Χρήση Υπηρεσιών Ιστού

Κεφάλαιο 13

Python για Όλους
www.py4e.com

Δεδομένα στον Ιστό

- Με το HTTP Αίτημα/Απάντηση καλά κατανοητό και καλά υποστηριζόμενο, υπήρξε μια φυσική κίνηση προς την ανταλλαγή δεδομένων μεταξύ προγραμμάτων, που χρησιμοποιούν αυτά τα πρωτόκολλα

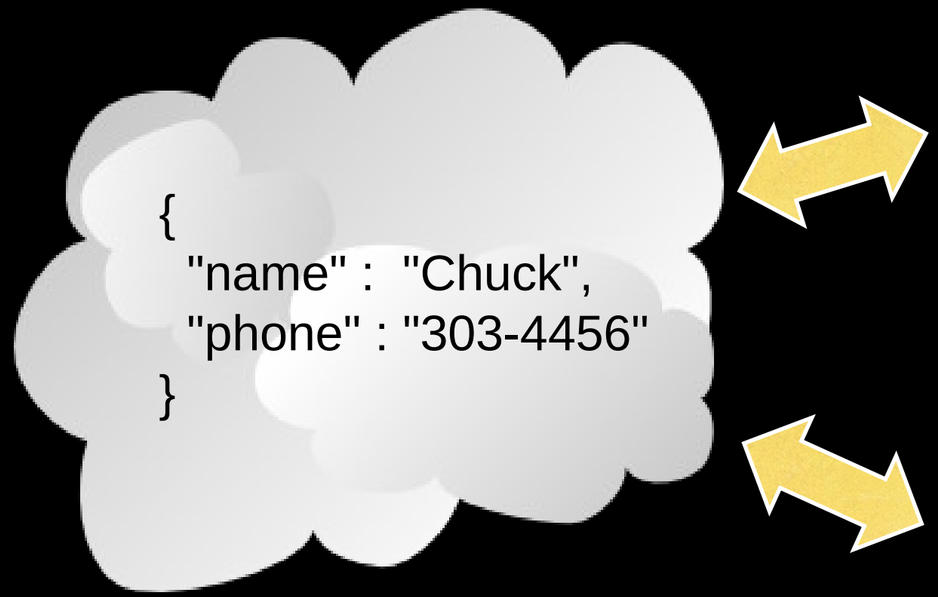
- Χρειάστηκε να καταλήξουμε/συμφωνήσουμε σε έναν τρόπο αναπαράστασης των δεδομένων που μεταφέρονται μεταξύ εφαρμογών και μεταξύ δικτύων

- Υπάρχουν δύο συνήθης μορφές: XML και JSON

Αποστολή Δεδομένων Μέσω του «Ιστού"

Πίνακας
PHP

Λεξικό
Python

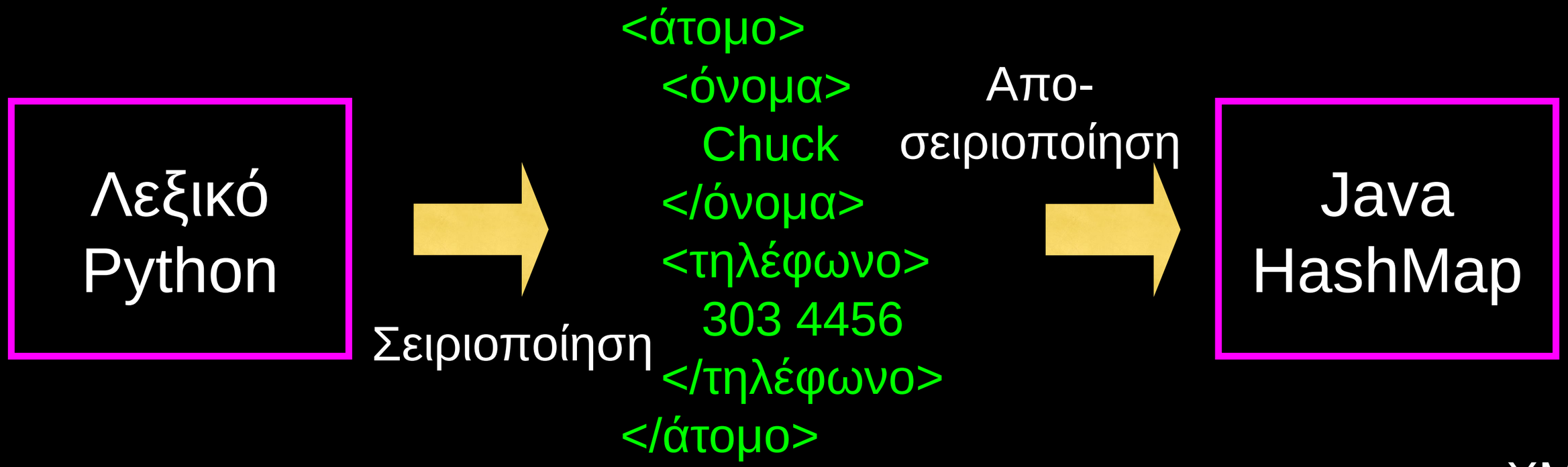


Αντικείμενο
JavaScript

Java
HashMap

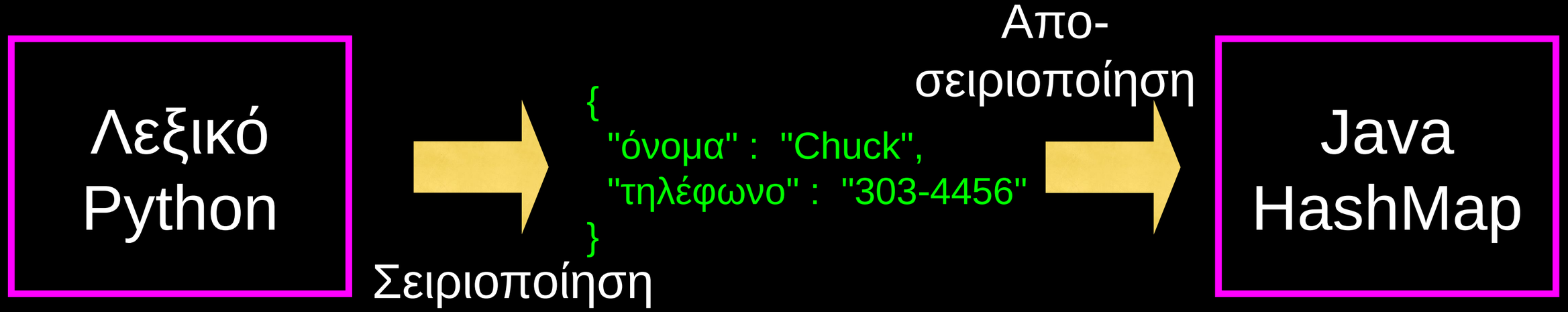
Γνωστό και ως «Wire Protocol» (Πρωτόκολλο Καλωδίων)
- Τι στέλνουμε στο "καλώδιο"

Συμφωνία για το “Wire Format” (Διαμόρφωση Καλωδίου)



XML

Συμφωνία για το “Wire Format” (Διαμόρφωση Καλωδίου)



JSON

XML

Επισήμανση δεδομένων προς αποστολή σε όλο το δίκτυο...

http://en.wikipedia.org/wiki/XML

XML «Στοιχεία» (ή Κόμβοι)

- Απλό Στοιχείο
- Σύνθετο Στοιχείο

```
<άνθρωποι>
  <άτομο>
    <όνομα>Chuck</όνομα>
    <τηλέφωνο>303 4456</τηλέφωνο>
  </άτομο>
  <άτομο>
    <όνομα>Noah</όνομα>
    <τηλέφωνο>622 7421</τηλέφωνο>
  </άτομο>
</άνθρωποι>
```

eXtensible Markup Language (επεκτάσιμη γλώσσα σήμανσης)

- Πρωταρχικός σκοπός είναι να βοηθήσει τα συστήματα πληροφοριών να μοιράζονται δομημένα δεδομένα

- Ξεκίνησε ως απλοποιημένο υποσύνολο της Γενικευμένο Πρότυπο Γλώσσας Σήμανσης (Standard Generalized Markup Language - SGML) και έχει σχεδιαστεί για να είναι ευανάγνωστο από τον άνθρωπο

http://en.wikipedia.org/wiki/XML

Βασικά της XML

- Ετικέτα Έναρξης
- Ετικέτα Τέλους
- Περιεχόμενο κειμένου
- Ιδιότητα
- «Αυτοκλεινόμενη» Ετικέτα

```
<άτομο>
  <όνομα>Chuck</όνομα>
  <τηλέφωνο type="intl">
     +1 734 303 4456
   </τηλέφωνο>
   <email hide="yes" />
</άτομο>
```

Λευκός Χώρος
(Μη ορατοί χαρακτήρες)

```
<άτομο>
  <όνομα>Chuck</όνομα>
  <τηλέφωνο type="intl">
     +1 734 303 4456
   </τηλέφωνο>
   <email hide="yes" />
</άτομο>
```

Τα άκρα της γραμμής δεν έχουν σημασία. Σε στοιχεία κειμένου οι μη ορατοί χαρακτήρες απορρίπτονται γενικά. Εμείς προσθέτουμε εσοχή μόνο για να είναι πιο εύκολα αναγνώσιμο.

```
<άτομο>
  <όνομα>Chuck</όνομα>
  <τηλέφωνο type="intl">+1 734 303 4456</τηλέφωνο>
   <email hide="yes" />
</άτομο>
```

