

# Table of Contents

[Table of Contents](#)

[The Elections API\(Voter Information API\) Documentation](#)

[Making a request for Token](#)

[How to get a token](#)

[Making a request for data](#)

[Method 1: Verification By Key](#)

[Method 2: Verification By Domain](#)

[API Requests](#)

[Managing an Election](#)

[API Reference](#)

[Post Filter](#)

[Requests](#)

[Properties](#)

[ElectoralDistrict Request](#)

[PollingStation Request](#)

[Properties](#)

[Candidate Request](#)

[Properties](#)

[Properties](#)

[Contest Request](#)

[Properties](#)

[Voter Request](#)

[Properties](#)

[Result Request](#)

[Properties](#)

[Properties](#)

# The Elections API(Voter Information API) Documentation

## Making a request for Token

To make a request to the API, there are several things that are required in addition to the request parameters i.e. an access Token supplied by the API, a key which is the hmac value of all the url parameters inclusive of the token.

### How to get a token

If at any time you need to fetch your application token, you first need to acquire a key. To get a key, hmac you **App Id** using your **App Secret** and **sha256**. Below is an example of how to acquire your key from your appid.

### Example Using Python:

Python 2.7.3 (default, Apr 10 2012, 23:31:26) [MSC v.1500 32 bit (Intel)] on win32

Type "copyright", "credits" or "license()" for more information.

```
>>> import hmac
>>> import hashlib
>>> appid = 'appid=KEA119'
>>> appsecret = '7d11beed7346bf00a8c6063dc4003c47'
>>> hmac.new(appsecret, appid, hashlib.sha256).hexdigest()
'f94801a462049f3a900476c9e7414fc0370df5d9bc5e37f53c6c3ffb48dc1d1f'
```

### Example Using PHP:

```
$appid = 'appid=KEA119';
$appsecret = '7d11beed7346bf00a8c6063dc4003c47';
$key = hash_hmac('sha256', $appid, $appsecret);
echo $key;
f94801a462049f3a900476c9e7414fc0370df5d9bc5e37f53c6c3ffb48dc1d1f
```

### Note:

The method above describes how to get a key for getting a token, by hashing the appid using the appsecret. For other requests, hash(filter+token). The filter must be used if it exists.

**NB: If you have more than 1 filters, when creating the filter string, append the filters to the string alphabetically, i.e.**

For a request:

<http://api.iebc.or.ke/candidate/?party=P001&contest=8E35B8D0-0343-E211-884B-70F3955EC982>

Assuming that the token is 2cbe8ab7619cd515337bab42243704926c86ecf75, then the key will be generated from the string =

“contest=8E35B8D0-0343-E211-884B-70F3955EC982party=P001token=2cbe8ab7619cd515337bab42243704926c86ecf75”

Now that you have a key, it's time to make the token request to the API. To get a token, do a GET on:

<http://api.iebc.or.ke/token/?appid=KEA119&key=f94801a462049f3a900476c9e7414fc0370df5d9bc5e37f53c6c3ffb48dc1d1f>

You should receive a JSON response that looks similar to this:

```
{"status": "SUCCESS", "token":  
"e4ef8cd8b6ab1052cbe8ab7619cd515337bab42243704926c86ecf750f4fe6db"}
```

If the request was not successful, you should receive a JSON response in the form:

```
{"status": "ERROR TITLE", "message": "ERROR DESCRIPTION"}
```

## Making a request for data

The path on the URL determines what data the API will fetch. Different URL paths access different classes on the API that handle requests for the various data. Below are examples on how to request data from the API. All responses are in JSON.

### Method 1: Verification By Key

\*code in the url refers to the id of the entity being queried. If used in the filter, then this will be the filter id.

