# Homework 8: Ajax, JSON, React, RWD and Node.js

(AJAX/JSON/HTML5/React-Bootstrap/React/Node.js/Cloud Exercise)

## 1. Objectives

- Get familiar with the AJAX and JSON technologies.
- Use a combination of HTML5, React-Bootstrap, and React.js on client side.
- Learn to use JavaScript / Node.js on server side.
- Get familiar with React-Bootstrap to enhance the user experience using responsive design.
- Get hands-on experience of Amazon Web Services/Google Cloud App Engine/Microsoft Azure.
- Learn to use APIs such as Guardian News API, New York Times API, Comment - box API, Bing Autosuggest API.

## 2. Background

### 2.1 AJAX and JSON

AJAX (Asynchronous JavaScript + XML) incorporates several technologies:

- Standards-based presentation using XHTML and CSS;
- Result display and interaction using the Document Object Model (DOM);
- Data interchange and manipulation using XML and JSON;
- Asynchronous data retrieval using XMLHttpRequest;
- JavaScript binding everything together.

See the class slides at https://csci571.com/slides/ajax.pdf

JSON, short for JavaScript Object Notation, is a lightweight data interchange format. Its main application is in AJAX web application programming, where it serves as an alternative to the use of the XML format for data exchange between client and server. See the class slides at:

https://csci571.com/slides/JSON1.pdf

### 2.2 React-Bootstrap

**React-Bootstrap** replaces the **Bootstrap** JavaScript. Each component has been built from scratch as a true **React** component, without unneeded dependencies like jQuery. As one of the oldest **React** libraries, **React-Bootstrap** has evolved and grown alongside **React**, making it an excellent choice as your UI foundation.

https://csci571.com/slides/React.pdf

### 2.3 Amazon Web Services (AWS)

AWS is Amazon’s implementation of cloud computing. Included in AWS is Amazon Elastic Compute Cloud (EC2), which delivers scalable, pay-as-you-go compute capacity in the cloud, and AWS Elastic Beanstalk, an even easier way to quickly deploy and manage applications in the AWS cloud. You simply upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. Elastic Beanstalk is built using familiar software stacks such as the Apache HTTP Server, PHP, and Python, Passenger for Ruby, IIS for .NET, and Apache Tomcat for Java.

To learn more about AWS support for Node.js visit this page:

https://aws.amazon.com/getting-started/projects/deploy-nodejs-web-app/

### 2.4 Google App Engine (GAE)

Google App Engine applications are easy to create, easy to maintain, and easy to scale as your traffic and data storage needs change. With App Engine, there are no servers to maintain. You simply upload your application and it's ready to go. App Engine applications automatically scale based on incoming traffic. Load balancing, micro services, authorization, SQL and noSQL databases, memcache, traffic splitting, logging, search, versioning, roll out and roll backs, and security scanning are all supported natively and are highly customizable.

To learn more about GAE support for Node.js visit this page:

https://cloud.google.com/appengine/docs/flexible/Node.js/

### 2.5 Microsoft Azure

Microsoft Azure is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through a global network of Microsoft-managed data centers. It provides software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems.

To learn more about Azure support for Node.js visit this page:

https://docs.microsoft.com/en-us/javascript/azure/?view=azure-node-latest

### 2.6 ReactJs

ReactJS is an open-source JavaScript library, developed by Facebook, which is used for building user interfaces specifically for single page applications. It differs from Angular in that React is a **view layer library**, **not** a **framework** like Angular. However, you can't use React alone to build a fully-functional web app. ReactJs uses 1-way data binding and a virtual DOM and the rendering is done on the server-side/client-side ( https://www.clariontech.com/blog/server-side-rendering-vs.-client-side-rendering ).

For this homework, **ReactJs and ReactBootstrap have to be used.**

To learn more about them, visit these links:

https://reactjs.org/
https://react-bootstrap.github.io/

## 2.7 Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js package ecosystem, **npm**, is the largest ecosystem of open source libraries in the world.

To learn more about Node.js, visit:

https://Node.js.org/en/

Also, **Express.js** is strongly recommended. Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is in fact the standard web server framework for Node.js.

To learn more about Express.js, visit:

http://expressjs.com/

**Note**: In this document when you see GAE/AWS/Azure it means that you can either use Google App Engine, Amazon Web Services or Microsoft Azure Services.

The only way to implement the client side: Use React ONLY

**All APIs calls should be done through your Node.JS server, except call to the autosuggest API.**

**( Making any other API calls from anywhere other than your backend will result in a 4 point penalty. )**

## 3. High Level Description

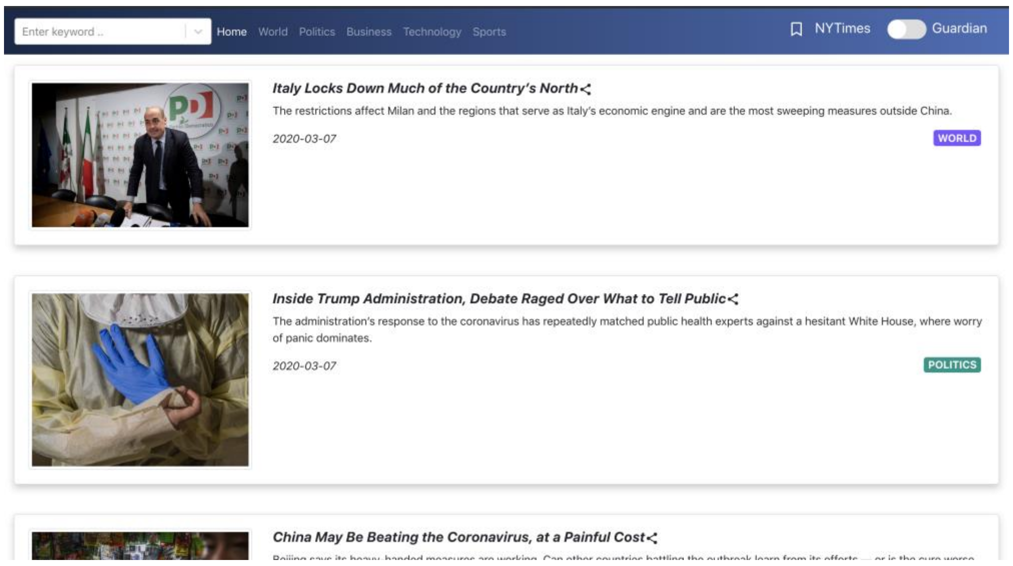
In this exercise you will create a webpage to display top headlines for both New York Times and Guardian News. Based on the selection by the user, the top-headlines should be for any of the following domains – **world, politics, business, technology and sports**. Each such article can be shared by the user on Facebook, Twitter or via Email.

The user can open up any of the article and have the option to share it on either Facebook or Twitter or via Email. For each article, after expanding, the user can even add comments on the articles if they like.

The user can also search for any keyword to read articles about any topic of his/her choice with autosuggestions available for his/her search query.

Your webpage should also support adding articles to and removing articles from the Bookmark tab. All the implementation details and requirements will be explained in the following sections.

When a user initially opens your webpage, your page should look like **Figure 1**.



*Italy Locks Down Much of the Country's North*

The restrictions affect Milan and the regions that serve as Italy's economic engine and are the most sweeping measures outside China.

*2020-03-07*

*Inside Trump Administration, Debate Raged Over What to Tell Public*

The administration's response to the coronavirus has repeatedly matched public health experts against a hesitant White House, where worry of panic dominates.

*2020-03-07*

*China May Be Beating the Coronavirus, at a Painful Cost*

Beijing says its heavy-handed measures are working. Can other countries battling the outbreak learn from its efforts — or is the cure worse

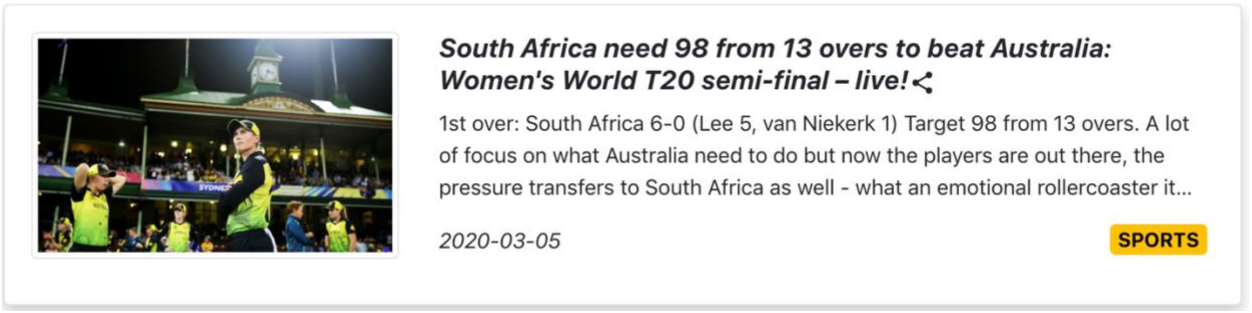
**Figure 1:** Initial Website

### 3.1 Tabs

There are 6 different tabs/sections that your website needs to have namely – **Home, World, Politics, Business, Technology, Sports.** Each of these sections/tabs display the top articles from any domain (for the Home tab) or the top articles belonging to the selected section/tab. The articles from each domain change based on the tab selected.

#### 3.1.1 Section based Articles

**Figure 2** and **Figure 3** show samples from both Guardian news and NY Times.

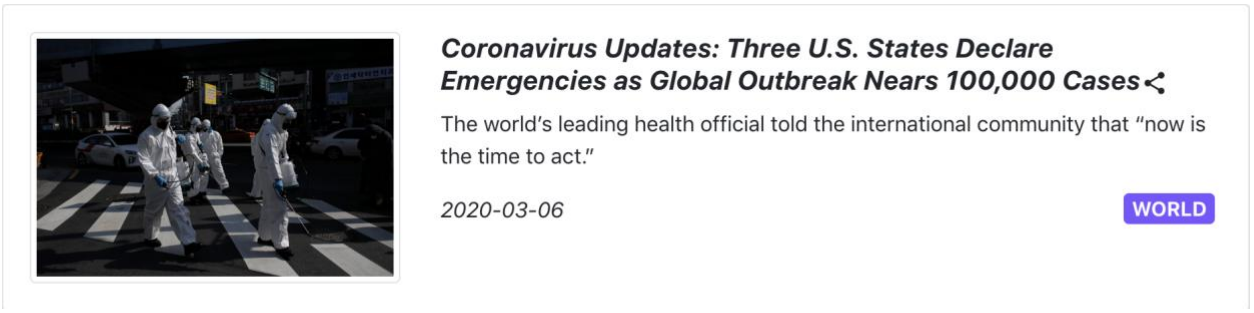


***South Africa need 98 from 13 overs to beat Australia: Women's World T20 semi-final – live!***

1st over: South Africa 6-0 (Lee 5, van Niekerk 1) Target 98 from 13 overs. A lot of focus on what Australia need to do but now the players are out there, the pressure transfers to South Africa as well - what an emotional rollercoaster it...

*2020-03-05*

**Figure 2:** Guardian section-based article



***Coronavirus Updates: Three U.S. States Declare Emergencies as Global Outbreak Nears 100,000 Cases***

The world's leading health official told the international community that "now is the time to act."

*2020-03-06*

**Figure 3:** NY Times section-based article

**Guardian URL for Home tab:**

https://content.guardianapis.com/search?api-key=[YOUR_API_KEY]&section=(sport|business|technology|politics)&show-blocks=all

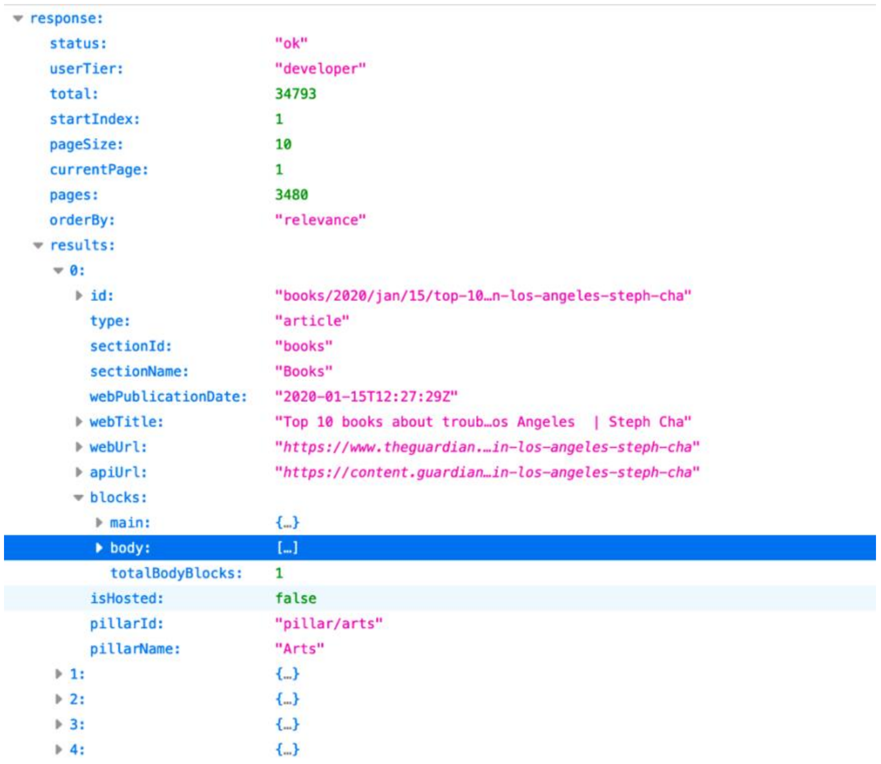
**Guardian URL for section tabs:**

https://content.guardianapis.com/[SECTION_NAME]?api-key=[YOUR_API_KEY]&show-blocks=all

The URL generation requires the following parameters depending on whether you're on the Home tab or any other:

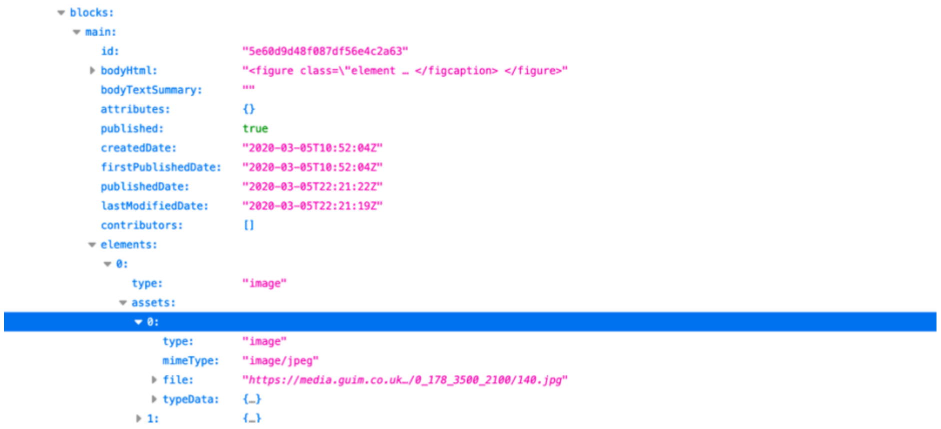
- The api-key parameter requires your Guardian API key
- For Home tab, you need to hit the '/search' endpoint of the Guardian API and provide section as a query parameter. The value MUST be exactly as shown in the above URL
- For the other tabs, based on the selection, the '/[SECTION_NAME]' endpoint of the Guardian API is to be accessed. For example, for the world tab, the URL would be - https://content.guardianapis.com/world?api-key=[YOUR_API_KEY]&show-blocks=all
- The show-blocks parameter is set to all in every tab

The JSON structure for both types of call is as shown in **Figure 4** below. The mapping between the article card and the JSON object is listed below.



```
▾ response:
    status:                 "ok"
    userTier:               "developer"
    total:                  34793
    startIndex:             1
    pageSize:               10
    currentPage:            1
    pages:                  3480
    orderBy:                "relevance"
  ▾ results:
    ▾ 0:
      ▸ id:                 "books/2020/jan/15/top-10…n-los-angeles-steph-cha"
        type:               "article"
        sectionId:          "books"
        sectionName:        "Books"
        webPublicationDate: "2020-01-15T12:27:29Z"
      ▸ webTitle:           "Top 10 books about troub…os Angeles  | Steph Cha"
      ▸ webUrl:             "https://www.theguardian.…in-los-angeles-steph-cha"
      ▸ apiUrl:             "https://content.guardian…in-los-angeles-steph-cha"
      ▾ blocks:
        ▸ main:             {…}
        ▸ body:             […]
          totalBodyBlocks:  1
        isHosted:           false
        pillarId:           "pillar/arts"
        pillarName:         "Arts"
    ▸ 1:                    {…}
    ▸ 2:                    {…}
    ▸ 3:                    {…}
    ▸ 4:                    {…}
```

**Figure 4:** Guardian News search result sample



```
▾ blocks:
  ▾ main:
      id:                 "5e60d9d48f087df56e4c2a63"
    ▸ bodyHtml:           "<figure class=\"element … </figcaption> </figure>"
      bodyTextSummary:    ""
      attributes:         {}
      published:          true
      createdDate:        "2020-03-05T10:52:04Z"
      firstPublishedDate: "2020-03-05T10:52:04Z"
      publishedDate:      "2020-03-05T22:21:22Z"
      lastModifiedDate:   "2020-03-05T22:21:19Z"
      contributors:       []
    ▾ elements:
      ▾ 0:
          type:           "image"
        ▾ assets:
          ▾ 0:
              type:       "image"
              mimeType:   "image/jpeg"
            ▸ file:       "https://media.guim.co.uk…/0_178_3500_2100/140.jpg"
            ▸ typeData:   {…}
          ▸ 1:            {…}
```

**Figure 5**: Guardian News search result sample for image

| **Domain Specific Article Card** | **Corresponding JSON key** |
|---|---|
| Title | Value in the hierarchy *“response”* -> *“results[index]”* -> *“webTitle”* |
| Image | Value of the hierarchy *“response”* -> *“results[index]”* ->*”blocks”*->*”main”*->*”elements”* ->*”0”*->*”assests”*->*last indexed assest* -> *“file”* (Refer Figure 5) |

| | |
|---|---|
| Section | Value in the hierarchy *“response”* -> *“results[index]”* -> *“sectionId”* |
| Date (yyyy/mm/dd) | Value in the hierarchy *“response”* -> *“results[index]”* -> *”webPublicationDate”* converted to the format as shown in Figure 2 |
| Description | Value in the hierarchy *“response”* -> *“results[index]”* -> *”blocks”*-> *”body”*-> *”0”* -> *bodyTextSummary”* |

**NOTE 1:** The above data needs to be used for each article. “index” represents article number. The article numbers begin from 0.

**NOTE 2:** Only the last image URL in assets key is to be used. If no such URLs are found, a default image for Guardian News is to be displayed. Refer section 5.2 for the default image for Guardian News which would be used anytime a Guardian article does not have an image mentioned in the specs (this ensures that we are always fetching the larger image).

**NOTE 3:** The section coloring is different for different sections. Refer section 5.1 for a mapping between section and color.

**NOTE 4:** In the article description, no words should be cut in the middle and there MUST be the ellipsis whenever needed (limit to 3 lines).

**NY Times URL for Home tab:**

https://api.nytimes.com/svc/topstories/v2/home.json?api-key=[YOUR_API_KEY]

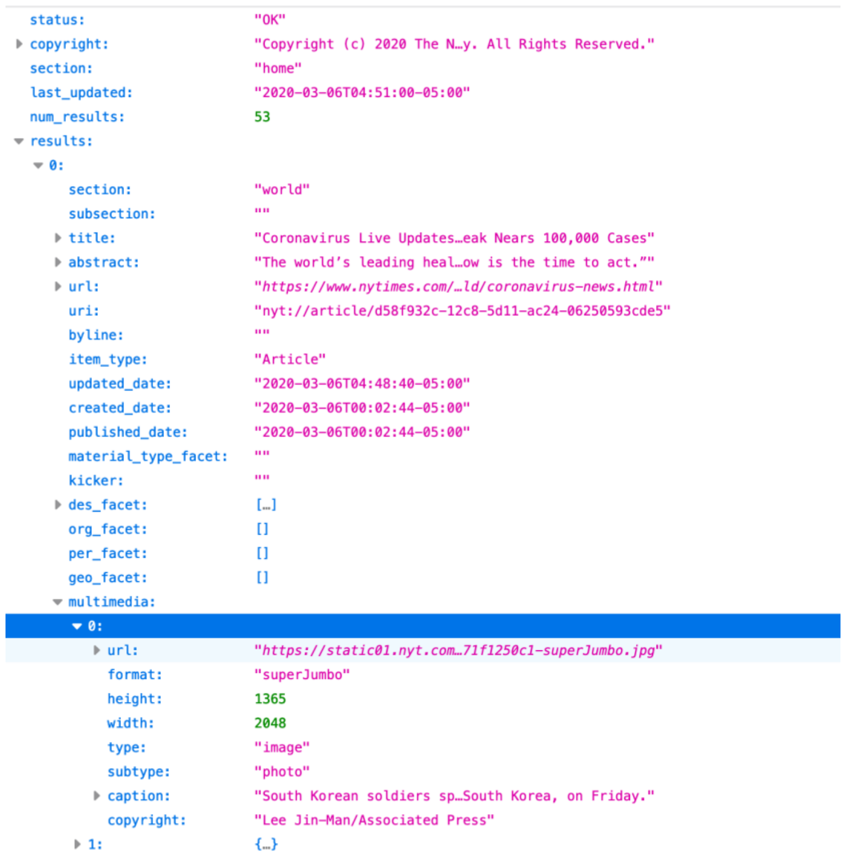
**NY Times URL for other tabs:**

https://api.nytimes.com/svc/topstories/v2/[SECTION_NAME].json?api-key=[YOUR_API_KEY]

The URL generation requires the following parameters depending on whether you’re on the Home tab or any other:

- The api-key parameter requires your NY Times API key
- For Home tab, you need to hit the ‘/home.json’ endpoint of the NY Times API and provide API key as query parameter.
- For the other tabs, based on the selection, the ‘/[SECTION_NAME].json’ endpoint of the NY Times API is to be accessed. For example, for the world tab, the URL would be – https://api.nytimes.com/svc/topstories/v2/world.json?api-key=[YOUR_API_KEY]
  The possible values for the SECTION_NAME are world, politics, business, technology, sports for the 5 different tabs we have.

The JSON structure for both types of calls is shown in **Figure 6** below. The mapping between the article card and the JSON object is listed below.



```
status:                 "OK"
▸ copyright:            "Copyright (c) 2020 The N…y. All Rights Reserved."
section:                "home"
last_updated:           "2020-03-06T04:51:00-05:00"
num_results:            53
▾ results:
  ▾ 0:
      section:              "world"
      subsection:           ""
    ▸ title:                "Coronavirus Live Updates…eak Nears 100,000 Cases"
    ▸ abstract:             "The world's leading heal…ow is the time to act.""
    ▸ url:                  "https://www.nytimes.com/…ld/coronavirus-news.html"
      uri:                  "nyt://article/d58f932c-12c8-5d11-ac24-06250593cde5"
      byline:               ""
      item_type:            "Article"
      updated_date:         "2020-03-06T04:48:40-05:00"
      created_date:         "2020-03-06T00:02:44-05:00"
      published_date:       "2020-03-06T00:02:44-05:00"
      material_type_facet:  ""
      kicker:               ""
    ▸ des_facet:            […]
      org_facet:            []
      per_facet:            []
      geo_facet:            []
    ▾ multimedia:
      ▾ 0:
        ▸ url:              "https://static01.nyt.com…71f1250c1-superJumbo.jpg"
          format:           "superJumbo"
          height:           1365
          width:            2048
          type:             "image"
          subtype:          "photo"
        ▸ caption:          "South Korean soldiers sp…South Korea, on Friday."
          copyright:        "Lee Jin-Man/Associated Press"
      ▸ 1:                  {…}
```

**Figure 6:** JSON Response for NY Times section-based search

| **Domain Specific Article Card** | **Corresponding JSON key** |
|---|---|
| Title | Value in the hierarchy *"results"* -> *"title"* |
| Image* | Value of the hierarchy *"results"* ->*"multimedia"*->*"url"* |
| Section | Value in the hierarchy *"results"* -> *"section"* |
| Date (yyyy/mm/dd) | Value in the hierarchy *"results"* ->*"published_date"* converted to the format as shown in Figure 3 |
| Description | Value in the hierarchy *"results"* ->*"abstract"* |

* - Pick the first multimedia image with width **atleast 2000.**

**NOTE 1:** The above data needs to be used for each article. The hierarchy would therefore go like "result"->[ARTICLE_NUMBER] followed by the ones given in the table above. The article numbers begin from 0.

**NOTE 2:** Only the first image URL with width >= 2000 is to be used. If no such URLs are found, a default image for NY Times News is to be displayed. Refer section 5.2 for the default image for NY Times News which would be used anytime a NY Times article does not have an image mentioned in the specs.

**NOTE 3:** The section coloring is different for different sections. Refer section 5.1 for a mapping between section and color.

**NOTE 4:** More data than mentioned in the table would need to be stored for future use mentioned hereafter in the document.

### 3.1.2 Sharing Modal

The share button is clickable and opens up a modal as shown in **Figure 7**.

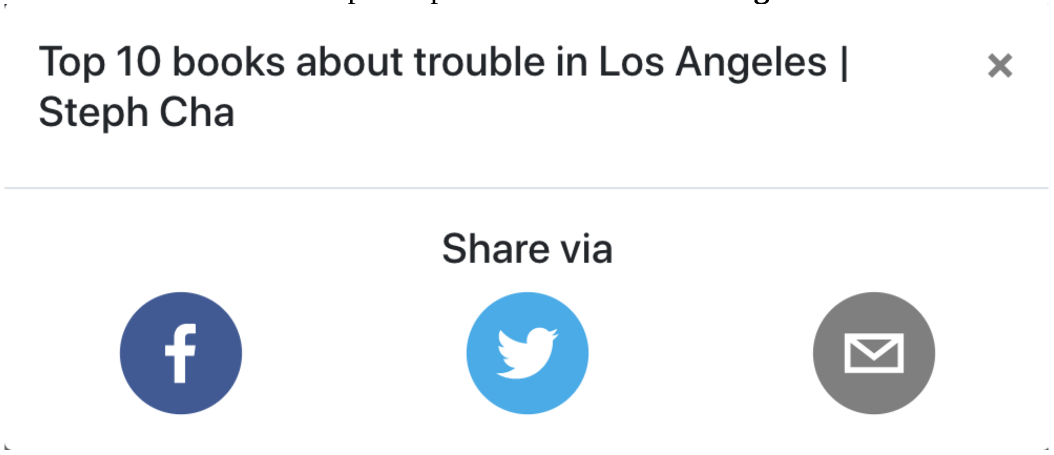


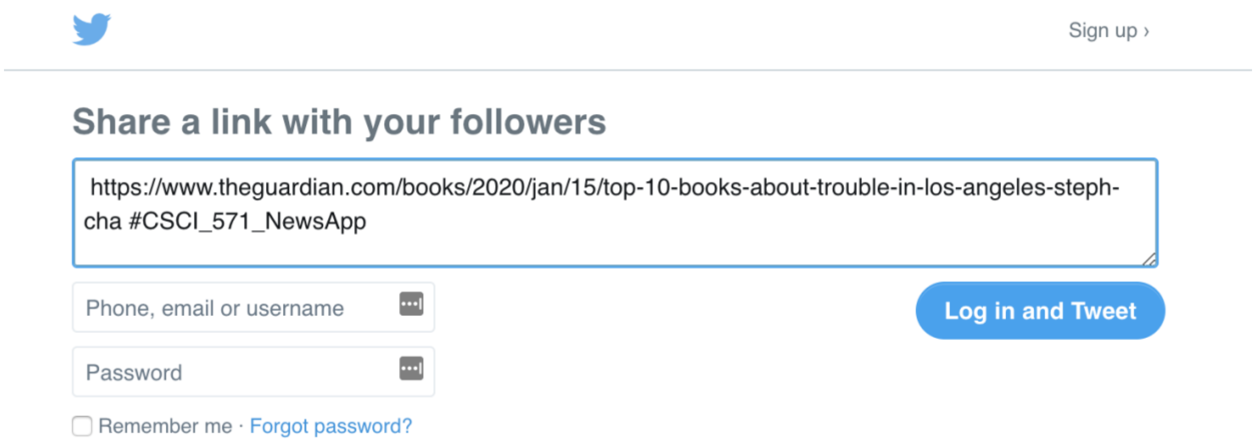
**Figure 7:** Modal display after clicking share button on article

The 3 share options for Facebook, Twitter and Email can be implemented by using **react-share.** The following GitHub link for react-share is all that you will need to setup the sharing options.

https://github.com/nygardk/react-share

The title for the sharing modal comes from the "Title" section in the table(s) for Guardian and NY Times above.