Ορολογία XML

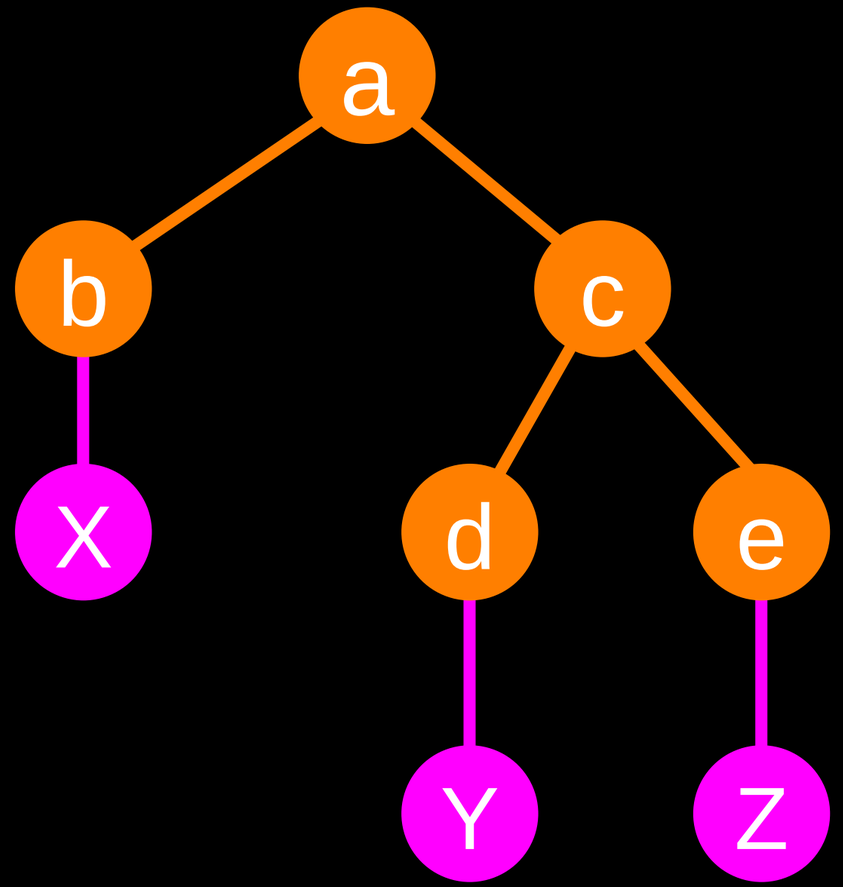
- Ετικέτες υποδεικνύουν την αρχή και το τέλος των στοιχείων
- Ιδιότητες - Ζεύγη κλειδιών/τιμών στην ετικέτα έναρξης του XML
- Σειριοποίηση / Απο- Σειριοποίηση - Μετατροπή δεδομένων ενός προγράμματος σε μια κοινή μορφή που μπορούν να αποθηκευτούν και/ή να μεταδοθούν μεταξύ συστημάτων με τρόπο ανεξάρτητο από τη γλώσσα προγραμματισμού

http://en.wikipedia.org/wiki/Serialization

XML ως Δέντρο

```
<a>
  <b>X</b>
  <c>
    <d>Y</d>
    <e>Z</e>
  </c>
</a>
```

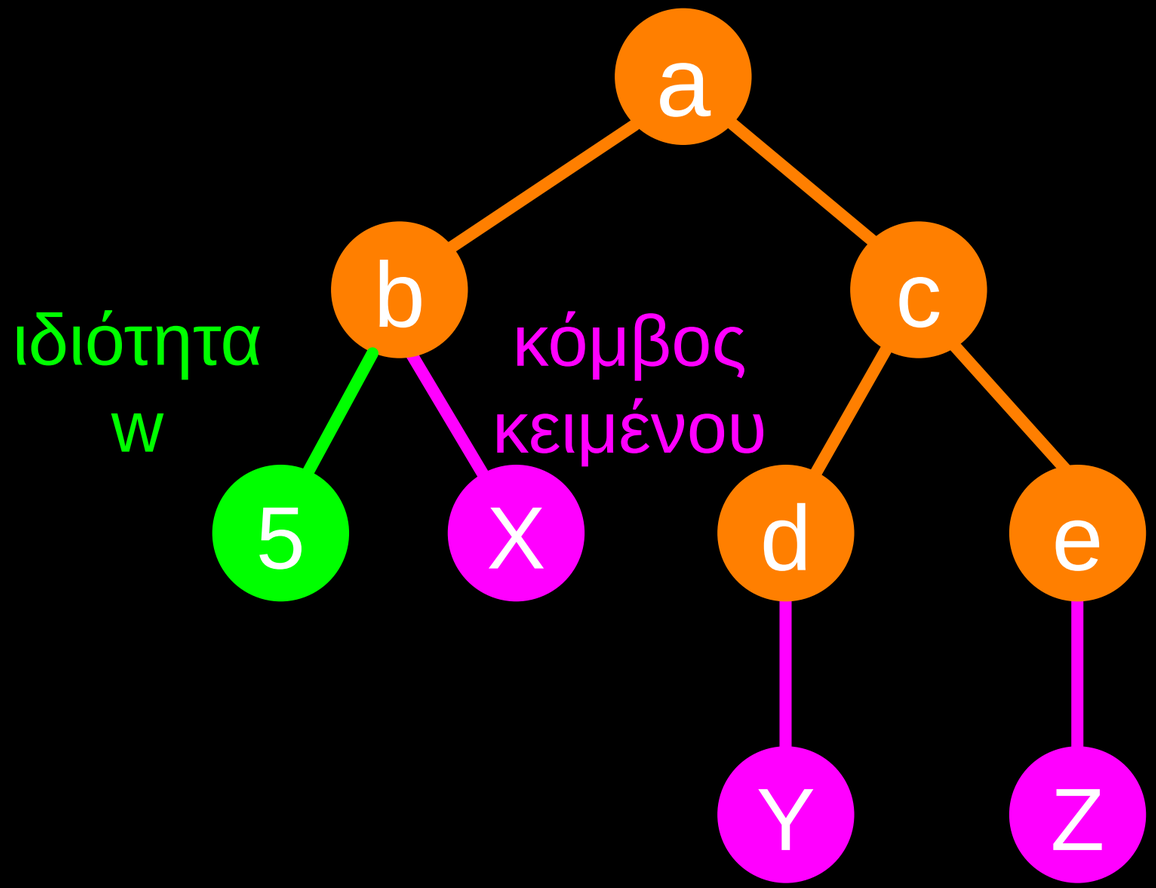
Στοιχεία Κείμενο



XML Κείμενο και Ιδιότητες

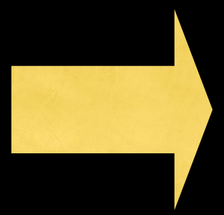
```
<a>
  <b w="5">X</b>
  <c>
    <d>Y</d>
    <e>Z</e>
  </c>
</a>
```

Στοιχεία Κείμενο

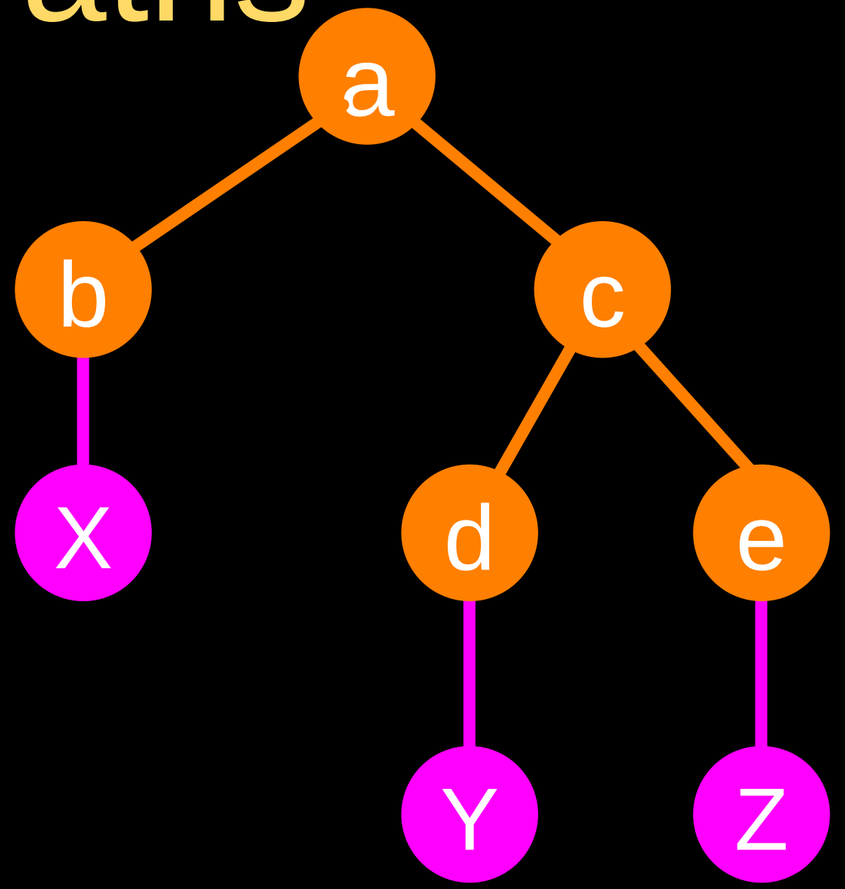


XML ως Μονοπάτια/Paths

```
<a>
  <b>X</b>
  <c>
    <d>Y</d>
    <e>Z</e>
  </c>
</a>
```



