Google Checkout!



Processing a Notification (1/3)

```
// NotificationServlet dispatches to the right Processor.
public void doPost(...)
...
    String notification =
getNotificationBody(request.getInputStream());
    String result = dispatch(notification);
    PrintWriter out = response.getWriter();
    out.print(result);
...
private String dispatch(String notification) throws Exception {
    if (notification.indexOf("new-order-notification") > -1) {
        NewOrderNotificationProcessor processor =
CheckoutRequestFactory
        .newNewOrderNotificationProcessor();
        return processor.process(notification);
    }
    ...
}
```

Processing a Notification (2/3)

```
// Implements an interface for pluggability.
public class NewOrderNotificationProcessorImpl extends
AbstractNotificationProcessor
    implements NewOrderNotificationProcessor {

    private MerchantConstants merchantConstants;

    // Dependency injection style constructor.
    public NewOrderNotificationProcessorImpl(MerchantConstants
merchantConstants) {
        this.merchantConstants = merchantConstants;
    }

    //This is where your business logic goes
    public String process(String callbackXML) throws CheckoutException {

        String ack = "";
        try {            // Parse the notification.
            Document document = Utils.newDocumentFromString(callbackXML);
```

Processing a Notification (3/3)

```
String orderNumber = Utils.getElementStringValue(document,
document.getDocumentElement(), "google-order-number");
...
// Build an order.
GoogleOrder order =
    GoogleOrder.findOrCreate(merchantConstants.getMerchantId(),
    orderNumber);
order.setLastFulStatus(lastFulStatus);
...
ack = getAckString();

// Log the order in a file or database.
order.addIncomingMessage(timestamp,
                           document.getDocumentElement().getNodeName(),
                           Utils.documentToStringPretty(document), ack);
} catch (Exception e) {
    throw new CheckoutException(e);
}
return ack;
```

Order Management: Add Tracking Number in Java Platform

```
// Setup Merchant Constants (usually from a properties file).
MerchantConstants mc = new MerchantConstants(...);

// Create a request to add UPS tracking data to the order.
AddTrackingDataRequest atdr =
new AddTrackingDataRequest(mc, "375053533130049", "UPS", "12345");
System.out.println("XML To Send: "+atdr.getXml());

// Send the request.
CheckoutResponse cResponse = atdr.send();

String cResponseStr = cResponse.getXml();

if (cResponseStr != null) {
    System.out.println(new StringBuffer(
        "XML Received: ").append(cResponseStr));
} else {
    System.out.println("No Response XML was sent.");
}
return;
```




DEMO

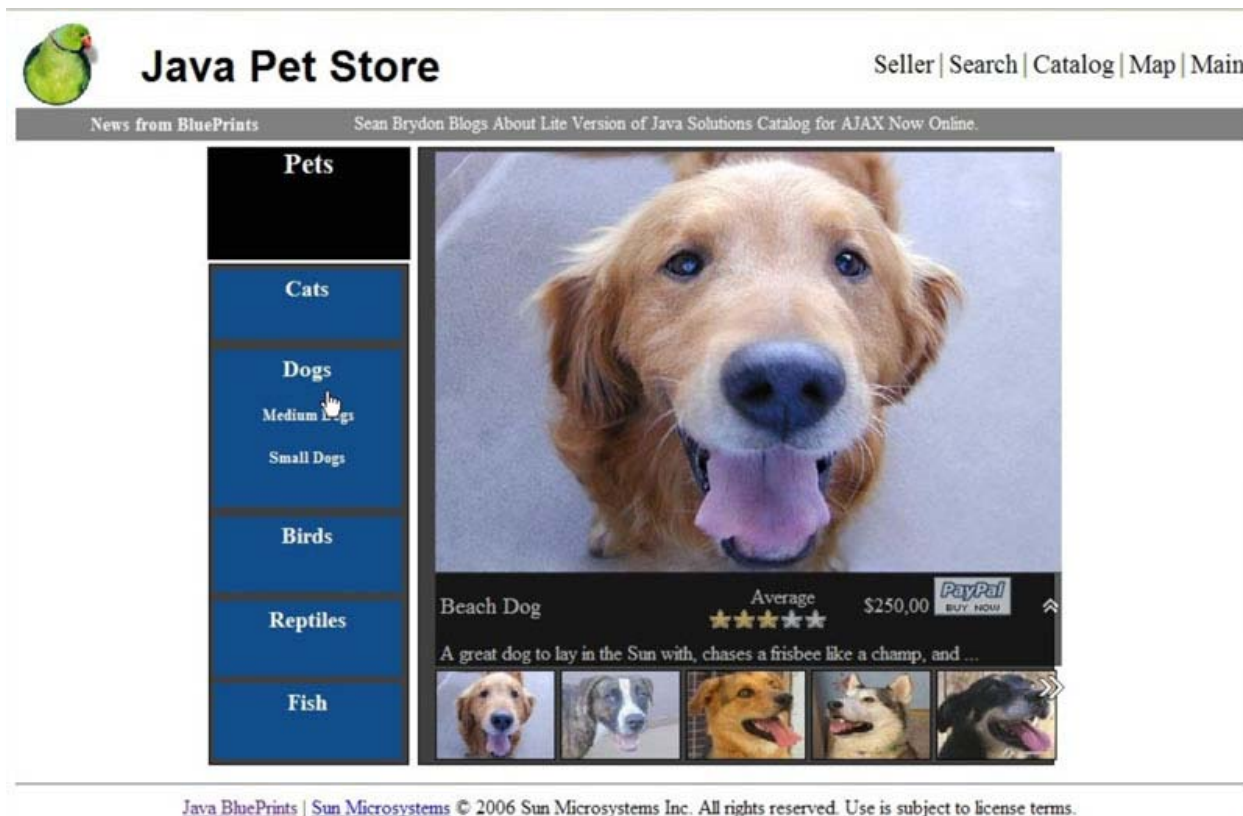
Google Checkout Sample
Java Technology-Based Console