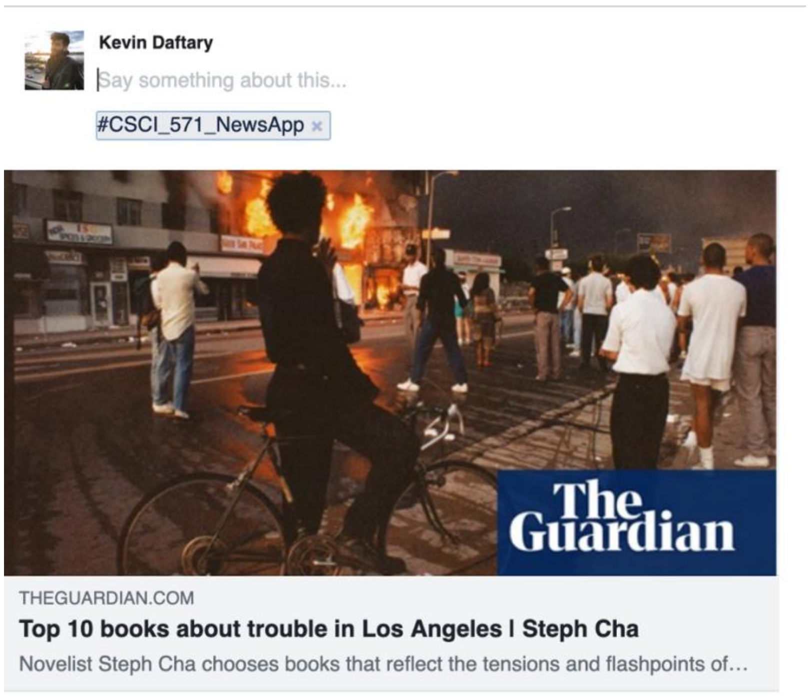
The following **Figures 8**, **Figure 9** and **Figure 10** show the result of clicking each icon in the sharing modal.


Sign up ›

Share a link with your followers

https://www.theguardian.com/books/2020/jan/15/top-10-books-about-trouble-in-los-angeles-steph-cha #CSCI_571_NewsApp

**Figure 8:** Twitter Share


Kevin Daftary

Say something about this...

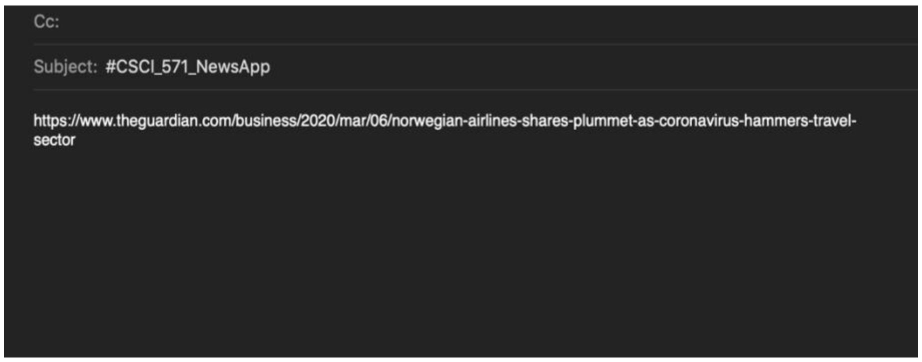
#CSCI_571_NewsApp

THEGUARDIAN.COM

Top 10 books about trouble in Los Angeles | Steph Cha

Novelist Steph Cha chooses books that reflect the tensions and flashpoints of…

**Figure 9:** Facebook Share


Cc:

Subject: #CSCI_571_NewsApp

https://www.theguardian.com/business/2020/mar/06/norwegian-airlines-shares-plummet-as-coronavirus-hammers-travel-sector

**Figure 10:** Email Share

**<u>For sharing from detailed article page and search result page:</u>**
The URL to be shared on each social media can be found at the ***"webUrl"*** key in ***"response"->"results"*** for Guardian response.

The URL to be shared on each social media can be found at the ***"url"*** key in ***"response"->"docs"*** for NY Times response.

**<u>For sharing from section based article page:</u>**
The URL to be shared on each social media can be found at the ***"webUrl"*** key in ***"response"->"results"*** for Guardian response.

The URL to be shared on each social media can be found at the ***"url"*** key in ***"results"->[index]->"url"*** for NY Times response.

Note the presence of the hashtag CSCI_571_NewsApp as well that needs to be replicated.

The number of section-based articles are to be restricted to **10 only. DO NOT** display anything if certain data is missing in the section-based article card except for the image (use default image provided otherwise).

When clicking on any article to reach the detailed article page, there is an intermediate screen that is displayed while the results are fetched as shown in **Figure 11**.

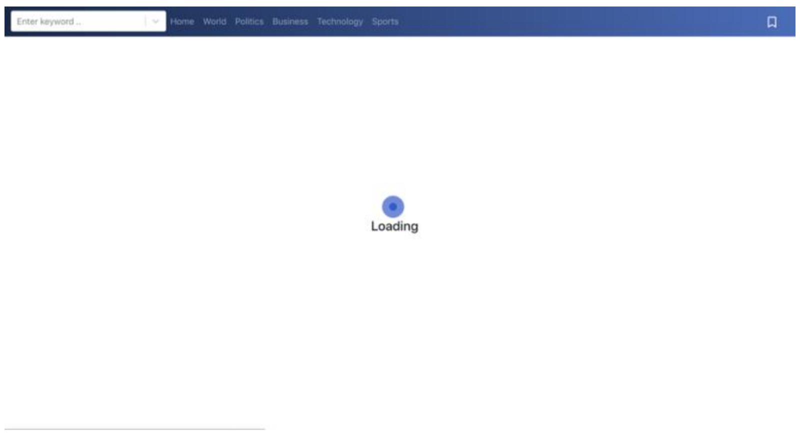
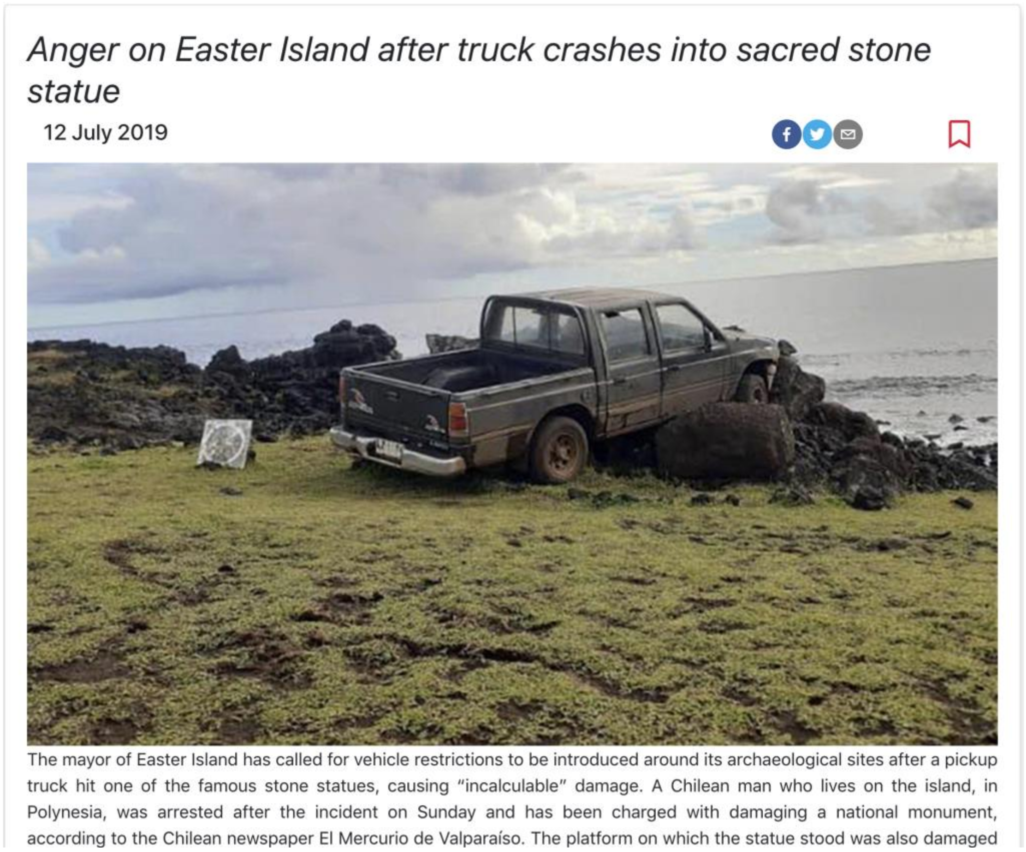


**Figure 11:** Loading screen

Please refer to the reference video for a clearer understanding of this.

### 3.2 Detailed Article Page
All articles are clickable and open a detailed view of the article as shown in **Figure 12** and **Figure 13**.

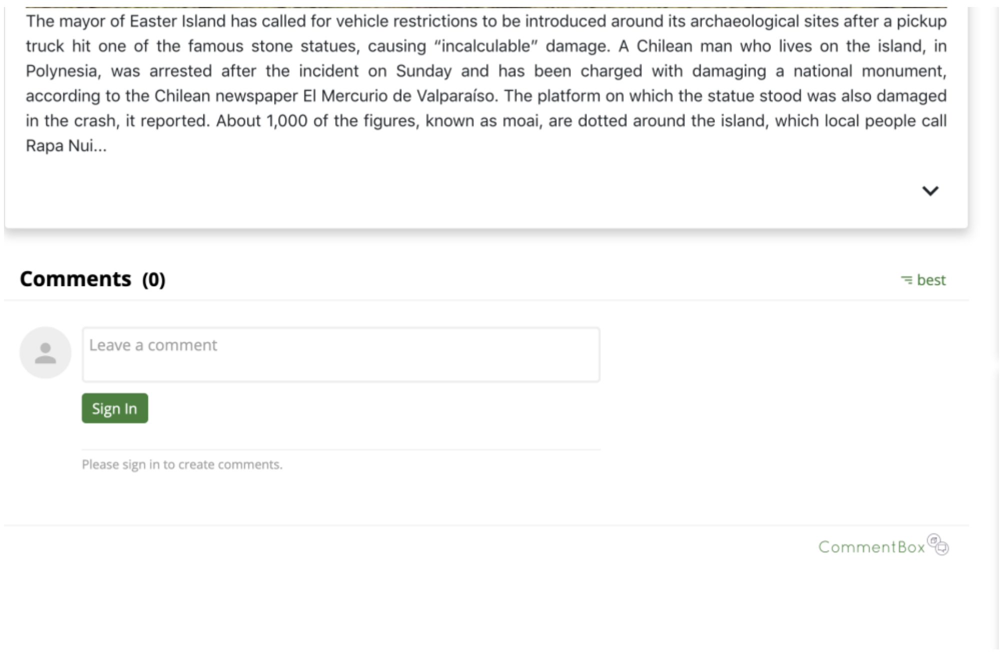


*Anger on Easter Island after truck crashes into sacred stone statue*

12 July 2019

The mayor of Easter Island has called for vehicle restrictions to be introduced around its archaeological sites after a pickup truck hit one of the famous stone statues, causing "incalculable" damage. A Chilean man who lives on the island, in Polynesia, was arrested after the incident on Sunday and has been charged with damaging a national monument, according to the Chilean newspaper El Mercurio de Valparaíso. The platform on which the statue stood was also damaged

**Figure 12:** Detailed Article View



The mayor of Easter Island has called for vehicle restrictions to be introduced around its archaeological sites after a pickup truck hit one of the famous stone statues, causing "incalculable" damage. A Chilean man who lives on the island, in Polynesia, was arrested after the incident on Sunday and has been charged with damaging a national monument, according to the Chilean newspaper El Mercurio de Valparaíso. The platform on which the statue stood was also damaged in the crash, it reported. About 1,000 of the figures, known as moai, are dotted around the island, which local people call Rapa Nui...

**Figure 13:** Detailed Article View (continued)

For the article, you need to use the *"id"* key in the JSON object returned by Guardian API (shown in **Figure 4**). For NY Times, you need to use *"url"* from the JSON object returned by the NY Times API (shown in **Figure 6** above). Using the corresponding data, a URL needs to be generated for the purpose of the detailed article.

