

Pennsylvania Voter Registration and Vote by Mail API Specification

Pennsylvania Department of State

Table of Contents

Contents

- Pennsylvania Voter Registration and Vote by Mail API Specification..... 1
 - Pennsylvania Department of State 1
 - Table of Contents 1
 - Introduction..... 2
 - Benefits of Using the API 2
 - What Does the API Do? 3
 - Register to Vote..... 3
 - Update Existing Voter Registration 3
 - Request a Mail-In Ballot 3
 - Who Can Use the API 4
 - Posting Organization 4
 - How to Register Your Organization with the Department of State 4
 - Terms of Use..... 5
 - Security..... 5
 - How it Works..... 6
 - Process..... 6
 - Modes of Operation 8
 - Flow of Information..... 9
 - Required Fields in Application 11
 - Signature Requirements..... 11
 - How Long Does the Process Take 12
 - Sample Timeline 12
 - Process Status..... 13
 - How to Make Calls..... 13
 - Base URLs 14
 - Return Responses..... 14
 - Error Responses..... 15

API Calls	15
[GET] GETAPPLICATIONSETUP.....	15
[GET] GETBALLOTAPPLICATIONSETUP	21
[GET] GETLANGUAGES	24
[GET] GETXMLTEMPLATE	25
[GET] GETBALLOTXMLTEMPLATE.....	31
[GET] GETERRORVALUES.....	35
[GET] GETMUNICIPALITIES	36
[POST] SETAPPLICATION.....	38
[POST] SETBALLOTAPPLICATION	39
References and Resources	41
Register to Use the API.....	41
Terms of use	41
Project Plan	41
Field Definition Library	41
Voter Registration Drives	41
Contact the Department of State.....	41
Revision History.....	41

Introduction

The Pennsylvania Department of State works to make voting more accessible. Voter registration drives are an effective way to help new voters register and current voters update their information. The Online Voter Registration and Vote by Mail API (PA OVR-VBM API) makes it easy for third-party organizations to build an application to register voters with mobile devices, including during voter registration drives. The vote by mail feature can also be used independent of voter registration. Absentee ballot applications are not currently supported through the API.

This API allows for the submission of [voter registration](#) and [vote by mail applications](#) in the same secure way as the standard online application processes through the Department of State. Organizations that build custom applications with this API can collect voter contact information for the use of their organization. This document provides interested organizations with the information they need to develop, test, and use the API.

The API was updated in May 2024 to include the new vote by mail application function for requesting mail-in ballots as a standalone function. The prior revision tied a voter registration application to an optional mail ballot application. This documentation has been updated for clarity and usability.

Benefits of Using the API

Many voter registration drives rely on collecting voter information on paper. By registering voters online through this application, your organization can reduce overhead costs and increase accuracy of voter data

submitted. The organization hosting the drive can also collect the contact information of applicants for their own communications purposes.

You can use the API to gather applicant information for your organization's use, including:

- Contact information
- Demographic information

Those who register to vote or request a mail-in ballot through the API will receive real-time confirmation of their submission. Organizations that use the API will have the ability to generate reports through their account with the Department of State to track the status of all submissions. Voter registration and vote by mail submissions are tracked separately.

What Does the API Do?

The API supports the following actions:

- Register to vote
- Update existing voter registration
- Request a mail-in ballot to vote by mail

Applicants completing these actions are required to provide either a Pennsylvania driver's license or PennDOT ID number. Signatures can also be accepted for voter registration applications. All actions must be completed by the dates outlined in the [PA Election Calendar](#) to be recognized for the current election cycle. The API will confirm the next election date and due dates in accordance with the PA Election Calendar.

Register to Vote

Anyone who meets [voter registration requirements](#) may register to vote through the API.

Update Existing Voter Registration

For those who are already registered to vote, the API can support them in updating their registration. Supported voter maintenance functions include changes to:

- Name
- Address
- Party affiliation
- If they require assistance to vote
- Federal or state employee registering in county of last residence

Request a Mail-In Ballot

Mail-in ballot requests can be submitted for those who are already registered to vote, as well as simultaneously with new or updated voter registration requests. Absentee ballot requests are not currently supported by the API.

The applicant must have a valid Pennsylvania driver's license or PennDOT ID to submit a mail-in ballot request.

Annual Mail-In Ballot

Through the vote by mail portion of the API, applicants have the option to get on a list to receive an annual mail ballot application. Once the voter completes the annual application to request mail-in ballot and the county approves the application, the voter will get a ballot for every election for the remainder of the year.

Who Can Use the API

The Online Voter Registration and Vote by Mail API is the best method of voter registration for organizations with technical resources to develop a web application program interface. This document is meant to support technical staff at your organization in setting up the API, including the development, testing, and use of an application for secure voter registration, voter record maintenance, and/or vote by mail ballot requests.

