

Intro to API's

Connecting with other applications

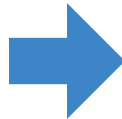
API

Application Programming Interface

API's are interfaces for code/computers to talk to one another.

```
app.get('/auth/facebook/callback',
  passport.authenticate('facebook-canvas', { failureRedirect: '/' }),
  function(req, res) {
    graph.setAccessToken(FB_ACCESS_TOKEN);
    // code is set
    // we'll send that and get the access token
    exports.graph = graph;
    res.redirect('/fbpage');
  });

//Facebook Canvas
app.post('/', function( req, res ){
  res.redirect(307, '/auth/facebook/canvas');
});
```



"FB, Talk to me!"

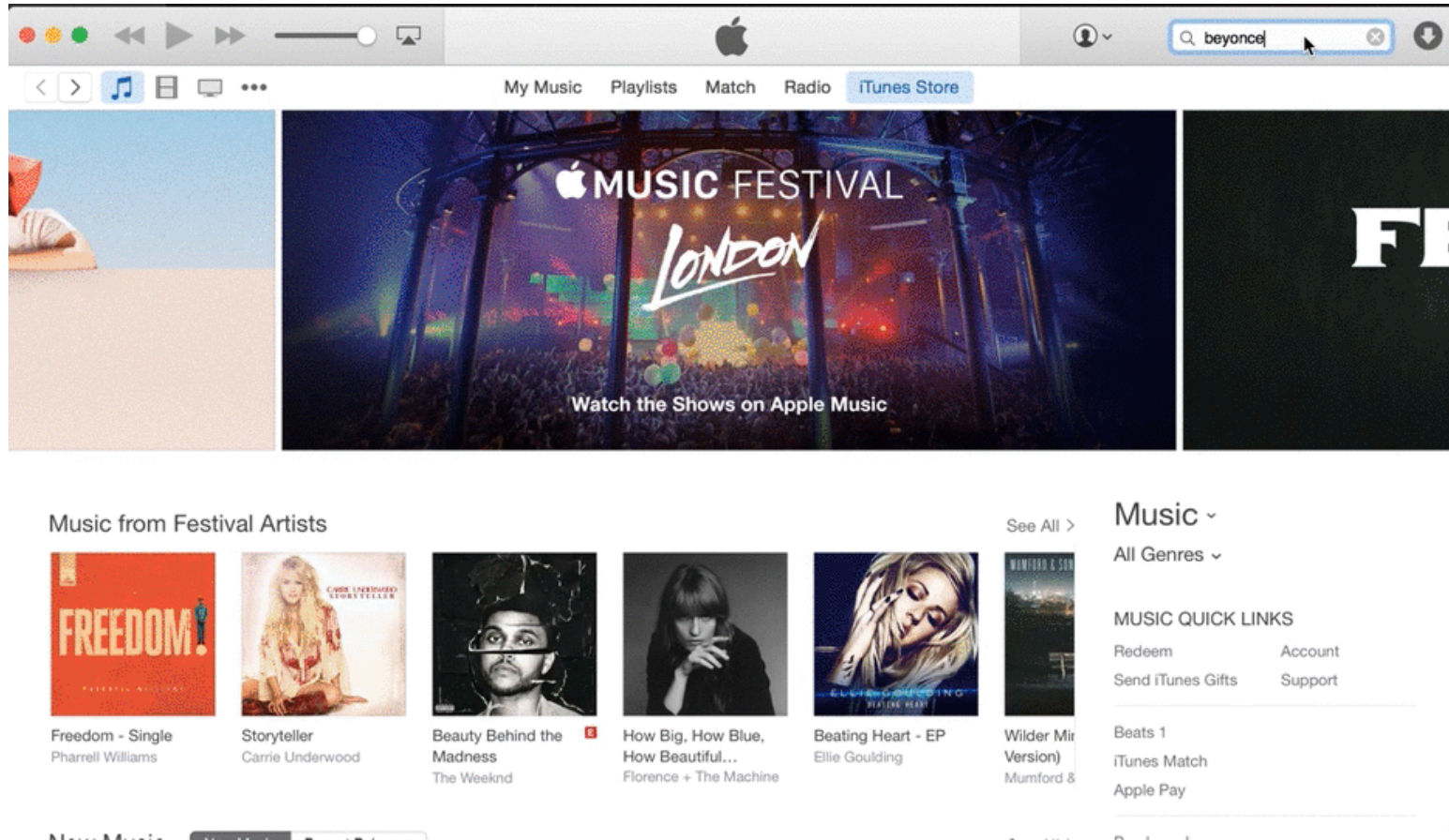


Web API's

Generally communicate via HTTP

- Twitter API, give me all tweets that mention "ice cream"
- Facebook API, send me the current user's profile picture
- Weather API, what is the weather in Missoula Montana?
- Reddit API, what is the current top post?
- GooglePlaces API, what gas stations are near the user?
- Yelp API, give me 10 restaurants in the zipcode 94110