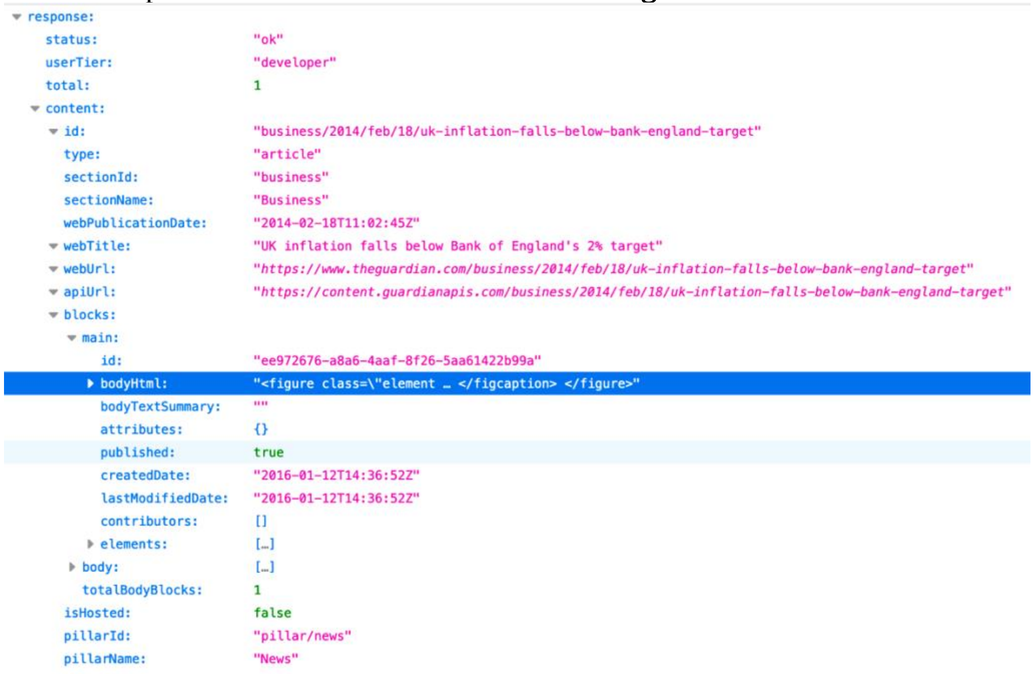
**Guardian News URL:**

https://content.guardianapis.com/[ARTICLE_ID]?api-key=[YOUR_API_KEY]&show-blocks=all

The above URL has the following parameters:

- The first parameter is the article ID as the guardian API endpoint. This value is retrieved from the *"response"*->*"results"*->*"id"* hierarchy of the Guardian API response shown in Figure 4
- The second parameter is the API key for Guardian News API
- The third parameter is a fixed key=value pair of show-blocks=all

A sample JSON response for the above call is shown in **Figure 14**.



```
▾ response:
    status:                 "ok"
    userTier:               "developer"
    total:                  1
  ▾ content:
    ▾ id:                   "business/2014/feb/18/uk-inflation-falls-below-bank-england-target"
      type:                 "article"
      sectionId:            "business"
      sectionName:          "Business"
      webPublicationDate:   "2014-02-18T11:02:45Z"
    ▾ webTitle:             "UK inflation falls below Bank of England's 2% target"
    ▾ webUrl:               "https://www.theguardian.com/business/2014/feb/18/uk-inflation-falls-below-bank-england-target"
    ▾ apiUrl:               "https://content.guardianapis.com/business/2014/feb/18/uk-inflation-falls-below-bank-england-target"
    ▾ blocks:
      ▾ main:
          id:               "ee972676-a8a6-4aaf-8f26-5aa61422b99a"
        ▸ bodyHtml:         "<figure class=\"element … </figcaption> </figure>"
          bodyTextSummary:  ""
          attributes:       {}
          published:        true
          createdDate:      "2016-01-12T14:36:52Z"
          lastModifiedDate: "2016-01-12T14:36:52Z"
          contributors:     []
        ▸ elements:         […]
      ▸ body:               […]
        totalBodyBlocks:    1
      isHosted:             false
      pillarId:             "pillar/news"
      pillarName:           "News"
```

**Figure 14:** Sample JSON for article expansion search

The mapping between the JSON data and the article is as follows.

| Detailed Article Card | Corresponding JSON key |
|---|---|
| Title | Value in the hierarchy *"response"* -> *"content"* -> *"webTitle"* |
| Image | Value of the hierarchy *"response"* -> *"content"* ->*"blocks"*->*"main"*->*"elements"* |

| | |
|---|---|
| | ->*"0"*->*"assests"*->last indexed assest -> *"file"* (Refer Figure 5) |
| Date | Value in the hierarchy *"response"* -> *"content"* ->*"webPublicationDate"* converted to the format as shown in Figure 12 |
| Description | Value in the hierarchy *"response"* -> *"content"* ->*"blocks"*->*"body"*->*"0"* ->*bodyTextSummary"* |

The article should also have the 3 sharing buttons as described previously with the same functionality and the hover functionality as shown in **Figure 15**.

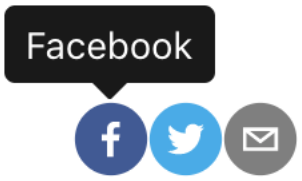


**Figure 15:** Share buttons with tooltip

**NOTE:** In the article description, no words should be cut in the middle and there MUST be the ellipsis whenever needed. The description should initially be restricted to 6 lines.

Each article has a bookmark button as shown in **Figure 16** below to add/remove article from favorites.

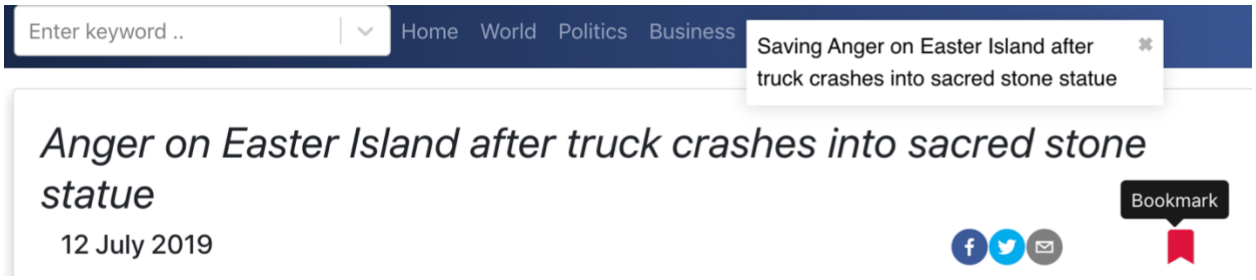

Enter keyword ..
Home
World
Politics
Business
Saving Anger on Easter Island after truck crashes into sacred stone statue
Anger on Easter Island after truck crashes into sacred stone statue
12 July 2019
Bookmark

**Figure 16:** Bookmark with toast message and tooltip

On clicking the bookmark button, the article should be added to favorites and a toast message should be displayed for the same. Here are a couple of links to help with the toast:

https://github.com/fkhadra/react-toastify

https://github.com/fkhadra/react-toastify/issues/374

Usage of local storage to save and persist articles in the bookmark tab is recommended. Here's a useful link for the local storage feature:

https://stackoverflow.com/questions/2010892/storing-objects-in-html5-localstorage

Each article has an arrow at the bottom right corner as shown in **Figure 13** above. The purpose of the arrow is to display the entire article description. This article description can be found in the *"response"->"content"->"body"->"0"->"bodyTextSummary"*

The toggling must be smooth and should follow the demonstration in the reference video. Here's a link that might be useful:

https://stackoverflow.com/questions/44375093/handling-scroll-animation-in-react

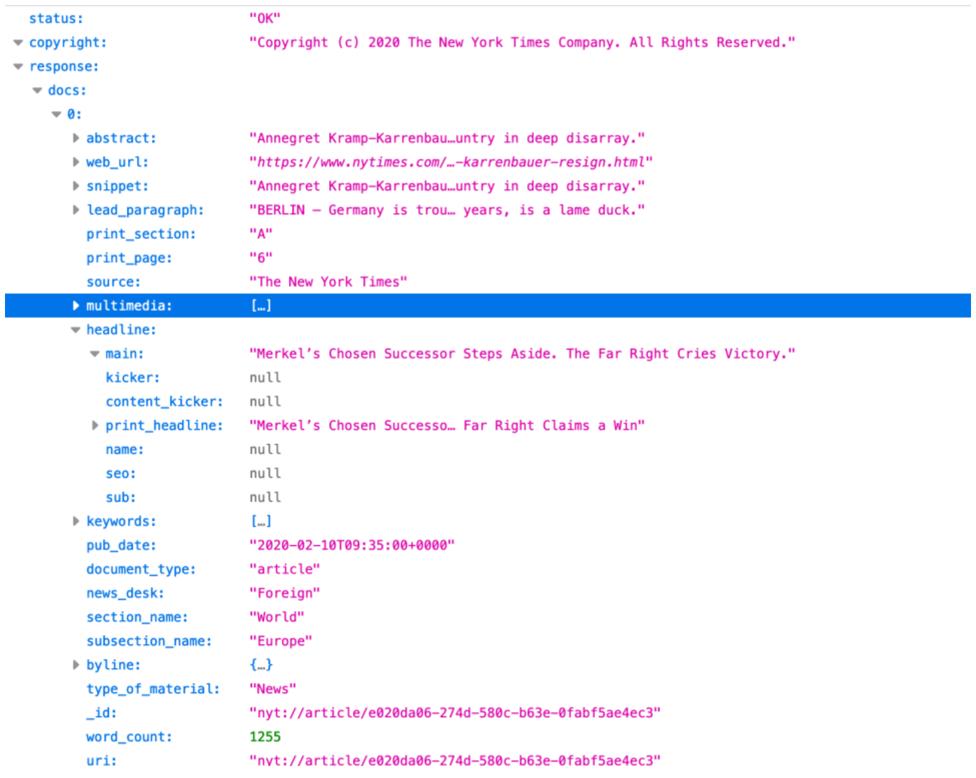
**NY Times URL:**

https://api.nytimes.com/svc/search/v2/articlesearch.json?fq=web_url:("[ARTICLE_WEB_URL]") &api-key=[API_KEY]

The above URL has the following parameters:

- The first parameter is the article web url. This value is retrieved from the *"response"->"docs"->"url"* hierarchy of the NY Times API response shown in **Figure 6**
- The second parameter is the API key for NY Times News API

A sample JSON response for the above call is shown in **Figure 17**.



```
status:                   "OK"
▼ copyright:              "Copyright (c) 2020 The New York Times Company. All Rights Reserved."
▼ response:
  ▼ docs:
    ▼ 0:
      ▶ abstract:         "Annegret Kramp-Karrenbau…untry in deep disarray."
      ▶ web_url:          "https://www.nytimes.com/…-karrenbauer-resign.html"
      ▶ snippet:          "Annegret Kramp-Karrenbau…untry in deep disarray."
      ▶ lead_paragraph:   "BERLIN — Germany is trou… years, is a lame duck."
        print_section:    "A"
        print_page:       "6"
        source:           "The New York Times"
      ▶ multimedia:       […]
      ▼ headline:
        ▼ main:           "Merkel's Chosen Successor Steps Aside. The Far Right Cries Victory."
          kicker:         null
          content_kicker: null
        ▶ print_headline: "Merkel's Chosen Successo… Far Right Claims a Win"
          name:           null
          seo:            null
          sub:            null
      ▶ keywords:         […]
        pub_date:         "2020-02-10T09:35:00+0000"
        document_type:    "article"
        news_desk:        "Foreign"
        section_name:     "World"
        subsection_name:  "Europe"
      ▶ byline:           {…}
        type_of_material: "News"
        _id:              "nyt://article/e020da06-274d-580c-b63e-0fabf5ae4ec3"
        word_count:       1255
        uri:              "nyt://article/e020da06-274d-580c-b63e-0fabf5ae4ec3"
```

**Figure 17:** Sample JSON response for detailed article search

The mapping between the JSON data and the article is as follows

| Detailed Article Card | Corresponding JSON key |
|---|---|
| Title | Value in the hierarchy *“response”* -> *“docs”* -> *“headline”*->*”main”* |
| Image* | Value of the hierarchy *“response”* -> *“docs”* ->*”multimedia”*->*”url”* |
| Date | Value in the hierarchy *“response”* -> *“docs”* ->*”pub_date”* converted to the format as shown in Figure 12 |
| Description | Value in the hierarchy *“response”* -> *“docs”* ->*”abstract”* |

* - Pick the first multimedia image with width **atleast 2000.**

### 3.2.1 Comment-Box:

Each article has a comment-box (shown in **Figure 13** above) that is **UNIQUE** to that article. To create the comment-box, you are required to use the CommentBox.io API.

https://commentbox.io/docs

**Hint:** For the purpose of keeping unique comment box for each article, use the article ID you used to search for the detailed article description. The comments should persist anytime a user closes the website and reopens it.

### 3.3 AutoSuggest on Search

The user can search for any keyword to look up articles related to that keyword from either Guardian News or NY Times depending on his/her selection. **Figure 18** shows an example of the autosuggesting functionality to be implemented for search.

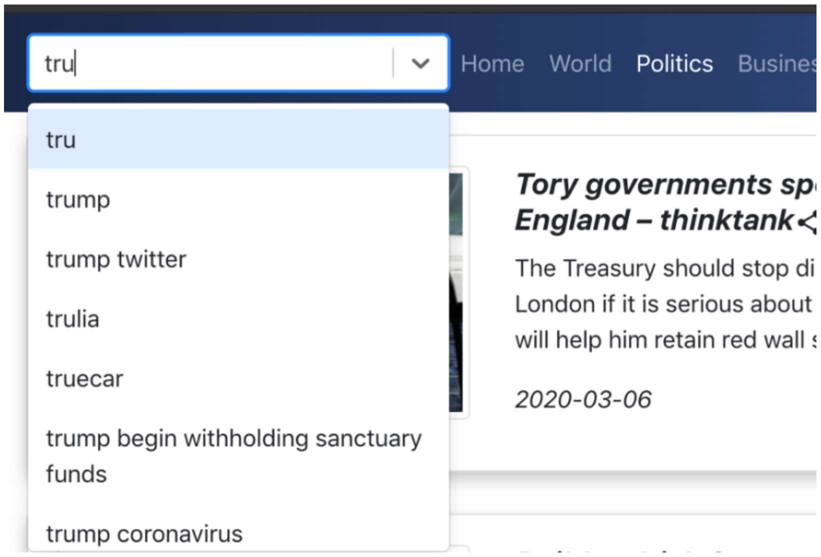

tru
Home
World
Politics
Busines
tru
trump
trump twitter
trulia
truecar
trump begin withholding sanctuary funds
trump coronavirus
Tory governments sp
England – thinktank
The Treasury should stop di
London if it is serious about
will help him retain red wall s
2020-03-06

**Figure 18:** Autosuggest example

For the purpose of the autosuggest, the *Bing Autosuggest API* is recommended.

https://azure.microsoft.com/en-us/services/cognitive-services/autosuggest/

**NOTE: The Autosuggest API call MUST be made from JavaScript and NOT your backend. This is to reduce latency and improve performance.**

Based on what the user types, use that query keyword and generate the URL to fetch suggestions from the API as follows:

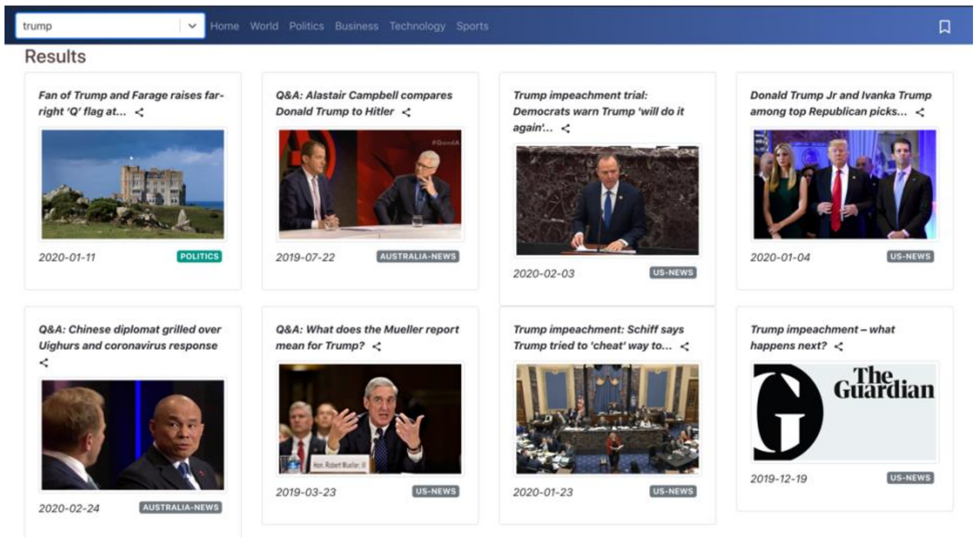
https://api.cognitive.microsoft.com/bing/v7.0/suggestions?q=[QUERY_KEYWORD]

Apart from the keyword, the API key needs to be passed in the request header instead of a request query parameter. The request header contains the API key in a key:value format with the 'key' being **'Ocp-Apim-Subscription-Key'** and the value will be your API key. Refer to section 4.3 for steps to obtain API key. Here is a link with a sample usage to help you with the autocomplete search.

https://gist.github.com/arnaudrenaud/70d21dcbc20acf2d287143880a18a63b

## 3.4 Displaying Search Results

After the user searches for a keyword, articles related to that keyword are to be displayed as shown in **Figure 19**.



