

CSC342 Raleigh Public Record Voters Guide Functional Spec

Backend Data Sources

[NC house / senate member voting details](#)

[NC campaign finance records](#)

[US campaign finance records](#)

[US house / senate member voting details](#)

Response codes

[General ajax request example](#)

Backend API

[List Candidates](#)

[Search Candidates By field](#)

[Search Candidates By Any Field](#)

[Get Candidate Details](#)

[Add New Candidate](#)

[Update Candidate](#)

[Delete Candidate \(by url\)](#)

[Delete Candidate \(by object\)](#)

[List Positions](#)

[Get Position Details](#)

[Search positions by field](#)

[Add articles for candidate](#)

[Create Position](#)

[Update Position](#)

[Delete Position](#)

[Add Question](#)

[Search Questions](#)

[List questions](#)

[Get question by ID](#)

[Delete question \(by url\)](#)

[Delete question \(by request body\)](#)

[Update Answer](#)

[Add Answer](#)

[Search Answers](#)

Contributors:

Samuel Fink

Nicholas Shelton

Backend Data Sources

NC house / senate member voting details

<http://www.ncleg.net/>

NC campaign finance records

http://app.ncsbe.gov/webapps/cf_rpt_search_org/

US campaign finance records

<http://www.fec.gov/disclosure.shtml>

US house / senate member voting details

<https://www.govtrack.us/congress/votes>

Response codes

All request should return appropriate HTTP/1.1 response codes

- Successful requests return 200 (OK)
- Request format errors return 400 (NOT ACCEPTABLE)
- Invalid URL requests return 404 (FILE NOT FOUND)
- Database connection failures return 500 (INTERNAL SERVER ERROR)
- Successful deletion returns 204 (NO CONTENT)
-

