Homework 8: Responsive Web Design

Congress Information Search

(AJAX/JSON/Bootstrap/jQuery/AngularJS/Cloud Exercise)

1. Objectives

- Become familiar with the AJAX and JSON technologies
- Use a combination of HTML5, CSS, Bootstrap, jQuery, AngularJS and PHP
- Get hands-on experience in Google Cloud App Engine or Amazon Web Services
- Get hands-on experience on how to use Bootstrap to enhance the user experience
- Provide an interface to perform congress search using Sunlight Congress APIs

2. Background

2.1 AJAX and JSON

Ajax (short for Asynchronous JavaScript and XML) is a mechanism which enables the communication between the client and the server without the need for a post-back or a complete page refresh. It can also be defined as a method of exchanging data with a server, and updating some parts of a web page – without reloading the entire page.

In this exercise, you have to use the AJAX component provided in JQuery. Please see the class slides at:

http://cs-server.usc.edu:45678/slides/jQueryTutorial.ppt

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:

http://cs-server.usc.edu:45678/slides/JSON1.ppt

2.2 Bootstrap

Bootstrap is a free collection of tools for creating responsive websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. To learn more details about Bootstrap please refer to the lecture material on Responsive Web Design (RWD). See the class slides at:

http://cs-server.usc.edu:45678/slides/Responsive.pdf

and

http://getbootstrap.com/

2.3 Sunlight Congress API

The Sunlight Congress API is a live JSON API for the people and work of Congress, provided by the Sunlight Foundation. With the API you can:

- Look up members of Congress by location or by zip code.
- Obtain the official Twitter, YouTube, and Facebook accounts.
- Look up the daily work of Congress: bills, amendments, nominations.
- Get the live activity of Congress: past and future votes, floor activity, hearings.

In order to make requests to Sunlight Congress API, you need to get an API key:

- Go to http://sunlightfoundation.com/api/
- Click on "Get a key!"
- Fill the sign up form
- You will get a confirm email. Click on the link in the email.

Then login at the Sunlight Foundation website and go to your "Profile Settings". You will see your API key string right above you name.

For information about the Sunlight Congress API, please go to:

https://sunlightlabs.github.io/congress/

and

http://tryit.sunlightfoundation.com/congress

2.4 Amazon Web Services (AWS)

AWS is Amazon's implementation of cloud computing. AWS includes 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 can 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 7.5 for .NET, and Apache Tomcat for Java.

The Amazon Web Services homepage is available at: http://aws.amazon.com/

2.5 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 PHP visit the page:

https://cloud.google.com/appengine/docs/php/

3. Description

Similar to homework 6, in this exercise you will create a webpage that allows users to search for congress information using the Sunlight Congress API.

The difference being, in this homework you will create a PHP script to return a JSON formatted data stream to the front-end. The client parses the JSON data and renders it in a nicer-looking, responsive UI (using Bootstrap).

A user first opens a page as shown below in Figure 1. The page includes a navigation bar at the left and the legislators' information table at the right.

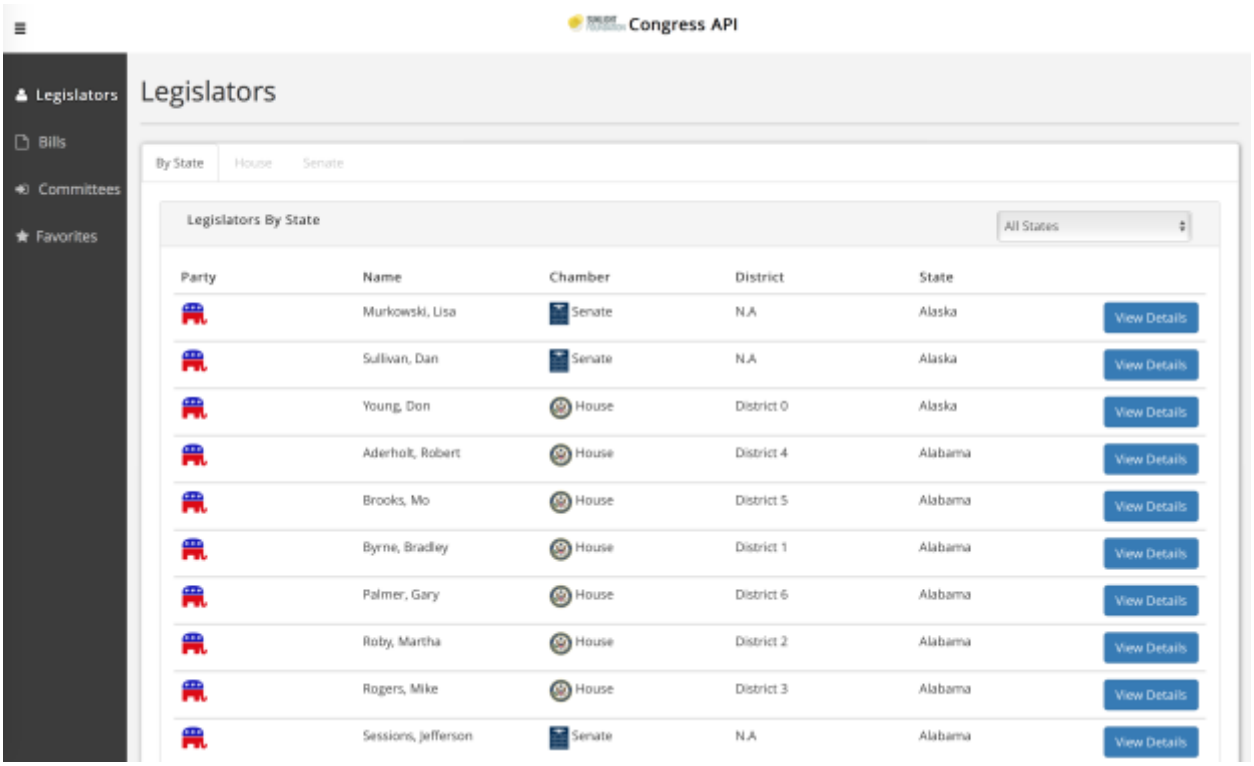


Figure 1 – Legislators page

Once the user opens this page, all of the legislators' information data is loaded using the AJAX component in JQuery. The AJAX-JQuery code calls your PHP script. The webpage must then use JavaScript to extract data from the JSON response and display the results on the same webpage. Description of how to display the results is given in the Hints section.

3.1 Page Header

At the top of the webpage, there is a page header which contains a stack icon at left and an image link in the center. The header is shown in Figure 2.

Figure 2 – Page Header

- When the stack icon is clicked, the navigation bar should be toggled (show / hide).
- The image is linked to http://sunlightfoundation.com/, the link should be opened in a new page instead of replace current page.

3.2 Navigation Bar

The navigation bar should contain 4 items (Legislators, Bills, Committees and Favorites), each item starts with an icon, followed by the text. The navigation bar is shown in Figure 3.

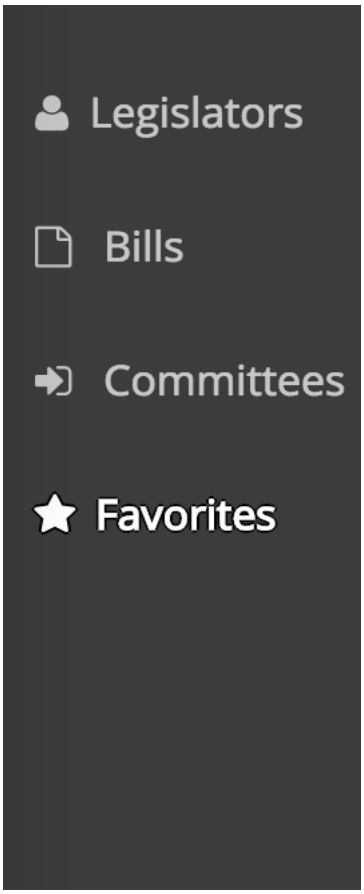


Figure 3

When the user clicks on an item in this navigation bar, the corresponding content should be loaded. For example, when the user clicks on Legislators, the legislators' information table should be loaded to the content part.

3.3 Legislators

3.3.1 Design

You must replicate the table displayed in Figure 4 using a **Bootstrap table**.