Results

Fan of Trump and Farage raises far-right 'Q' flag at... 2020-01-11 POLITICS

Q&A: Alastair Campbell compares Donald Trump to Hitler 2019-07-22 AUSTRALIA-NEWS

Trump impeachment trial: Democrats warn Trump 'will do it again'... 2020-02-03 US-NEWS

Donald Trump Jr and Ivanka Trump among top Republican picks... 2020-01-04 US-NEWS

Q&A: Chinese diplomat grilled over Uighurs and coronavirus response 2020-02-24 AUSTRALIA-NEWS

Q&A: What does the Mueller report mean for Trump? 2019-03-23 US-NEWS

Trump impeachment: Schiff says Trump tried to 'cheat' way to... 2020-01-23 US-NEWS

Trump impeachment – what happens next? 2019-12-19 US-NEWS

**Figure 19:** Search Results

Note the absence of the switch between Guardian and NY Times on the search result page. To obtain the results, the URLs to be used for Guardian and NY Times are described below.

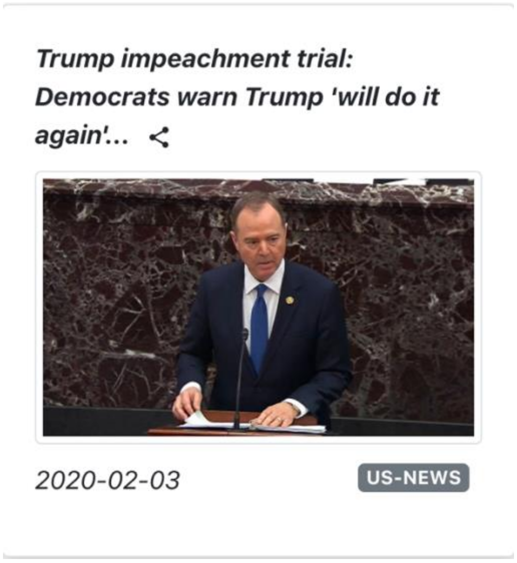
**Guardian News URL:**

https://content.guardianapis.com/search?q=[QUERY_KEYWORD]&api-key=[YOUR_API_KEY]&show-blocks=all

The URL generation above requires 3 parameters:

- The first parameter is the query term to search for
- The second parameter is your API key for Guardian News. The steps to generate your API key are described in section 4.1
- The third parameter is the **"show-blocks"** key with value set to all

The JSON response object is shown in **Figure 4**.



*Trump impeachment trial: Democrats warn Trump 'will do it again'...*

2020-02-03 US-NEWS

**Figure 20:** Guardian News Search Result Article Card

The following table is the mapping between the JSON response and the data to be displayed on the card shown in **Figure 20**.

| Search based Article Card | Corresponding JSON key |
|---|---|
| Title | Value in the hierarchy *"response"* -> *"results[index]"* -> *"webTitle"* |
| Image | Value of the hierarchy *"response"* -> *"results[index]"* ->*"blocks"*->*"main"*->*"elements"* ->*"0"*->*"assests"*->*last indexed assest* -> *"file"* (Refer Figure 5) |
| Section | Value in the hierarchy *"response"* -> *"results[index]"* -> *"sectionId"* |
| Date (yyyy-mm-dd) | Value in the hierarchy *"response"* -> *"results[index]"* ->*"webPublicationDate"* converted to the format as shown in Figure 19 |

**NOTE 1:** The above data needs to be used for each article. “index” represents article number. The article numbers begin from 0.

**NOTE 2:** Only the last image URL is to be used. If no such URLs are found, a default image for Guardian News is to be displayed. Refer section 5.2 for the default image for Guardian News which would be used anytime a Guardian article does not have an image mentioned in the specs.

**NOTE 3:** The section coloring is different for different sections. Refer section 5.1 for a mapping between section and color.

**NOTE 4:** More data than mentioned in the table would need to be stored for future use mentioned hereafter in the document.

**New York Times URL:**

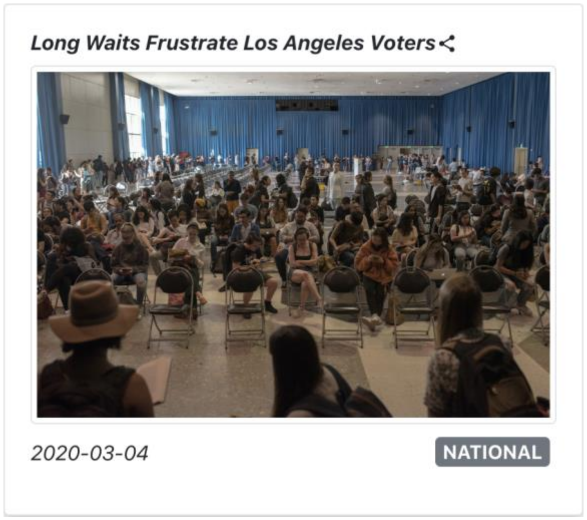


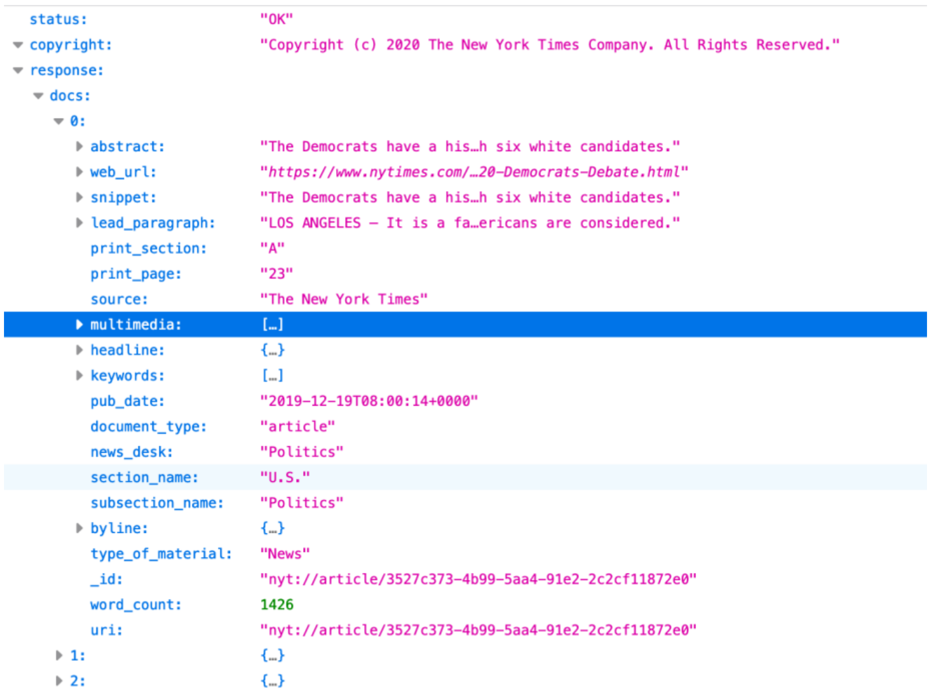
**Figure 21:** NY Times Article card

https://api.nytimes.com/svc/search/v2/articlesearch.json?q=[CITY]&api-key=[YOUR_API_KEY]

The URL generation above requires 3 parameters:

- The first parameter is the query term to search for, here, user city obtained from IP-API
- The second parameter is your API key for New York Times News API. The steps to generate your API key are described in section 4.2

The JSON response is as shown in **Figure 22**.



```
status:                 "OK"
▼ copyright:            "Copyright (c) 2020 The New York Times Company. All Rights Reserved."
▼ response:
  ▼ docs:
    ▼ 0:
      ▶ abstract:          "The Democrats have a his…h six white candidates."
      ▶ web_url:           "https://www.nytimes.com/…20-Democrats-Debate.html"
      ▶ snippet:           "The Democrats have a his…h six white candidates."
      ▶ lead_paragraph:    "LOS ANGELES — It is a fa…ericans are considered."
        print_section:     "A"
        print_page:        "23"
        source:            "The New York Times"
      ▶ multimedia:        […]
      ▶ headline:          {…}
      ▶ keywords:          […]
        pub_date:          "2019-12-19T08:00:14+0000"
        document_type:     "article"
        news_desk:         "Politics"
        section_name:      "U.S."
        subsection_name:   "Politics"
      ▶ byline:            {…}
        type_of_material:  "News"
        _id:               "nyt://article/3527c373-4b99-5aa4-91e2-2c2cf11872e0"
        word_count:        1426
        uri:               "nyt://article/3527c373-4b99-5aa4-91e2-2c2cf11872e0"
    ▶ 1:                   {…}
    ▶ 2:                   {…}
```

**Figure 22:** JSON response from NY Times

The following table is the mapping between the JSON response and the data to be displayed on the card shown in **Figure 21**:

| **Search based Article Card** | **Corresponding JSON key** |
|---|---|
| Title | Value in the hierarchy *"response"* -> *"docs"* -> *"headline"* -> *"main"* |
| Image* | Value of the hierarchy *"response"* -> *"docs"* ->*"multimedia"*->*"url"* |
| Section | Value in the hierarchy *"response"* -> *"docs"* -> *"news_desk"* |
| Date (yyyy-mm-dd) | Value in the hierarchy *"response"* -> *"docs"* ->*"pub_date"* converted to the format as shown in Figure 21 |

* - Pick the first multimedia image with width **atleast 2000.** Simply using the value won't fetch the image. It can be found at the URL https://www.nytimes.com/[IMAGE_URL_FROM_JSON]

**NOTE 1:** The above data needs to be used for each article. The hierarchy would therefore go like "response"->"docs"->[ARTICLE_NUMBER] followed by the ones given in the table above. The article numbers begin from 0.

**NOTE 2:** Only the first image URL with width >= 2000 is to be used. If no such URLs are found, a default image for NY Times News is to be displayed. Refer section 5.2 for the default image for NY Times News which would be used anytime an NY Times article does not have an image mentioned in the specs.

**NOTE 3:** The section coloring is different for different sections. Refer section 5.1 for a mapping between section and color.

**NOTE 4:** More data than mentioned in the table would need to be stored for future use mentioned hereafter in the document.

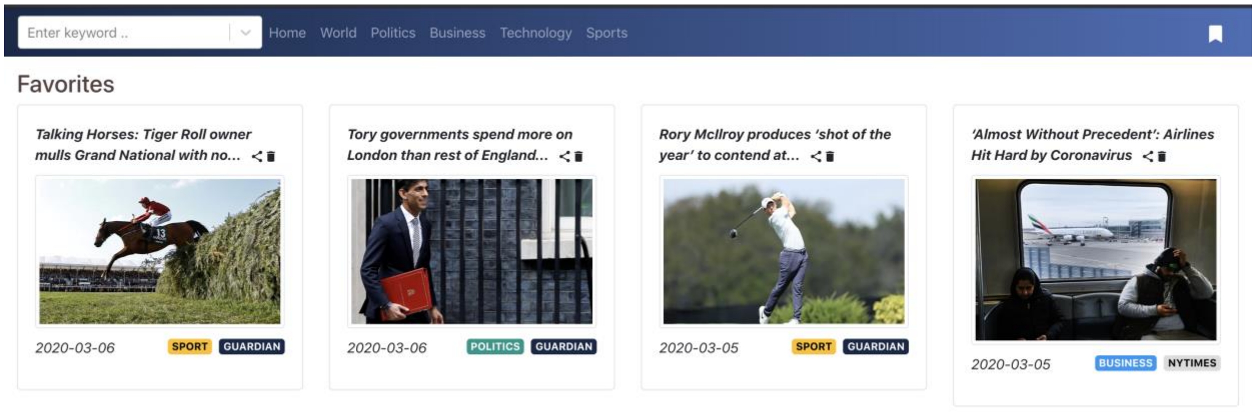
**NOTE 5:** Depending on the section, the color should alter. If section is missing, do not display anything.

**NOTE 6: The vertical size of the cards can vary depending on the image height. That is OK.**

**NOTE 7:** If any image is not found, use the default image provided.

## 3.5 Favorites/Bookmark Tab

From the homepage of the website, the detailed article page and the search result page, the user can navigate to the bookmarks tab. **Figure 23** shows a sample of the bookmarks tab.



*Talking Horses: Tiger Roll owner mulls Grand National with no...*
2020-03-06 SPORT GUARDIAN

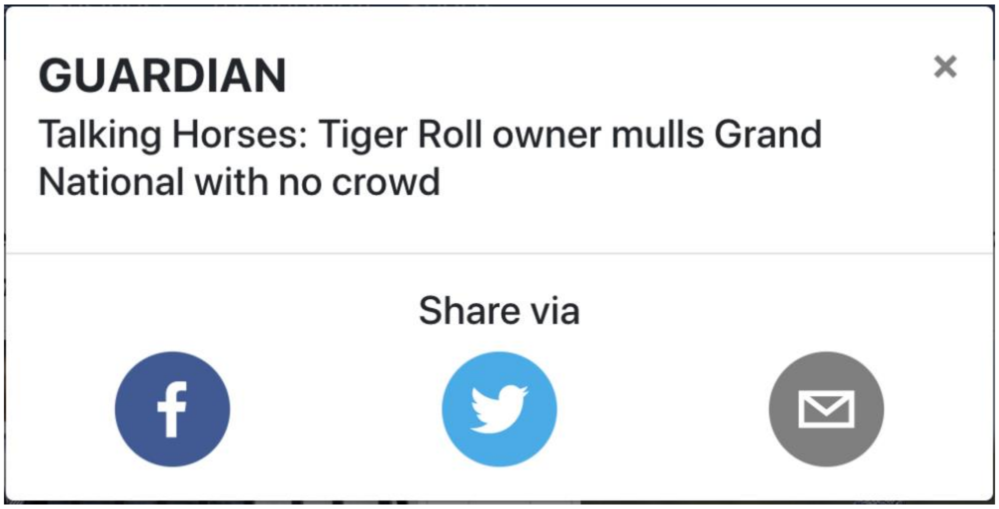
*Tory governments spend more on London than rest of England...*
2020-03-06 POLITICS GUARDIAN

*Rory McIlroy produces 'shot of the year' to contend at...*
2020-03-05 SPORT GUARDIAN

*'Almost Without Precedent': Airlines Hit Hard by Coronavirus*
2020-03-05 BUSINESS NYTIMES

**Figure 23:** Favorites tab

Note the difference in color of the bookmark tab. Apart from the displaying the section, the news source is also displayed with a difference in color for Guardian and NYTimes as shown above. The card format is similar to that in section 3.1.1 with the addition of either Guardian or NY Times and the delete button to remove the article from bookmarks.