/a/b	X
/a/c/d	Y
/a/c/e	Z



Στοιχεία Κείμενο

Σχήμα XML

Περιγράφοντας μια «σύμβαση» ως προς το τι είναι αποδεκτό στην XML

http://en.wikipedia.org/wiki/Xml_schema

http://en.wikibooks.org/wiki/XML_Schema

Σχήμα XML

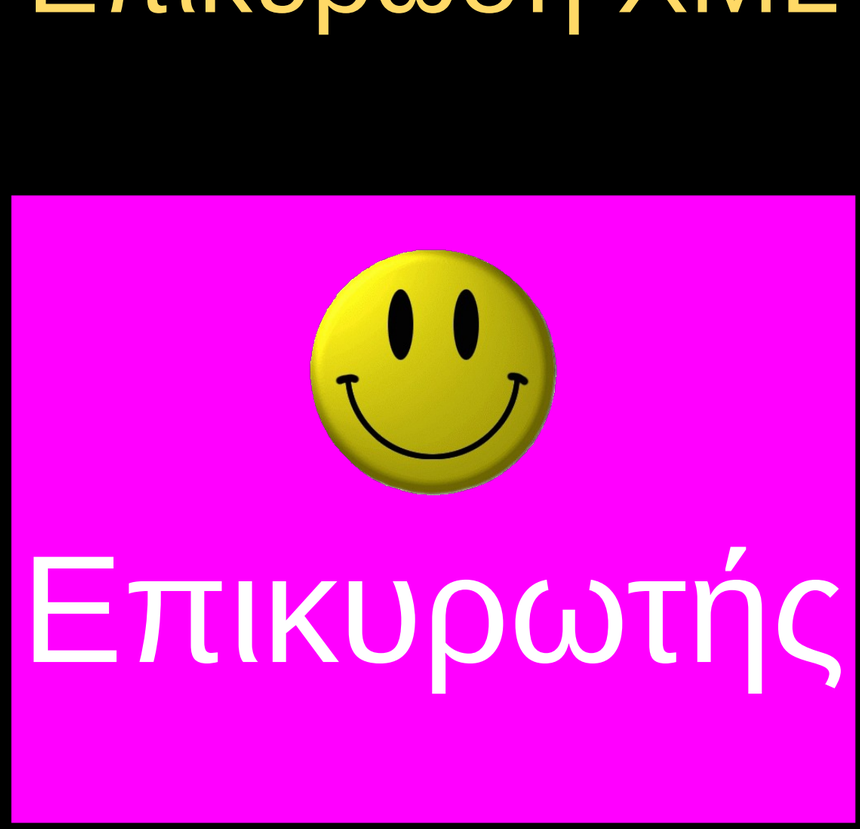
- Περιγραφή της αποδεκτής μορφής ενός εγγράφου XML
- Εκφράζει περιορισμούς στη δομή και το περιεχόμενο των εγγράφων
- Συχνά χρησιμοποιείται για τον καθορισμό μιας «σύμβασης» μεταξύ συστημάτων – «Το σύστημά μου θα δέχεται μόνο XML που συμμορφώνεται με αυτό το συγκεκριμένο Σχήμα».
- Εάν ένα συγκεκριμένο κομμάτι XML πληροί τις προδιαγραφές του Σχήματος - λέγεται ότι το «επικυρώνει»

http://en.wikipedia.org/wiki/Xml_schema

Επικύρωση XML

Έγγραφο XML

Σύμβαση
Σχήματος XML



Επικύρωση XML

Έγγραφο XML

```
<άτομο>
  <επίθετο>Severance</επίθετο>
  <ηλικία>17</ηλικία>
  <ημγεννησης>2001-04-17</ημγεννησης>
</άτομο>
```

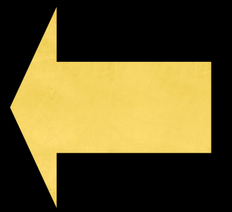
Σύμβαση Σχήματος XML

```
<xs:complexType name="άτομο">
 <xs:sequence>
  <xs:element name="επίθετο" type="xs:string"/>
  <xs:element name="ηλικία" type="xs:integer"/>
  <xs:element name="ημγεννησης" type="xs:date"/>
 </xs:sequence>
</xs:complexType>
```

Επικυρωτής

Πολλές Γλώσσες Σχήματος XML

- Ορισμός Τύπου Εγγράφου (Document Type Definition - DTD)
  - http://en.wikipedia.org/wiki/Document_Type_Definition
- Γενικευμένο Πρότυπο Γλώσσας Σήμανσης (ISO 8879:1986 SGML)
  - http://en.wikipedia.org/wiki/SGML
- Σχήμα XML από το W3C - (XSD)
  - http://en.wikipedia.org/wiki/XML_Schema_(W3C)



http://en.wikipedia.org/wiki/Xml_schema

XSD Σχήμα XML (προδιαγραφές W3C)

- Θα εστιάσουμε στην έκδοση της Κοινοπραξίας του Ιστού Παγκόσμιας Εμβέλειας (World Wide Web Consortium W3C)
- Συχνά ονομάζεται «Σχήμα W3C» επειδή το «Σχήμα» θεωρείται γενικό
- Συνηθέστερα ονομάζεται XSD επειδή τα ονόματα αρχείων τελειώνουν σε .xsd

http://www.w3.org/XML/Schema

http://en.wikipedia.org/wiki/XML_Schema_(W3C)

Δομή XSD

- xs:στοιχείο
- xs:αλληλουχία
- xs:σύνθετοςΤύπος

```
<άτομο>
  <επίθετο>Severance</επίθετο>
  <ηλικία>17</ηλικία>
  <ημγέννησης>2001-04-17</ημγέννησης>
</άτομο>
```

```
<xs:complexType name="άτομο">
 <xs:sequence>
  <xs:element name="επίθετο" type="xs:string"/>
  <xs:element name="ηλικία" type="xs:integer"/>
  <xs:element name="ημγέννησης" type="xs:date"/>
 </xs:sequence>
</xs:complexType>
```

Περιορισμοί XSD

```
<xs:element name="άτομο">
  <xs:complexType>
    <xs:sequence>
      <xs:element name= "πλήρες_όνομα" type="xs:string"
          minOccurs="1" maxOccurs="1" />
      <xs:element name= "όνομα_παιδιού" type="xs:string"
            minOccurs="0" maxOccurs="10" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<άτομο>
  <πλήρες_όνομα>Tove Refsnes</πλήρες_όνομα>
  <όνομα_παιδιού>Hege</όνομα_παιδιού>
  <όνομα_παιδιού>Stale</όνομα_παιδιού>
  <όνομα_παιδιού>Jim</όνομα_παιδιού>
  <όνομα_παιδιού>Borge</όνομα_παιδιού>
</άτομο>
```

http://www.w3schools.com/Schema/schema_complex_indicators.asp

Τύποι Δεδομένων XSD

```
<xs:element name= "πελάτης" type="xs:string"/>
<xs:element name= "έναρξη" type="xs:date"/>
<xs:element name= "ημέναρξης" type="xs:dateTime"/>
<xs:element name= "τιμή" type="xs:decimal"/>
<xs:element name= "εβδομάδες" type="xs:integer"/>
```

Είναι σύνηθες να αναπαρίσταται ο χρόνος σε UTC/GMT, δεδομένου ότι οι διακομιστές είναι συχνά διασκορπισμένοι σε όλο τον κόσμο

```
<πελάτης>John Smith</πελάτης>
<έναρξη>2002-09-24</έναρξη>
<ημέναρξης>2002-05-30T09:30:10Z</ημέναρξης>
<τιμή>999.50</τιμή>
<εβδομάδες>30</εβδομάδες>
```

http://www.w3schools.com/Schema/schema_dtypes_numeric.asp

ISO 8601 Μορφή Ημερομηνίας/Ώρας

2002-05-30T09:30:10Z

Έτος-μήνας-ημέρα

Ώρα της μέρας

Ζώνη ώρας - συνήθως καθορίζεται σε UTC / GMT και όχι σε τοπική ζώνη ώρας

http://en.wikipedia.org/wiki/ISO_8601

http://en.wikipedia.org/wiki/Coordinated_Universal_Time

```xml
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Address">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Recipient" type="xs:string" />
        <xs:element name="House" type="xs:string" />
        <xs:element name="Street" type="xs:string" />
        <xs:element name="Town" type="xs:string" />
        <xs:element minOccurs="0" name="County" type="xs:string" />
        <xs:element name="PostCode" type="xs:string" />
        <xs:element name="Country">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="FR" />
              <xs:enumeration value="DE" />
              <xs:enumeration value="ES" />
              <xs:enumeration value="UK" />
              <xs:enumeration value="US" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