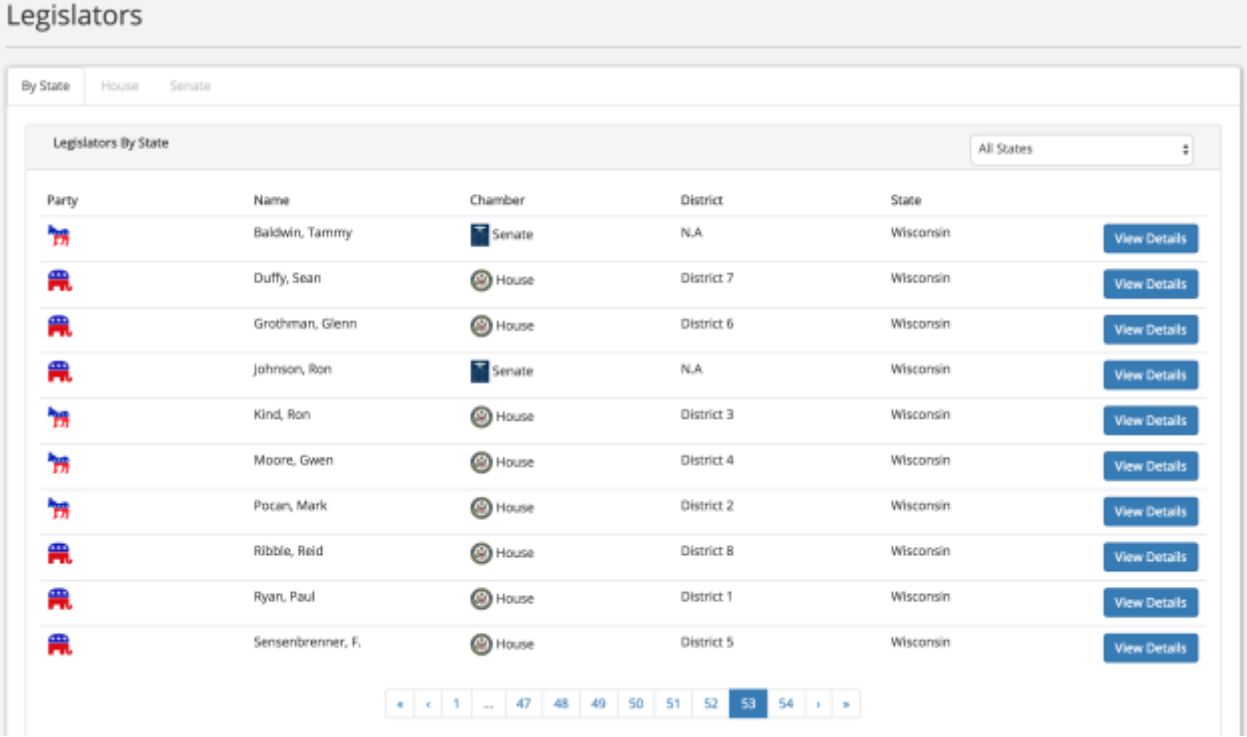


Figure 4 – Legislators Table

- The page starts with a page heading displays "Legislators" using **Bootstrap headings**.
- Then displays three tabs which are *State*, *House* and *Senate*. The tabs should be implemented using **Bootstrap Togglable tabs**.
- Each tab contains a table with filters (top right) and paginations (bottom).
- **You need to load all legislators in total to display (approximately 538 of them).**

3.3.1 State tab

You should replicate the result table as shown in Figure 4.

The table contains the information shown in Table 1.

Table Field	**Description**
Party	Displays the party of the legislator. The party should be shown in image format. The image assets are provided in the "Hints" section.
Name	Displays the name of the legislator. The format should be *Last Name, First Name*.
Chamber	Displays the chamber of the legislator. You should display the chamber image first. Then followed by the chamber name.
District	Displays the district of the legislator. Display N.A., if not available.

State	Displays the state of the legislator.
View Details	Displays a button. Information about this button will be enclosed in later section.

Table 1: Legislator table description

3.3.1.1 Filter By States

At the top right of the table, there is a HTML select tag which enables the user to filter legislators by state, or select the (default) All States. See Figure 5 and 6.

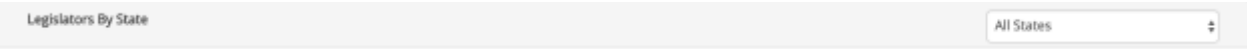


Figure 5 – State Selector

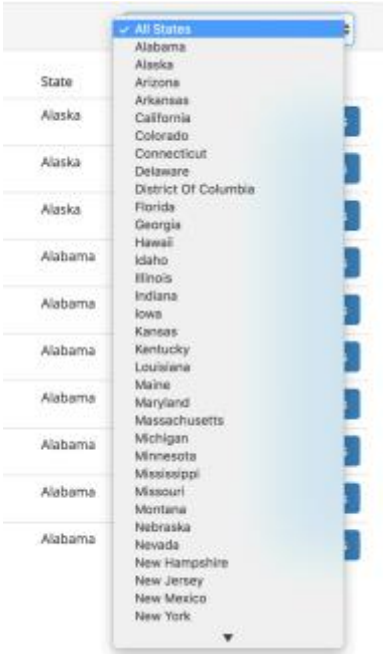


Figure 6 – State Selector Drop-down

When a state is selected, the information table should only display legislators in that state. As shown in Figure 7.

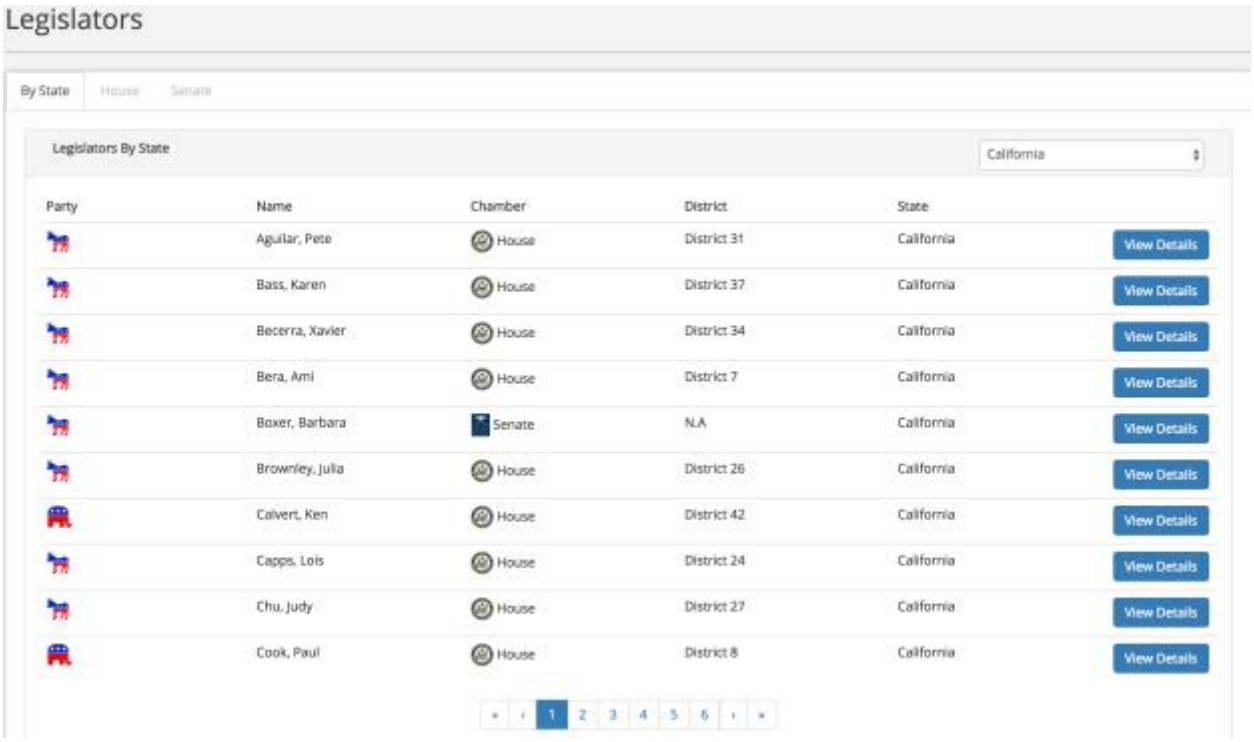


Figure 7 – Legislators filtered by State

You can implement this feature using AngularJS's dirPagination plugin. Details about this plugin are explained in the Hints section.

3.3.1.2 Pagination

The table should be paginated. Each page displays a maximum of 10 results.

You can also implement this feature using AngularJS's dirPagination plugin. Details about this plugin are explained in the Hints section.

3.3.1.3 View Details

When the "View Details" button is clicked, the page should navigate to the detail page of this legislator. You can implement this feature using **Bootstrap Carousel**.

The detail page is shown in Figure 8.

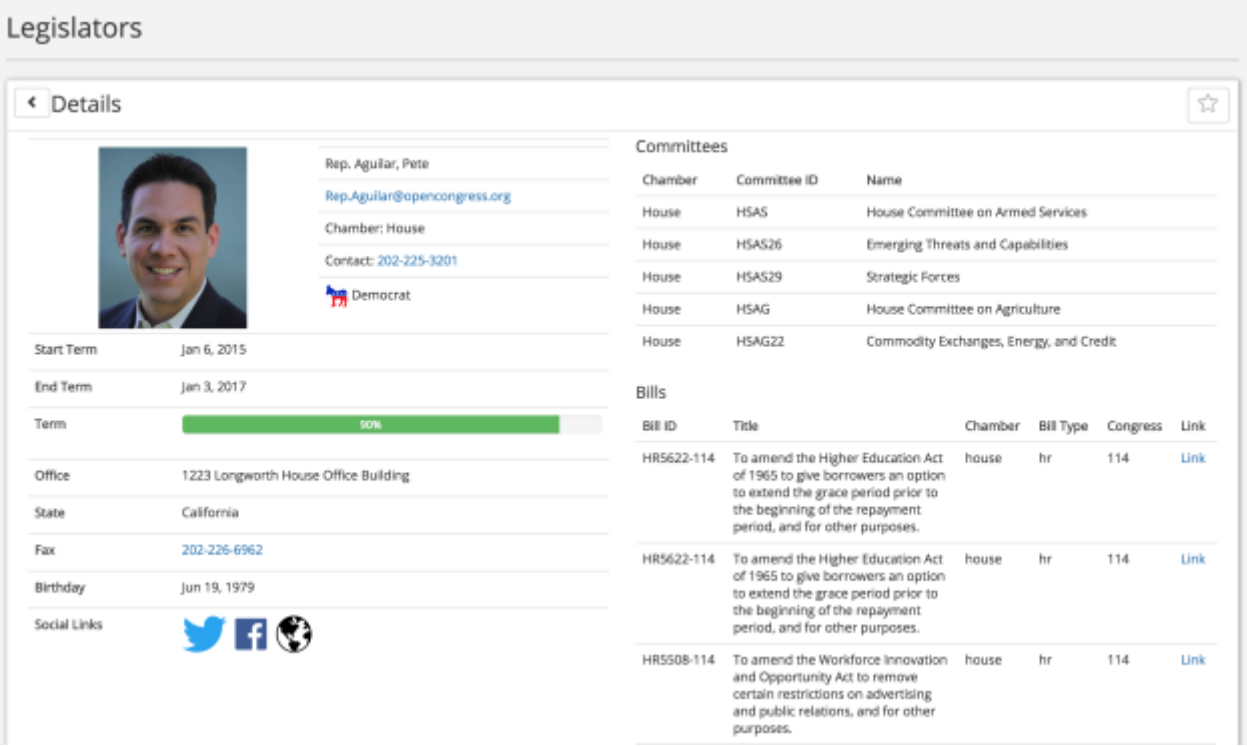


Figure 8 – Legislator Details

The detail page contains 3 parts:

- A table contains the personal information of the legislator
- A table contains the top 5 committees that the legislator belongs to
- A table contains the top 5 bills that the legislator sponsors

Following are the descriptions of these 3 tables.

