

# Access Web APIs with XMLHttpRequest

---



**Richard Monson-Haefel**

SR. SOFTWARE ENGINEER

[www.monsonhaefel.com](http://www.monsonhaefel.com)



# Overview



## XMLHttpRequest

- > 25 years
- Documentation, reliability, ubiquitous
- Fetch, Axios, superagent, etc.
- XML and JSON
- XMLHttpRequest == XHR

## In this module

- Create and initialize an XHR object
- Send a Web API request
- Process a Web API response
  - Parsing the JSON payload
  - Displaying the data
- OMDB : Open Movie Database



# Understanding the OMDB API

---



# OMDB API URL Example

<http://www.omdbapi.com/?t=Star+Wars&y=1977>

HTTP Protocol

- GET
- POST
- PUT
- DELETE

Domain Name

Subdomain  
Path

Query Parameters



# OMDB API Query Parameters

`http://www.omdbapi.com/?t=Star+Wars&y=1977`



The diagram illustrates the structure of the URL `http://www.omdbapi.com/?t=Star+Wars&y=1977`. A horizontal orange line with vertical tick marks at the end of each segment is positioned below the URL. The segments are: `http://`, `www.`, `omdbapi.com/`, and `?t=Star+Wars&y=1977`. Below these segments are labels: `http://` is labeled 'HTTP Protocol', `www.` is labeled 'Domain Name', `omdbapi.com/` is labeled 'Subdomain' and 'Path', and `?t=Star+Wars&y=1977` is labeled 'Query Parameters'.

HTTP Protocol

- GET
- POST
- PUT
- DELETE

Domain Name

Subdomain

Path

Query Parameters



# OMDB API Query Parameters

<http://www.omdbapi.com/?t=Star+Wars&y=1977>



# OMDB API Query Parameters

<http://www.omdbapi.com/?t=Star+Wars&y=1977>



Param 1      Param 2

?



# OMDB API Query Parameters

<http://www.omdbapi.com/?t=Star+Wars&y=1977>

?key





# OMDB API Query Parameters

<http://www.omdbapi.com/?t=Star+Wars&y=1977>

? key=



# OMDB API Query Parameters

<http://www.omdbapi.com/?t=Star+Wars&y=1977>



? key=value



# OMDB API Query Parameters

<http://www.omdbapi.com/?t=Star+Wars&y=1977>

?key=value&key=value



# Connecting to a Web API

---



# On Loading Event System

## **onload**

- When the web api call succeeds

## **onabort**

- When xhr.abort() is called

## **onerror**

- When the web api call fails

## **onloadstart**

- As soon as the data returns

## **onprogress**

- As the data is received

## **onloadend**

- Always called no matter the outcome



# Old XHR Event System

## **onreadystatechange**

- Replaced by on loading events
- Ready states
  - UNSET
  - OPENED
  - HEADERS\_RECIEVED
  - LOADING
  - DONE



# Making Asynchronous Web Calls

---



# Handling the HTTP Response

---





# Parsing the JSON and Displaying the data

---



# XHR Data Types

Text

arraybuffer

blob

document

JSON

DOM String

Streams (Microsoft IE only. Deprecated)



# Summary



## **XMLHttpRequest**

- Instantiate
- Initialize
- Send

## **useEffect**

- To make Web API requests once

## **useState**

- To share Web API data

## **Callback functions**

- To process the Web API responses
- onload, onerror, etc.



# Building Mobile Apps with React Native

---

Thank you!



**Richard Monson-Haefel**

SR. SOFTWARE ENGINEER

[www.monsonhaefel.com](http://www.monsonhaefel.com)

