Welcome to
The ABCs of APIs

Presented by:
Lori Guillory & Allison Ellington
Overview

This presentation will:

• Explain what APIs are and their benefits
• Show how APIs are growing and how that growth could benefit you
• List the information that good API documentation should provide
• Show API examples
What is an API?

• An **Application Programming Interface** (API) is a protocol for allowing one software program to communicate with another.

• APIs let the programs share information and influence each others’ behavior.

• This presentation addresses web service APIs, which involve programs that communicate across the internet.
APIs = Access to data

Data is at the center of APIs. APIs provide access to data, both public and private.

90% of the data in the world today has been created in the last two years alone.  

*IBM, 2012*

Companies create APIs to give others the opportunity to interface with their data and to broaden their audience.

Companies that develop APIs are not worried about how the data is presented; instead, they allow other developers to create the front-end that uses their data.
Why use an API?

By using data that other companies already have and maintain, your company’s development resources can focus on your company’s offerings.

**Worldwide Rentals**

San Francisco Rentals

- List with a photo, location, availability, and price for each rental
- Location map
- Weather

**Google’s Map API**

**Weather Underground’s Weather API**
How does an API work?

APIs use a “request and response” method of conversation. APIs function much like a conversation between people but with defined rules about the type of communication allowed.

1. Greetings
2. Ask for information
3. Information is provided
An API conversation

1. Greetings
2. Ask for information
3. Information provided

What’s the weather in San Francisco?

Partly cloudy, 66.3°F (19.1°C), 65% humidity.

http://api.wunderground.com/api/Your_Key/conditions/q/CA/San_Francisco.json

{ "response": { "version": "0.1", "display_location": { "city": "San Francisco", "state": "CA" }, "weather": "Partly Cloudy", "temp_f": 66.3, "temp_c": 19.1, "relative_humidity": "65%", "feelslike_string": "66.3 F (19.1 C)", "feelslike_f": "66.3", "feelslike_c": "19.1", "UV": "5" } }
API informational flow

Database/Servers

API

HTTP

End-User Application

You document here.

You do not document here.

Client App
- REST Library (optional)
- HTTP Library
API web services

SOAP (Simple Object Access Protocol)

SOAP is an XML-based (Extensible Markup Language) communication protocol for accessing web services that often uses HTTP (Hypertext Transfer Protocol).

REST (Representational State Transfer)

REST is an architectural style that often relies on a URL to make a request. REST does not rely on XML responses.
API terminology

**XSD (XML Schema Definition)** describes what an XML document can contain, the structure of the XML document, and the rules for data content. Also referred to as *schema*.

**Elements** are the main building blocks of all XML documents, containing the data and determining the structure of the instance document. Elements are defined within the XSD.

**Attributes** provide extra information within an element. Attributes have name and type properties and are defined within an XSD.

**WSDL (Web Service Definition Language)** is an XML-based interface definition language that is used for describing the functionality offered by a web service.

Liquid Technologies (http://www.liquid-technologies.com) has an XML Schema Tutorial for learning about how schemas work.
SOAP and REST examples

SOAP

```xml
<?xml version="1.0"?>

<soap:Envelope xmlns:soap="http://www.w3.org/2001/12/soap-envelope" soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  <soap:body weather="http://www.acme.com/weather">
    <weather:GetLocalTemp>
      <weather:ZipCode>12345</weather:ZipCode>
    </weather:GetLocalTemp>
  </soap:Body>
</soap:Envelope>
```

REST

GET http://www.acme.com/weather/LocalTemp/12345
REST example

This REST example uses the Photobucket API to retrieve the URL of an album for a particular user (AllisonLori).

Request

The GET operation tells the web service to retrieve data.

GET
http://api.photobucket.com/album/AllisonLori/url?format=xml&oauth_consumer_key=000000000&oauth_nonce=b7d20411c21d051f9e2e59de6ae4f5538&oauth_signature_method=HMAC-SHA1&oauth_timestamp=1236628744&oauth_version=1.0

The format=xml section defines that the response should be returned in XML format.

Authorization information is required to ensure that only an authorized user is accessing the data.
Photobucket GUI

Welcome to Photobucket
All your memories in one place.

Email
Password

Sign up

OR

Sign up with Facebook

Share your boldest, brightest most colorful photo for a chance to win a $500 prize package!

LEARN MORE
Photobucket GUI

Welcome to Photobucket
All your memories in one place.

Email
Password
Sign up

OR
Sign up with Facebook

Log in
allisonlori
Authentication

Need an account?
Sign up!

Remember me
Forgot?

Log in

Share your boldest, brightest most colorful photo for a chance to win a $500 prize package!

