APIs

Allie Sauppé | CS4HS

February 3, 2018

Consider the following…

You want to…

- have your website display a Google calendar of events
- leverage common programming functionality (e.g., mathematical operations)
- use publicly available web fonts on your website

…but you don’t want to program it yourself

Building Blocks of Programming

Programs reuse common functionality

- mathematical operations
- using files
- accessing websites

Programming leverages these units to build complex programs

Application Programming Interfaces (APIs)

Allows programmers to access public functionality of a program

Provided by many programming languages, websites

- Java: https://docs.oracle.com/javase/9/docs/api/index.html?overview-summary.html
- Python: https://docs.python.org/3.6/c-api/index.html
- Pinterest: https://developers.pinterest.com/docs/getting-started/introduction/
- Twitter: https://developer.twitter.com/en/docs/api-reference-index

Website APIs also called web services

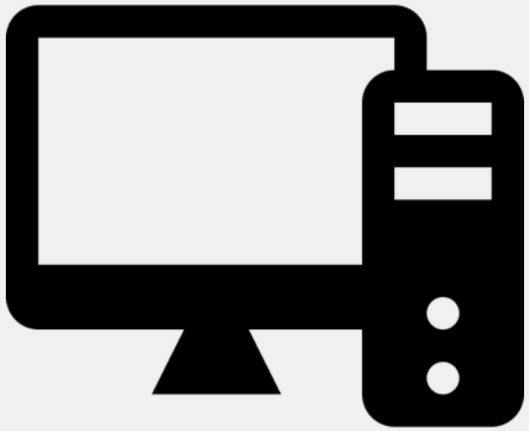
Why Web APIs?

Allows for communication with a common web service

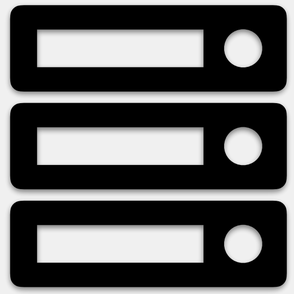
- display content from one web service in another context
- universal logins
- changing between mobile apps

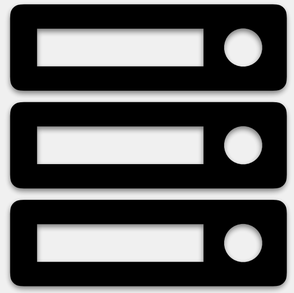
The Internet

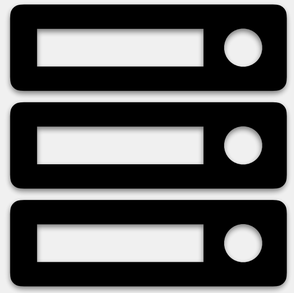
client



server

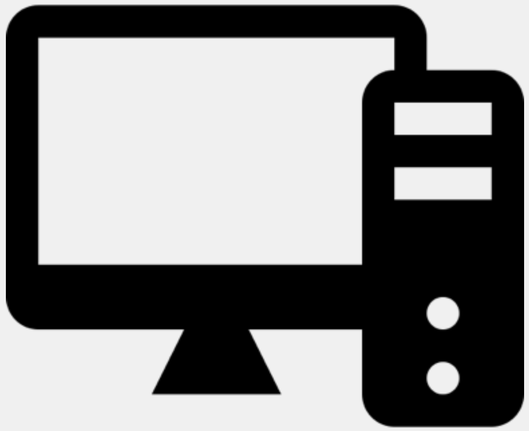






The Internet

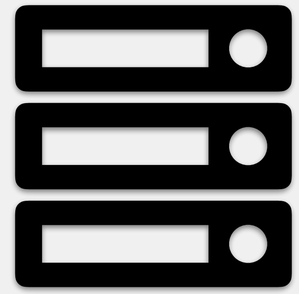
client

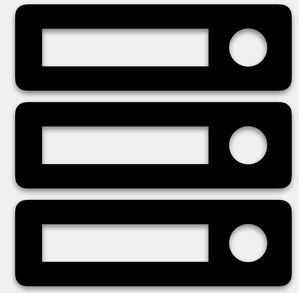


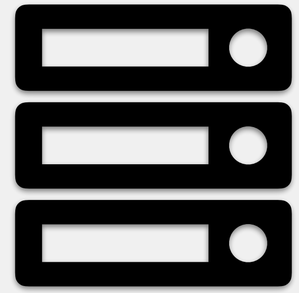
request for info

- url
- data/file
- ...