```xml
<?xml version="1.0" encoding="utf-8"?>
<Address
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="SimpleAddress.xsd">
  <Recipient>Mr. Walter C. Brown</Recipient>
  <House>49</House>
  <Street>Featherstone Street</Street>
  <Town>LONDON</Town>
  <PostCode>EC1Y 8SY</PostCode>
  <Country>UK</Country>
</Address>
```

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="shiporder">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="orderperson" type="xs:string"/>
   <xs:element name="shipto">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="address" type="xs:string"/>
      <xs:element name="city" type="xs:string"/>
      <xs:element name="country" type="xs:string"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
   <xs:element name="item" maxOccurs="unbounded">
    <xs:complexType>
     <xs:sequence>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="note" type="xs:string" minOccurs="0"/>
      <xs:element name="quantity" type="xs:positiveInteger"/>
      <xs:element name="price" type="xs:decimal"/>
     </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
  <xs:attribute name="orderid" type="xs:string" use="required"/>
 </xs:complexType>
</xs:element>
</xs:schema>
```

http://www.w3schools.com/Schema/schema_example.asp

xml1.py

```
import xml.etree.ElementTree as ET
data = '''<άτομο>
  <όνομα>Chuck</όνομα>
  <τηλέφωνο type="intl">
     +1 734 303 4456
   </τηλέφωνο>
   <email hide="yes"/>
</άτομο>'''

tree = ET.fromstring(data)
print('Όνομα:',tree.find('όνομα').text)
print('Attr:',tree.find('email').get('hide'))
```

xml2.py

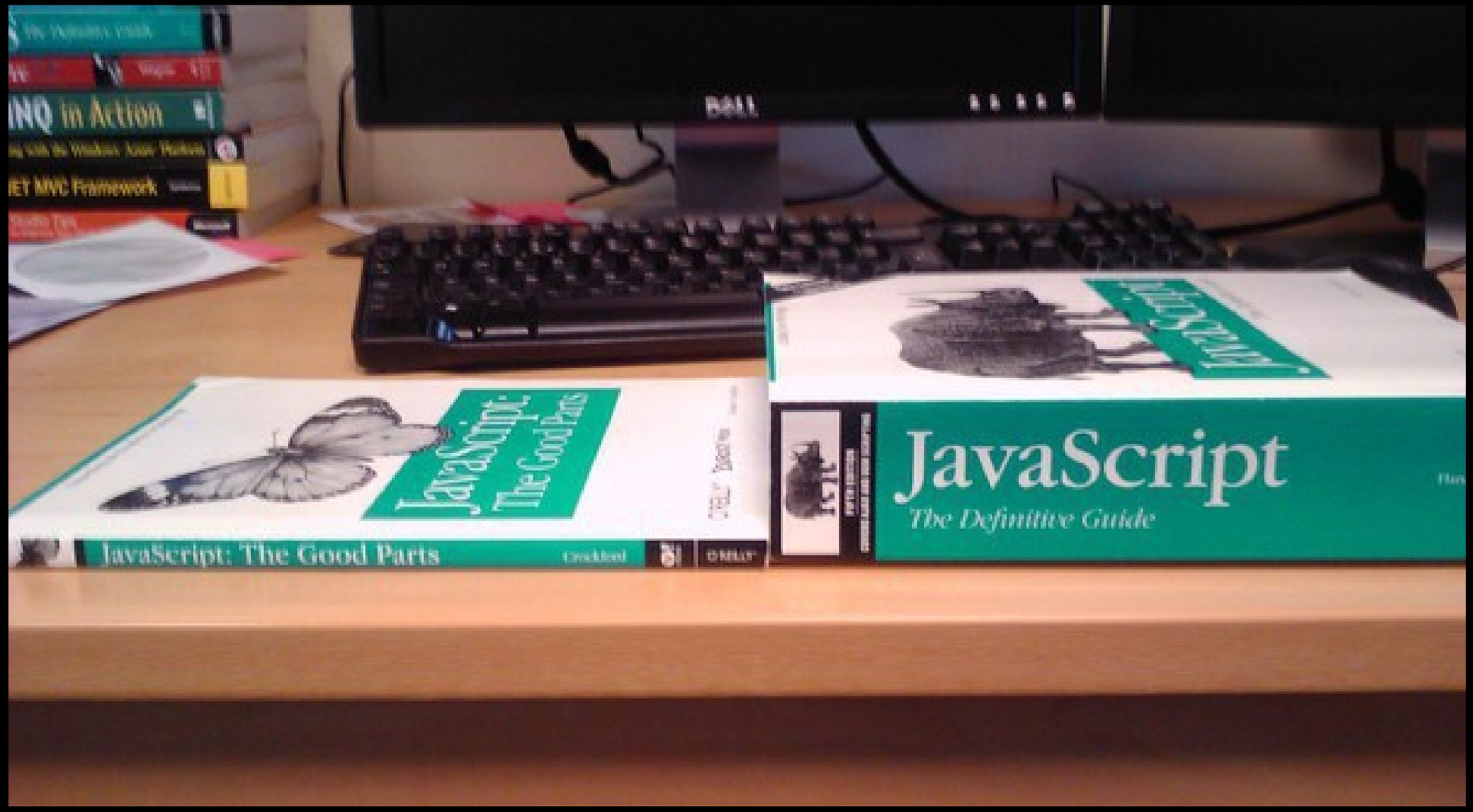
```
import xml.etree.ElementTree as ET
input = '''<stuff>
    <χρήστες>
        <χρήστης x="2">
            <id>001</id>
            <όνομα>Chuck</όνομα>
        </χρήστης>
        <χρήστης x="7">
            <id>009</id>
            <όνομα>Brent</όνομα>
        </χρήστης>
    </χρήστες>
</stuff>'''

stuff = ET.fromstring(input)
lst = stuff.findall('χρήστες/χρήστης')
print('Πλήθος χρηστών:', len(lst))
for στοιχείο in lst:
    print('Όνομα', στοιχείο.find('όνομα').text)
    print('Id', στοιχείο.find('id').text)
    print('Attribute', στοιχείο.get("x"))
```

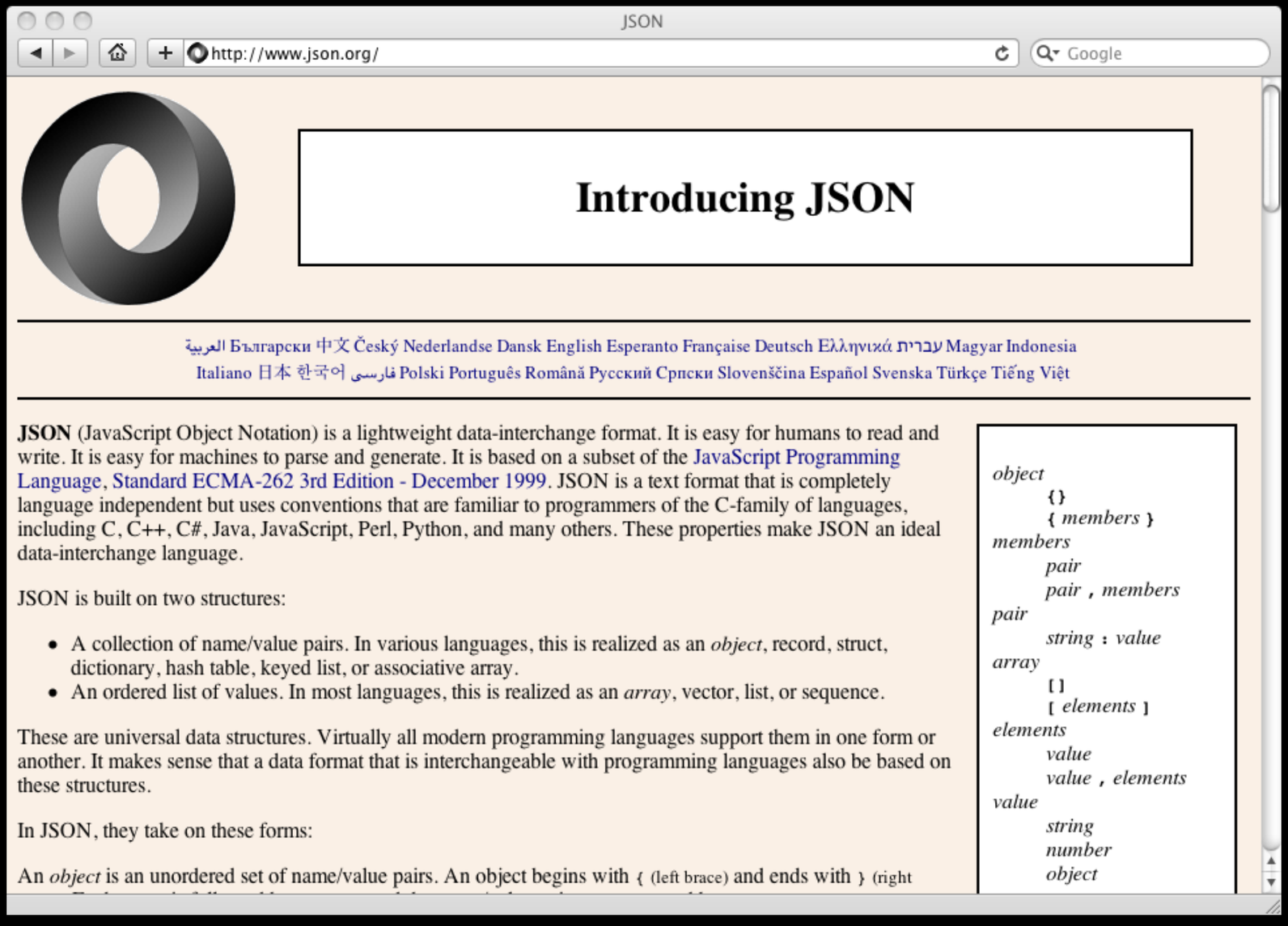
Σημειογραφία Αντικειμένου JavaScript - JavaScript Object Notation

JavaScript Object Notation

- Ο Douglas Crockford - “Ανακάλυψε” την JSON
- Σημειογραφία σταθερού αντικειμένου σε JavaScript

http://www.youtube.com/watch?v=kc8BAR7SHJI


NQ in Action
ET MVC Framework
JavaScript: The Good Parts
O'REILLY
JavaScript
The Definitive Guide


JSON
http://www.json.org/
Google
Introducing JSON
العربية Български 中文 Český Nederlandse Dansk English Esperanto Française Deutsch Ελληνικά עברית Magyar Indonesia
Italiano 日本 한국어 فارسی Polski Português Română Русский Српски Slovenščina Español Svenska Türkçe Tiếng Việt
JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.
JSON is built on two structures:
A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.
These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.
In JSON, they take on these forms:
An object is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right
object
{}
{ members }
members
pair
pair , members
pair
string : value
array
[]
[elements]
elements
value
value , elements
value
string
number
object

json1.py

```
import json
data = '''{
  "όνομα" : "Chuck",
  "τηλέφωνο" : {
    "τύπος" : "intl",
    "αριθμός" : "+1 734 303 4456"
   },
   "email" : {
    "κρυφό" : "yes"
   }
}'''

info = json.loads(data)
print('Όνομα:',info["όνομα"])
print('Κρυφό:',info["email"]["κρυφό"])
```