\*key refers to the hashed value of a filter and the token. i.e hash(filter+token) in that order

\*ed in the url will refer to data types of type electoral districts i.e. county, constituency & ward

\* If no filters the key is the hashed value of the token ie value = "token={{token}}"

Example getting key for a filter county = C001 and token =  
**e4ef8cd8b6ab1052cbe8ab7619cd515337bab42243704926c86ecf750f4fe6db**

Example in Python:

```
>>> import hashlib  
  
>>> import hmac  
  
>>> token = "token=e4ef8cd8b6ab1052cbe8ab7619cd515337bab42243704926c86ecf750f4fe6db"  
  
>>> appsecret = '7d11beed7346bf00a8c6063dc4003c47'  
  
>>> filterr = "county=C001"  
  
>>> keyString = "%s%s"%(filterr, token)  
  
>>> hmac.new(appsecret, keyString, hashlib.sha256).hexdigest()  
  
'8526091f265e93767ec69eece766b54c327b85c78c41203a4247fb0b0bee8c3a'
```

Your key is **8526091f265e93767ec69eece766b54c327b85c78c41203a4247fb0b0bee8c3a**

**A Sample Request:**

<http://api.iebc.or.ke/constituency/?county=C001&token=e4ef8cd8b6ab1052cbe8ab7619cd515337bab42243704926c86ecf750f4fe6db&key=8526091f265e93767ec69eece766b54c327b85c78c41203a4247fb0b0bee>

[8c3a](#)

If there is a match in the key generated for the request and the generated by the verification system, then the response will be a list of constituency summary information objects or an object containing the key verification error message.

## Method 2: Verification By Domain

In cases where the appsecret is discoverable e.g. in client side scripting languages like Javascript, the key is not required and in this case, verification is done through domain verification. The request will only take the token and any other url parameters like the filter and the election parameters that apply to that request.

For this method, the request must originate from the domain within which the app is registered to. So while registering the application ensure that the domain you register with, **is the intended domain for the application** i.e. for <http://www.example.com>, use [example.com](http://www.example.com) as the domain.

### Using example in Method 1:

A request for the same data as in method 1, **using method 2** will have the request shown in the example below.

#### A Sample Request:

<http://api.iebc.or.ke/constituency/?county=C001&token=e4ef8cd8b6ab1052cbe8ab7619cd515337bab42243704926c86ecf750f4fe6db>

If domain verification is successful, then the response will be a list of constituency summary information objects or an object containing the domain verification error message.

## API Requests

The Table below illustrates how to make requests to the api, url and urls parameters required, and the response expected from each request.

### Note:

For each request you make to the api, please remember to append the token and key as required by the request method you are using to the url.

Request	Required	Results	Filters
<a href="http://api.iebc.or.ke/county/">http://api.iebc.or.ke/county /</a> <a href="http://api.iebc.or.ke/county/code/">http://api.iebc.or.ke/county /code/</a>	token and key*	- The response is an Array of ElectoralDistrict Summary Information objects for each county  -If the county code is used, the response is an object	No filters

		containing the specified county information	
<a href="http://api.iebc.or.ke/constituency/">http://api.iebc.or.ke/constituency/</a>  <a href="http://api.iebc.or.ke/constituency/code/">http://api.iebc.or.ke/constituency/code/</a>	token, key* and filter(optional )	<p>- The response is an Array of ElectoralDistrict Summary Information objects for each constituency</p> <p>- if the constituency code is used, then the response is an object containing the specified constituency's information.</p>	Filter by county. This is in the form: <a href="http://api.iebc.or.ke/constituency/?county=code">http://api.iebc.or.ke/constituency/?county=code</a>
<a href="http://api.iebc.or.ke/ward/">http://api.iebc.or.ke/ward/</a>  <a href="http://api.iebc.or.ke/ward/code/">http://api.iebc.or.ke/ward/code/</a>	token, key* and filter(optional )	<p>- The response is an Array of ElectoralDistrict summary Information objects for each ward</p> <p>- if the ward code is used, then the response is an object containing the specified ward's information.</p>	Filter by county or constituency. This is in the form: <a href="http://api.iebc.or.ke/ward/?constituency=code">http://api.iebc.or.ke/ward/?constituency=code</a>  <a href="http://api.iebc.or.ke/ward/?county=code">http://api.iebc.or.ke/ward/?county=code</a>
<a href="http://api.iebc.or.ke/pollingstation/">http://api.iebc.or.ke/pollingstation/</a>  <a href="http://api.iebc.or.ke/pollingstation/code/">http://api.iebc.or.ke/pollingstation/code/</a>	token, key* and filter(not required for polling station information)	<p>-Returns an Array of PollingStation summary information objects for each polling station in the result</p> <p>-If polling station code is used, then the response is an object of polling station information for the polling station defined by the code</p>	<p>- Polling stations are filtered across the electoral districts; county, constituency and the ward.</p> <p>This is of the form: <a href="http://api.iebc.or.ke/pollingstation/?ward=code">http://api.iebc.or.ke/pollingstation/?ward=code</a>  <a href="http://api.iebc.or.ke/pollingstation/?constituency=code">http://api.iebc.or.ke/pollingstation/?constituency=code</a>  <a href="http://api.iebc.or.ke/pollingstation/?county=code">http://api.iebc.or.ke/pollingstation/?county=code</a></p>
<a href="http://api.iebc.or.ke/party/">http://api.iebc.or.ke/party/</a>	token and	- returns an Array of party	Parties can be filtered