Table Field	Description
Photo	Displays the photo of the legislator. You should use the base link: https://theunitedstates.io/images/congress/original and the "bioguide_id" to create the image URL.
Name	Displays the name of the legislator. The format should be *Title, Last Name, First Name*.
Email	The email address of the legislator. The email should be a link, click on the link to write a new email to this address.
Chamber	Displays the chamber of the legislator. Format should be *Chamber: chamber name*.
Contact	The phone number of the legislator

Party	The party of the legislator. Format should be: *Party image, Party name*.
Start Term	The start term of the legislator. Format of date should be: *MM-dd-yyyy*
End Term	The end term of the legislator. Format of date should be: *MM-dd-yyyy*.
Term	A progress bar with percentage showing the term progress. The progress percentage should be (now - start) / (end - start) * 100
Office	The office of the legislator
State	The state the legislator belongs to
Fax	The fax number
Birthday	The birthday of the legislator. Format of date should be: *MM-dd-yyyy*.
Social Links	The twitter link, Facebook link and website of the legislator.

Table 2: Legislator information table description

Table Field	**Description**
Chamber	Chamber of the committee
Committee ID	ID of the committee
Name	Name of the committee

Table 3: Committees information table description

Table Field	**Description**
Bill ID	ID of the Bill
Title	Title of the Bill
Chamber	Chamber of the Bill

Bill Type	Type of the Bill
Congress	Congress of the Bill
Link	Link to the pdf version of the Bill

Table 4: Bills information table description

3.3.1.4 Back Button

At the top left of the detail page, there is a back button. When clicked, the page should go back to the legislators table. See Figure 9.

Figure 9 – Back Button

3.3.1.5 Favorite Button

At the top right of the detail page, there is favorite button. When clicked, the legislator is added to the Favorites list. If clicked again, the legislator is removed from the Favorites list.

Figure 10 - Before and after click

- The favorite button should allow the user to add the legislator to the Favorites list and store it in the browser's local storage.
- If the stock is in the Favorites list, the button should have a yellow star, otherwise a white star (see Figure 10).

3.3.2 House Tab

When the House tab is selected, only the legislators of the US House of Representatives will be displayed in the table. See Figure 11.

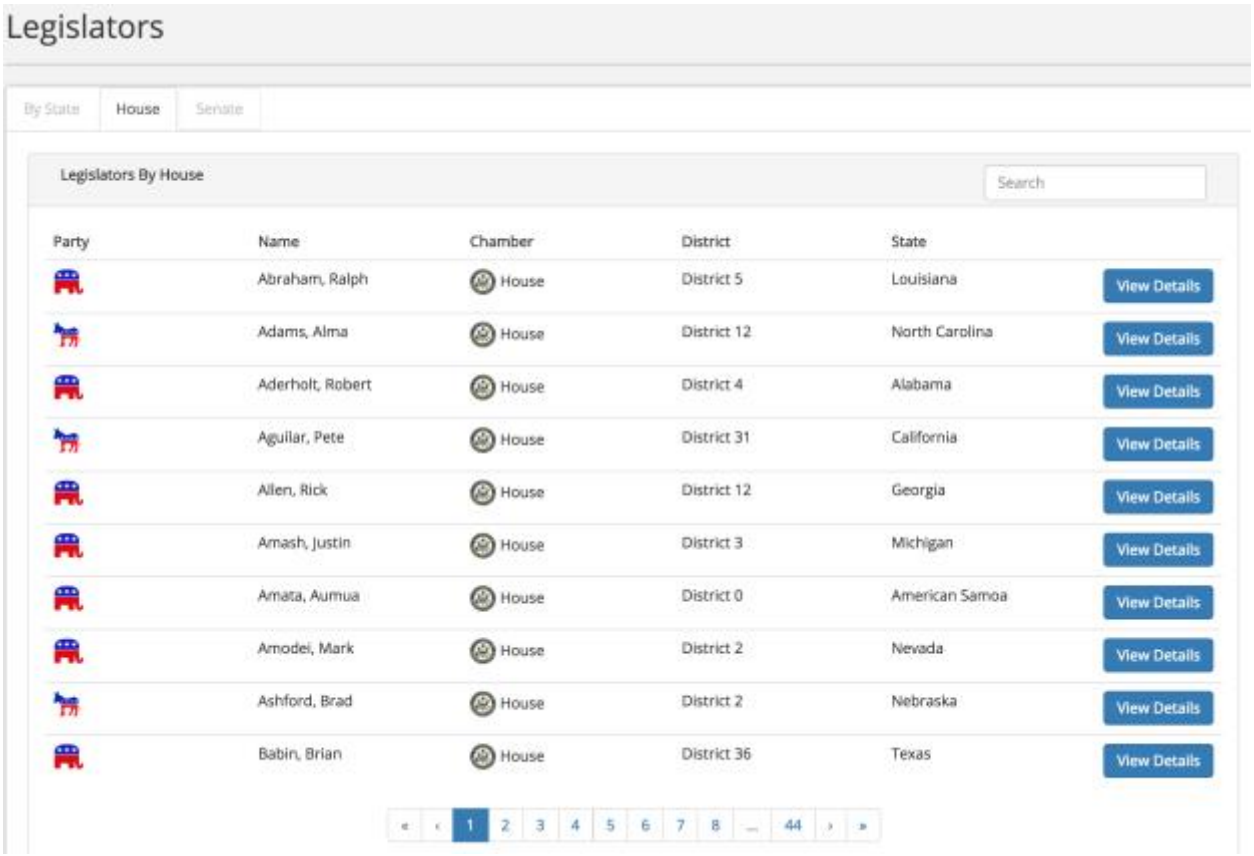


Figure 11 – House table

The table fields are exactly the same as for the By State Tab.

The difference is that, at the top right of the table, there is a Search input box instead of the state select dropdown. The user can enter any string to filter the legislators. The fuzzy search can optimally be performed by the AngularJS's dirPagination plugin. Details about this plugin will be enclosed in the Hints section.

Other details are the same as the By State tab, including the details page.

3.3.3 Senate Tab

When the Senate tab is selected, only the legislators of the US Senate will be displayed in the table. See Figure 12.

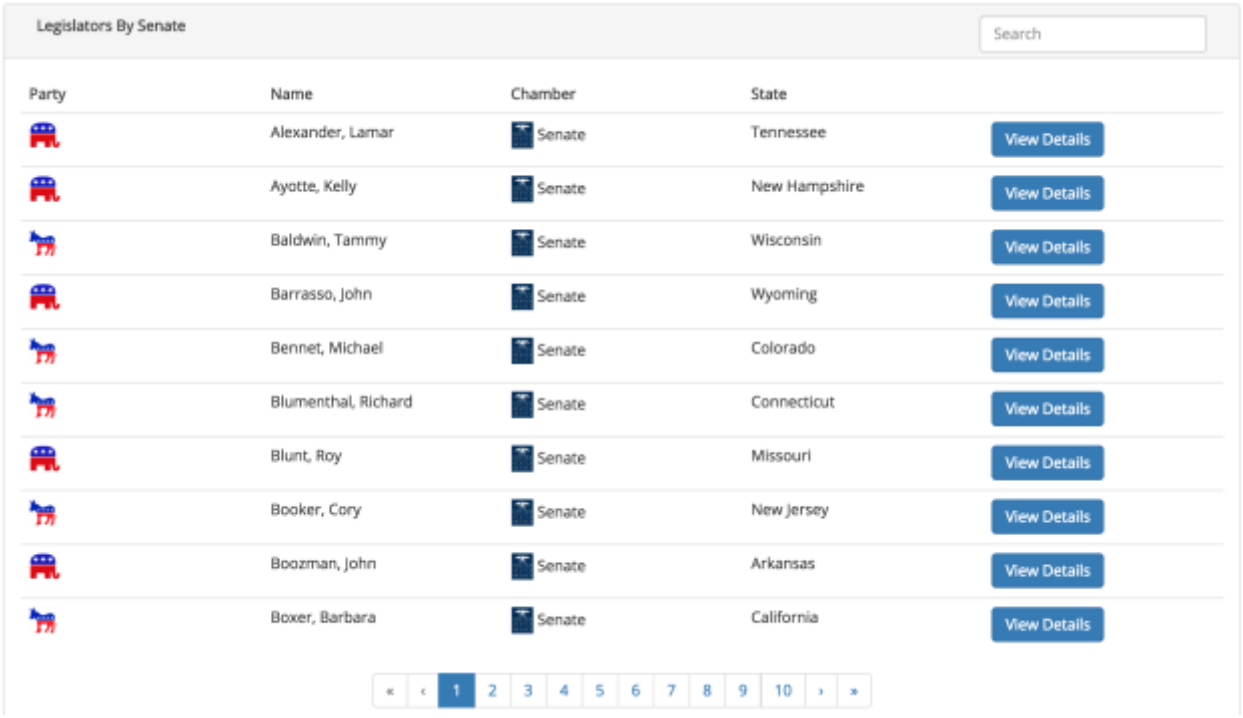


Figure 12 – Senate table

The table fields are exactly the same as the By State Tab, except that there is no "District" field. There is also a Search input box at the top right, performs exactly the same logic as for the House tab. Other details are the same as for the By State tab, including the Details page.

3.4 Bills

3.4.1 Design

You must replicate the table displayed in Figure 13 using a **Bootstrap table**.

