7-22-2015

YeaNay: An Open Source Tool to Rate the Votes of Members of the United States House of Representatives and Senate

Eric Venlet

Robert Adams
Grand Valley State University, adamsr@gvsu.edu

Open Journal of Political Science, 2015, 5, 277-283
Published Online July 2015 in SciRes. http://www.scirp.org/journal/ojps
http://dx.doi.org/10.4236/ojps.2015.54029

YeaNay: An Open Source Tool to Rate the Votes of Members of the United States House of Representatives and Senate

Eric Venlet, D. Robert Adams

School of Computing and Information Systems, Grand Valley State University, Allendale, USA
Email: eric.venlet@gmail.com, adams@cis.gvsu.edu

Received 5 June 2015; accepted 19 July 2015; published 22 July 2015

Abstract

Government transparency is typically regarded as the most viable way to strengthen its accountability to the public (Shkabatur, 2012). Even on the international stage, the right to access government information is regarded as fundamental to democracy (Bertot, Jaeger, & Grimes, 2011). In order to improve transparency, the US government made data, like bills and votes, available online (Brito, 2008b). One popular way to organize the data available to the public is through the creation of voter guides. The method an organization used for developing a voter guide was analyzed for this project. In response to the method, a web application (YeaNay) was developed to take the largely manual process and make a highly automated solution. YeaNay utilizes HTML, CSS, and JavaScript to build the user interface and ColdFusion and PL/SQL to query the data necessary for the development of a voter guide. The data are queried either from the database or from Congress API v3 (provided by the Sunlight Foundation). One user, with minimal training, is able to use YeaNay to find and score legislation within minutes for use in a voter guide. YeaNay focuses the firehose of congressional information that is now available and presents it in a manageable and usable environment.

Keywords

Voter Guide, Transparency, Voting History, House of Representatives, Senate

1. Introduction

Government transparency is typically regarded as the most viable way to strengthen its accountability to the public. As technology and the availability of information have increased, so it is able to overcome past hurdles

How to cite this paper: Venlet, E., & Adams, D. R. (2015). YeaNay: An Open Source Tool to Rate the Votes of Members of the United States House of Representatives and Senate. *Open Journal of Political Science, 5,* 277-283.
http://dx.doi.org/10.4236/ojps.2015.54029

to transparency (Shkabatur, 2012). In fact, government transparency was even part of the platform of the Obama administration. On his first day in office, President Obama signed an Open Government Memorandum in order to provide "an unprecedented level of openness in government" (Shkabatur, 2012).

In response to the memorandum, Congress passed legislation to further expose its own activities. The underlying purpose of the legislation was to "restore accountability, honesty, and openness at all levels of government (Brito, 2008a)". Even on the international stage, the right to access government information is regarded as fundamental to democracy. It is necessary in making informed decisions, improving trust in the government, and providing information to the public (Bertot, Jaeger, & Grimes, 2011). For a fully functional democracy, the public should have the ability to see what the government is doing and how it is spending money. In addition, transparency further aids to expose corruption, bolster public confidence in government and promote greater fiscal responsibility (Arizona Republic, 2010).

2. Challenges for Improving Government Transparency

In order for transparency to come full cycle, there needs to be a dialogue between the government and the citizens. For their part, governments can provide its citizens with information about its activities. Citizens then need to become more knowledgeable about the available data, and then the dialogue between them and the government can become more collaborative and meaningful (Chun et al., 2010).

The challenge becomes how to make the data "fit for use" for people outside the government (Dawes & Helbig, 2014). Despite increased availability, accessing and interpreting the information can still be difficult due to organization, structure, search, metadata, and other factors (Bertot, Jaeger, & McClure, 2008). For example, simply searching for information on www.usa.gov leads to unrefined and disorganized results. This serves to illustrate that the information was not designed with the average citizen in mind (Jaeger & Bertot, 2010). To further demonstrate the point, information regarding bills and votes on bills is available online. However, when looking up an individual legislator's vote (i.e. Justin Amash), a good starting point is amash.house.gov. After a brief description regarding Representative Amash, there is a redirection to view his votes that brings the user to clerk.house.gov. Votes can then be found through a listing of roll calls which are displayed in a table. If more information regarding the bill is required, further redirection brings the user to thomas.loc.gov. The challenge of turning raw government data into useful citizen-friendly information is apparent.