General ajax request example

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $.ajax({
            contentType: 'application/json',
            data: /*put json object of request here*/ ,
            dataType: 'json',
            success: function(data){
                alert("success");
            },
            error: function(data){
                if(data.status >= 400){
                    alert("Error " + data.status);
                } else {
                    alert("OK " + data.status);
                }
                $("#response").html(data.responseText);
                /* do something with the response data in data.responseText */
            },
            processData: false,
            type: /* put request type here */,
            url: /* put rest url here */
        });
    });
});
</script>
```

Backend API

List Candidates

Request URL	/rest/candidates
Request type	GET
Parameters	None
Response format	JSON array of of objects each containing the following information: id: the candidate's assigned id number firstName: the candidate's first name lastName: the candidate's last name position: IDs of the position the candidate is running for level : 1 2 3 (int rpresentation for Federal, State, and Local) party: the party the candidate is affiliated with shortDesc : a 1 of 2 sentence description of the candidate img : a link to an image to use as a thumbnail

Search Candidates By field

Request URL	/rest/candidates
Request type	SEARCH
Parameters	<p>JSON object with attributes:</p> <ul style="list-style-type: none">firstName [optional]lastName [optional]position [optional]level [optional]party [optional]shortDesc [optional]longDesc [optional] <p>AT LEAST ONE FIELD MUST BE SPECIFIED</p> <p>search is conducted as a substring search of the field the data corresponds to</p> <p>field search results are AND together to produce final result</p>
Response format	<p>JSON array of of objects each containing the following information:</p> <ul style="list-style-type: none">id: the candidate's assigned id numberfirstName: the candidate's first namelastName: the candidate's last nameposition: IDs of the position the candidate is running forlevel : 1 2 3 (int rpresentation for Federal, State, and Local)party: the party the candidate is affiliated withshortDesc : a 1 of 2 sentence description of the candidateimg : a link to an image to use as a thumbnail

Search Candidates By Any Field

Request URL	/rest/candidates
Request type	SEARCH
Parameters	JSON object with attributes: str : a space separated list of keywords to search for. results of each keyword search are combined using AND
Response format	JSON array of of objects each containing the following information: id: the candidate's assigned id number firstName: the candidate's first name lastName: the candidate's last name position: IDs of the position the candidate is running for level : 1 2 3 (int rpresentation for Federal, State, and Local) party: the party the candidate is affiliated with shortDesc : a 1 of 2 sentence description of the candidate img : a link to an image to use as a thumbnail

Get Candidate Details

Request URL	/rest/candidates/id
Request type	GET
Parameters	none
Response format	<p>JSON object having the following attributes</p> <ul style="list-style-type: none">id: the candidate's assigned id numberfirstName : The candidate's first namelastName : The candidate's last nameposition : A list of IDs of the positions the candidate is running forparty : The party the candidate is affiliated withshortDesc : A short description of the candidateimg : A URL to an image of the candidatelongDesc: a long description / bio (a paragraph or two)pastPositions: a list of past governmental positions the candidate heldarticles: a list of links to articles about the candidate and article titles in order of most recent first (RPR articles only)prevFinance: if the candidate has available campaign finance information the link to it goes herevoteRecord: if available, a link to a list of bills that the candidate voted on in the past, and the number of votes by all members for/against the bill, and its outcome (only applies for incumbent house/senate)committees: if available, a list of committees that the candidate serves on (on applies for incumbent house/senate)QandA: an array of objects with "question" and "answer" fieldsadditionalData: a blob of data set from administrative panel for adding any data missed by the existing fields.

Add New Candidate

(Requires Authentication)

Request URL	/rest/candidates
Request type	PUT
Parameters	<p>firstName : The candidate's first name</p> <p>lastName : The candidate's last name</p> <p>position : IDs of the position the candidate is running for</p> <p>party : The party the candidate is affiliated with</p> <p>shortDesc : A short description of the candidate</p> <p>img: [optional] a URL for an image of the candidate</p> <p>longDesc: [optional] a long description / bio (a paragraph or two)</p> <p>additionalData : [optional] a blob of data for any data missed by existing fields</p> <p>prevFinance: [optional] if the candidate has available campaign finance information the link to it goes here</p> <p>voteRecord: [optional] if available, a link to a list of bills that the candidate voted on in the past, and the number of votes by all members for/against the bill, and its outcome (only applies for incumbent house/senate)</p>
Response	NONE

Update Candidate

Request URL	/rest/candidates
Request type	POST
Parameters	<p>JSON object with the following information</p> <p>id: [required] The id of the candidate to update firstName : [optional] The candidate's first name lastName : [optional] The candidate's last name position : [optional] A list of IDs of the positions the candidate is running for party : [optional] The party the candidate is affiliated with shortDesc : [optional] A short description of the candidate img : [optional] A URL to an image of the candidate longDesc: [optional] a long description / bio (a paragraph or two) pastPositions: [optional] a list of past governmental positions the candidate held articles: a list of links to articles about the candidate and article titles in order of most recent first (RPR articles only) prevFinance: [optional] if the candidate has available campaign finance information the contributors will be returned in a list voteRecord: [optional] if available, a list of bills that the candidate voted on in the past, and the number of votes by all members for/against the bill, and its outcome (only applies for incumbent house/senate) committees: [optional] if available, a list of committees that the candidate serves on (only applies for incumbent house/senate) additionalData: [optional] a blob of data set from administrative panel for adding any data missed by the existing fields.</p>