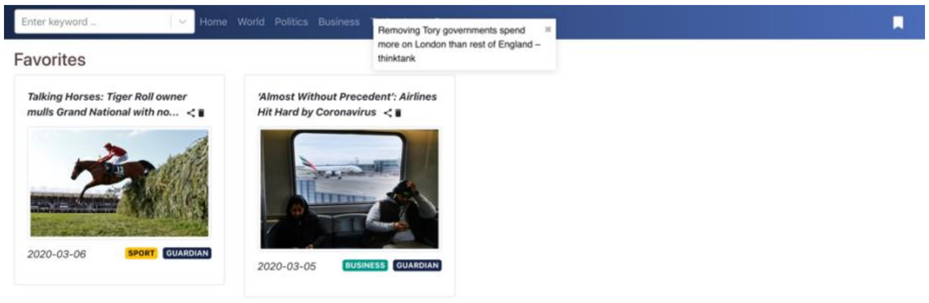

GUARDIAN

Talking Horses: Tiger Roll owner mulls Grand National with no crowd

Share via

**Figure 24:** Sharing modal in bookmarks tab

The share button brings up a modal similar to that in section 3.1.2 with the additional mention of whether it was reported in Guardian or NY Times (see **Figure 24**). On removing an article, a toast pops up as shown in **Figure 25**.



Removing Tory governments spend more on London than rest of England – thinktank

Talking Horses: Tiger Roll owner mulls Grand National with no...

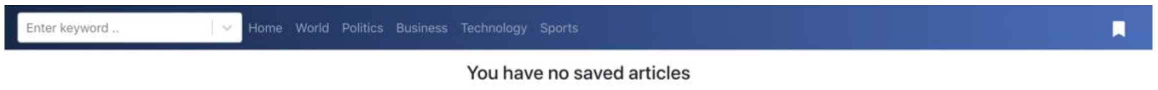
2020-03-06

'Almost Without Precedent': Airlines Hit Hard by Coronavirus

2020-03-05

**Figure 25:** Removing article from bookmarks/favorites tab

**Figure 26** shows the case when there are no articles that have been bookmarked.



You have no saved articles

**Figure 26:** No bookmarked articles

### 3.6 Switching between Guardian and NY Times

**Figures 27** and **Figure 28** show the switch condition when you switch between the two news resources.

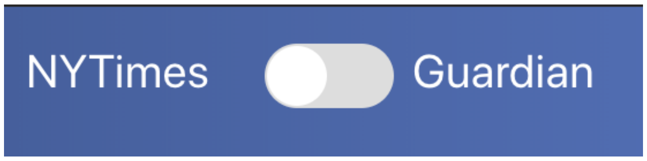


**Figure 27:** NYTimes selected

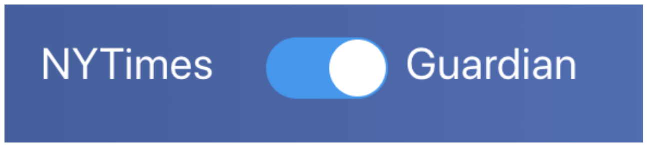
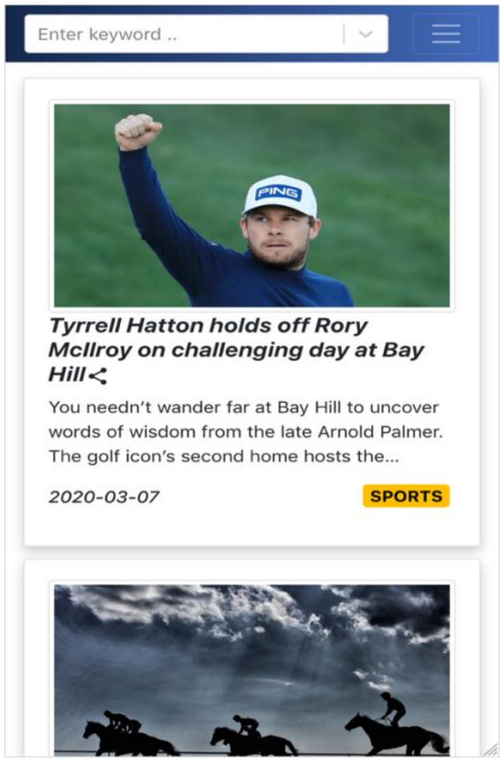


**Figure 28:** Guardian selected

Note here that the page refreshes anytime the switch is changed. Also note the difference in color of the switch.

**<u>NOTE: The selection of source should persist anytime a user closes and reopens the website. If he/she closed the website with Guardian selected, the next time he/she opens it, the switch should be on Guardian only.</u>**

## 3.6 Mobile View Screenshots:



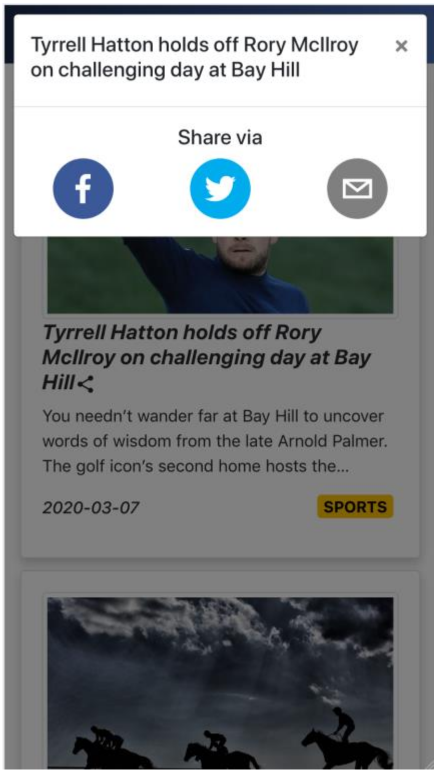
Tyrrell Hatton holds off Rory McIlroy on challenging day at Bay Hill

You needn't wander far at Bay Hill to uncover words of wisdom from the late Arnold Palmer. The golf icon's second home hosts the...

2020-03-07

SPORTS

**Figure 29:** Initial website


Tyrrell Hatton holds off Rory McIlroy on challenging day at Bay Hill

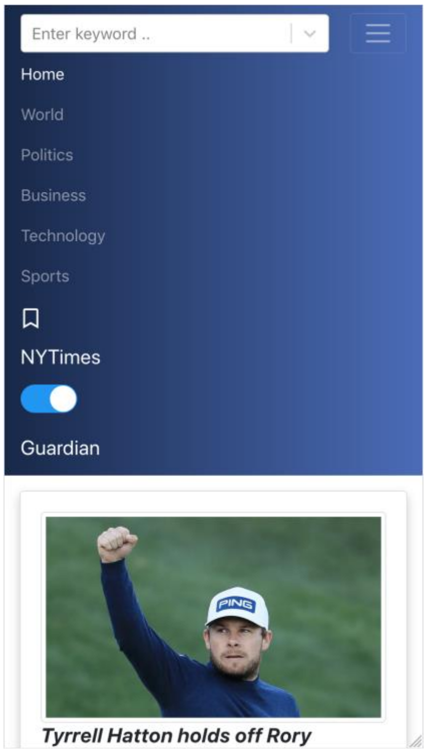
Share via

*Tyrrell Hatton holds off Rory McIlroy on challenging day at Bay Hill*

You needn't wander far at Bay Hill to uncover words of wisdom from the late Arnold Palmer. The golf icon's second home hosts the...

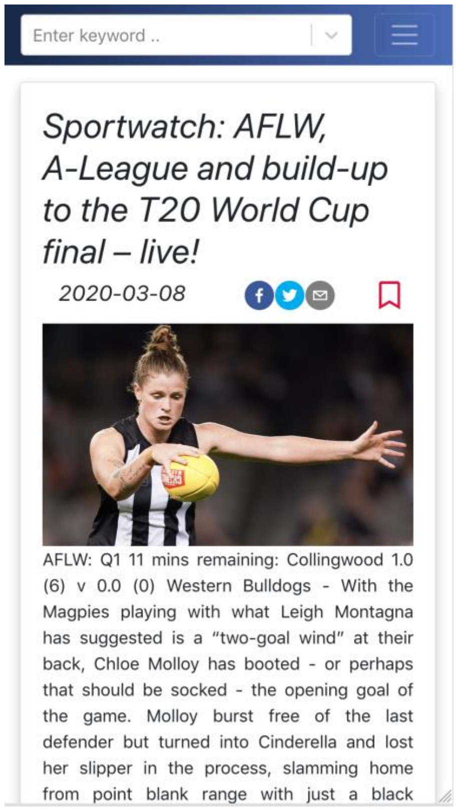
*2020-03-07* SPORTS

**Figure 30:** Sharing Modal



*Tyrrell Hatton holds off Rory*

**Figure 31:** Opening the rightmost tab

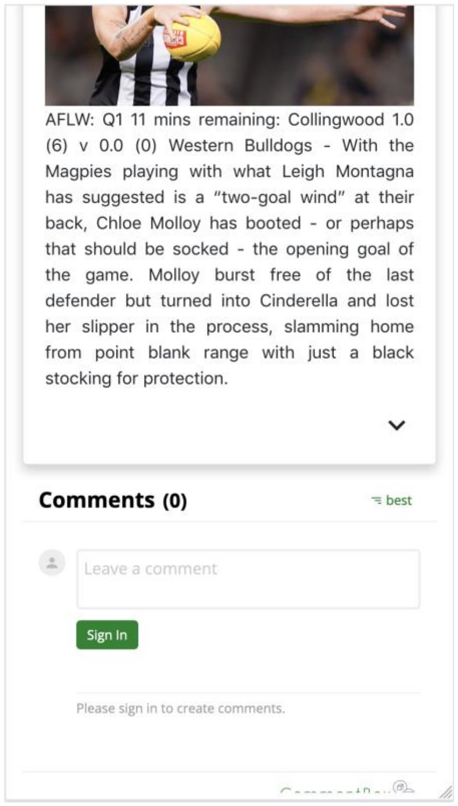


*Sportwatch: AFLW, A-League and build-up to the T20 World Cup final – live!*

2020-03-08

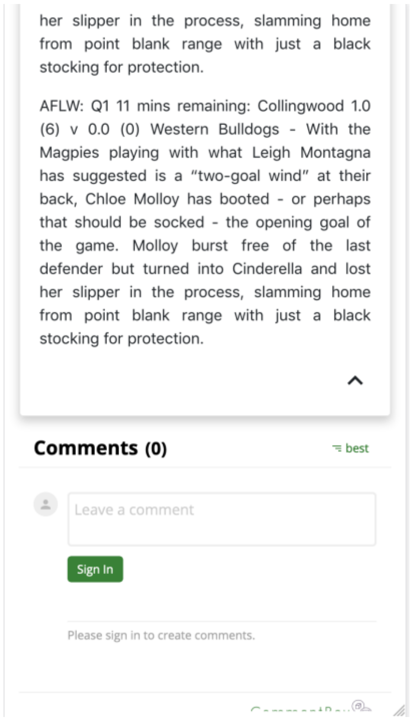
AFLW: Q1 11 mins remaining: Collingwood 1.0 (6) v 0.0 (0) Western Bulldogs - With the Magpies playing with what Leigh Montagna has suggested is a "two-goal wind" at their back, Chloe Molloy has booted - or perhaps that should be socked - the opening goal of the game. Molloy burst free of the last defender but turned into Cinderella and lost her slipper in the process, slamming home from point blank range with just a black

**Figure 32:** Detailed article



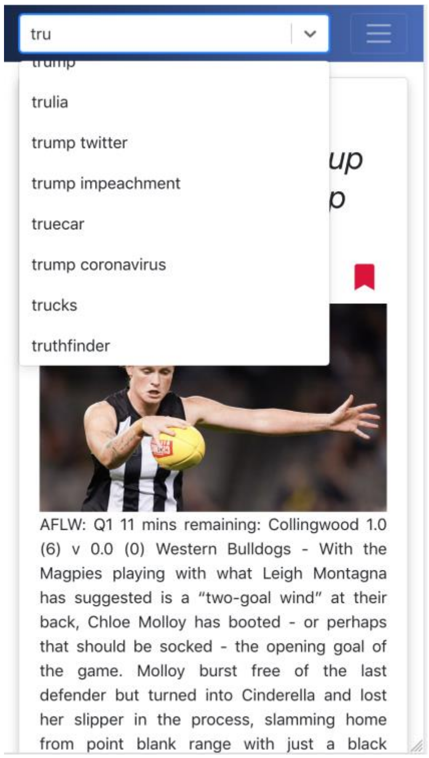
AFLW: Q1 11 mins remaining: Collingwood 1.0 (6) v 0.0 (0) Western Bulldogs - With the Magpies playing with what Leigh Montagna has suggested is a "two-goal wind" at their back, Chloe Molloy has booted - or perhaps that should be socked - the opening goal of the game. Molloy burst free of the last defender but turned into Cinderella and lost her slipper in the process, slamming home from point blank range with just a black stocking for protection.

**Figure 33:** Detailed article (continued)


her slipper in the process, slamming home from point blank range with just a black stocking for protection.