The Java Pet Store Demo 2.0

Full of annotations and Ajax goodness

Let's integrate it with Google Checkout!



DEMO

Adding a Checkout Button
in the Java Pet Store Demo





DEMO

Handling Notifications
in Java Pet Store Demo

<code/>

Mendoza: Google Checkout Test Server

Integration testing for your Checkout implementation

- Problem: testing synchronous and asynchronous http messages, some triggered by Ajax interactions
- Solution: What if we just add something between you and Checkout to ease the burden of manual testing integration?
- Cf. Cedric Beust “Unit or Functional”
<http://beust.com/weblog/archives/000444.html>

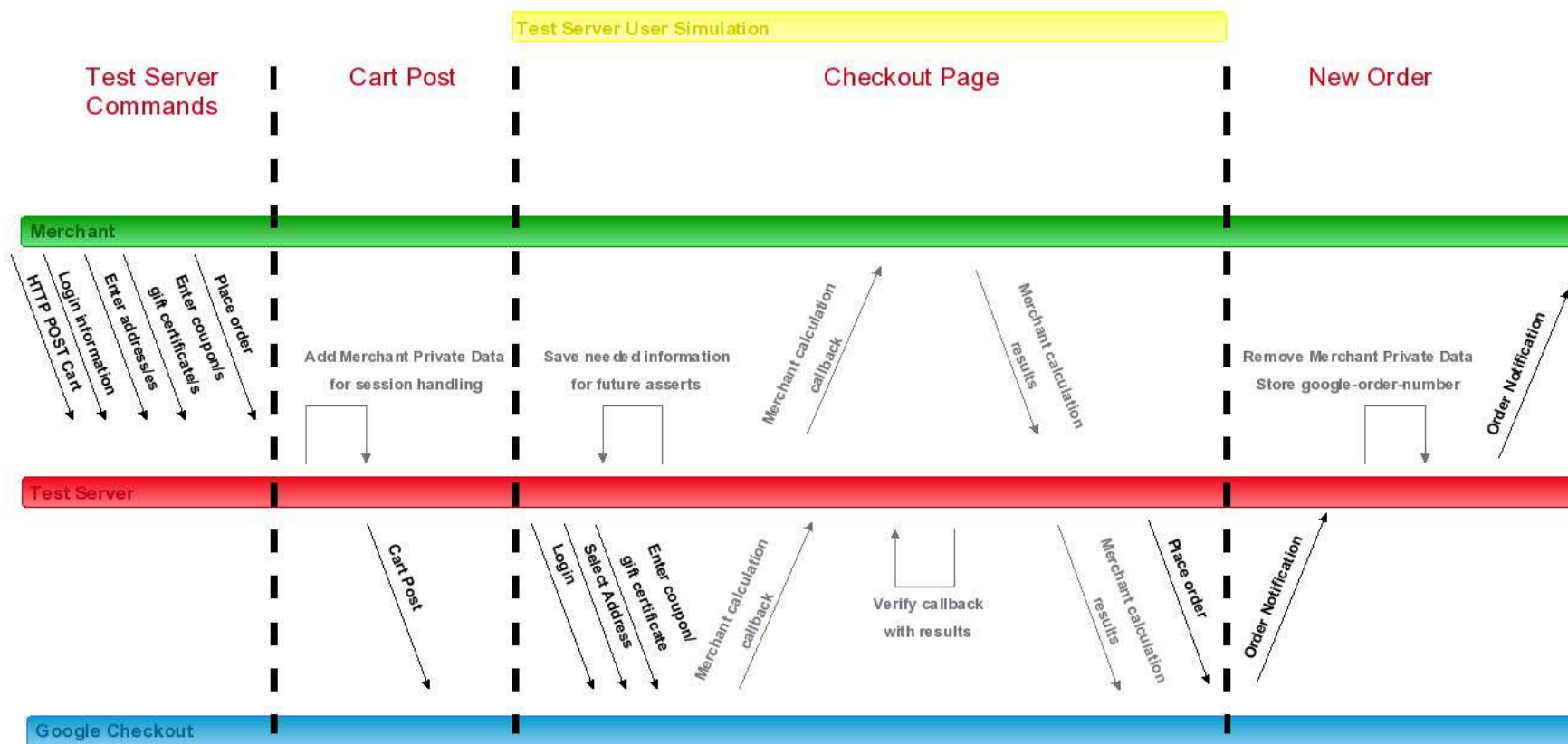
Mendoza: Integration Testing for Your Checkout Implementation

- Mendoza sits between you and Checkout and “acts as a transparent proxy”
- Uses Selenium to place orders + user interaction
 - As if it was a buyer using a browser
 - Test tool for web applications
 - Tests run directly in a browser, just as real users do (run in Internet Explorer, Mozilla and Firefox on Windows, Linux, and Macintosh)

Mendoza: Integration Testing for Your Checkout Implementation

- Simple standalone Java application (Java Archive [JAR])
- Receives http commands from the Merchant then simulates user interaction + interacts with Checkout using the API
- Verifies syntax + semantics + flow of the messages

Mendoza: Interaction Flow



Mendoza: Future Development

- Improved web UI
- Unit testing tools
- Test suites
- Performance testing capabilities
- Malbec