Voter Guides

If governments continue to make data, like bills and votes, available online citizens will be able to use the Internet to inform themselves and shed light on what the government is doing (Brito, 2008b, *Columbia Science and Technology Law Review*). Many organizations such as the Family Research Council, the American Civil Liberties Union, the National Rifle Association, and the Center for American Progress desire to improve the accountability of congress. One way they do this is by providing voter guides to the public. These voter guide rate legislators based on their voting histories. If a legislator (representative or senator) votes according to the organization's opinion, the legislator will be rated with a "+", otherwise they are rated with a "−". At the end of a congressional session, each legislator is then given a rating (0% - 100%) for how closely they voted to the organization's opinion. This provides citizens with an easy way to determine the voting tendencies of the legislators that represent them. The difficulty for these organizations is in gathering the data needed to create a voter guide, and then organizing that data into a useful format.

3. YeaNay

YeaNay is an open source web application for organizations to create voter guides for their members. YeaNay provides a fast and understandable way to gather and display information regarding the voting habits of legislators on specific topics. It does this by querying data sources regarding representatives and senators as well as the bills, nominations, resolutions that they vote on. YeaNay then analyzes this information to automatically score the votes based on whether or not the topic being voted on was supported by the organization. In addition, YeaNay illustrates techniques that other applications and organizations could use to gather and represent data from government sources.

3.1. Implementation

YeaNay is organized as a rather typical client/server web application. The client (running in a browser) is responsible for providing the user interface, as well as performing simple form validation. It is written in Javascript. The server is responsible for managing YeaNay's data, as well as providing the application logic of scoring votes. The server logic is written in ColdFusion, while the data is managed by Oracle. In addition, PL/SQL is used to perform functions strictly on the database level.

When a user begins using YeaNay, the flow of a request is handled in the following way (**Figure 1(a)**):

- A client makes a request to YeaNay;
- Using ColdFusion, the server sets session variables and authenticates the user on the first request;
- The server makes queries to a remote Oracle database to verify user credentials and retrieve information;
- The scoring of votes and loading of legislators into the database occurs independently through stored PL/SQL procedures that make calls to the API (**Figure 1(b)**);
- If the data required does not exist in the database, it is requested from an external API;
- The server then builds the page with the resulting data and presents it to the client;
- Javascript is used on the client-side to validate form data and manipulate HTML elements (e.g., display/ hide).

JavaScript is implemented for client-side operations. The primary use of JavaScript is to provide form validation, manipulate html document elements, and automate the setting of form variables based off of information returned by the server.

ColdFusion was selected for the server side language. The organization for whom YeaNay was originally developed utilizes ColdFusion and that was the primary reason for its usage. ColdFusion serves three main purposes in YeaNay. The first is to interact with the database. Secondly, ColdFusion is used to make the vast majority of calls to the API if the database does not contain the necessary data. Finally, YeaNay authenticates users to the database and stores the user credentials on the session level.

The primary use of the database is storing a voter guide (a "scorecard" in the vernacular of the organization for whom YeaNea was developed). However, several PL/SQL procedures were written to supplement the tool, as described below. Four tables within an Oracle (10g) database are used to store the data for the application. The four tables are scorecard, scorecard_bill, scorecard_legislator, and scorecard_legislator_vote (**Figure 2**).

- Scorecard represents a voter guide and holds a unique identifier and description for a scorecard relating to a specific congressional session;
- The bills are stored in the scorecard_bill table by scorecard_id and bill_id, along with accompanying data like bill_description and status;
- Scorecard_legislator stores a list of senators and representatives for each scored session as identified by sco