A Human Interface



The itunes API

Make an HTTP request to:

<https://itunes.apple.com/search?term=beyonce&entity=musicVideo>

```
{
  resultCount: 50,
  - results: [
    - {
      wrapperType: "track",
      kind: "music-video",
      artistId: 1419227,
      collectionId: 780330041,
      trackId: 780332981,
      artistName: "Beyoncé",
      collectionName: "BEYONCÉ",
      trackName: "Partition (Video)",
      collectionCensoredName: "BEYONCÉ",
      trackCensoredName: "Partition (Video)",
      artistViewUrl: "https://itunes.apple.com/us/artist/beyonce/id1419227?uo=4",
      collectionViewUrl: "https://itunes.apple.com/us/music-video/partition-video/id780332981?uo=4",
      trackViewUrl: "https://itunes.apple.com/us/music-video/partition-video/id780332981?uo=4",
      previewUrl: "http://a143.v.phobos.apple.com/us/r1000/145/Video/v4/ff/94/62/ff946294-d872-7203-840d-b16c26ad9b6c/mzvf_6426187122334993115.640x480.h264lc.U.p.m4v",
      artworkUrl30: "http://is2.mzstatic.com/image/thumb/Video/v4/bf/3a/32/bf3a32d7-8c61-ab79-852b-6380d470a97d/d1.ocjuvnhd.jpg/40x30bb-85.jpg",
      artworkUrl60: "http://is4.mzstatic.com/image/thumb/Video/v4/bf/3a/32/bf3a32d7-8c61-ab79-852b-6380d470a97d/d1.ocjuvnhd.jpg/80x60bb-85.jpg",
      artworkUrl100: "http://is4.mzstatic.com/image/thumb/Video/v4/bf/3a/32/bf3a32d7-8c61-ab79-852b-6380d470a97d/d1.ocjuvnhd.jpg/100x100bb-85.jpg",
      collectionPrice: 15.99,
      trackPrice: 1.99,
      releaseDate: "2013-12-13T08:00:00Z",
      collectionExplicitness: "explicit",
      trackExplicitness: "explicit",
      discCount: 1,
      discNumber: 1,
      trackCount: 33,
      trackNumber: 22,
      trackTimeMillis: 229417,
      country: "USA",
      currency: "USD",
      primaryGenreName: "Pop",
      contentAdvisoryRating: "Explicit",
      radioStationUrl: "https://itunes.apple.com/station/idra.780332981"
    },
    - {
      wrapperType: "track",
      kind: "music-video",
      artistId: 1419227,
      collectionId: 780330041,
      trackId: 780330565,
      artistName: "Beyoncé",
      collectionName: "BEYONCÉ",
```

[+ - View source](#)

Make an HTTP request to an API

Get Some Data Back!

Search for Beyonce Albums

<https://itunes.apple.com/search?term=beyonce&entity=album>

Search for Beatles Songs

<https://itunes.apple.com/search?term=beatles&entity=song>

Search for Podcasts About "code"

<https://itunes.apple.com/search?term=code&entity=podcast>

Search for Harry Potter Movies

<https://itunes.apple.com/search?term=harry-potter&entity=movie>

Data Formats

When we use the internet, we make HTTP request and get HTML back.

API's don't respond with HTML. HTML contains information about the structure of a page. API's respond with data, not structure.

They use simpler data formats like **XML** and **JSON**

XML

Extended Markup Language

XML is syntactically similar to HTML, but it does not describe presentation like HTML does

```
<person>  
  <age>21</age>  
  <name>Travis</name>  
  <city>Los Angeles</city>  
</person>
```


JSON

Javascript Object Notation

JSON looks exactly like JavaScript objects, but everything is a string

```
{  
  "person": {  
    "age": "21",  
    "name": "Travis",  
    "city": "Los Angeles"  
  }  
}
```

XML vs. JSON

```
<person>
  <age>21</age>
  <name>Travis</name>
  <city>Los Angeles</city>
</person>
```

```
{
  "person": {
    "age": "21",
    "name": "Travis",
    "city": "Los Angeles"
  }
}
```

XML vs. JSON

```
<person>  
  <age>21</age>  
  <name>Travis</name>  
  <city>Los Angeles</city>  
</person>
```

```
{  
  "person": {  
    "age": "21",  
    "name": "Travis",  
    "city": "Los Angeles"  
  }  
}
```