LEARN MORE
REST example

Response
<response>
  <status>OK</status>  -- Indicates that the request was successful
  <content>
    <username>AllisonLori</username>  -- The username we asked for in the GET request
    <subdomain>
      <album>library</album>  -- The album name: “library “is the default album for a user on Photobucket
    </subdomain>
    <path>s95.photobucket.com/user/AllisonLori/library/</path>  -- The path to the album
  </content>
  <format>xml</format>
  <method>GET</method>
  <timestamp>1236628744</timestamp>
</response>
Travelport Universal API Demo

Travelport Universal API Demo

The Travelport Demo has been created to stimulate new ideas and help you develop new applications to stay ahead of the competition. Through this demo site, we are able to showcase new content and functionality with code that’s ready to use. Plus, we are continuously expanding our supplier offerings through our Travelport Universal API platform.

Travelport Universal API offers multi-source content via a single API: air, hotel, car rental, and rail. Ancillaries may also be offered, depending on the type of content that is requested.

Ready to begin your development? Request test access to Travelport Universal API now!

Important! When using the Demo Site on the Production environment, there is no option to create a booking.
This SOAP example uses the proprietary Travelport Universal API Demo site (https://demo.travelportuniversalapi.com/) to create a request and response.

**Request**

```xml
<HotelSearchAvailabilityReq xmlns="http://www.travelport.com/schema/hotel_v32_0" TraceId="7ebe1c2c-61d0-4f63-a600-0f90695912c0" AuthorizedBy="Travelport" TargetBranch="TRGT_BRCH">
  <BillingPointOfSaleInfo xmlns="http://www.travelport.com/schema/common_v32_0" OriginApplication="uAPI" />
  <HotelLocation Location="KBV" />
  <HotelSearchModifiers NumberOfAdults="1" NumberOfRooms="1" AvailableHotelsOnly="true">
    <PermittedProviders xmlns="http://www.travelport.com/schema/common_v32_0">
      <Provider Code="1G" />
    </PermittedProviders>
  </HotelSearchModifiers>
  <HotelStay>
    <CheckinDate>2015-09-18</CheckinDate>
    <CheckoutDate>2015-09-21</CheckoutDate>
  </HotelStay>
</HotelSearchAvailabilityReq>
```

This section requests a hotel in Krabi (in Thailand) from Sep 18 to 21, 2015.
SOAP example

The header looks similar to this:

```plaintext
B2BGateway/connect/uAPI/HotelService HTTP/2.0
Accept-Encoding: gzip, deflate
Content-Type: text/xml;charset=UTF-8
SOAPAction: ""
Authorization: Basic UniversalAPI/UserName:Password
Content-Length: length
```

- **Authorization** sends the username and password for authentication.
- **The POST operation** tells the web service to send data and provides the URL of where to send the data.
- **The Content-Type section** defines that the request and response use XML format.
# Travelport Universal API Demo

## Shop Results

**KBV 18/09/2015 - 21/09/2015**

<table>
<thead>
<tr>
<th>Hotel</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBIS STYLES KRABI AO NANG</td>
<td>1,040</td>
<td>4,200</td>
</tr>
<tr>
<td>VOGUE RESORT AO NANG</td>
<td>1,450</td>
<td>2,750</td>
</tr>
<tr>
<td>HOLIDAY INN RESORT KRABI AO NAN</td>
<td>2,380</td>
<td>4,800</td>
</tr>
<tr>
<td>SHERATON KRABI BEACH RESORT</td>
<td>3,000</td>
<td>12,300</td>
</tr>
<tr>
<td>SOFITEL KRABI PHOKEETHRA</td>
<td>3,089</td>
<td>18,119</td>
</tr>
<tr>
<td>AMARI VOGUE KRABI</td>
<td>3,570</td>
<td>10,200</td>
</tr>
<tr>
<td>NAKAMANDA RESORT AND SPA</td>
<td>6,200</td>
<td>6,200</td>
</tr>
<tr>
<td>RAYAVADEE</td>
<td>10,925</td>
<td>27,000</td>
</tr>
<tr>
<td>PHULAY BAY RITZ-CARLTON RESERY</td>
<td>13,200</td>
<td>47,500</td>
</tr>
<tr>
<td>PIMANN BURI LUXURY POOL VILLA RES</td>
<td>17,000</td>
<td>28,000</td>
</tr>
</tbody>
</table>
SOAP example

Response

<x hotel:HotelSearchResult>
  <common_v32_0:VendorLocation ProviderCode="1G" VendorCode="SI" VendorLocationID="53742"
  Key="caz3FGu8S2K00Ve64rEmrw==" />

  <hotel:HotelProperty HotelChain="SI" HotelCode="53742" HotelLocation="KBV" Name="SHERATON KRABI BEACH RESORT"
  VendorLocationKey="caz3FGu8S2K00Ve64rEmrw==" HotelTransportation="Limo"
  ReserveRequirement="Other" ParticipationLevel="Best Available Rate and Inside Shopper participant"
  Availability="Available">
    <hotel:PropertyAddress>
      <hotel:Address>155 MOO 2  NONG THAL</hotel:Address>
    </hotel:PropertyAddress>

