Agenda

1. Definition
2. Why web services?
3. Connectivity
4. SOAP and RESTful services

Xamarin University

Definition

Xamarin University

Xamarin Evolve 2014

"A software system designed to support interoperable machine-to-machine interaction over a network."

W3C - http://www.w3.org/TR/ws-arch/
Wikipedia: https://en.wikipedia.org/wiki/Web_service

Xamarin University

Common Characteristics

- Language agnostic
- Supported over HTTP
- Exposes an API to clients
- Discoverable

Xamarin University

Why web services?

Xamarin University

Distributed Computing

- Utilize 3rd party services
- Securely access data from our mobile apps
- It's a big, big world and user's expect our apps to connect to other apps and services they use

Xamarin University

Connectivity

Xamarin University

Connection Types

Cellular

Roaming

Wi-Fi

Xamarin University

Watching Connectivity

- Monitor network connections
- Detect network type
- Host reachability

Reachability

ConnectivityManager

NetworkInterface

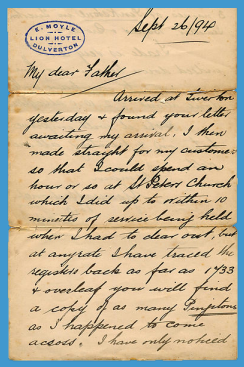
Xamarin University

SOAP and RESTful services

Xamarin University

SOAP

- Highly structured
- XML based
- WSDL
- Multiple protocols



Xamarin University

Demo

Xamarin University

SOAP – Request

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 299
SOAPAction: "http://www.w3.org/2003/05/soap-envelope"

<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
  </soap:Header>
  <soap:Body>
    <s:GetStockPrice xmlns:s="http://www.example.org/stock">
      <s:StockName>IBM</s:StockName>
    </s:GetStockPrice>
  </soap:Body>
</soap:Envelope>
```

Xamarin University

SOAP – Response

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 201

<?xml version="1.0"?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope"
   xmlns:s="http://www.example.org/stock">
   <env:Body>
     <s:GetStockPriceResponse>
          <s:StockPrice>45.25</s:StockPrice>
     </s:GetStockPriceResponse>
   </env:Body>
</env:Envelope>
```

Xamarin University

SOAP – Generated Proxy

```
public double GetStockPrice(string stockName)
{
    // Invocation of the service method
}
```

Xamarin University

Using SOAP from an App

- Server typically uses a framework, like WCF
- Tools to create a *proxy file*
- App uses the proxy methods to create, send and receive the XML data

Xamarin University

Demo

Xamarin University

SOAP – Pros and Cons

+ Platform and Language agnostic
+ Additional specifications for standardized security, etc.
+ Supports different transports
+ Easy to setup and to use

– Verbose payload
– Proxy files hard to maintain and can be brittle to change
– Not cacheable

Xamarin University

RESTful Services

- Representational State Transfer
- Simple XML over HTTP at it's core
- Uses the standard HTTP verbs

Chris McClave, Connecticut, United States
https://www.flickr.com/photos/13763109@N00/2329727859

Xamarin University

Xamarin University

Demo

Xamarin University

REST – Request

```
http://www.example.org/stock/IBM
```

```
GET /stock/IBM HTTP/1.1
Host: example.org
Accept: text/xml
Accept-Charset: utf-8
```

Xamarin University

REST – XML Response

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 100

<?xml version="1.0"?>
<s:Quote xmlns:s="http://example.org/stock">
     <s:StockPrice>45.25</s:StockPrice>
</s:Quote>
```

Xamarin University

REST – JSON Response

```
HTTP/1.1 200 OK
Content-Type: text/json; charset=utf-8
Content-Length: 30

{
    "StockPrice" : 45.25
}
```

Xamarin University

REST – Pros and Cons

+ Accessible to any client that supports HTTP
+ Supports caching
+ Concise, closer to the web – uses HTTP verbs and response codes
+ No tooling required

– HTTP(s) only
– No official standards
– Harder to map results to objects

Xamarin University

Summary

- Web services are easy to consume in mobile apps
- There is no good or bad
- Add an abstraction layer

Xamarin University