server

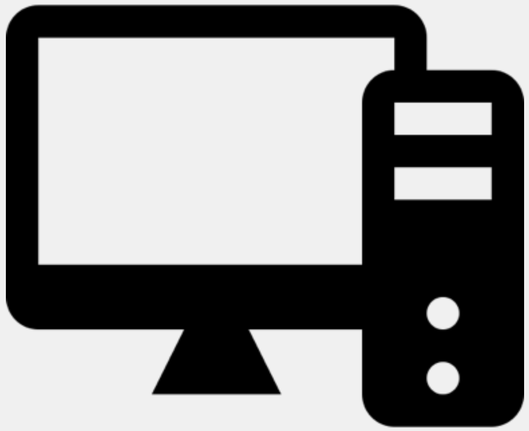






The Internet

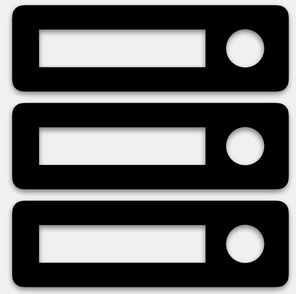
client

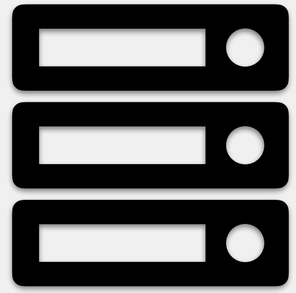


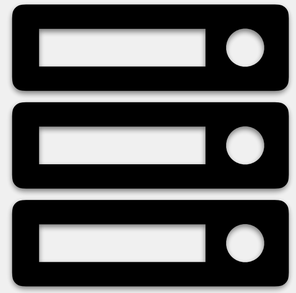
returns...

- website
- data/file
- ...

server

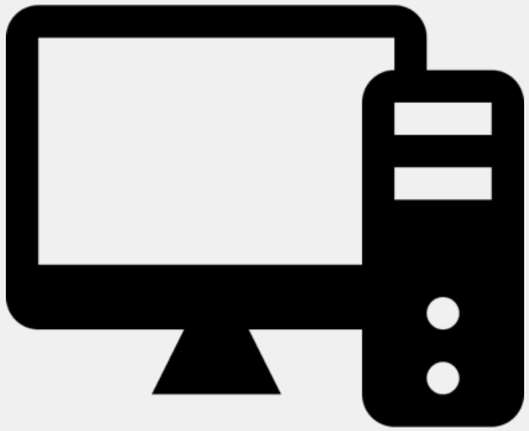






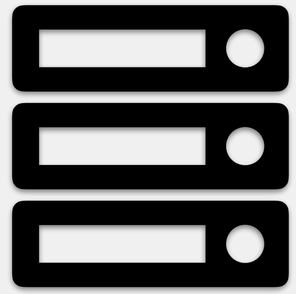
The Internet

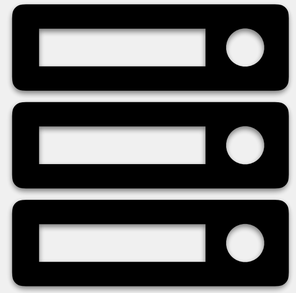
client

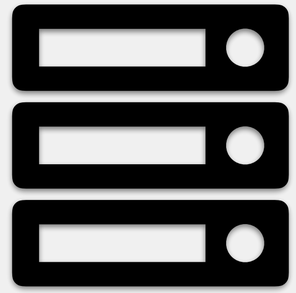


text

server







Data Formats

Several common formats for transmitting textual data

- csv, xml, json, url

Programming languages often have easy ways to import this data

Example: Data Formats

Storing restaurant data: name, avg. star rating, type of food

Name	AvgRating	FoodType
The Mint	4.5	American
Iguana's	4.25	Mexican

Data Formats: CSV

comma separated values

Name	AvgRating	FoodType
The Mint	4.5	American
Iguana's	4.25	Mexican

```
“Name”, “AvgRating”, “FoodType”
“The Mint”, 4.5, “American”
“Iguana’s”, 4.25, “Mexican
```

Data Formats: XML

extended markup language

Name	AvgRating	FoodType
The Mint	4.5	American
Iguana's	4.25	Mexican

```
<restaurant>
    <name>The Mint</name>
    <avgRating>4.5</avgRating>
    <foodType>American</foodType>
</restaurant>
```