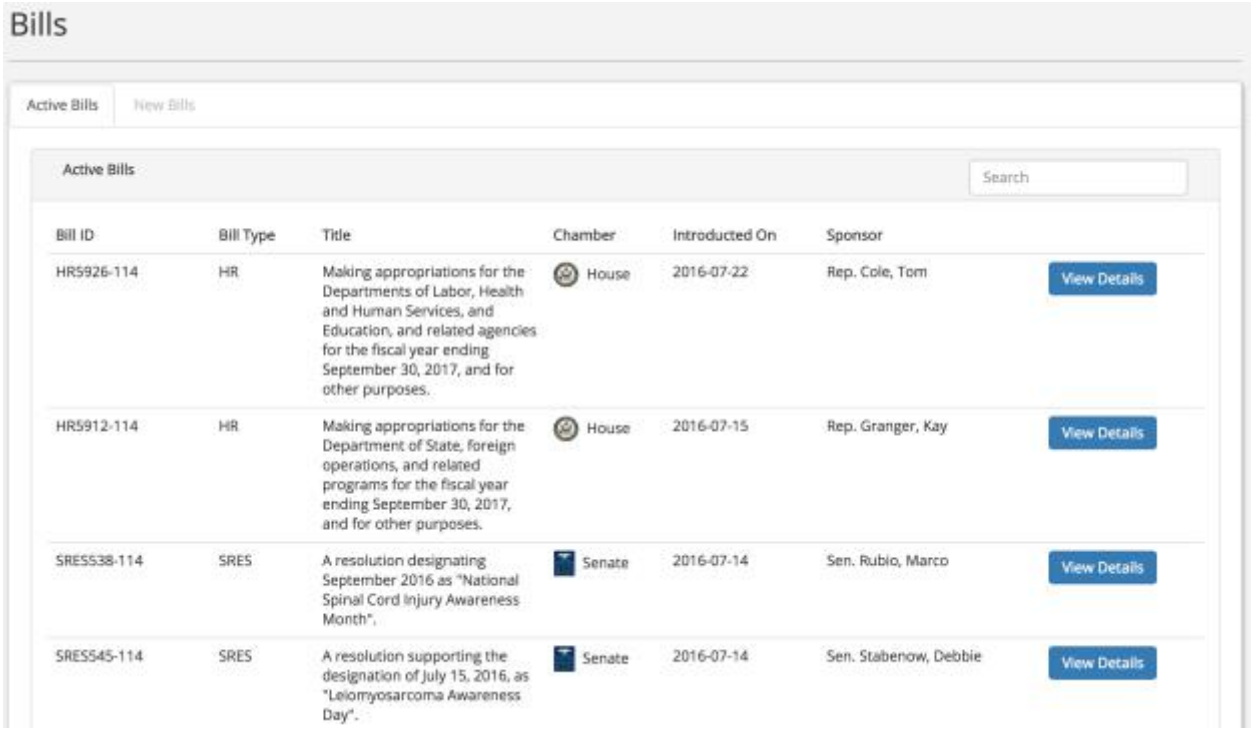


Figure 13 – Active Bills page

- The Bills page contains 2 tabs: "Active Bills" and "New Bills"
- Each tab contains a table with search input and pagination
- **You only need to load 50 bills in total for both Active and new bills to display in the tables**

3.4.2 Active Bills Tab

You must replicate the table displayed in Figure 13. The table fields are listed in Table 5.

Table Field	**Description**
Bill ID	ID of the Bill
Bill Type	Type of the Bill
Chamber	Chamber of the Bill. Start with the image of the Chamber. Then the name of the chamber.
Introduced	The date that the bill was introduced. Date format should be: *YYYY-MM-DD*
Sponsor	Sponsor of the Bill. Format should be: Title, Last Name, First Name

View Details	View Details button. When clicked, the page should navigate to the detail page of the Bill

Table 5: Active Bills table description

3.4.2.1 Pagination

The result table should be paginated. Details can be looked up in Section 3.3.1.2.

3.4.2.2 Search Filter

The result table should have a search input box at the top right of the page. Details can be looked up in Section 3.3.2

3.4.2.3 View Details

When the "View Details" button is clicked, the page should go to the details page of the current bill (as shown in Figure 14.

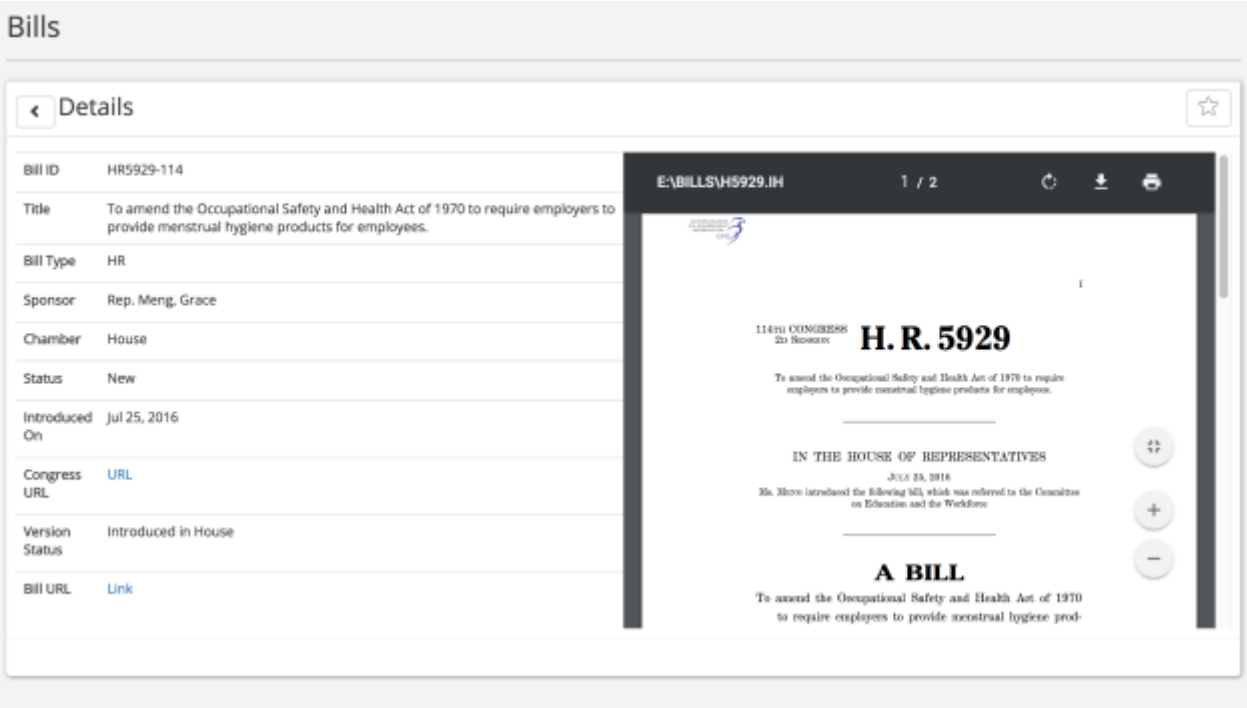


Figure 14 - Bill Detail page

The table fields are as shown in Table 6.

Table Field	Description
Bill ID	ID of the Bill

Bill Type	Type of the Bill
Sponsor	Sponsor of the Bill. Format should be: Title, Last Name, First Name
Chamber	Chamber name of the Bill.
Status	The status should be “Active” or “New”, depends on the “active” field in JSON response.
Introduced On	The date that the bill was introduced. Date format should be: *MM-DD-YYYY*.
Congress URL	The link of congress URL
Version Status	The “version name” field in “last version” field of JSON response
Bill URL	The link to the pdf version of the bill

Table 6: Bill Detail table description

Apart from the table, you should embed the PDF of this Bill in the page. Place it at the right of the details table. See Figure 14.

3.4.2.4 Back button

Similar as described in Section 3.3.1.4. When clicked, it should navigate back to the Bill tab table.

3.4.2.5 Favorite button

Similar as described in Section 3.3.1.5. If this Bill is in the Favorites list, the user should see a yellow button when entering this Bill’s detail page.

3.4.3 New Bills Tab

You should replicate the table as shown in Figure 15.