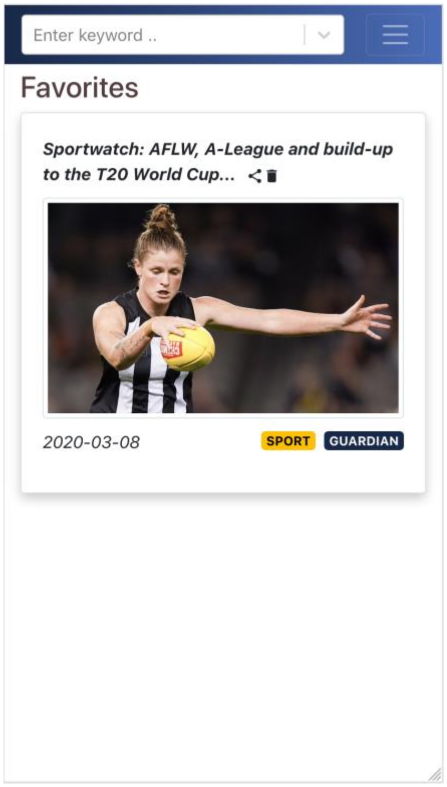
AFLW: Q1 11 mins remaining: Collingwood 1.0 (6) v 0.0 (0) Western Bulldogs - With the Magpies playing with what Leigh Montagna has suggested is a "two-goal wind" at their back, Chloe Molloy has booted - or perhaps that should be socked - the opening goal of the game. Molloy burst free of the last defender but turned into Cinderella and lost her slipper in the process, slamming home from point blank range with just a black stocking for protection.

**Figure 34:** Detailed article description expansion



AFLW: Q1 11 mins remaining: Collingwood 1.0 (6) v 0.0 (0) Western Bulldogs - With the Magpies playing with what Leigh Montagna has suggested is a "two-goal wind" at their back, Chloe Molloy has booted - or perhaps that should be socked - the opening goal of the game. Molloy burst free of the last defender but turned into Cinderella and lost her slipper in the process, slamming home from point blank range with just a black

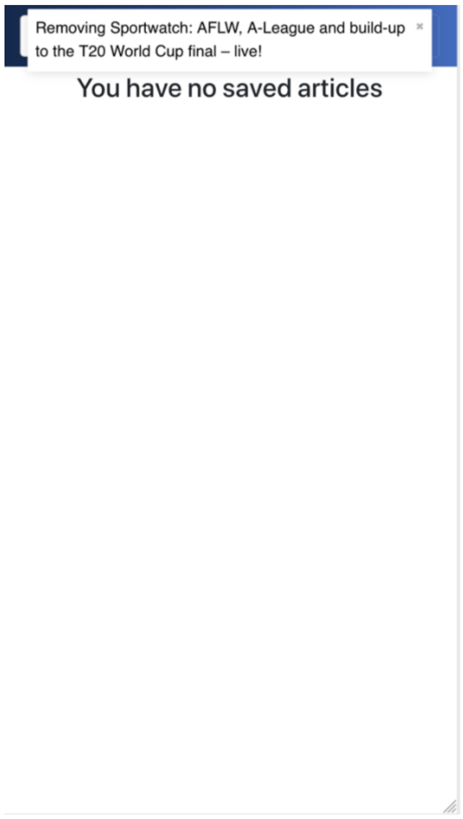
**Figure 35:** Autosuggest example for search



Sportwatch: AFLW, A-League and build-up to the T20 World Cup...

2020-03-08 SPORT GUARDIAN

**Figure 36:** Favorites/Bookmark tab



Removing Sportwatch: AFLW, A-League and build-up to the T20 World Cup final – live!

You have no saved articles

**Figure 37:** No results in bookmark tab

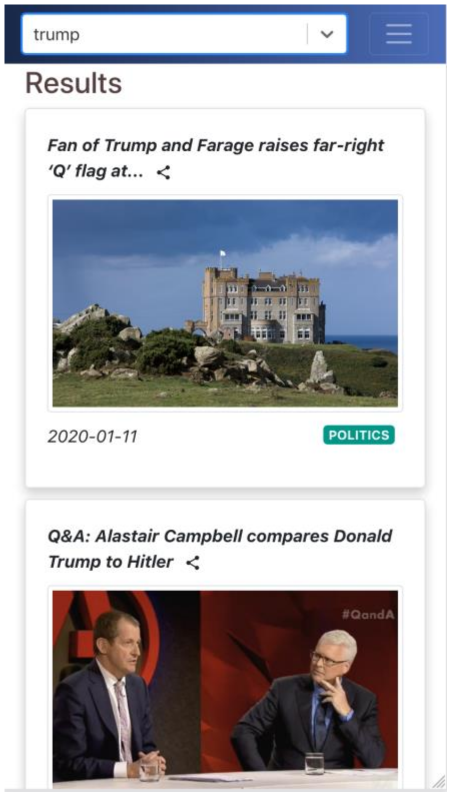

trump
Results
Fan of Trump and Farage raises far-right 'Q' flag at...
2020-01-11
POLITICS
Q&A: Alastair Campbell compares Donald Trump to Hitler

**Figure 38:** Search result page

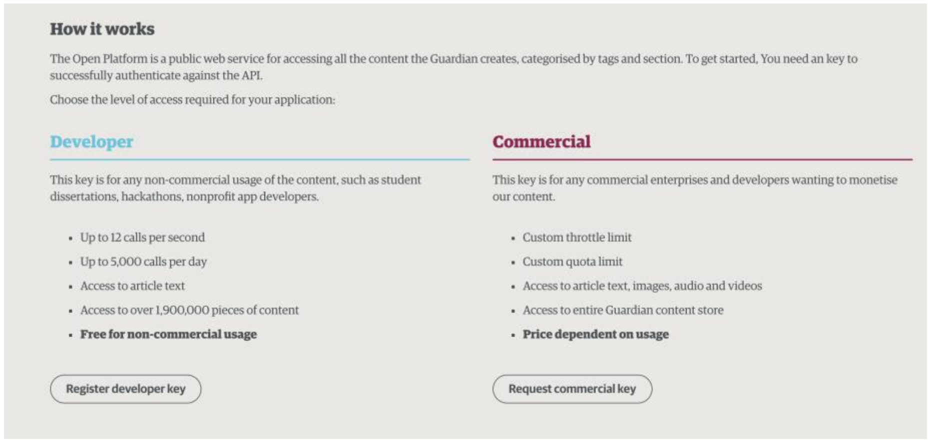
## 4. APIs Used

### 4.1 Guardian News API

To get the API key for Guardian News, visit:

https://open-platform.theguardian.com/documentation/

From here, navigate to "sign up for an API key" which takes you to the following page



How it works

The Open Platform is a public web service for accessing all the content the Guardian creates, categorised by tags and section. To get started, You need an key to successfully authenticate against the API.

Choose the level of access required for your application:

Developer

This key is for any non-commercial usage of the content, such as student dissertations, hackathons, nonprofit app developers.

- Up to 12 calls per second
- Up to 5,000 calls per day
- Access to article text
- Access to over 1,900,000 pieces of content
- **Free for non-commercial usage**

Register developer key

Commercial

This key is for any commercial enterprises and developers wanting to monetise our content.

- Custom throttle limit
- Custom quota limit
- Access to article text, images, audio and videos
- Access to entire Guardian content store
- **Price dependent on usage**

Request commercial key

**Figure 39:** Guardian News Register for key page

Register for developer key and on the subsequent page, enter details as asked. Once you click register, a key will be emailed to you on the email address you use.

### 4.2 New York Times API

Visit https://developer.nytimes.com/get-started and create an account by providing the details asked. Then:

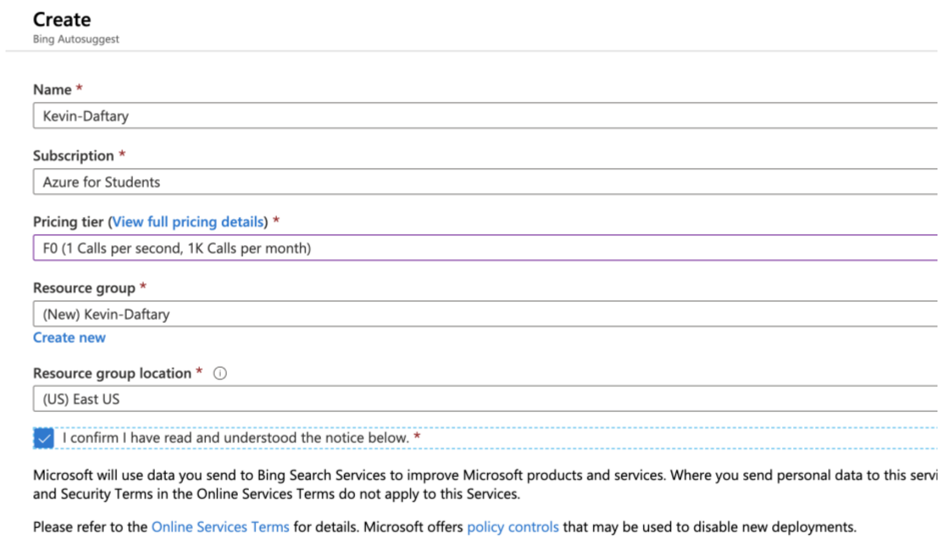
- Verify the account using the link emailed to you.
- Sign in to the account you just created.
- From the top right corner where your username/email is present, select the "Apps" option in the dropdown and click on "New App".
- Provide a suitable name for your app and amongst the APIs, select the **Article Search API** and the **Top Stories API.**
- Click **'Create'.**

You will be taken to a page where you can find your API key.

### 4.3 Bing Autosuggest API:

Visit the given link - https://azure.microsoft.com/en-us/services/cognitive-services/autosuggest/

Click 'Try Bing Autosuggest' and opt for either the Free Azure account version or the existing account version. If you are creating a new account, use your USC email address to do so. For the existing azure account holders, sign into your account.


Create
Bing Autosuggest

Name *
Kevin-Daftary

Subscription *
Azure for Students

Pricing tier (View full pricing details) *
F0 (1 Calls per second, 1K Calls per month)

Resource group *
(New) Kevin-Daftary
Create new

Resource group location *
(US) East US

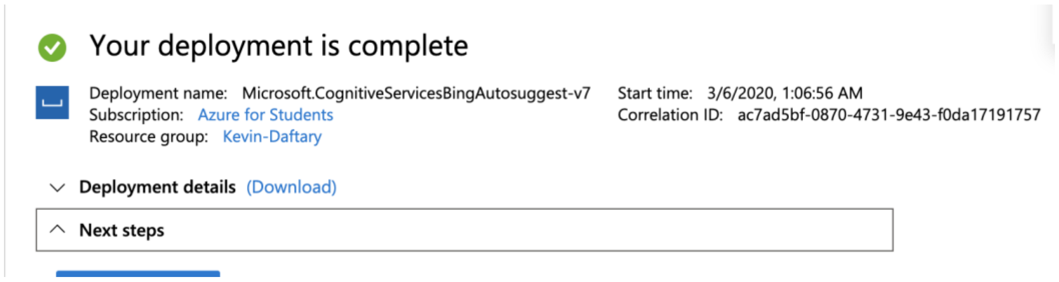
✓ I confirm I have read and understood the notice below. *

Microsoft will use data you send to Bing Search Services to improve Microsoft products and services. Where you send personal data to this servi
and Security Terms in the Online Services Terms do not apply to this Services.

Please refer to the Online Services Terms for details. Microsoft offers policy controls that may be used to disable new deployments.

**Figure 40:** NY Times registration step

Fill in the details as shown above. **Make sure the pricing tier is F0.** Click **'Create'** and wait for it to finish. You MAY reserve T tier if you have 200$ free credits. Keep in mind the autosuggest pricing is 3$/1000 requests and allows more API requests/sec.



Your deployment is complete

Deployment name: Microsoft.CognitiveServicesBingAutosuggest-v7
Subscription: Azure for Students
Resource group: Kevin-Daftary
Start time: 3/6/2020, 1:06:56 AM
Correlation ID: ac7ad5bf-0870-4731-9e43-f0da17191757

**Figure 41:** Final confirmation for NY Times

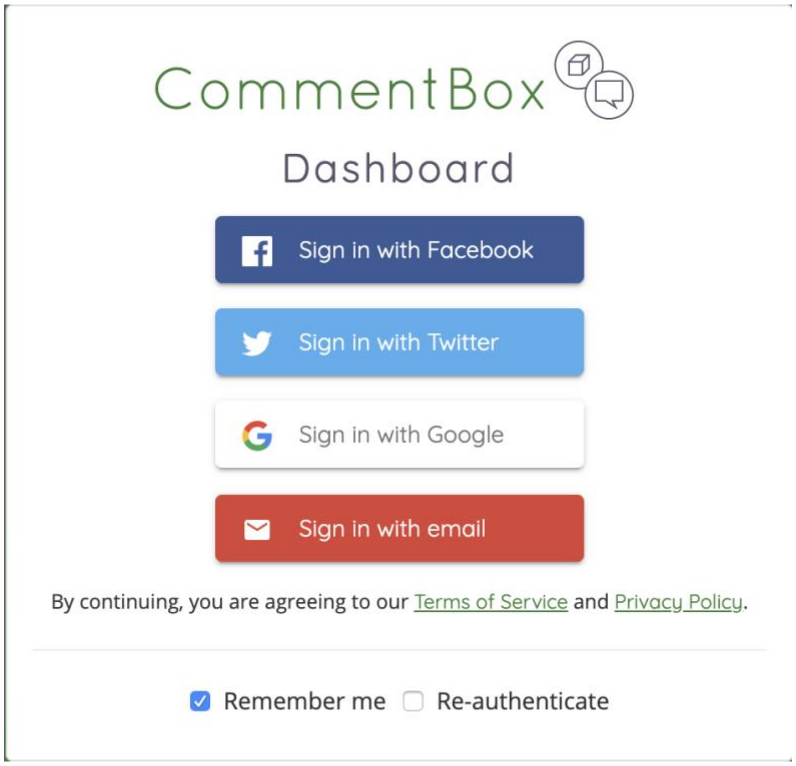
Click “Go To Resource” and find the API key you will require.