 - Batch automated testing tool for the Google Checkout API integration level 2
 - Includes batch OSCommerce and ZenCart integration testing

DEMO

Using Mendoza to Test Our Integration

Summary

- Google Checkout is worth integrating
 - Better click-through rate on ads sporting the shopping cart
 - Free processing in 2007
- Google Checkout API makes it possible
 - Simple things easy: Level 1 integration
 - Static Buy Now buttons, HTML buttons, signed XML
 - Difficult things possible: Level 2 integration
 - Uniform HTTP + XML API
 - Two-way, synchronous and asynchronous
- Sample Java code makes it easy
- Developers can make money with Checkout
 - Incentive program for helping merchants integrate
 - https://services.google.com/inquiry/checkout_ecommerce

For More Information

- Main site: <http://checkout.google.com>
- Google Checkout Video / Demo:
<https://checkout.google.com/seller/demo.html>
- Developer's Center:
<https://checkout.google.com/seller/developers.html>
- Google Checkout API Center: <http://code.google.com/apis/checkout/>
- Developer's Guide:
<http://code.google.com/apis/checkout/developer/index.html>
- Sample Java Code:
<http://code.google.com/p/google-checkout-java-sample-code/>
- Google Checkout API Blog:
<http://googlecheckoutapi.blogspot.com/>

Q&





JavaOne

Fun and Profit With the Google Checkout API in Java™ Technology

Patrick Chanezon, API Evangelist, Google

Inderjeet Singh, Software Engineer, Google

Ignacio Blanco, Software Engineer, Globant

<http://code.google.com/apis/checkout/>

TS-8849