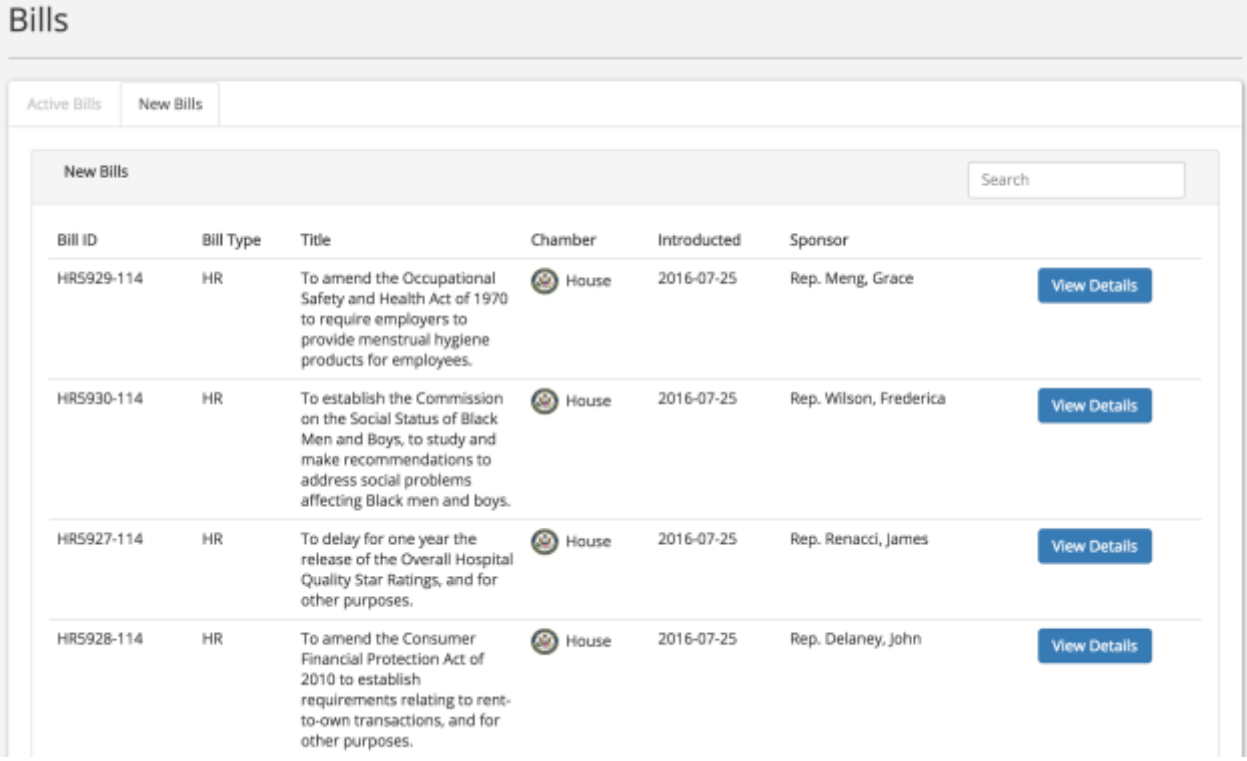


Figure 15 – New Bills page

The only difference between the "Active Bills" tab and the "New Bills" tab is that all of the bills in the New Bills tab have the 'active' field set to false in the JSON response.

Other details are exactly the same as for the Active Bills tab, including the Details page.

3.5 Committees

3.5.1 Design

You must replicate the table displayed in Figure 16 using a **Bootstrap table**.

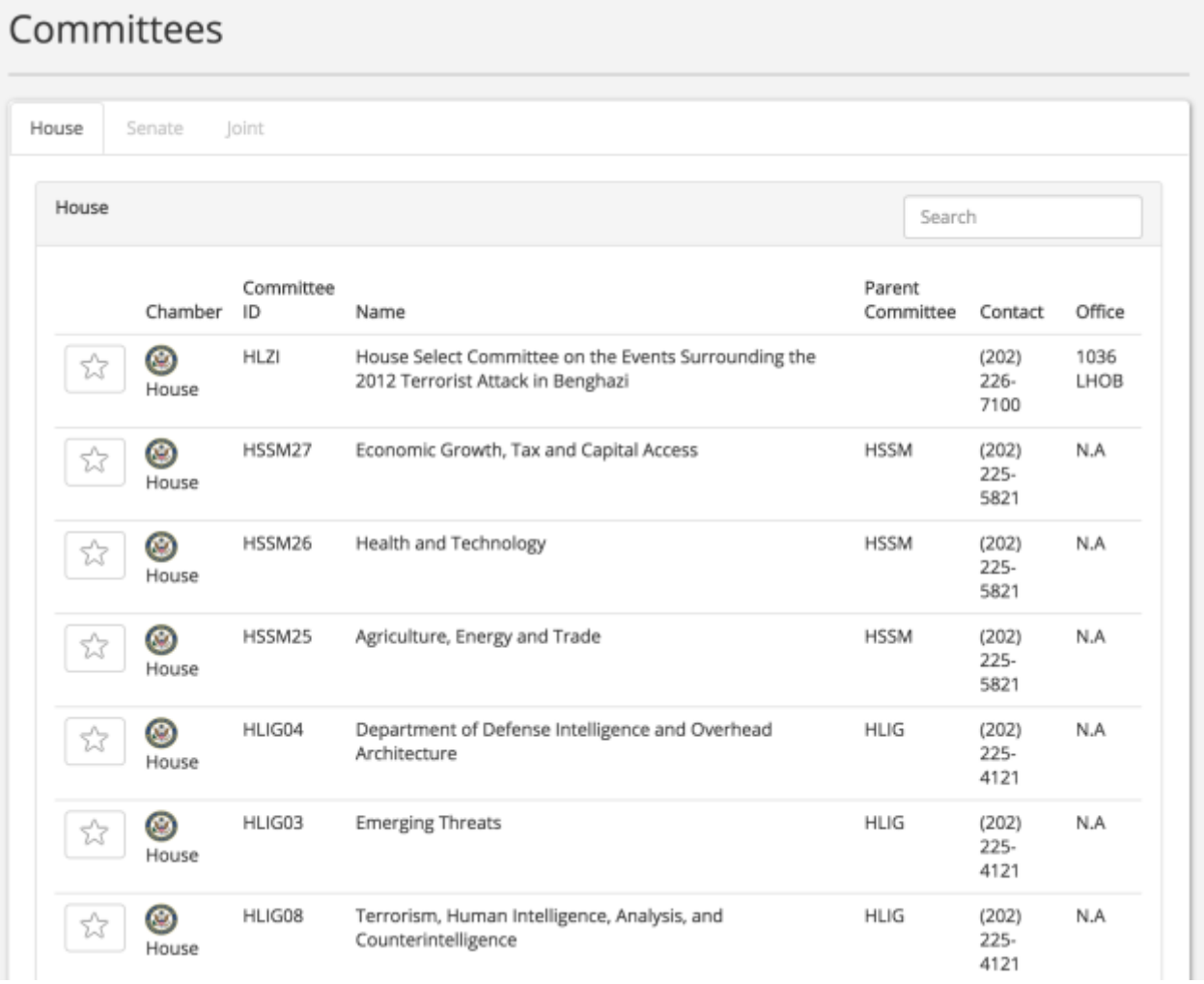


Figure 16 – Committee (House) page

- The Committees page contains 3 tabs: House, Senate and Joint
- Each tab contains a table with pagination and a search input filter box
- **You need to load all committees in total to display (approximately 228)**

3.5.2 House Tab

You must replicate the table displayed in Figure 16. The table fields are as shown in Table 7.

Table Field	**Description**
Favorite Button	The function is similar as described in Section 3.3.1.5
Chamber	Chamber of the Committee. Start with the image of the Chamber. Then the name of the chamber.

Committee ID	ID of the Committee
Name	Name of the Committee
Parent Committee	Parent Committee of the Committee. If not available, leave empty
Contact	Phone of the Committee
Office	Office of the Committee. If not available, display N.A

Table 7: House table description

3.5.2.1 Pagination

The result table should be paginated. Details can be looked up in Section 3.3.1.2.

3.5.2.2 Search Filter

The result table should have a search input box at the top right. Details can be looked up in Section 3.3.2.

3.5.3 Senate Tab

You must replicate the table displayed in Figure 17.

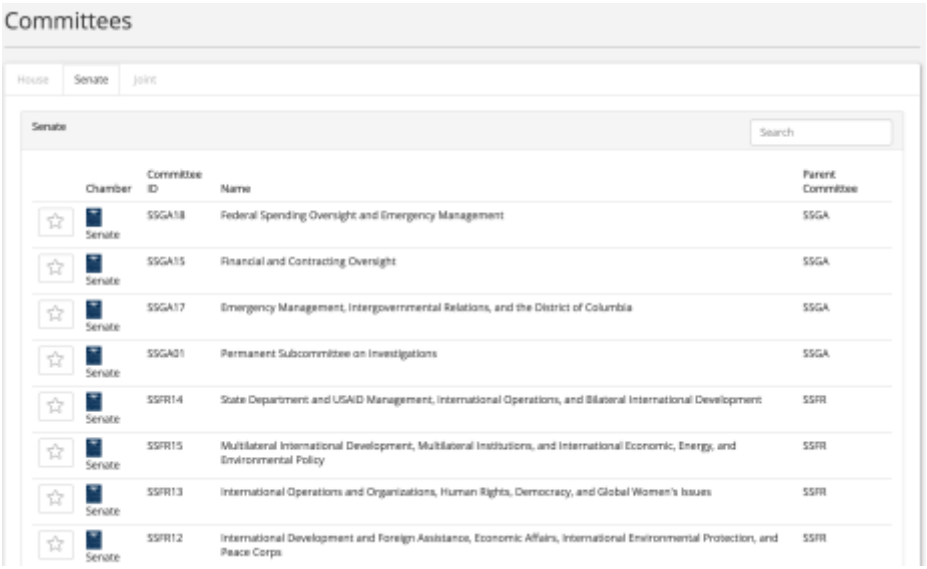


Figure 17 – Committees Senate table

The table fields are as shown in Table 8.

Table Field	Description
Favorite Button	The function is similar as described in Section 3.3.1.5
Chamber	Chamber of the Committee. Start with the image of the Chamber. Then the name of the chamber.
Committee ID	ID of the Committee
Name	Name of the Committee
Parent Committee	Parent Committee of the Committee. If not available, leave empty

Table 8: Senate table description

3.5.3.1 Pagination

The result table should be paginated. Details can be looked up in Section 3.3.1.2.

3.5.3.2 Search Filter

The result table should have a search input at top right. Details can be looked up in Section 3.3.2.

3.5.4 Joint Tab

You must replicate the table displayed in Figure 18.

