APIs: Crossing the Boundaries of Distance, Hardware and Technology

Presenter: James Vitale, LACMA
What are “APIs”? - Not just an ancient Egyptian bull deity

Application Programming Interfaces

- **Application**
  - What we’re doing. The kind of system or set of features and functionality with which we are interacting

- **Programming**
  - The rules (algorithms and heuristic methods) that describe the “how” behind what we’re doing to achieve our desired outcome

- **Interfaces**
  - The specific mechanisms we decide are best-suited to build a pathway for communication between our application and the outside world (other people, platforms and applications)

APIs - Crossing the Boundaries of Distance, Hardware and Technology
APIs: Common Goals

● **Access to Information.** Questions raised:
  ○ Authentication (Who?)
  ○ Authorization (What?)
  ○ Encryption (Secured?)

● **Exchange of Information.** Questions raised:
  ○ Format? (XML, JSON, Flat-file, etc.)
  ○ Internationalization (Unicode and multibyte character concerns)
  ○ Binary Data (BLOBs), etc.

● **Ease of Use.** Questions raised:
  ○ Richness of the interface (Searching, filtering, pagination, data retrieval options, etc.)
  ○ Simplicity of the interface (Parameters, default values, error-handling, “advanced” features)
  ○ Impact on performance, stability and physical resource requirements to host the APIs
A Brief History of APIs

- APIs have gone through several evolutions over the past 6 decades, very much paralleling the ever-changing shape of our global technology infrastructure, data systems and information exchange needs:
  
  - 1960s - 1980s → EDI (Electronic Data Interchange);
  
  - 1990s → CORBA (Common Object Request Broker Architecture)
  
  - Late 90’s - 2000s → Web Services Model
    - SOAP interfaces (WSDLs, XSDs, XML exchange)
    - RESTful interfaces (HTTP GET/POST, simple request/response architecture)
APIs: Common Characteristics

- **Require a Protocol**
  - Communication mechanism between 2 points

- **Sending a Request**
  - How we “ask” for what we want
  - Simple vs. Complex Requests
  - Adv: Synchronous vs. Asynchronous Requests

- **Receiving a Response**
  - Getting back the results of our request
  - The ups & downs of the Internet
  - Adv: “Chatty” vs. “Chunky” APIs

- **The Data Structure and packaging it**
  - What gets delivered in the response and why
APIs: Crossing the Boundaries

- **Language-independent**
  - Consistent behavior regardless of the programming language is used in the requestor’s environment

- **Platform-independent**
  - Consistent behavior regardless of the requestor’s O.S. (UNIX, MacOS, Windows)

- **Hardware agnostic**
  - APIs should generally be agnostic (unimpacted) by the hardware device/platform communicating with them (server, mainframe, mobile device, tablet, etc.)
Choose an appropriate protocol based on the following considerations:

- Frequency of use (# of calls/hour to the API)
- Complexity/work required on the server that hosts the API to fulfill the request
- Average/Max size of data set returned
- Availability requirements (High/Med/Low)
- Security/Encryption/Compression

Illustration 5.1 - A simple URL using the HTTP protocol to reference the server that will host our APIs
http://apis.museum.org/

The protocol we choose is referred to as our “Transport” mechanism, the vehicle by which data (information) is passed between the client (requester) and the server (responder). We can introduce Transport-level Security, by opting to use HTTPS instead of HTTP, which employs SSL encryption technology, if required.
The “Request” is a key part of the Interface in our API. APIs typically support the ability to make a variety of requests. Requests are the mechanism through which a 3rd-party presents a question or query to your information system. Part of our role is to define a flexible and sensible Request Interface.

**Illustration 6.1** - A very simple, albeit inflexible Request Interface

http://apis.museum.org/searchobjectbyfirstname/picasso

**Illustration 6.2** - With some planning and good design, a flexible and more intuitive Request Interface evolves

http://apis.museum.org/objectsearch?filterby=title&value=blue%20boy
http://apis.museum.org/objectsearch?filterby=artist,placemade,title&value=picasso,spain,woman
APIs: Use Case Exploration, Pt. 3 - The Response

- Considerations when designing the response mechanism
  - Consistency and reliability
  - Speed and performance
  - Usability
  - Preserving data integrity (Unicode characters, binary data, etc.)
  - Size / Format
APIs: Use Case Exploration, Pt. 4 - The Data

XML

```xml
<response>
  <list name="responseHeader">
    <int name="status">0</int>
    <int name="OUI">0</int>
    <list name="param">
      <int name="q">"Title:Blue boy</int>
      <int name="f">XML</int>
      <object ID="AccessionNumber,Artists,Title,PlaceMade,DateMade,Culture,CreditLine"/>
    </list>
  </list>
  <list name="obj">
    <long name="ObjectID">24592</long>
    <string name="Title">"Blue"
    <string name="Artists">"William Turnbull", "Clifford Smith"
    <string name="PlaceMade">"Scotland"
    <string name="DateMade">1966
    <string name="Culture">""
    <string name="CreditLine">"Museum Purchase with Membership Fund, Accession Number: M.66.23.146."
  </list>
</response>
```

JSON

```json```
{  
    "responseHeader": {  
        "status":0,  
        "OUI":0,  
        "param": [  
            {  
                "q": ""Title:Blue boy",  
                "f": "XML"  
            }  
        ],  
        "response": [{  
            "ObjectID":24592,  
            "Title": "Blue",  
            "Artists": ["William Turnbull", "Clifford Smith"],  
            "PlaceMade": "Scotland",  
            "DateMade": "1966",  
            "CreditLine": "Museum Purchase with Membership Fund, Accession Number: M.66.23.146, Culture:",  
            "ObjectID":121567,  
            "Title": "Blue",  
            "Artists": ["Elliworth Kelly"],  
            "PlaceMade": "United States",  
            "DateMade": "2007",  
            "CreditLine": "Gift of the artist and Gemini G.E.L., Accession Number: M.2007.59.30",  
            "Culture": ""
    }  
},
```

CSV

```
ObjectID,AccessionNumber,Artists,Title,PlaceMade,DateMade,Culture,CreditLine
```

APIs - Crossing the Boundaries of Distance, Hardware and Technology
APIs: Use Case Exploration, Pt. 5 - Integration

- One set of APIs, many points of consumption, including:

  Mobile Applications
  Websites / Intranet
  Analytics / Reporting
  Enterprise Applications
  Academia & Research
APIs: Their benefits and values (Internal)

- Improve operational efficiency
- Automate routine processes
- Improved visibility and access to the institution’s most valued data
- Encourages support for routine data quality / cleansing activities
APIs: Their benefits and values (External)

- Very low cost of implementation
- Facilitates creative, collaborative marrying of the institution’s data assets with other partners to create new, enriched information
- Provides a public information resource and service to the community
- An opportunity to express the interaction of art & technology through the further exploration of digital visualizations married to data-driven APIs
APIs: Supporting Your Institution’s Goals

- Startout by identifying opportunities (review self-interview handout)
- Collaborate with appropriate departments
- Brainstorm a list of goals & objectives
  - Who’s your audience/consumer?
  - What data is of interest?
  - Who manages this data at your institution?
  - Who’s in the best position to work with you to design, build and support the APIs you have in mind?
  - Solicit a project sponsor, commitment of resources and assembling a project plan
  - Kick it off and start the work!
Thank you for attending.

Your questions are welcomed and encouraged!