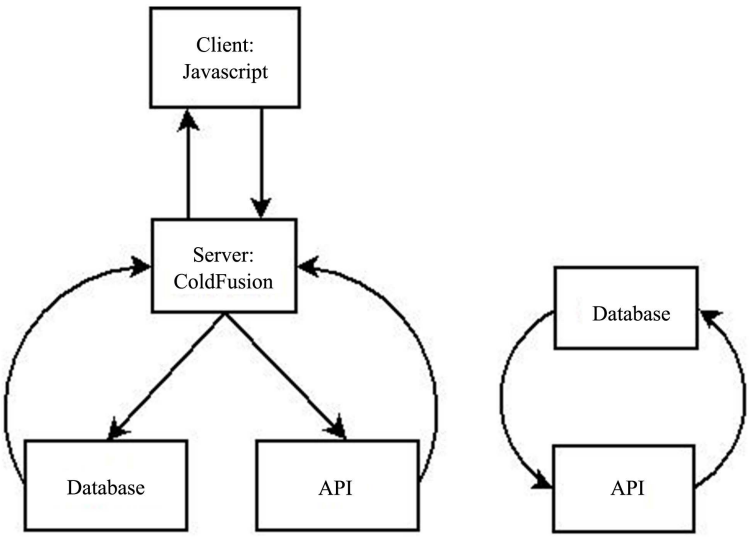

Figure 1. YeaNay request overview.

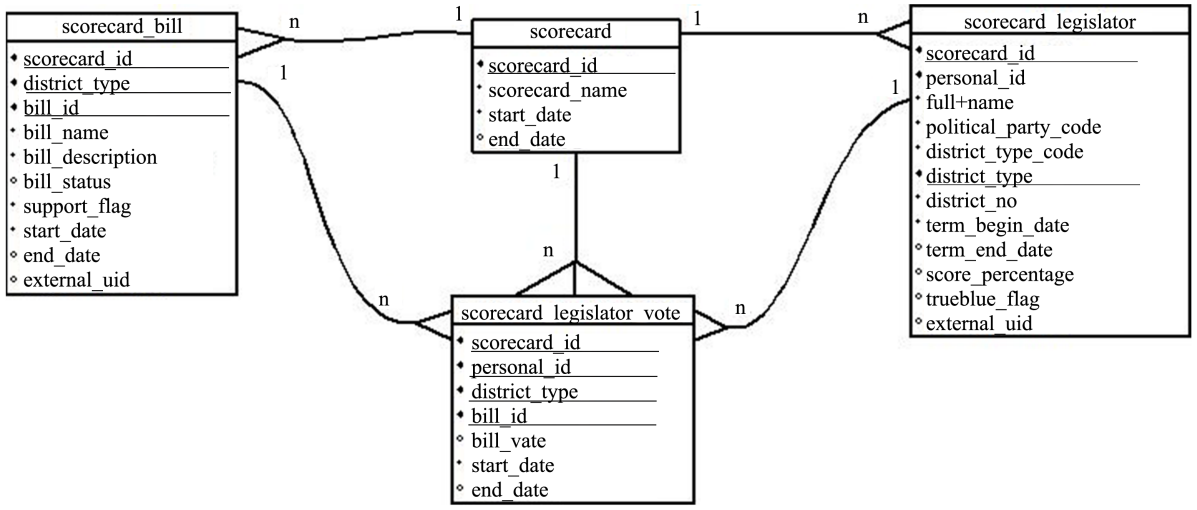

Figure 2. Scorecard TABLE SChema.

recard_id;

- Finally scorecard_legislator_vote is used to store the legislators' votes on a specific bill_id in a specific congressional session.

3.2. APIs Used

YeaNay uses an API that is provided by the Sunlight Foundation. The Sunlight Foundation describes itself as "a national, nonpartisan, nonprofit organization that looks to make the government and politics more accountable and more transparent" (http://sunlightfoundation.com/about/, retrieved on 4/4/15). The API offers access to real-time information regarding legislators, bills, and votes in both the House of Representatives and the Senate. The Sunlight Foundation has a variety of APIs available for access, Congress API v3 is the API utilized by YeaNay.

Congress API v3 offers a variety of methods for retrieving congressional information. The methods that YeaNay utilizes are /legislators, /bills, /bills/search, /votes, and /nominations (**Table 1**).