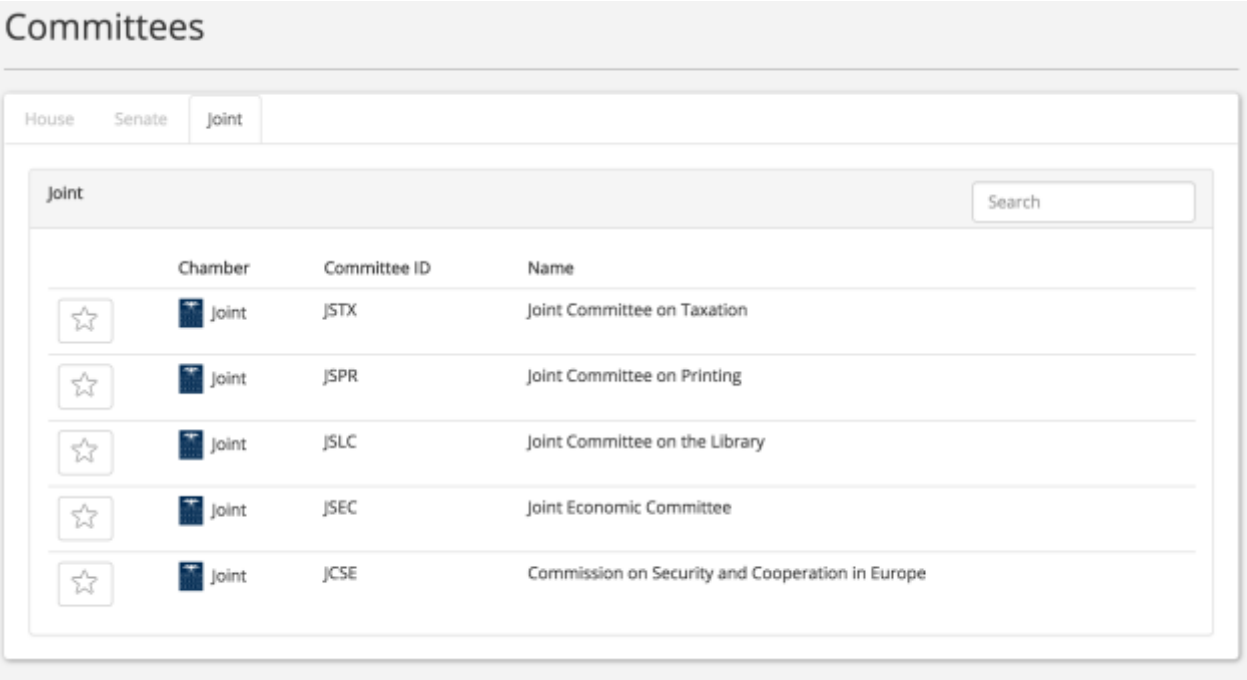


Figure 18 – Committees Joint table

The table fields are as shown in Table 9.

Table Field	Description
Favorite Button	The function is similar as described in Section 3.3.1.5
Chamber	Chamber of the Committee. Start with the image of the Chamber. Then the name of the chamber.
Committee ID	ID of the Committee
Name	Name of the Committee

Table 9: Senate table description

3.5.4.1 Pagination

The result table should be paginated. Details can be looked up in Section 3.3.1.2.

3.5.4.2 Search Filter

The result table should have a search input at top right. Details can be looked up in Section 3.3.2.

3.6 Favorites

3.6.1 Design

You must replicate the table displayed in Figure 19 using a **Bootstrap table**.

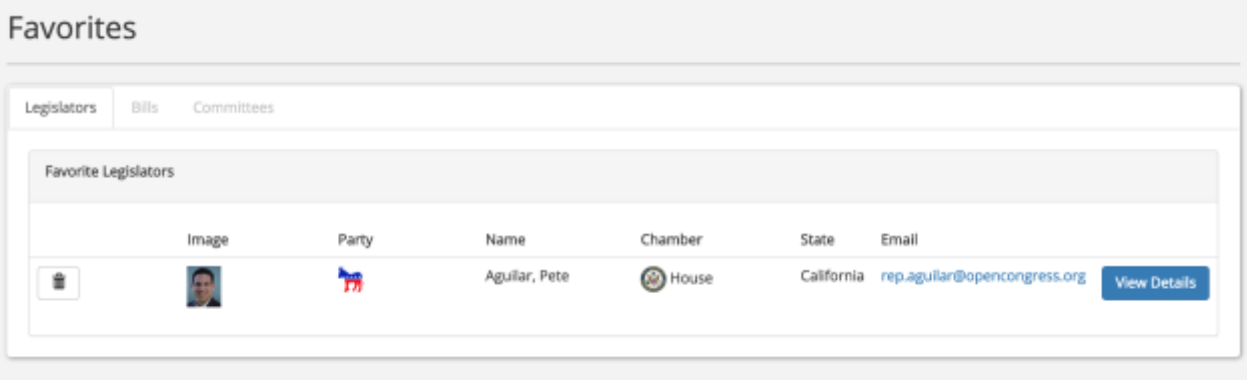


Figure 19 Favorites (Legislators) table

- The Favorites page contains 3 tabs: Legislators, Bills and Committees
- Each tab contains a table
- You don't need to implement pagination and search input field for the tables

3.6.2 Legislators Tab

You must replicate the table displayed in Figure 19.

The table fields are as shown in Table 10.

Table Field	Description
Delete Button	When clicked, the legislator will be deleted from the favorite list
Image	The photo of the legislator
Party	The party that the legislator belongs to
Name	Name of the legislator. Format should be Last Name, First Name
Chamber	Chamber that the legislator belongs to. Start with the image of the chamber. Followed by the name of the chamber
State	State that the legislator belongs to
Email	A link of the email address. When clicked, write a new email to the legislator.
View Details	When clicked, navigate to the details page of the legislator. (When back button clicked at the detail page, don't need to go back to Favorites page, just the same as described before)

Table 10: Legislators table description

3.6.3 Bills Tab

You must replicate the table displayed in Figure 20.

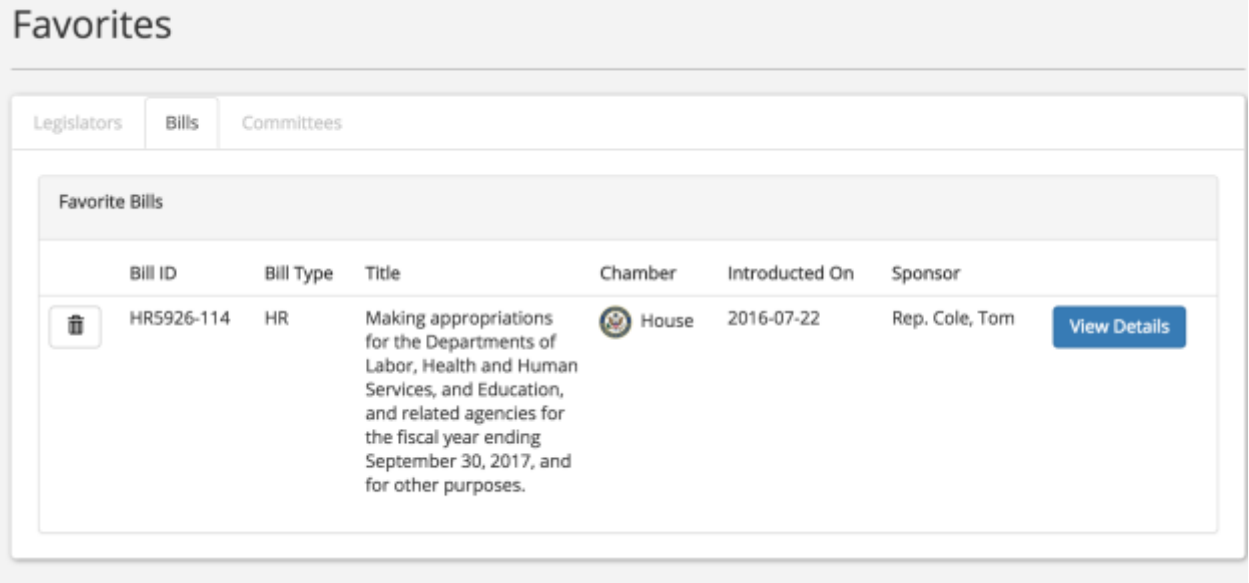


Figure 20 – Favorites Bills table

The table fields are as shown un Table 11.

Table Field	Description
Delete Button	When clicked, the Bill will be deleted from the favorite list
Bill ID	ID of the Bill
Bill Type	Type of the Bill
Title	Title of the Bill
Chamber	Chamber of the Bill. Start with the image of the Chamber. Then the name of the chamber.
Introduced On	The date that the bill was introduced. Date format should be: *YYYY-MM-DD*.
Sponsor	Sponsor of the Bill. Format should be: *Title, Last Name, First Name*.
View Details	When clicked, navigate to the details page of the Bill. (When back button clicked at the detail page, don't need to go back to Favorites page, just the same as described before)

Table 11: Bills table description

3.6.4 Committees Tab

You must replicate the table displayed in Figure 21.