Η JSON αναπαριστά τα δεδομένα ως εμφωλευμένες "λίστες" και "λεξικά"

json2.py

```
import json
είσοδος = '''[
  { "id" : "001",
    "x" : "2",
    "όνομα" : "Chuck"
  } ,
  { "id" : "009",
    "x" : "7",
    "όνομα" : "Chuck"
  }
]'''

πληροφορία = json.loads(είσοδος)
print('Πλήθος Χρηστών:', len(πληροφορία))
for είδος in πληροφορία:
    print('Name', είδος['όνομα'])
    print('Id', είδος['id'])
    print('Ιδιότητα', είδος['x'])
```

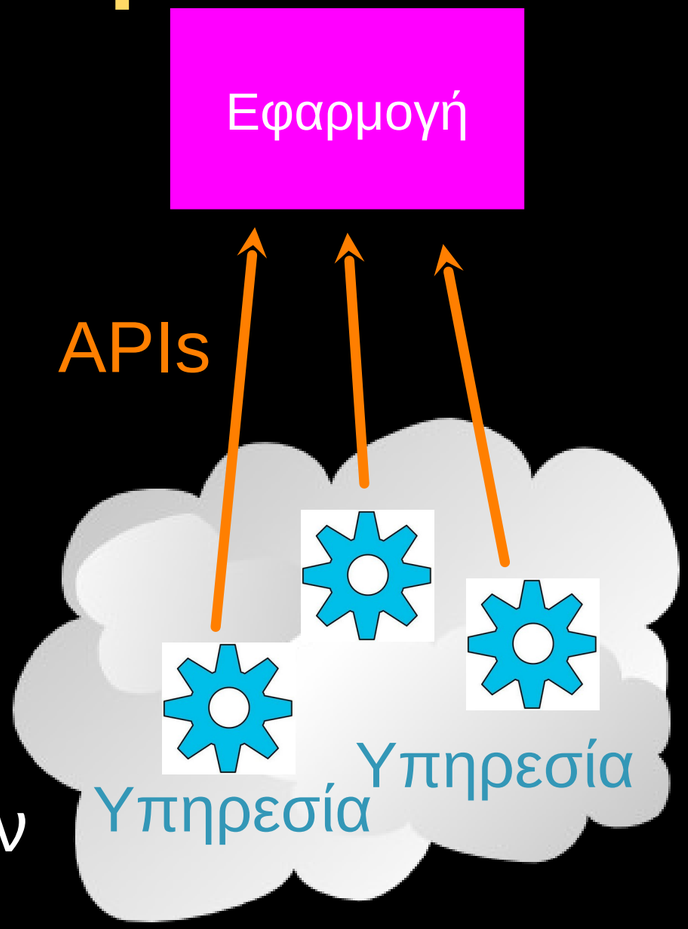
Η JSON αναπαριστά τα δεδομένα ως εμφωλευμένες "λίστες" και "λεξικά"

Προσέγγιση Προσανατολισμένη στην Εξυπηρέτηση

http://en.wikipedia.org/wiki/Service-oriented_architecture

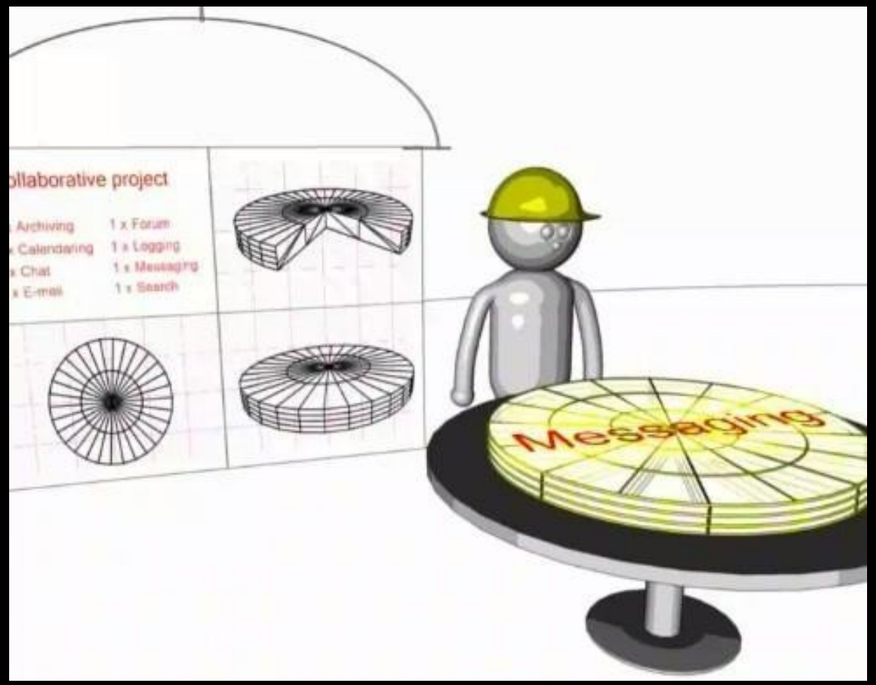
Προσέγγιση Προσανατολισμένη στην Εξυπηρέτηση

- Οι περισσότερες μέτριου μεγέθους ή μεγαλύτερες εφαρμογές ιστού χρησιμοποιούν υπηρεσίες
- Χρησιμοποιούν υπηρεσίες από άλλες εφαρμογές
  - Χρέωση Πιστωτικής Κάρτας
  - Συστήματα Κρατήσεων ξενοδοχείων
- Οι υπηρεσίες δημοσιεύουν τους «κανόνες» που πρέπει να ακολουθούν οι εφαρμογές για να κάνουν χρήση της υπηρεσίας (API)



Πολλαπλά Συστήματα

- Αρχικά - δύο συστήματα συνεργάζονται και κατανέμουν το πρόβλημα
- Καθώς τα δεδομένα / η υπηρεσία γίνονται χρήσιμα - πολλές εφαρμογές θέλουν να χρησιμοποιήσουν τις πληροφορίες / την εφαρμογή



http://www.youtube.com/watch?v=mj-kCFzF0ME

5:15

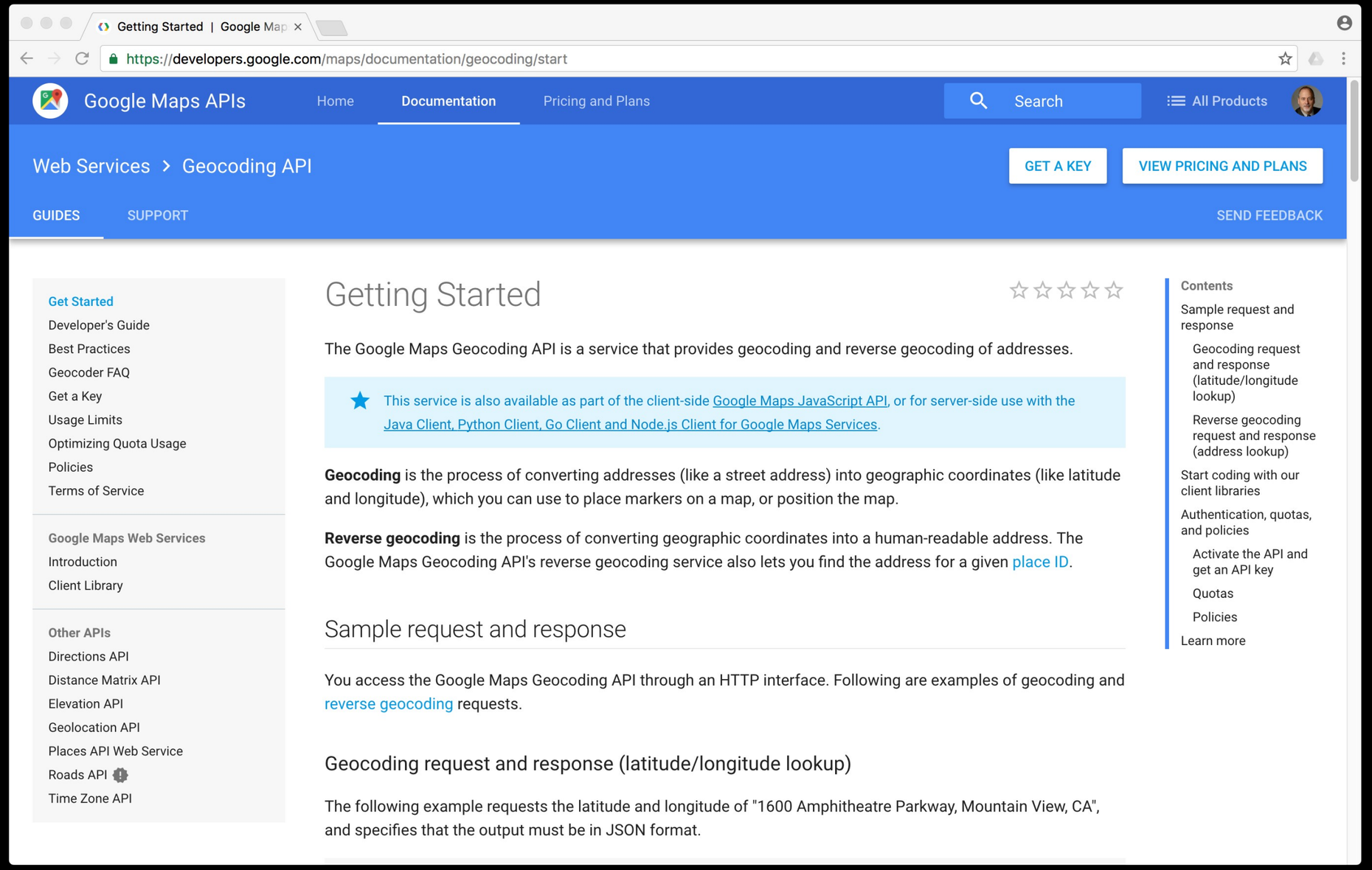
Υπηρεσίες Ιστού

http://en.wikipedia.org/wiki/Web_services

Διεπαφή Προγράμματος Εφαρμογής (API)