    <common_v32_0:Distance Units="MI" Value="2" Direction="S" />

    <hotel:Amenities>
      <hotel:Amenity Code="AICO" />
      << Over 35 amenity codes returned >>
    </hotel:Amenities>

  </hotel:HotelProperty>

  <hotel:RateInfo MinimumAmount="THB3000.00" MinAmountRateChanged="false" MaximumAmount="THB12300.00" MaxAmountRateChanged="false" />
</x>
Why should I care about APIs?

The number of APIs is constantly growing. APIs speed development and allow companies to use services to enhance their products. Many companies are offering APIs to reach a wider audience with their products.

On June 15, 2015, ProgrammableWeb had 13,581 APIs listed on its site, an increase of 787 since February 1.
What can APIs do for me?

- Because APIs have no GUI, documentation is key.
- API documentation skills could lead to more money.

Data from SimplyHired.com as of February 2015

![Bar chart showing salary comparison between US and Denver for different positions: Technical Writer, Sr. Technical Writer, API Technical Writer, Sr. API Writer.

- Technical Writer: US $53,000, Denver $67,000
- Sr. Technical Writer: US $67,000, Denver $71,000
- API Technical Writer: US $73,000, Denver $81,000
- Sr. API Writer: US $84,000, Denver $87,000]
Documenting your first API

What you don’t need to know:
• How to program

What you do need to know:
Most of what you need to know is already in your technical writer toolbox. Do what you normally do when documenting a GUI:
• Gather information.
• Filter the information you receive down to what a user needs to know.
• Lay out information such that it can be easily found and searched.

What you need to learn:
• API terminology
• Minimal familiarity with code
What information do I provide?

• Getting started information
  – Credentials
  – Connectivity
  – Security
  – Quick start task

• Industry information

• Why use this API
What information do I provide?

• How to use the API
  – Workflows for complex functionality
  – Interactions between the transactions
  – For each transaction, describe the required and optional information, acceptable values, return values, errors/exceptions
  – Tutorials and message examples are highly desirable

• Reference documentation
  – WSDLs and schema
  – Sample code

• Versioning information
What’s the big deal with versions?

New API versions can break existing functionality.

• Define the version support timeline.

• Ensure customers are notified in advance of breaking changes, so they can assign resources to upgrade to the new version.

• Decide how you will support multiple versions with the same source documentation. Or, will you create a help version for each API release?

• Provide a detailed list of changes between versions and clearly identify breaking changes.
Where do I find this information?

Like information about a GUI product, API product information comes from a variety of sources, but you may have to dig a little more to find it.

- Interview subject matter experts
- Attend functionality walkthroughs and developer meetings
- Read product documents, such as functional designs, Wiki pages and use cases, or JIRA stories

Pare down the information to what is needed by the developer (customer).
I have to read code?!

• Poke around code (schema files and sample code)
• Ask questions
• Use tools to make it easier
  – Altova XMLSpy
  – Eclipse (free)
  – XSD Diagram (free)
  – <oXygen/> XML Editor
  – Notepad ++
API documentation examples

Good documentation examples:

- Flickr API
- Lufthansa Open API
- Stripe

How do they compare to our documentation?

- Travelport Universal API
Wrap-up

• APIs are growing, as is the need for those who can provide user documentation.

• API documentation skills can make you a more desirable job candidate and could yield higher pay.

• The skills you have as a technical writer are easily adaptable to documenting APIs.
THANK YOU FOR ATTENDING TODAY’S WEBINAR!

As a webinar attendee, receive **$100 OFF** our next advanced training course. Just $499 per student!

---

**MadCap Flare CSS Training**
July 21-22, 2015 (web-based)

**MadCap Flare Responsive HTML5, Mobile, and EPUB Training**
August 11-12, 2015 (web-based)

*Offer valid through Friday, July 17, 2015.*

Note: Courses subject to change. Availability based on student registration. Contact us for the latest schedule and course availability.

TO RECEIVE YOUR DISCOUNT, CONTACT:
sales@madcapsoftware.com | +1 858.320.0387 opt.1
MADWORLD

2016 | CALL FOR PAPERS

SUBMISSION DEADLINE: FRIDAY, JULY 24, 2015

The Premier Technical Communication and Content Strategy Conference

APRIL 10-12, 2016 | SAN DIEGO, CALIFORNIA

www.madworldconference.com
Allison Ellington
Allison.Ellington@Travelport.com

Lori Guillory
Lori.Guillory@Travelport.com
Every accomplishment starts with the decision to try.

Author unknown