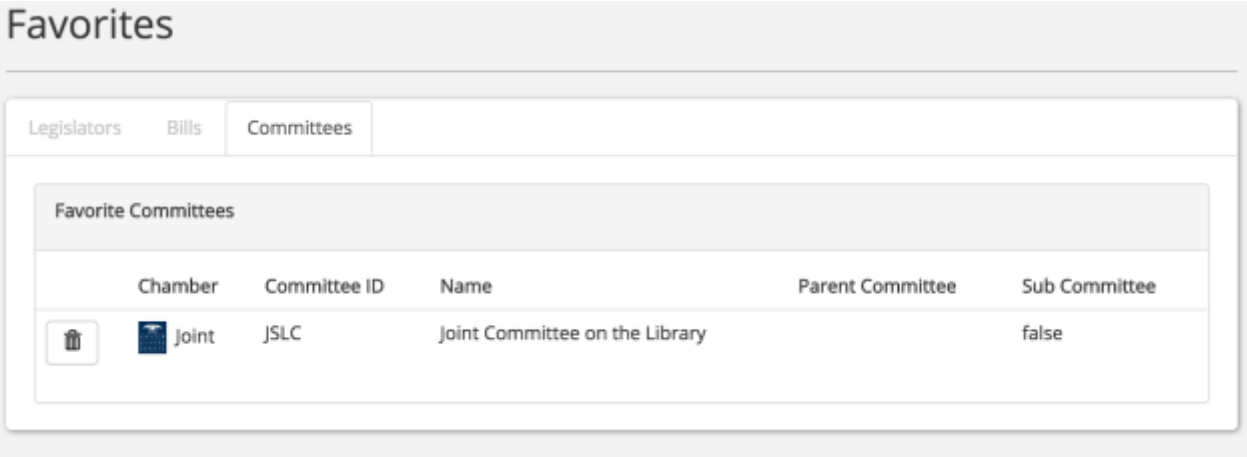


Figure 21 – Favorites Committees table

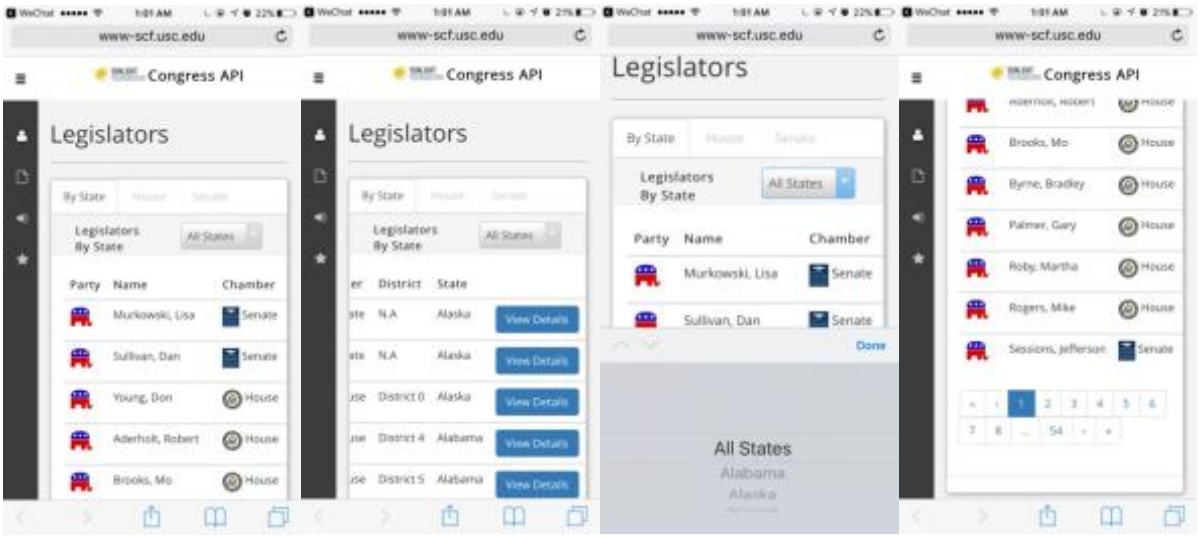
The table fields are as shown in Table 12.

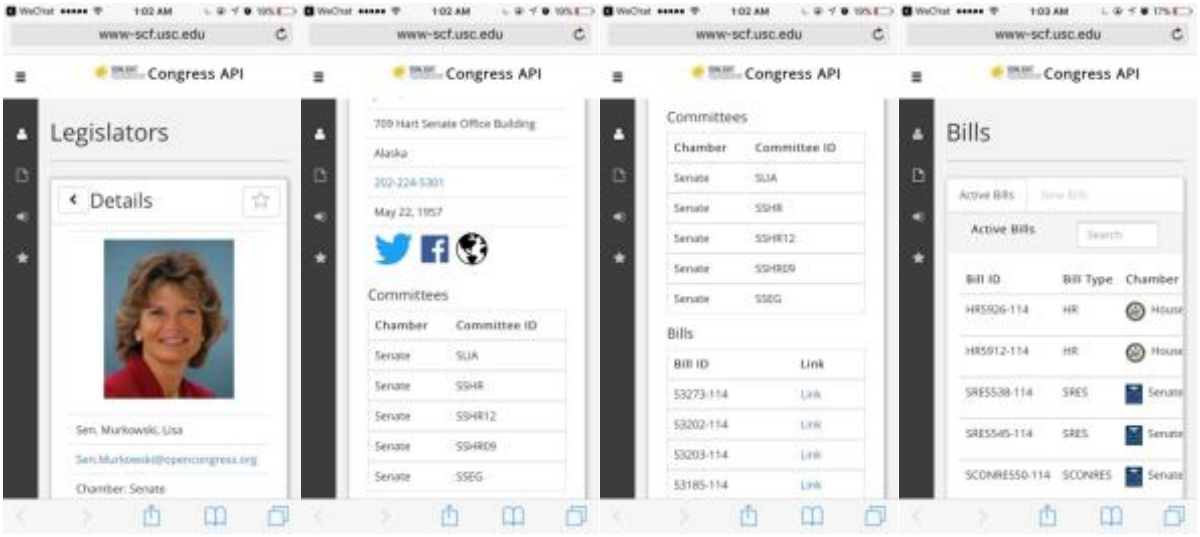
Table Field	Description
Delete Button	When clicked, the Committee will be deleted from the favorite list
Chamber	Chamber of the Committee. Start with the image of the Chamber. Then the name of the chamber.
Committee ID	ID of the Committee
Name	Name of the Committee
Parent Committee	Parent Committee of the Committee. If not available, leave empty
Sub Committee	Whether it's sub committee or not. True or false.

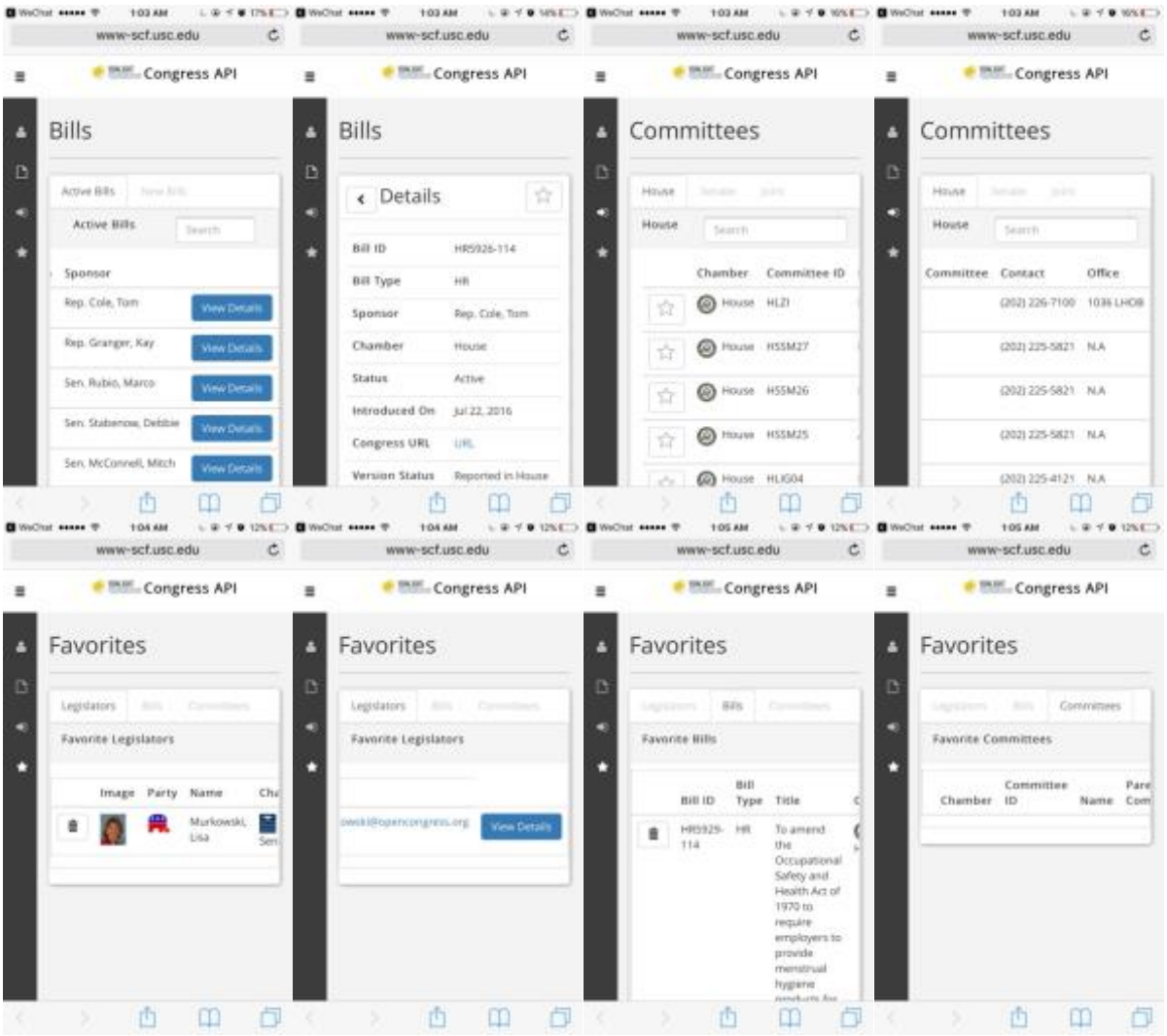
Table 12: Committees table description

3.6 Responsive Behavior

The following are snapshots of the major screens taken on an iPhone.


www-scf.usc.edu
Congress API
Legislators
By State
Legislators By State
All States
Party Name Chamber
Murkowski, Lisa Senate
Sullivan, Dan Senate
Young, Don House
Aderholt, Robert House
Brooks, Mo House
District State
N.A Alaska
View Details
District 0 Alaska
District 4 Alabama
District 5 Alabama
Done
All States
Alabama
Alaska
Byrne, Bradley House
Palmer, Gary House
Roby, Martha House
Rogers, Mike House
Sessions, Jefferson Senate


www-scf.usc.edu
Congress API
Legislators
Details
Sen. Murkowski, Lisa
Chamber: Senate
709 Hart Senate Office Building
Alaska
202-224-5301
May 22, 1957
Committees
Chamber Committee ID
Senate SLIA
Senate SSHR
Senate SSHR12
Senate SSHR09
Senate SSEG
Bills
Bill ID Link
S3273-114 Link
S3202-114 Link
S3203-114 Link
S3185-114 Link
Active Bills
Bill ID Bill Type Chamber
HR5926-114 HR House
HR5912-114 HR House
SRES538-114 SRES Senate
SRES545-114 SRES Senate
SCONRES50-114 SCONRES Senate



Some requirements in the mobile view are listed here:

- The navigation bar should only display icons.
- Several tables in the legislator's detail page should be stacked.
- Each table content can be scrolled horizontally.
- In the legislator's detail page, the committees table has only "Chamber" and "Committee ID" fields.
- In the legislator's detail page, he bills table has only "Bill ID" and "Link" fields
- In the Bill's detail page, the PDF preview should be hidden.