<p><a href="http://api.iebc.or.ke/party/code/">http://api.iebc.or.ke/party/code/</a></p>	<p>key* - key is the hashed value of the token</p>	<p>summary information objects</p> <p>- if the party code is used, then the response is an object containing party information for the party defined by the code.</p>	<p>by an election.</p>
<p><a href="http://api.iebc.or.ke/contest/">http://api.iebc.or.ke/contest/</a></p>	<p>token, key* and filter</p>	<p>- The response is an Array of contest summary information objects</p>	<p>Filter is either an election, electoral district or an elective post.</p> <p>This request is of the form: <a href="http://api.iebc.or.ke/contest/?post=code">http://api.iebc.or.ke/contest/?post=code</a></p>
<p><a href="http://api.iebc.or.ke/candidate/">http://api.iebc.or.ke/candidate/</a></p> <p><a href="http://api.iebc.or.ke/candidate/code/">http://api.iebc.or.ke/candidate/code/</a></p>	<p>-token, key* and filter</p>	<p>- The response is an Array of candidate summary information objects.</p> <p>- If the candidate code is used then the response is an object of candidate information for the candidate defined by the code.</p>	<p>Filter is either an election, a contest, an electoral district, an elective post or a party.</p> <p>This request will take the form: <a href="http://api.iebc.or.ke/candidate/?contest=code">http://api.iebc.or.ke/candidate/?contest=code</a></p>
<p><a href="http://api.iebc.or.ke/voter/code/?type=info">http://api.iebc.or.ke/voter/code/?type=info</a></p> <p><a href="http://api.iebc.or.ke/voter/code/?type=results">http://api.iebc.or.ke/voter/code/?type=results</a></p>	<p>token, key* and filter</p>	<p>- For filter type = info, the response is an object of voter information.</p> <p>- For filter type = results, the response is an object of voter results information.</p>	<p>The filters for the voter information will be info and results.</p>
<p><a href="http://api.iebc.or.ke/results/ED/CODE/?election=code">http://api.iebc.or.ke/results/ED/CODE/?election=code</a></p>	<p>a token, key* and a filter.</p>	<p>- The response is an object containing an array of results information objects.</p>	<p>The filters for the results are the election, the electoral district, the elective post .</p> <p>This is of the form: <a href="http://api.iebc.or.ke/r">http://api.iebc.or.ke/r</a></p>

			<p>results/county/code/? post=code&amp;election=c ode</p> <p>Filters apply to all the results url paths except where the filter and the url path parameter are the same.</p> <p>This is incorrect: http://api.iebc.or.ke/r esults/county/code/? county=code</p> <p>*code refers to a unique identifier for each type</p>
<p>http://api.iebc.or.ke/info/ http://api.iebc.or.ke/info/co de/</p>	<p>a token and a filter  token and key*</p>	<p>- The response is an Array of voter information summary objects.</p> <p>- if a code is passed then the result is an object of voter information objects for the voter information defined by code.</p>	<p>No filters</p>
<p>http://api.iebc.or.ke/reporti ng/ http://api.iebc.or.ke/reporti ng/code/</p>	<p>a token and a filter  token and key*</p>	<p>- The response is an Array of reporting polling stations.</p> <p>- If a post is passed then the response is an Array of reporting polling stations that have reported results for that post.</p> <p>-If a code is passed then the response is an Array of reporting polling stations that have reported results in the electoral area defined by the code.</p>	<p>- post</p>

\* required only for requests using method 1



## Managing an Election

The API is capable of serving election data for different elections. This is done by adding the url parameter **election** for each party, results and/or candidate request that you make to the API. Below is the a table of the election data currently in the API and their election code.