Το ίδιο το API είναι σε μεγάλο βαθμό αφηρημένο διότι καθορίζει μια διεπαφή και ελέγχει τη συμπεριφορά των αντικειμένων που καθορίζονται σε αυτήν τη διεπαφή. Το λογισμικό που παρέχει τη λειτουργικότητα που περιγράφεται από ένα API λέγεται ότι είναι «υλοποίηση» του API. Ένα API συνήθως ορίζεται με βάση τη γλώσσα προγραμματισμού που χρησιμοποιείται για τη δημιουργία μιας εφαρμογής.

http://en.wikipedia.org/wiki/API



https://developers.google.com/maps/documentation/geocoding/

```
{
    "status": "OK",
     "results": [
        {
            "geometry": {
                "location_type": "APPROXIMATE",
                 "location": {
                    "lat": 42.2808256,
                     "lng": -83.7430378
                }
            },
            "address_components": [
                {
                    "long_name": "Ann Arbor",
                     "types": [
                        "locality",
                         "political"
                    ],
                    "short_name": "Ann Arbor"
                }
             ],
             "formatted_address": "Ann Arbor, MI, USA",
             "types": [
                "locality",
                "political"
            ]
        }
    ]
}
```


http://maps.googleapis.com/maps/api/geocode/json?address=Ann+Arbor%2C+MI

geojson.py

```
import urllib.request, urllib.parse, urllib.error
import json

serviceurl = 'http://maps.googleapis.com/maps/api/geocode/json?'

while True:
    διεύθυνση = input('Εισάγετε τοποθεσία: ')
    if len(διεύθυνση) < 1: break

    url = serviceurl + urllib.parse.urlencode({'διεύθυνση': διεύθυνση})

    print('Ανάκτηση', url)
    uh = urllib.request.urlopen(url)
    δεδομένα = uh.read().decode()
    print('Ανακτήθηκαν', len(δεδομένα), 'χαρακτήρες')

    try:
        js = json.loads(δεδομένα)
    except:
        js = None

    if not js or 'status' not in js or js['status'] != 'OK':
        print('==== Αποτυχία Ανάκτησης ====')
        print(δεδομένα)
        continue

    lat = js["results"][0]["geometry"]["location"]["lat"]
    lng = js["results"][0]["geometry"]["location"]["lng"]
    print('Γ.πλάτος ', lat, 'Γ.μήκος ', lng)
    τοποθεσία = js['results'][0]['formatted_address']
    print(τοποθεσία)
```

```
Εισάγετε τοποθεσία: Ann Arbor, MI
Ανάκτηση http://maps.googleapis.com/...
Ανακτήθηκαν 1669 χαρακτήρες
Γ.πλάτος 42.2808256 Γ.μήκος  -83.7430378
Ann Arbor, MI, USA
Εισάγετε τοποθεσία :
```

geojson.py

Ασφάλεια API και περιορισμός πρόσβασης

- Οι υπολογιστικοί πόροι για την εκτέλεση αυτών των API δεν είναι «ανέξοδοι»
- Τα δεδομένα που παρέχονται από αυτά τα API είναι συνήθως πολύτιμα
- Οι πάροχοι δεδομένων ενδέχεται να περιορίσουν τον αριθμό αιτημάτων ανά ημέρα, να απαιτούν ένα «κλειδί» API ή ακόμη και να χρεώνουν τη χρήση
- Μπορεί να αλλάξουν τους κανόνες καθώς τα πράγματα εξελίσσονται...

Usage Limits

The Google Geocoding API has the following limits in place:

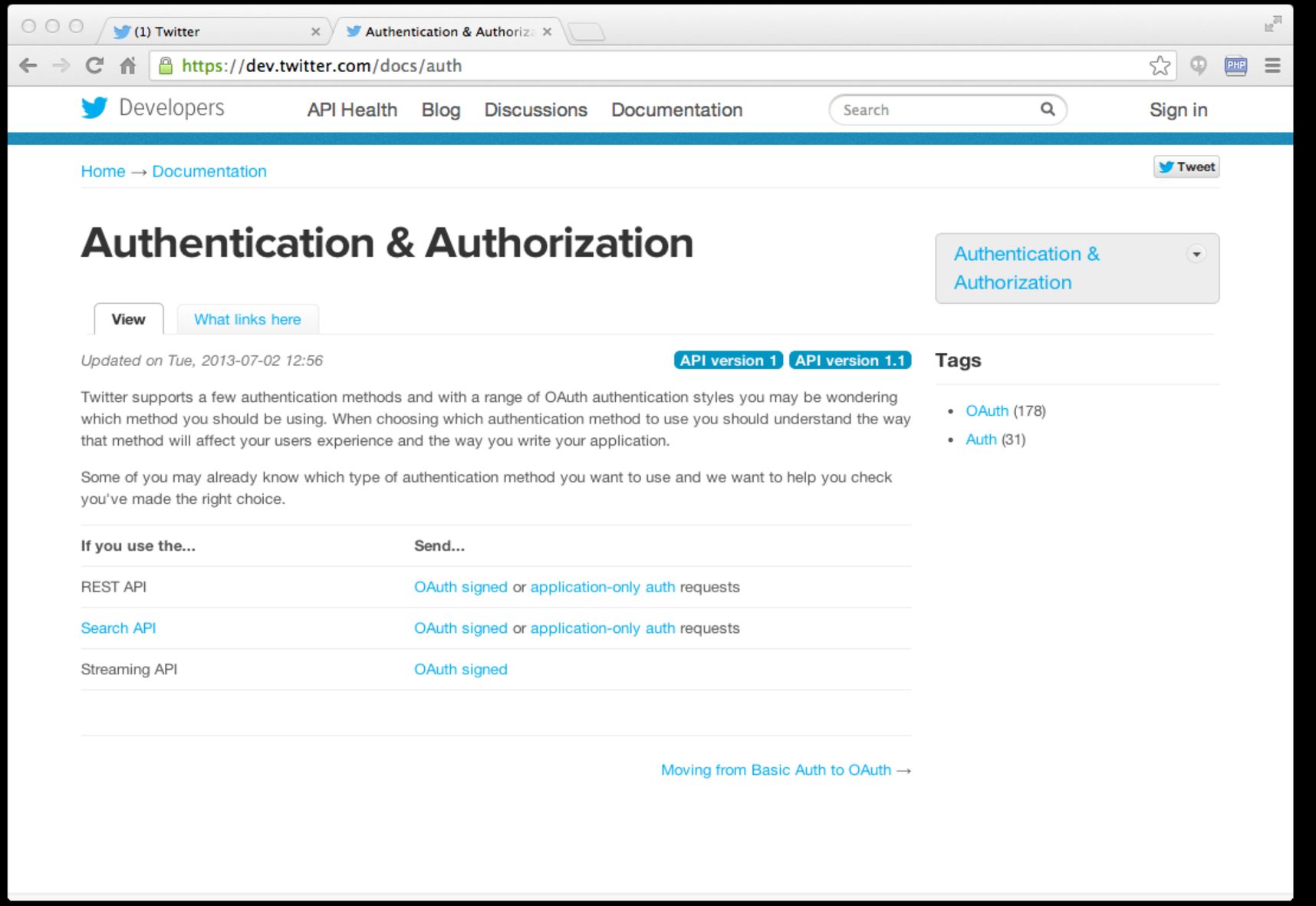
- 2,500 requests per day.

Google Maps API for Business customers have higher limits:

- 100,000 requests per day.

These limits are enforced to prevent abuse and/or repurposing of the Geocoding API, and may be changed in the future without notice. Additionally, we enforce a request rate limit to prevent abuse of the service. If you exceed the 24-hour limit or otherwise abuse the service, the Geocoding API may stop working for you temporarily. If you continue to exceed this limit, your access to the Geocoding API may be blocked.

The Geocoding API may only be used in conjunction with a Google map; geocoding results without displaying them on a map is prohibited. For complete details on allowed usage, consult the Maps API Terms of Service License Restrictions.


(1) Twitter
Authentication & Authoriz
https://dev.twitter.com/docs/auth
Developers
API Health
Blog
Discussions
Documentation
Search
Sign in
Home → Documentation
Tweet
Authentication & Authorization
Authentication & Authorization
View
What links here
Updated on Tue, 2013-07-02 12:56
API version 1
API version 1.1
Twitter supports a few authentication methods and with a range of OAuth authentication styles you may be wondering which method you should be using. When choosing which authentication method to use you should understand the way that method will affect your users experience and the way you write your application.
Some of you may already know which type of authentication method you want to use and we want to help you check you've made the right choice.
If you use the...
Send...
REST API
OAuth signed or application-only auth requests
Search API
OAuth signed or application-only auth requests
Streaming API
OAuth signed
Moving from Basic Auth to OAuth →
Tags
OAuth (178)
Auth (31)

Tweets | Twitter Developer

https://dev.twitter.com/docs/platform-objects/tweets

Developers API Health Blog Discussions Documentation Search Sign in

Home → Documentation → Platform Objects Tweet

Tweets

View What links here

Updated on Tue, 2013-08-13 17:29 API version 1 API version 1.1

Tweets are the basic atomic building block of all things Twitter. Users tweet Tweets, also known more generically as "status updates." Tweets can be embedded, replied to, favorited, unfavorited and deleted.