4. API Documentation

The same APIs that were used for homework 6 can be used for homework 8 as well.

4.1 Legislators API

The API to query all of the legislators is:

https://congress.api.sunlightfoundation.com/legislators?apikey=**YOUR_API_KEY_HERE&per_page=all**

The response is as follows:


```
{
  - results: [
      - {
            bioguide_id: "D000626",
            birthday: "1970-03-01",
            chamber: "house",
            contact_form: null,
            crp_id: "N00038767",
            district: 8,
            fax: null,
          - fec_ids: [
                "H6OH08315"
            ],
            first_name: "Warren",
            gender: "M",
            govtrack_id: "412675",
            in_office: true,
            last_name: "Davidson",
            leadership_role: null,
            middle_name: null,
            name_suffix: null,
            nickname: null,
            oc_email: null,
            ocd_id: "ocd-division/country:us/state:oh/cd:8",
            office: "1011 Longworth House Office Building",
            party: "R",
            phone: "202-225-6205",
            state: "OH",
            state_name: "Ohio",
            term_end: "2017-01-03",
            term_start: "2016-06-09",
            thomas_id: "02296",
            title: "Rep",
            votesmart_id: 166760,
            website: null
        },
      - {
            bioguide_id: "L000585",
            birthday: "1968-07-05",
```


Figure 22

4.2 Bills API

The API to query the first page of 50 Bills is:

https://congress.api.sunlightfoundation.com/bills?apikey=**YOUR_API_KEY_HERE&per_page=50**

The response is as follows:

```
{
 - results: [
    - {
        bill_id: "hr5929-114",
        bill_type: "hr",
        chamber: "house",
      - committee_ids: [
            "HSED"
        ],
        congress: 114,
        cosponsors_count: 0,
        enacted_as: null,
      - history: {
            active: false,
            awaiting_signature: false,
            enacted: false,
            vetoed: false
        },
        introduced_on: "2016-07-25",
        last_action_at: "2016-07-25",
      - last_version: {
            version_code: "ih",
            issued_on: "2016-07-25",
            version_name: "Introduced in House",
            bill_version_id: "hr5929-114-ih",
          - urls: {
                html: "https://www.gpo.gov/fdsys/pkg/BILLS-114
                pdf: "https://www.gpo.gov/fdsys/pkg/BILLS-114h
                xml: "https://www.gpo.gov/fdsys/pkg/BILLS-114h
            },
            pages: 2
        },
        last_version_on: "2016-07-25",
        last_vote_at: null,
        number: 5929,
        official_title: "To amend the Occupational Safety and
        for employees.",
        popular_title: null,
      - related_bill_ids: [
            "hr5915-114"
        ],
        short_title: "Menstrual Products for Employees Act",
      - sponsor: {
            first_name: "Grace",
            last_name: "Meng",
            middle_name: null,
            name_suffix: null,
            nickname: null,
```


Figure 23

4.3 Committees API

The API to query the first page of all committees is:

`https://congress.api.sunlightfoundation.com/committees?apikey=YOUR_API_KEY_HERE&per_page=all`

The response is as follows:

```
{
 - results: [
    - {
        chamber: "senate",
        committee_id: "SSGA18",
        name: "Federal Spending Oversight and Emergency Management",
        parent_committee_id: "SSGA",
        subcommittee: true
      },
    - {
        chamber: "senate",
        committee_id: "SSGA15",
        name: "Financial and Contracting Oversight",
        parent_committee_id: "SSGA",
        subcommittee: true
      },
    - {
        chamber: "senate",
        committee_id: "SSGA17",
        name: "Emergency Management, Intergovernmental Relations, and the D:
        parent_committee_id: "SSGA",
        subcommittee: true
      },
    - {
        chamber: "senate",
        committee_id: "SSGA01",
        name: "Permanent Subcommittee on Investigations",
        parent_committee_id: "SSGA",
        subcommittee: true
      },
    - {
        chamber: "senate",
        committee_id: "SSFR14",
        name: "State Department and USAID Management, International Operati
        parent_committee_id: "SSFR",
        subcommittee: true
      },
```


Figure 24

5. Hints

5.1 Images

The images for this homework have been provide in http://cs-server.usc.edu:45678/hw/hw8/images/

5.2 Get started with the Bootstrap Library

To know how to get started with Bootstrap, please refer to the page at http://getbootstrap.com/getting-started/. You need to import the necessary CSS file and JS file provided by Bootstrap.

5.3 Bootstrap UI Components

Bootstrap provides a complete mechanism to make Web pages responsive to different mobile devices. In this exercise you will get hands-on experience with responsive design using the Bootstrap Grid System.

At a minimum, you will need to use Bootstrap Form, Tab, Wells, Carousel and Glyphicons to implement the required functionality. Information about these components can be found here:

Bootstrap Form http://getbootstrap.com/css/ - forms

Bootstrap Tabs	http://getbootstrap.com/javascript/ - tabs
Bootstrap Wells	http://getbootstrap.com/components/ - wells
Bootstrap Carousel	http://getbootstrap.com/javascript/ - carousel
Bootstrap Glyphicons	http://getbootstrap.com/components/ - glyphicons

5.4 Google App Engine/Amazon Web Services

You should use the domain name of the Google App Engine/Amazon Web Services you created in HW#7 to make the request. For example, if you're GAE/AWS server domain is called example.appspot.com or example.elasticbeanstalk.com, and the user performs a GET request with operation name="legislator", then a query of the following type will be generated:

(GAE) - http://example.appspot.com/?operation=legislators

(AWS) - http://example.elasticbeanstalk.com/?operation=legislators

5.5 AJAX call

You can send the request to the PHP script by passing the URL to `$.ajax()`. 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 PHP code is running on Google GAE or AWS (please refer to the grading guideline for details).

The is the JQuery AJAX call:

```
$.ajax({
  url: 'URL in GAE',
  //parameter list
  data: {symbol:"AAPL"},
  type: 'GET',
  success: function(response,status,xhr){
      //parse the output
  },
  error: function(xhr, status, error){
      //parse the error
  }
});
```

Figure 25 – JQuery AJAX call

The is the AngularJS AJAX call:

```
// Simple GET request example:
$http({
  method: 'GET',
  url: '/someUrl'
}).then(function successCallback(response) {
    // this callback will be called asynchronously
    // when the response is available
  }, function errorCallback(response) {
    // called asynchronously if an error occurs
    // or server returns response with an error status.
  });
```

Figure 26 – AngularJS AJAX call

5.6. 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 method, getItem() and setItem(), that can be used for this purpose. The local storage could only store strings. So you need to convert the data to strings before storing it in the local storage. See:

https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

http://www.w3schools.com/html/html5_webstorage.asp

5.7 Get Started with AngularJS

AngularJS extends HTML with new attributes.

AngularJS is perfect for Single Page Applications (SPAs).

AngularJS is easy to learn.

To get started with AngularJS. The W3SCHOOL site has easy tutorials:

http://www.w3schools.com/angular/

Or, you can follow the tutorials at the official website:

https://docs.angularjs.org/tutorial

5.8 AngularJS Pagination and Filters

The Pagination Directive of AngularJS is truly plug-n-play - no need to do any set-up or logic in your controller. Just add an attribute, drop in your navigation wherever you like, and boom - instant, full-featured pagination.

The Github page is at:

https://github.com/michaelbromley/angularUtils/tree/master/src/directives/pagination

The Demo page is at:

http://plnkr.co/edit/Wtkv71LIqUR4OhzhgpqL?p=preview

5.9 AngularJS Progress Bar

The AngularJS Progress Bar Directive allows you to auto compute the progress with a simple set up.

The documentation is located at:

https://angular-ui.github.io/bootstrap/ - /progressbar

5.10 External Libraries

This is the list of external libraries that you may also want to use:

- JQuery – https://code.jquery.com/
- Moment JS -http://momentjs.com/ for time conversion

6. FAQ's

Q1. Which server should I use, where would server be hosted?

You can use either host it in Google App Engine or Amazon Web Services. While grading we just need to see the data

Q2. Can I use any other way to store data?

You are allowed only to use browsers local storage. No other way of storing data like cookies, session storage is allowed.

Q3. How long should the local storage be persistent?

The local storage data should be persistent until the browser is closed. Even if the page is refreshed, the local storage data should not be deleted.

Q4. Is it required to use Bootstrap Carousel?

Yes. It is mandatory to use bootstrap carousel.

Q5. Facing issues with Cross Origin Request?

There are several ways to avoid cross origin issues. You can use CORS. If you like to use $.ajax() you can use datatype: jsonp with callback function appended with the URL.

Q6. If there is a scroll issue in iPhone or iPad?

Use the following style to the parent or body tag

```
-webkit-overflow-scrolling: touch;
```

7. Files to Submit

In your course homework page, you should update the HW8 link to refer to your new initial web page for this exercise. You will likely create 4 files: an HTML, a CSS, a JavaScript and a "pure" PHP file. Additionally, you need to provide a link to your homework page, and a link to the GAE/AWS server where the AJAX call is made with a sample parameter value. Also, use 'submit' to provide your files electronically to the csci571 account so that they can be graded and compared to all other students' code using MOSS. You may have to copy the PHP file hosted in the cloud to cs-server, to submit it with the rest of your files.

****IMPORTANT**:**

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.