Response format	NONE

Delete Candidate (by url)

(Requires Authentication)

Request URL	/rest/candidates/id
Request type	DELETE
Parameters	NONE
Response format	NONE

Delete Candidate (by object)

(Requires Authentication)

Request URL	/rest/candidates
Request type	DELETE
Parameters	JSON object containing: id: the id of the candidate to delete
Response format	NONE

List Positions

Request URL	/rest/positions
Request type	GET
Parameters	NONE
Response format	JSON array of of matches each containing the following information: id : The position ID for detail lookup level : 1 2 3 (for Federal State Local) for the locality of the position termLength: length of the position's term (in years) name: The full name of the position description: a description of the position

Get Position Details

Request URL	/rest/positions/id
Request type	GET
Parameters	NONE
Response format	JSON object containing the following information: name : The string for the position title level : 1 2 3 (for Federal State Local) for the locality of the termLength: length of the position's term (in years) description: a description of the position's duties

Search positions by field

Request URL	/rest/positions
Request type	SEARCH
Parameters	<p>JSON object containing</p> <ul style="list-style-type: none">level : [optional] 1 2 3 (for Federal State Local) for the locality of the positionname: [optional] The full name of the positiontermLength: [optional] the length of a term (in years)description: [optional] a description of the position <p>AT LEAST ONE FIELD MUST BE SPECIFIED</p> <p>search is conducted as a substring search of the field the data corresponds to</p> <p>field search results are AND together to produce final result</p>
Response format	<p>JSON array of of matches each containing the following information:</p> <ul style="list-style-type: none">id : The position ID for detail lookuplevel : 1 2 3 (for Federal State Local) for the locality of the positionname: The full name of the positiontermLength: the length of a term (in years)description: a description of the position

Add articles for candidate

Request URL	/rest/articles
Request type	PUT
Parameters	<p>JSON object containing</p> <ul style="list-style-type: none">id: the ID of the candidatelink: the link to the article to add
Response format	NONE

Create Position

Request URL	/rest//positions
Request type	PUT
Parameters	JSON object containing the following information: name : The string for the position title level : 1 2 3 (for Federal State Local) for the locality of the position termLength: length of the position's term (in years) description: a description of the position's duties
Response format	NONE

Update Position

Request URL	/rest//positions
Request type	PUT
Parameters	JSON object containing at least one of the following fields to update id: the ID of the position to update [optional] name : The string for the position title [optional] level : 1 2 3 (for Federal State Local) for the locality of the position [optional] termLength: length of the position's term (in years) [optional] description: a description of the position's duties
Response format	NONE

Delete Position

(Requires Authentication)

Request URL	/rest/positions
Request type	DELETE
Parameters	JSON object containing id: the ID of the position to delete
Response format	NONE

Add Question

Request URL	/rest/questions
Request type	PUT
Parameters	JSON object containing question: the question string
Response format	NONE

Search Questions

Request URL	/rest/questions
Request type	SEARCH
Parameters	JSON object containing at least one field [optional] questionID: the id of the question [optional] question: substring to search for in question bodies
Response format	An array of JSON objects containing which match the search questionID: the id of the question question: The question text

List questions

Request URL	/rest/questions
Request type	GET
Parameters	NONE
Response format	An array of JSON objects containing which match the search questionID: the id of the question question: The question text answers: an array of answers containing; candidateID: the id of the candidate's response to update response: the candidate response string

Get question by ID

Request URL	/rest/questions/id
Request type	GET
Parameters	NONE
Response format	A JSON object containing questionID: the id of the question question: The question text answers: an array of answers containing; candidateID: the id of the candidate's response to update response: the candidate response string

Delete question (by url)

Request URL	/rest/questions/id
Request type	DELETE
Parameters	NONE
Response format	NONE

Delete question (by request body)

Request URL	/rest/questions
Request type	DELETE
Parameters	JSON object containing id: the id of the question to delete
Response format	NONE
Notes	This operation will delete the question and all corresponding answers

Update Answer

Request URL	/rest/answers
Request type	POST
Parameters	JSON object containing candidateID: the id of the candidate's response to update questionID: the id of the question to update response to response: the candidate response string
Response format	NONE

Add Answer

Request URL	/rest/answers
Request type	PUT
Parameters	JSON object containing candidateID: the id of the candidate's response to update questionID: the id of the question to update response to response: the candidate response string
Response format	NONE

Search Answers

Request URL	/rest/answers
Request type	SEARCH
Parameters	JSON object containing at least one field [optional] candidateID: the id of the candidate's response to update [optional] questionID: the id of the question to update response to [optional] response: substring to search in the candidate response strings
Response format	An array of JSON objects containing which match the search candidateID: the id of the candidate's response to update questionID: the id of the question to update response to response: the candidate response string