Test data for each election will now be available under the election id generated by an **election code** with the suffix **-TEST**

i.e. For the October 17th by elections test data will be available under the election code :  
**20131017-TEST**

<b>Election</b>	<b>Code</b>
The General Election 2013	GEK2013
The Makueni Senate By Election	MAKSEN2013
The October 17th By Elections	20131017

i.e. for upcoming October 17th By Elections:

<http://api.iebc.or.ke/results/county/85BBDC91-AA5B-4985-BAB1-ED5E90EB448F/?post=3&token={{YourToken}}&election=20131017>

## API Reference

### Post Filter

This defines the contest types and the id of each contest type. Used when filtering with post.

Post	Post Id(contest type)
Presidential	1
Governor	2
Senator	3
National Assembly	4
Women Representative	5
County Assembly Representative	6

A sample response filtered by post would be in the form:

```
{
  "status": "SUCCESS",
  "candidates": [
    {
      "race": "KENYA",
      "candidates": [
        {
          "picture": "http://api.iebc.or.ke/images/candidate/imagename.jpg",
          "contest_id": "FE8DB1B1",
          "code": "5FFCD75B",
          "surname": "surname",
          "other_name": "names",
          "party": {
            "picture": "http://api.iebc.or.ke/images/parties/imagename.jpg",
            "abr": "abr",
            "code": "A26505A1",
            "name": "Party a"
          },
          "running_mate": "names",
          "contest_type": "1"
        }
      ]
    }
  ]
}
```

## Requests

### Token Request

The request returns the access token for the requesting application.

### Properties

Property	Type	Description
appid	string	The identifier for a given application.
key	string	The key is the hashed value of the appid.

The request returns a JSON response containing either an **error message** or **token**.

A Sample response:

```
{"status": "SUCCESS", "token":  
"e4ef8cd8b6ab1052cbe8ab7619cd515337bab42243704926c86ecf750f4fe6db"}
```

### ElectoralDistrict Request

This request returns a list of electoral districts dependent on the filter or information about a given electoral district as specified by the code.

Property	Type	Description
token	string	This is the token generated for the application to access the api
ed	string	Refers to the electoral districts ( <b>county, constituency or ward</b> )
code	string	Refers to an identifier unique to an electoral district (ed)
key *	string	Hashed value of the filter and token i.e hash(filter+token)
filter	string	This is used to filter the response

\* required only for requests using method 1

ED	Filters
county	None
constituency	county
ward	constituency and county

A sample response would be in the form:

Electoral District List:

```
{  
  "status": "SUCCESS",
```

```

    "region": {
      "polygon": "http://api.iebc.or.ke/geojson/county_1.geojson",
      "code": "C001",
      "name": "Baringo",
      "registered": "174708",
      "center": {"lat": 0.512912, "lon": 35.952537}
    }
  }
}

```

## PollingStation Request

This request returns a list of polling station summary information objects or an object of polling station information.

### Properties

Property	Type	Description
token	string	This is the token generated for the application to access the api
code	string	Refers to an identifier unique to an electoral district (ed)
key*	string	Hashed value of the filter and token i.e hash(filter+token)
filter	string	This is used to filter the response

\* required only for requests using method 1

Filters
county, constituency, ward

Sample response:

```

{
  "status": "SUCCESS",
  "info": {
    "code": "052A4B3B-211B-4ED8-BF4E-6D67D17AC9CC",
    "accessibility": null,
    "voters": null,
    "female": null,
    "networks": null,
    "name": "ST. PAUL'S PRIMARY SCHOOL",

```

```

    "electricity": null,
    "disability": null,
    "facility_type": null,
    "streams": null,
    "location": {"lat": -1.2865009999999999, "lon": 36.831015000000001},
    "male": null
  }
}

```

## Candidate Request

The request returns a list of candidate summary information objects or a candidate information object.