The data can be retrieved from the API in either JSON or XML format. The web server used for YeaNay (ColdFusion MX7) does not have methods to parse JSON (native format of the external API) data and therefore receives the data in XML format. The database has JSON functions and therefore, requests made by the database receive and parse the data in JSON. Each of the available methods returns a large amount of data, not all of which is used. For example, /bills by default returns a large quantity of information that is largely unnecessary when developing a voter guide (**Figure 3**). For YeaNay, only specific fields were requested when making the calls to the various methods (**Table 1**).

3.3. Scoring a Bill

When the application first starts, the database is empty, so the first task is to collect a list of legislators. A stored procedure in the database is used to query an external API (described above) for the most recent list of legislators. If the database has already been populated, the list of legislators is then checked against the current scorecard's list of legislators. If any new legislators exist, they are added. Legislators who are no longer in office remain associated with the current scorecard in order to keep their history of votes. Once the list of legislators has been populated into the database, it is then possible to begin scoring votes.

The user has the ability to search for bills based on bill_id, roll_id, or keyword. When searching by bill_id or roll_id, the specific bill is displayed. If the keyword search is utilized, a list of bills that have been tagged with the associated keyword are displayed. Once a bill is selected, the bill's information is displayed and the user has the option to score it and select whether or not the bill is supported by the organization.

When the user submits the bill to be scored, the DBMS schedules a stored procedure to be executed. An asynchronous stored procedure is used to avoid long load times caused by the time required to score a bill. The

```
{
  "results":[
     {
        "bill_id":"s895-114",
        "bill_type":"s",
        "chamber":"senate",
        "committee_ids":[
           "SSVA"
        ],
        "congress":114,
        "cosponsors_count":0,
        "enacted_as":null,
        "history":{
           "active":false,
           "awaiting_signature":false,
           "enacted":false,
           "vetoed":false
        },
        "introduced_on":"2015-03-26",
        "last_action_at":"2015-03-27",
        "last_version_on":"2015-03-26",
        "last_vote_at":null,
        "number":895,
        "official_title":"A bill to allow members of the Armed Forces to defer
principal on Federal student loans for a certain period in connection with receipt
of orders for mobilization for war or national emergency, and for other purposes.",
        "popular_title":null,
        "related_bill_ids":[

        ],
        "short_title":null,
        "sponsor":{
           "first_name":"Jon",
           "last_name":"Tester",
           "middle_name":null,
           "name_suffix":null,
           "nickname":null,
           "title":"Sen"
        },
        "sponsor_id":"T000464",
        "urls":{
           "congress":"http://beta.congress.gov/bill/114th/senate-bill/895",
           "govtrack":"https://www.govtrack.us/congress/bills/114/s895",
           "opencongress":"https://www.opencongress.org/bill/s895-114"
        },
        "withdrawn_cosponsors_count":0
     }]
}
```


Figure 3. /bills JSON.

Table 1. API Methods.

Path	Description
/legislators	Current legislators' names, IDs, biography, and social media. **Fields**: bioguide_id, first_name, middle_name, last_name, nickname, state, party, name_suffix, district
/bills	Legislation in the House and Senate, back to 2009. Updated daily. **Fields**: bill_id, summary_short, short_title, official_title, last_vote_at, chamber, bill_status
/bills/search	Full text search over legislation. /bills/search enables the ability to search for bills by keywords. The list of keywords were developed by the Library of Congress, 1,023 unique keywords since 2009, as of late 2012 (https://sunlightlabs.github.io/congress/bills.html#summary-and-keywords, retrieved on 3/12/15). **Fields**: bill_id, short_title, official_title
/nominations	Presidential nominations before the Senate, back to 2009. Updated daily. **Fields**: position, name, last_action.text, nomination_id
/votes	Roll call votes in Congress, back to 2009. Updated within minutes of votes. **Fields**: bill_id, voted_at, voter_ids, chamber, vote_type, roll_type, question

stored procedure will make a request to the API that returns a JSON listing of legislators and their votes on the selected bill. The procedure then iterates through the returned JSON and scores individual legislator's votes by matching the external_uid, stored in the database, with the bioguide_id from the API (an official ID assigned by congress, **Figure 3**). The results of the scoring are stored in the database.