Brian Sutorius Follow
@bsuto

The "http://" at the beginning of URLs is a command to the browser. It stands for "head to this place:" followed by two laser-gun noises.

4:29 PM - 21 Feb 2012

4,218 RETWEETS **1,768** FAVORITES

Field Guide

Consumers of Tweets should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing. Please note that Tweets found in Search results vary somewhat in structure from this document.

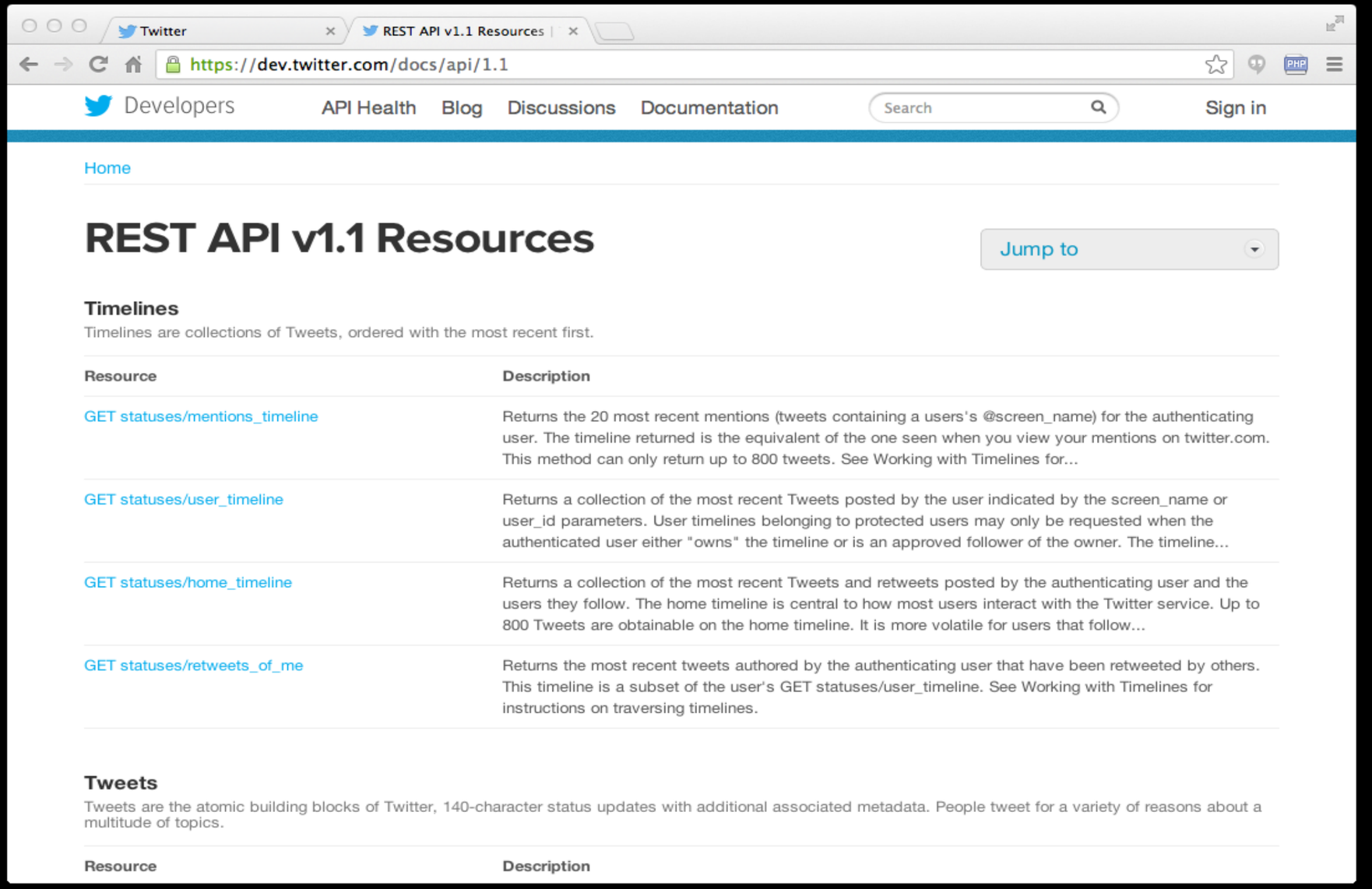
Field	Type	Description
annotations	Object	*Unused.* Future/beta home for status annotations.

Natural habitat

Tweets can be found alone, within user objects, but most often within timelines.

Related API Resources

- GET favorites

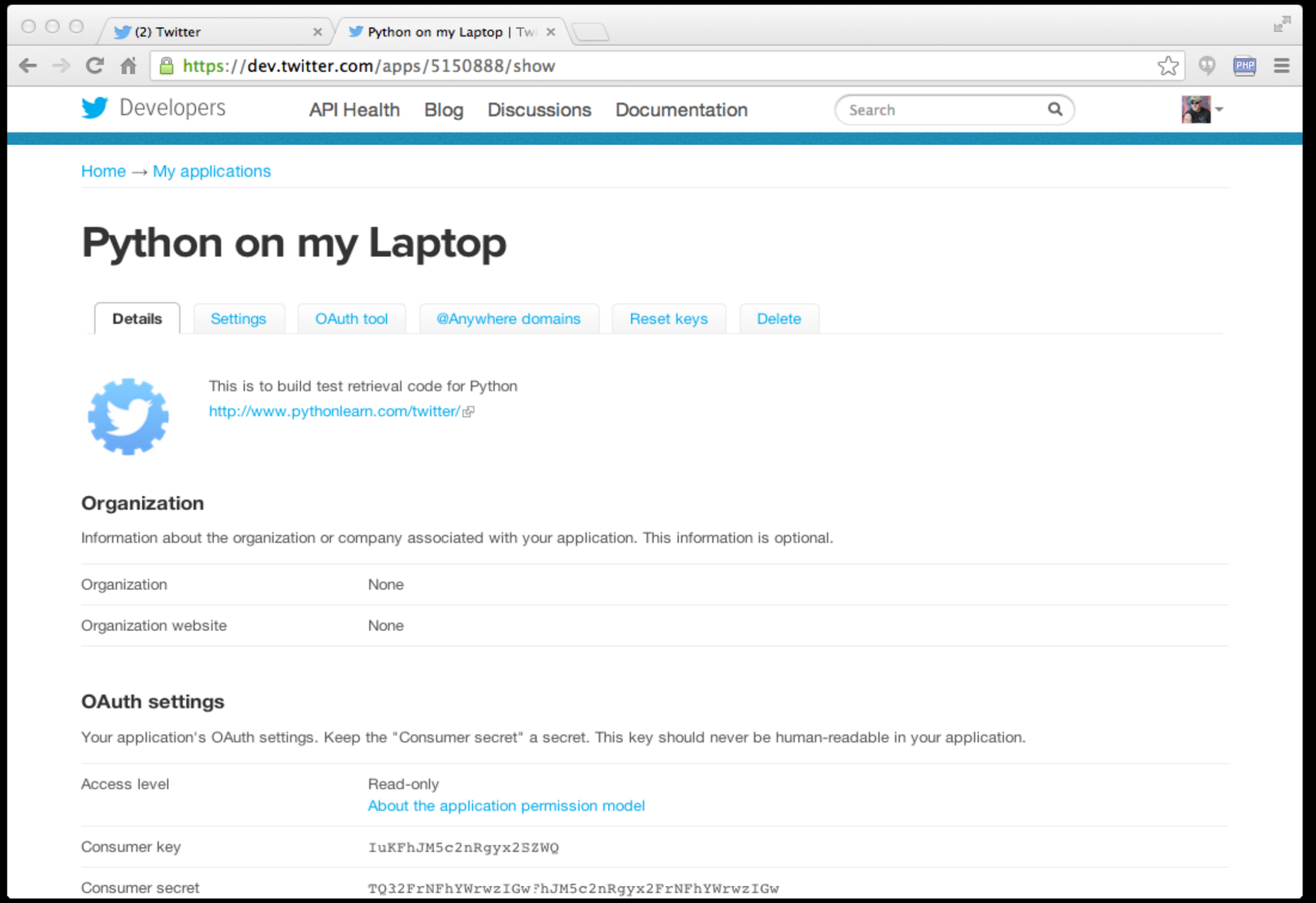

Twitter
REST API v1.1 Resources
https://dev.twitter.com/docs/api/1.1
Developers
API Health
Blog
Discussions
Documentation
Search
Sign in
Home
REST API v1.1 Resources
Jump to
Timelines
Timelines are collections of Tweets, ordered with the most recent first.
Resource
Description
GET statuses/mentions_timeline
Returns the 20 most recent mentions (tweets containing a users's @screen_name) for the authenticating user. The timeline returned is the equivalent of the one seen when you view your mentions on twitter.com. This method can only return up to 800 tweets. See Working with Timelines for...
GET statuses/user_timeline
Returns a collection of the most recent Tweets posted by the user indicated by the screen_name or user_id parameters. User timelines belonging to protected users may only be requested when the authenticated user either "owns" the timeline or is an approved follower of the owner. The timeline...
GET statuses/home_timeline
Returns a collection of the most recent Tweets and retweets posted by the authenticating user and the users they follow. The home timeline is central to how most users interact with the Twitter service. Up to 800 Tweets are obtainable on the home timeline. It is more volatile for users that follow...
GET statuses/retweets_of_me
Returns the most recent tweets authored by the authenticating user that have been retweeted by others. This timeline is a subset of the user's GET statuses/user_timeline. See Working with Timelines for instructions on traversing timelines.
Tweets
Tweets are the atomic building blocks of Twitter, 140-character status updates with additional associated metadata. People tweet for a variety of reasons about a multitude of topics.
Resource
Description