## 4.4 CommentBox.io API:

To get started with the CommentBox.io API, visit:

https://dashboard.commentbox.io/projects

**Figure 42** shows the page it sends you to.

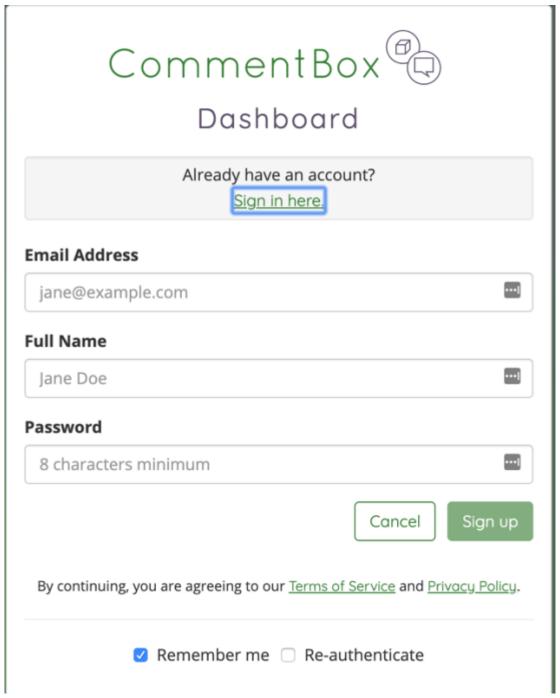


By continuing, you are agreeing to our Terms of Service and Privacy Policy.

☑ Remember me ☐ Re-authenticate

**Figure 42:** CommentBox.io

Select any of the options and **SIGN UP (NOT SIGN IN!)** for the API. **Figure 43** shows the **Sign up** modal.



CommentBox

Dashboard

Already have an account?
Sign in here

Email Address
jane@example.com

Full Name
Jane Doe

Password
8 characters minimum

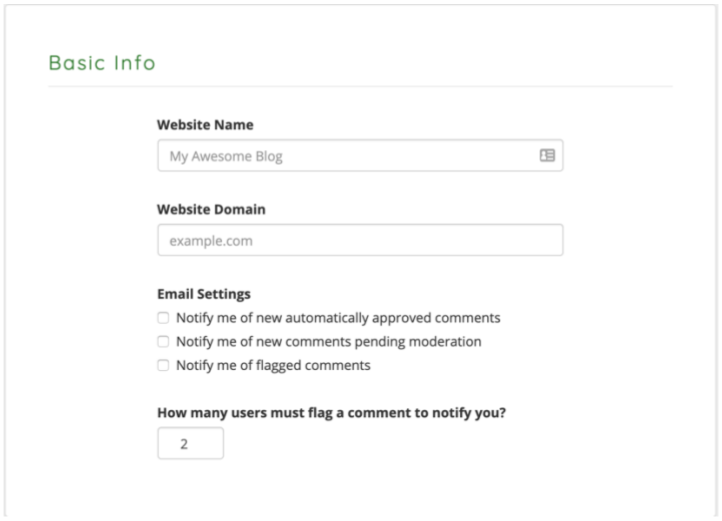
Cancel Sign up

By continuing, you are agreeing to our Terms of Service and Privacy Policy.

☑ Remember me ☐ Re-authenticate

**Figure 43**: Sign up modal

Enter your details. You should receive a confirmation link. Go ahead and verify your account.



Basic Info

Website Name
My Awesome Blog

Website Domain
example.com

Email Settings
☐ Notify me of new automatically approved comments
☐ Notify me of new comments pending moderation
☐ Notify me of flagged comments

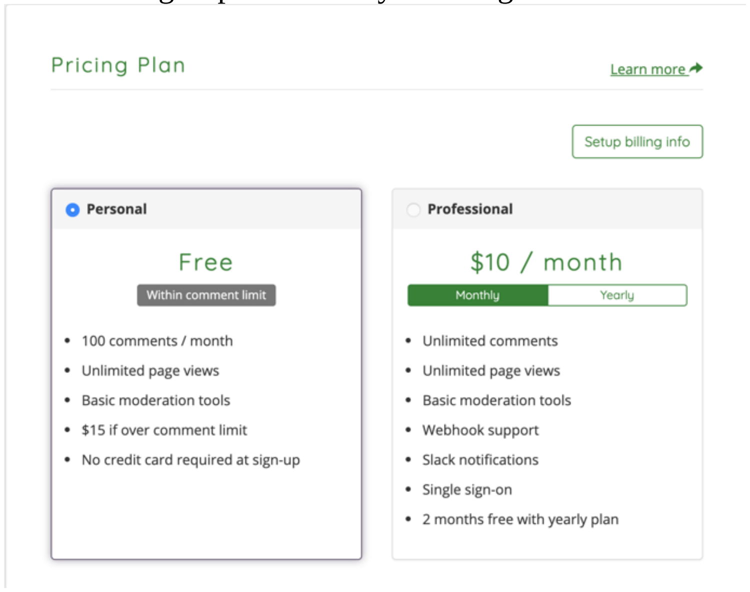
How many users must flag a comment to notify you?
2

**Figure 44**: CommentBox.io Project Creation Step 1

**Figure 44** shows what you should have in front of after signing up. Select any website name. The Website Domain is **VERY IMPORTANT**. This field is what allows you to have comment boxes in your website. Add in your website's domain here (Something like localhost:port_number or 127.0.0.1:port_number, etc. before you deploy).

**Remember to update this domain to the domain of your cloud deployment. Failure to do so will affect the availability of comment box in your final homework and would lead to deduction in points.**

For the last 2 questions, pick the first option and keep the default value respectively. **Figure 45** to **Figure 47** show the remaining steps reached by scrolling down.



[● Personal]

Free

Within comment limit

- 100 comments / month
- Unlimited page views
- Basic moderation tools
- $15 if over comment limit
- No credit card required at sign-up

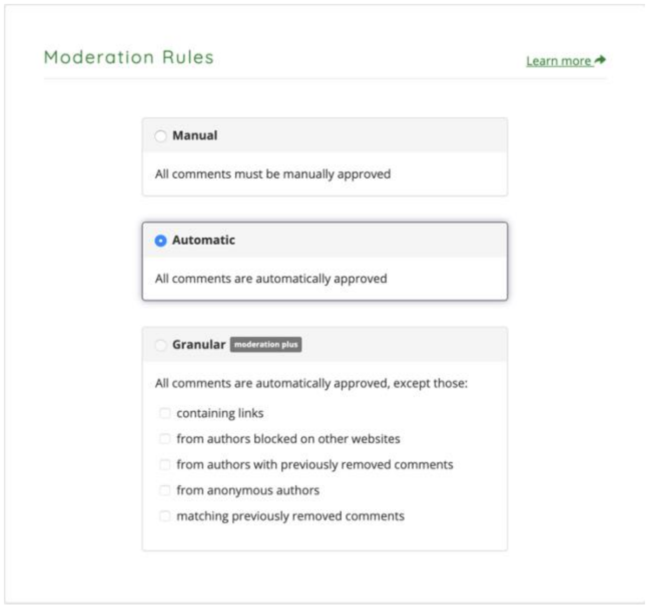
○ Professional

$10 / month

Monthly | Yearly

- Unlimited comments
- Unlimited page views
- Basic moderation tools
- Webhook support
- Slack notifications
- Single sign-on
- 2 months free with yearly plan

**Figure 45**: CommentBox.io Project Creation Step 2



○ Manual

All comments must be manually approved

[● Automatic]

All comments are automatically approved

○ Granular moderation plus

All comments are automatically approved, except those:

- ☐ containing links
- ☐ from authors blocked on other websites
- ☐ from authors with previously removed comments
- ☐ from anonymous authors
- ☐ matching previously removed comments

**Figure 46**: CommentBox.io Project Creation Step 3

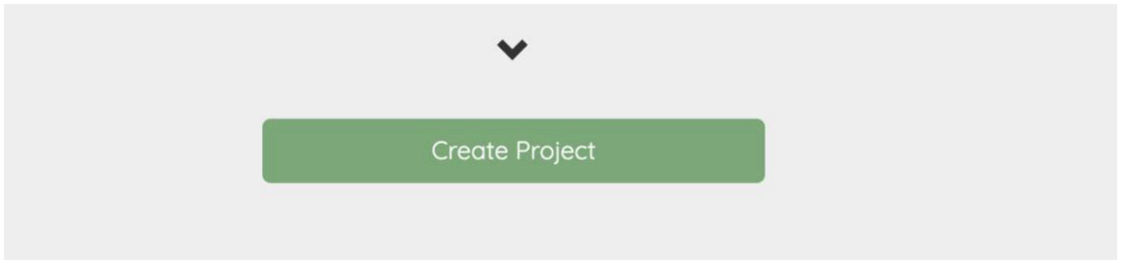
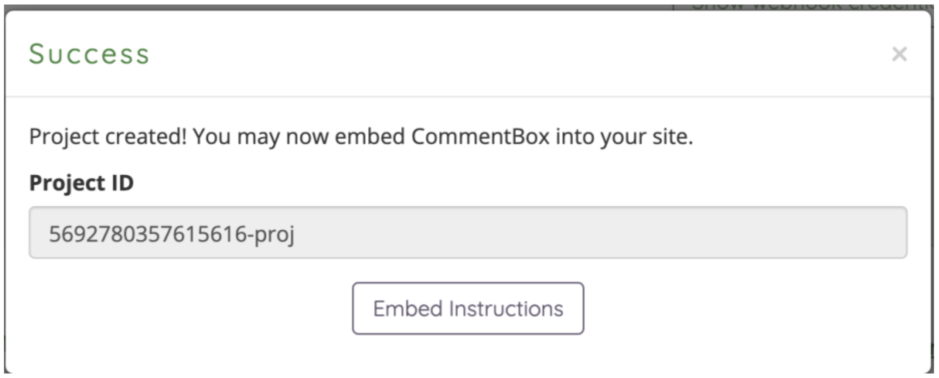


**Figure 47**: CommentBox.io Project Creation Step 4

For steps 2 and 3, keep the default selections and skip over the remaining questions which are optional. Click **"Create Project"** in step 4. This only works if you have verified your account. **Figure 48** should show up. Note down the project ID. The embed instructions takes you to the documentation link provided above.



Success

Project created! You may now embed CommentBox into your site.

**Project ID**

5692780357615616-proj

**Figure 48**: CommentBox.io Project Creation Step 5

## 5. Hints and Useful Links

### 5.1 Difference in tags for sections

The mapping below shows the different colors for each tab.

| | |
|---|---|
| World | WORLD |
| Politics | POLITICS |
| Business | BUSINESS |
| Technology | TECHNOLOGY |
| Sports | SPORTS |
| Any other | HEALTH |
| Guardian Favorite | GUARDIAN |
| NY Times Favorite | NYTIMES |

### 5.2 Default Images for Guardian and New York Times:

Guardian News –
https://assets.guim.co.uk/images/eada8aa27c12fe2d5afa3a89d3fbae0d/fallback-logo.png

NY Times –
https://upload.wikimedia.org/wikipedia/commons/0/0e/Nytimes_hq.jpg

### 5.3 Links that might help you:

- React Bootstrap:
  - https://react-bootstrap.github.io/components/navbar/
- React Router DOM:
  - https://www.freecodecamp.org/news/react-router-in-5-minutes/
  - https://reacttraining.com/react-router/web/example/url-params
  - https://alligator.io/react/react-router-parameters/
  - https://learnwithparam.com/blog/how-to-handle-query-params-in-react-router/
  - https://codesandbox.io/s/n7q6kw9n8m?from-embed
  - https://github.com/react-bootstrap/react-bootstrap/issues/3944
- React-Select:
  - https://alligator.io/react/react-select/
  - https://react-select.com/home#async
- React Switch:
  - https://www.npmjs.com/package/react-switch
- React Share:
  - https://github.com/nygardk/react-share
- React Icons for Bookmark:
  - https://github.com/react-icons/react-icons
  - https://react-icons.netlify.com/#/search
- React Responsive Library:
  - https://www.npmjs.com/package/react-responsive
- React Toast:
  - https://github.com/fkhadra/react-toastify
- React Tooltip:
  - https://github.com/wwayne/react-tooltip
- React Spinners:
  - https://www.npmjs.com/package/react-spinners
  - https://github.com/emotion-js/emotion

### 5.4 Google App Engine/Amazon Web Services/ Microsoft Azure

You should use the domain name of the GAE/AWS/Azure service you created in Homework #7 to make the request. For example, if your GAE/AWS/Azure server domain is called example.appspot.com/example.elasticbeanstalk.com/ example.azurewebsites.net, the JavaScript program will perform a GET request with keyword=”xxx”, and an example query of the following type will be generated:

GAE - http://example.appspot.com/articleSearch?query=xxx
AWS - http://example.elasticbeanstalk.com/articleSearch?query=xxx
Azure – http://example.azurewebsites.net/articleSearch?query=xxx

Your URLs don't need to be the same as the ones above. You can use whatever paths and parameters you want. Please note that in addition to the link to your Homework #8, you should also **provide a link like this URL in the table of your Node.JS backend link**. When your grader clicks on this additional link, a valid link should return a JSON object with appropriate data.

## 5.5 Deploy Node.js application on GAE/AWS/Azure

Since Homework #8 is implemented with Node.js on AWS/GAE/Azure, you should **select Nginx as your proxy server (if available)**, which should be the default option.

## 5.6 AJAX call

You should send the request to the Node.js script(s) by calling an Ajax function (ReactJs). You **must use a GET method** to request the resource since you are required to provide this link to your homework list to let graders check whether the Node.js script code is running in the "cloud" on Google GAE/AWS/Azure (see 5.3 above). Please refer to the grading guidelines for details.

**Note: jQuery .ajax cannot be used in this project.**

## 5.7 HTML5 Local Storage

Local storage is more secure, and large amounts of data can be stored locally, without affecting website performance. Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server. There are two methods, getItem() and setItem(), that you can use. The local storage can only store strings. Therefore, you need to convert the data to string format before storing it in the local storage. For more information, see:

https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage
http://www.w3schools.com/html/html5_webstorage.asp
https://stackoverflow.com/questions/2010892/storing-objects-in-html5-localstorage

## 6. Files to Submit

In your course homework page, you should update the Homework #8 link to refer to your new initial web page for this exercise. Additionally, you need to provide **an additional link** to the URL of the GAE/AWS/Azure service where the AJAX call is made with sample parameter values (i.e. a valid query, with keyword, location, etc. See 6.6).

Also, submit all files (HTML, JS, CSS) you have written yourself, electronically to the GitHub Classroom repository so that can be compared to all other students' code. **Don't include library, node-js modules, any config files or any images that we provided or that are included in any library or any code generated by the tools.**

**Please note, you need to submit your files (HTML, JS, CSS), both backend and frontend, directly to the root folder of your homework 8 GitHub Classroom repository**. That is, **do not submit any subfolders on GitHub Classroom**. If you submit some subfolders, you will receive a 2-points penalty.

****IMPORTANT**:**
All videos are part of the homework description. All discussions and explanations in Piazza related to this homework are part of the homework description and will be accounted into grading. So please review all Piazza threads before finishing the assignment.