Data Formats: XML

extended markup language

Name	AvgRating	FoodType
The Mint	4.5	American
Iguana's	4.25	Mexican

```
<restaurant>
    <name>The Mint</name>
    <avgRating>4.5</avgRating>
    <foodType>American</foodType>
</restaurant>
```

Data Formats: XML

extended markup language

Name	AvgRating	FoodType
The Mint	4.5	American
Iguana's	4.25	Mexican

```
<restaurant>
    <name>The Mint</name>
    <avgRating>4.5</avgRating>
    <foodType>American</foodType>
</restaurant>
```

Data Formats: XML

extended markup language

```
<restaurant>
    <name>The Mint</name>
    <avgRating>4.5</avgRating>
    <foodType>American</foodType>
</restaurant>
<restaurant>
    <name>Iguana’s</name>
    <avgRating>4.25</avgRating>
    <foodType>Mexican</foodType>
</restaurant>
```

Data Formats: JSON

JavaScript object notation

Name	AvgRating	FoodType
The Mint	4.5	American
Iguana's	4.25	Mexican

```
{
    “name” : “The Mint”,
    “avgRating” : 4.5,
    “foodType” : “American”
}
```

Data Formats: JSON

Javascript object notation

```
{
    “name” : “The Mint”,
    “avgRating” : 4.5,
    “foodType” : “American”
}
{
    “name” : “Iguana’s”,
    “avgRating” : 4.25,
    “foodType” : “Mexican”
}
```

Web Service Approaches

Numerous communication strategies exist

- web services will commit to one

Popular examples

- simple object access protocol (SOAP)
- representation state transfer (REST)

SOAP Basics

Tried-and-true format

Uses XML to structure communication

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/…">
  <soapenv:Body>
    <hs:GetHolidaysAvailable>
      <hs:countryCode>UnitedStates</hs:countryCode>
    </hs:GetHolidaysAvailable>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Basics

Tried-and-true format

Uses XML to structure communication

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/…">
  <soapenv:Body>
    <hs:GetHolidaysAvailable>
      <hs:countryCode>UnitedStates</hs:countryCode>
    </hs:GetHolidaysAvailable>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Basics

Tried-and-true format

Uses XML to structure communication

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/…">
  <soapenv:Body>
    <hs:GetHolidaysAvailable>
      <hs:countryCode>UnitedStates</hs:countryCode>
    </hs:GetHolidaysAvailable>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Basics

Tried-and-true format

Uses XML to structure communication

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/…">
  <soapenv:Body>
    <hs:GetHolidaysAvailable>
      <hs:countryCode>UnitedStates</hs:countryCode>
    </hs:GetHolidaysAvailable>
  </soapenv:Body>
</soapenv:Envelope>
```

SOAP Basics

Tried-and-true format

Uses XML to structure communication

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/…">
  <soapenv:Body>
    <hs:GetHolidaysAvailable>
      <hs:countryCode>UnitedStates</hs:countryCode>
    </hs:GetHolidaysAvailable>
  </soapenv:Body>
</soapenv:Envelope>
```

REST Basics

Newer approach

Uses a less-verbose communication structure

- bundles everything into the URL

Basic functionality

- GET: returns information
- POST: executes some addition/modification
- DELETE: removes something

Tokens

Utilizing REST as a developer requires *tokens*

Keys used for authentication

Need to sign up for an account

- possibly register as developing an application

Anatomy of a URL

REST utilizes key/value pairs in the URL to send additional information

```
https://verilymag.com/2017/04/ted-talks-fast-fashion-ethical-fashion-youtube?
utm_source=Verily%20Newsletter&utm_campaign=d270dfc25e-
EMAIL_CAMPAIGN_2018_02_01&utm_medium=email&utm_term=0_e08a3e62a0-
d270dfc25e-88890565
```

Anatomy of a URL

REST utilizes key/value pairs in the URL to send additional information

```
https://verilymag.com/2017/04/ted-talks-fast-fashion-ethical-fashion-youtube
?
utm_source=Verily%20Newsletter&
utm_campaign=d270dfc25e-EMAIL_CAMPAIGN_2018_02_01&
utm_medium=email&
utm_term=0_e08a3e62a0-d270dfc25e-88890565
```

Standard vs Mobile URLs

CNN: https://www.cnn.com/

mobile CNN: http://m.cnn.com/en

Pipeline

Client

- formats and sends request to server (programming language, API)

Server

- sends back data

Client

- parses data (JSON, CSV, ...)

Pinterest Exercise

Pinterest API explorer: https://developers.pinterest.com/tools/api-explorer/?