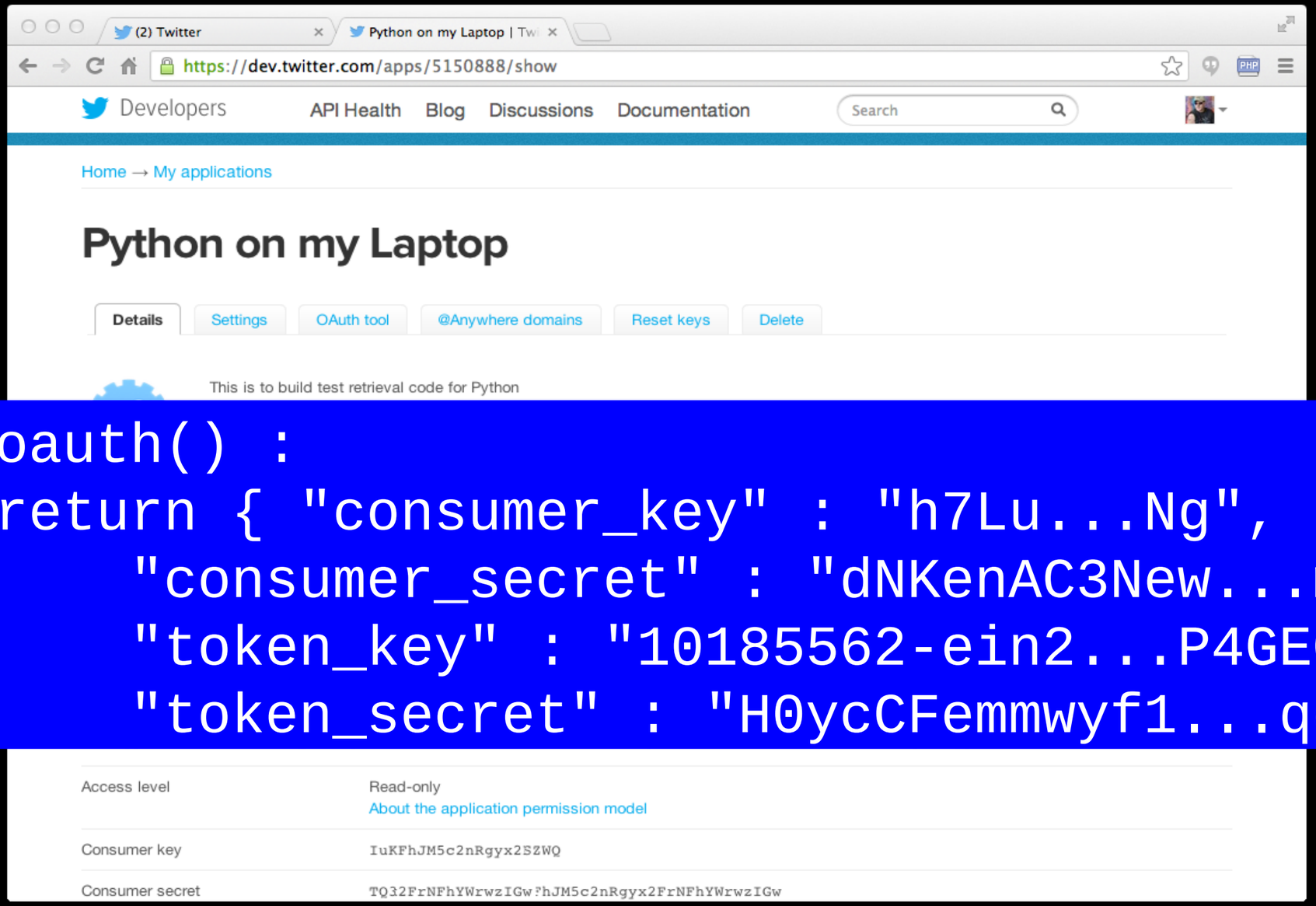
twitter2.py

```
import urllib.request, urllib.parse, urllib.error
import twurl
import json

TWITTER_URL = 'https://api.twitter.com/1.1/friends/list.json'

while True:
    print('')
    acct = input('Εισάγετε τον λογαριασμό Twitter:')
    if (len(acct) < 1): break
    url = twurl.augment(TWITTER_URL,
                        {'screen_name': acct, 'count': '5'})
    print('Ανάκτηση', url)
    connection = urllib.request.urlopen(url)
    data = connection.read().decode()
    headers = dict(connection.getheaders())
    print('Απομένουν', headers['x-rate-limit-remaining'])
    js = json.loads(data)
    print(json.dumps(js, indent=4))

    for u in js['users']:
        print(u['screen_name'])
        s = u['status']['text']
        print('  ', s[:50])
```

twitter2.py

```
Εισάγετε τον λογαριασμό Twitter:drchuck
Ανάκτηση https://api.twitter.com/1.1/friends ...
Απομένουν 14
{
    "users": [
        {
            "status": {
                "text": "@jazzychad I just bought one .__.",
                 "created_at": "Fri Sep 20 08:36:34 +0000 2013",
             },
             "location": "San Francisco, California",
             "screen_name": "leahculver",
             "name": "Leah Culver",
         },
         {
            "status": {
                "text": "RT @WSJ: Big employers like Google ...",
                 "created_at": "Sat Sep 28 19:36:37 +0000 2013",
             },
             "location": "Victoria Canada",
             "screen_name": "_valeriei",
             "name": "Valerie Irvine",
     ],
}
Leahculver
   @jazzychad I just bought one .__._
Valeriei
   RT @WSJ: Big employers like Google, AT&amp;amp;T are h
Ericbollens
   RT @lukew: sneak peek: my LONG take on the good &a
halherzog
  Learning Objects is 10. We had a cake with the LO,
```


(2) Twitter
Python on my Laptop | Tw
https://dev.twitter.com/apps/5150888/show
Developers
API Health
Blog
Discussions
Documentation
Search
Home → My applications
Python on my Laptop
Details
Settings
OAuth tool
@Anywhere domains
Reset keys
Delete
This is to build test retrieval code for Python
http://www.pythonlearn.com/twitter/
Organization
Information about the organization or company associated with your application. This information is optional.
Organization
None
Organization website
None
OAuth settings
Your application's OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.
Access level
Read-only
About the application permission model
Consumer key
IuKFhJM5c2nRgyx2SZWQ
Consumer secret
TQ32FrNFhYWrwzIGw?hJM5c2nRgyx2FrNFhYWrwzIGw



hidden.py

```
def oauth() :
    return { "consumer_key" : "h7Lu...Ng",
        "consumer_secret" : "dNKenAC3New...mmn7Q",
        "token_key" : "10185562-ein2...P4GEQQOSGI",
        "token_secret" : "H0ycCFemmwyf1...qoIpBo" }
```

(1) Twitter | OAuth | Twitter Developers

https://dev.twitter.com/docs/auth/oauth

Developers | API Health | Blog | Discussions | Documentation | Search | Sign in

Home → Documentation

Tweet

OAuth

OAuth

View | What links here

Updated on Mon, 2013-03-11 12:22

API version 1 | API version 1.1

Send secure authorized requests to the Twitter API

Twitter uses OAuth to provide authorized access to its API.

Features

- **Secure** - Users are not required to share their passwords with 3rd party applications, increasing account security.
- **Standard** - A wealth of client libraries and example code are compatible with Twitter's OAuth implementation.

Related Questions

- Do Twitter's OAuth 1.0A access tokens expire?
- Will an application have to request user authorization just to make public API calls?

Tags

- OAuth (178)
- Auth (31)

twurl.py

```
import urllib
import oauth
import hidden

def augment(url, parameters) :
    secrets = hidden.oauth()
    consumer = oauth.OAuthConsumer(secrets['consumer_key'], secrets['consumer_secret'])
    token = oauth.OAuthToken(secrets['token_key'],secrets['token_secret'])
    oauth_request = oauth.OAuthRequest.from_consumer_and_token(consumer,
        token=token, http_method='GET', http_url=url, parameters=parameters)
    oauth_request.sign_request(oauth.OAuthSignatureMethod_HMAC_SHA1(), consumer, token)
    return oauth_request.to_url()
```

https://api.twitter.com/1.1/statuses/user_timeline.json?count=2&oauth_version=1.0&oauth_token=101...SGI&screen_name=drchuck&oauth_nonce=09239679&oauth_timestamp=1380395644&oauth_signature=rLK...BoD&oauth_consumer_key=h7Lu...GNg&oauth_signature_method=HMAC-SHA1

Σύνοψη

- Αρχιτεκτονική Προσανατολισμένη στην Υπηρεσία - επιτρέπει σε μια εφαρμογή να χωριστεί σε μέρη και να διανεμηθεί σε ένα δίκτυο
- Η Διεπαφή Προγράμματος Εφαρμογών (Application Program Interface API) είναι μια σύμβαση αλληλεπίδρασης
- Οι Υπηρεσίες Ιστού παρέχουν υποδομή για εφαρμογές που συνεργάζονται (με API) μέσω δικτύου - το SOAP και το REST είναι δύο στυλ υπηρεσιών Ιστού
- Οι XML και JSON είναι μορφές σειριοποίησης

Ευχαριστίες / Συνεισφορές

Αρχική ανάπτυξη : Charles Severance, University of Michigan School of Information

Απόδοση στα Ελληνικά: Κιουρτίδου Δ. Κωνσταντία

… Εισαγάγετε νέους Μεταφραστές και άτομα που έχουν συνεισφέρει εδώ

Συνέχεια…