Creating a voter guide from the information stored in the database can now be readily accomplished. By using scorecard_id to join the scorecard_legislator and scorecard_legislator_vote tables (**Figure 2**), a listing of legislators and their votes on each scored bill can be compiled. The resulting listing can be sorted and ordered (by state, legislator name, party, etc.) and can be used on a website as a digital voter guide or formatted and printed for a hard copy voter guide.

4. Original Process

Prior to YeaNay, organizations had to employ a largely manual process involving the collaboration of several people to create voter guides. For this project an existing organization was analyzed to provide a baseline with which to compare YeaNay. The process they used for creating voting guides is described below. In comparison to the original process, YeaNay provides a highly automated solution that removes steps and simplifies the scoring of votes and ultimately the creation of voter guides.

Once a year, a file was received from a vendor with a list of representatives and senators along with their titles and contact information. This file was formatted and loaded into the database. Each individual legislator was then matched up with an existing database entry or a new entry would be manually created. The list of legislators would then be added to scorecard_legislator for use in scoring a new congressional session.

Throughout a congressional session, an individual would manually keep a list of potential bills to score. This list would be stored in an Excel spreadsheet and was not entered into the database until the scorer desired to make a voter guide. The spreadsheet holding the bills to be scored would then be delivered to the database developer containing the bill title, description, whether the organization approved of the bill, as well as whether the bill passed or failed. This spreadsheet would then be entered into the database and matched up with a congressional session.

Scoring each legislator's vote was handled in a similar manner to the bills. At the end of each congressional session a file would be created capturing the vote of every legislator on every bill. This file would then be loaded into the database. Once it was loaded, a person had to inspect all of the votes to make sure that each legislator's vote had been correctly scored by comparing the spreadsheet data with the actual votes due to errors with the scorer entering data into the spreadsheet or the loading process. From beginning to end, the entire process was time consuming and involved a significant amount of hands-on work.

5. Conclusion

In conclusion, YeaNay provides an automated process to create voter guides. Through keeping the list of legislators up to date and allowing for scoring votes based on a bill id/roll call, what previously involved several people and consumed excessive amounts of time is now a more streamlined process. One user, with minimal training, is able to use YeaNay to find and score legislation within minutes. The scores provided are then easily translatable for use in a voter guide or for personal use and knowledge. YeaNay focuses the firehose of congressional information that is now available and presents it in a manageable and usable environment.

References

Arizona Republic (2010). *Government Transparency Needed*.

Bertot, J. C., Jaeger, P. T., & McClure, C. R. (2008). *User-Centered E-Government Services: Benefits, Costs, and Research Needs*. Paper Presented at the 9th Annual International Conference on Digital Government Research.

Bertot, J., Jaeger, P. T., & Grimes, J. M. (2011). Promoting Transparency and Accountability through ICTs, Social Media, and Collaborative E-Government. *Transforming Government: People, Process and Policy, 6,* 78-91.

Brito, J. (2008a). Improving Government Transparency Online. *Public Manager, 37,* 22-26.

Brito, J. (2008b). Hack, Mash, & Peer: Crowdsourcing Government Transparency. *Columbia Science and Technology Law Review, 9,* 119-122.

Chun, S. A., Shulman, S., Sandoval, R., & Hovy, E. (2010). Government 2.0: Making Connections between Citizens, Data and Government. *Information Polity, 15,* 1-9.

Dawes, S. S., & Helbig, N. (2010). Information Strategies for Open Government: Challenges and Prospects for Deriving Public Value from Government Transparency. In M. A. Wimmer, J.-L. Chappelet, M. Janssen, & H. J. Scholl (Eds.), *Electronic Government* (pp. 50-60). Berlin: Springer.

Jaeger, P. T., & Bertot, J. C. (2010). Designing, Implementing, and Evaluating User-Centered and Citizen-Centered E-Government. *International Journal of Electronic Government Research, 6,* 1-17. http://dx.doi.org/10.4018/jegr.2010040101

Shkabatur, J. (2012). Transparency With(out) Accountability: Open Government in the United States (March 25, 2012). *Yale Law & Policy Review, 31,* 79-140.