### Properties

Property	Type	Description
token	string	This is the token generated for the application to access the api
code	string	Refers to an identifier unique to a candidate
key*	string	Hashed value of the filter and token i.e hash(filter+token)
filter	string	This is used to filter the response

\* required only for requests using method 1

Filters
party, contest, ward, constituency or county

Sample Response:

```
{
  "status": "SUCCESS",
  "candidates": [
    {
      "race": "MAKADARA",
      "candidates": [
        {
          "picture": "http://api.iebc.or.ke/images/candidate/imagename.jpg",
          "contest_id": "C5D9CC",
          "code": "6870012",
          "surname": "surname",
          "other_name": "names",
          "party": {
            "picture": "http://api.iebc.or.ke/images/parties/imagename.jpg",
            "abr": "abr",
            "code": "BB490E47-850",
            "name": "Party c"
          },
          "running_mate": " ",
          "contest_type": "4"
        }
      ]
    }
  ]
}
```

## Party Request

This request returns a list of parties' summary information objects or an object of party information.

### Properties

Property	Type	Description
token	string	This is the token generated for the application to access the api
code	string	Refers to an identifier unique to a party
key*	string	Hashed value of the token only. Party request does not take any filters.

\* required only for requests using method 1

```
{
Sample response:
  "status": "SUCCESS",
  "parties": [ {
    "picture": "http://api.iebc.or.ke/images/parties/imagename.jpg",
    "code": "5EC982",
    "name": "Party a",
    "abr": "abr },
    {
      "picture": "http://api.iebc.or.ke/images/parties/imagename.jpg",
      "code": "A16505",
      "name": "Party b",
      "abr": "abr }
  ]
}
```

## Contest Request

This request returns a list of contests summary information objects or an object of contest information.

### Properties

Property	Type	Description
token	string	This is the token generated for the application to access the api
code	string	Refers to an identifier unique to a contest
key*	string	Hashed value of the filter and token i.e. hash(filter+token)
filter	string	This is used to filter the response

\* required only for requests using method 1

Filters
post, county, constituency and county

A contest response will be in the form:

```
{
  "status": "SUCCESS",
  "contests": [
    {
      "contestType": "2",
      "code": "C535B8D0-0343-E211-884B-70F3955EC982",
      "contest": "Governor - BARINGO (030)"
    },
    {
      "contestType": "3",
      "code": "C635B8D0-0343-E211-884B-70F3955EC982",
      "contest": "Senator - BARINGO (030)"
    },
    {
      "contestType": "5",
      "code": "C735B8D0-0343-E211-884B-70F3955EC982",
      "contest": "Women Representative - BARINGO (030)"
    }
  ]
}
```

### Voter Request

This request returns an object of voter registration information or an object of voter results information.

### Properties

Property	Type	Description
token	string	This is the token generated for the application to access the api
code	string	Refers to an identifier unique to a voter, may be their Voter ID or Passport No. <b>Required</b>
key*	string	Hashed value of the filter and token i.e hash(filter+token)



type	string	Determines the type of information being fetched for the voter. Maybe results or info.
------	--------	--

\* required only for requests using method 1

The response will be in the form:

```
{
  "status": "SUCCESS",
  "voter_info": {
    "name": "",
    "gender": "",
    "pollingStation": {
      "point": "",
      "electricity": "",
      "disability": "",
      "accessibility": "",
      "facility_type": "",
      "streams": "",
      "networks": ""
    },
    "county": "",
    "date": "",
    "contestant": "",
    "ward": "",
    "contests": [],
    "constituency": ""
  }
}
```

## Result Request

This request returns a list of results summary information objects

### Properties

Property	Type	Description
token	string	This is the token generated for the application to access the api
category	string	defines a filter for results that you want
code	string	Refers to an identifier unique to a category
key*	string	Hashed value of the filter and token i.e hash(filter+token)
filter	string	Another filter for the results request

\* required only for requests using method 1

<b>Category * required</b>	<b>Filters</b>
----------------------------	----------------

county, constituency, ward	party, contest, post(contest type)
contest	party, county, constituency, ward
party	county, constituency, ward, contest, post(contest type)
post(contest type)	party, county, constituency, ward

This request returns a list of voting information summary information object or an object of information specific to a given voter information

### Properties

Property	Type	Description
token	string	This is the token generated for the application to access the api
code	string	Refers to an identifier unique to a voter, may be their Voter ID or Passport No. <b>Required</b>
key*	string	Hashed value of the filter and token i.e hash(filter+token)

\* required only for requests using method 1

Sample response:

```
{
  "status": "SUCCESS",
  "list": [
    {
      "code": "VI001",
      "title": "Presidential Mandate"
    }
  ]
}
```