Prior to using the Online Voter Registration and Vote by Mail API, your organization must be approved by the Department of State.

Is this API the right choice for your organization? [Read the PA Voter Registration Guide to learn about other voter registration options.](#)

Posting Organization

The posting organization is the entity interested in developing their own custom voter registration and/or mail-in ballot application with the Online Voter Registration and Vote by Mail API. Organizations approved as posting organizations are responsible for the compliance and maintenance of their application and its data exchange with the API.

Posting organizations can request access to the API from the Department of State for the following uses:

- Voter registration access
- Vote by mail access
- Both voter registration and vote by mail access

If your organization would like to use both voter registration and vote by mail, you must complete separate registration processes. Because of the many associated security requirements and reports, correspondence logging is specific to the type of registration, therefore the processes are separate.

You may use the same contact information, affiliates, default language, usernames, etc. for both registrations.

Affiliates of the Posting Organization

Posting organizations can enroll other organizations, also known as affiliates, to use the posting organization's application. The posting organization must enroll affiliates and go through the same tests to vet the affiliate and their understanding of the posting organization's application. As the affiliate conducts its testing and is approved to submit voter registration and vote by mail submissions, the posting organization must authorize affiliate application submissions using the affiliate access key assigned. This is different than the posting organization's access key.

There is no limit to the number of active affiliates a posting organization can have.

How to Register Your Organization with the Department of State

To use the API, your organization must be approved by the Department of State. There are separate registration processes and terms of use for the online voter registration and vote by mail features.

Registration and Set Up

To become an approved posting organization with the Department of State, your organization must [complete the registration process](#). Once completed, the posting organization must review, sign, and return the Terms of Use.

The Department of State will provide initial status and credential access upon completion.

STAGING Environment Access

Once pre-approved by the Department of State, organizations will receive an authentication access key for read-only access to the STAGING environment. Initial access to the STAGING environment is intended to aid in the development of the custom voter registration and/or vote by mail application. This will support in the building of dropdowns, listing of error codes, tag catalog for submissions, and more, including some pre-populated information from the SURE system.

The STAGING environment will be used for:

- User review of new features and ongoing bug fixes
- User acceptance testing of new features and ongoing bug fixes
- Deployment testing
- Demos
- Posting organization testing and approval

When the organization has a successful design and can retrieve information, they should contact the Department of State. The Department of State will then update the access key to be in the APPROVED status to allow read/write access to the same STAGING environment.

Testing

Once in the APPROVED status, the Department of State will provide data for end-to-end testing. The posting organization will use the provided test data and share the API responses generated back to the Department of State. All tests must be successful with documentation before a CERTIFIED access key will be granted for the production, or PROD, environment. The Department of State will verify that the applications were successfully submitted.

PROD Environment Access

Once verification is completed in the APPROVED status, the Department of State will update the posting organization's status to CERTIFIED. This removes access to STAGING, while providing access to the PROD environment. The PROD environment is for the operational functions of live data involving voter information and its associated systems. The access key for the PROD environment will be provided once the organization has satisfied all testing requirements and the respective Terms of Use has been agreed to and completed. The access key will be used to authenticate each posting organization for security and reporting purposes.

To allow for continual development and maintenance, Department of State will change the status to CERTIFIED TESTING. This status allows read and write capabilities for both PROD and STAGING, with separate access keys.

Terms of Use

The posting organization must read and agree to the applicable terms of use to gain access to the corresponding module of the PA Online Voter Registration and Vote by Mail API service:

- [Voter Registration API Terms of Use](#)
- [Vote by Mail API Terms of Use](#)

The custom application must pass all content, workflow, and acceptance tests outlined.

Security

Current security measures employ both the environment access key (per posting organization) and calls using HTTPS and SSL protocols. Posting organizations are responsible for the encryption of data, both at rest and while

in transit. The encryption must be at least at a level as identified by Commonwealth IT Policies, [specifically IT SEC-031](#). Please note that separate access keys are granted and required for:

- Access to UAT and PROD environments
- Access levels (read-only versus read/write) within an environment
- Type of application submission (voter registration with optional mail-in versus vote by mail only). This means that there will be a separate set of access keys for online voter registration and vote by mail.

The Department of State reserves the right to rescind or re-issue new PROD or STAGING access keys with advanced notice. Re-issuance of access keys may be required due to security concerns.

How it Works

Once developed and certified, the posting organization’s application submits Voter Registration and Mail Ballot Applications to SURE to be processed following the validation rules required for all applications.

Process

Figure 1: Online Voter Registration Process

This figure outlines the process of an applicant using your application for voter registration, including the information collected throughout the process.

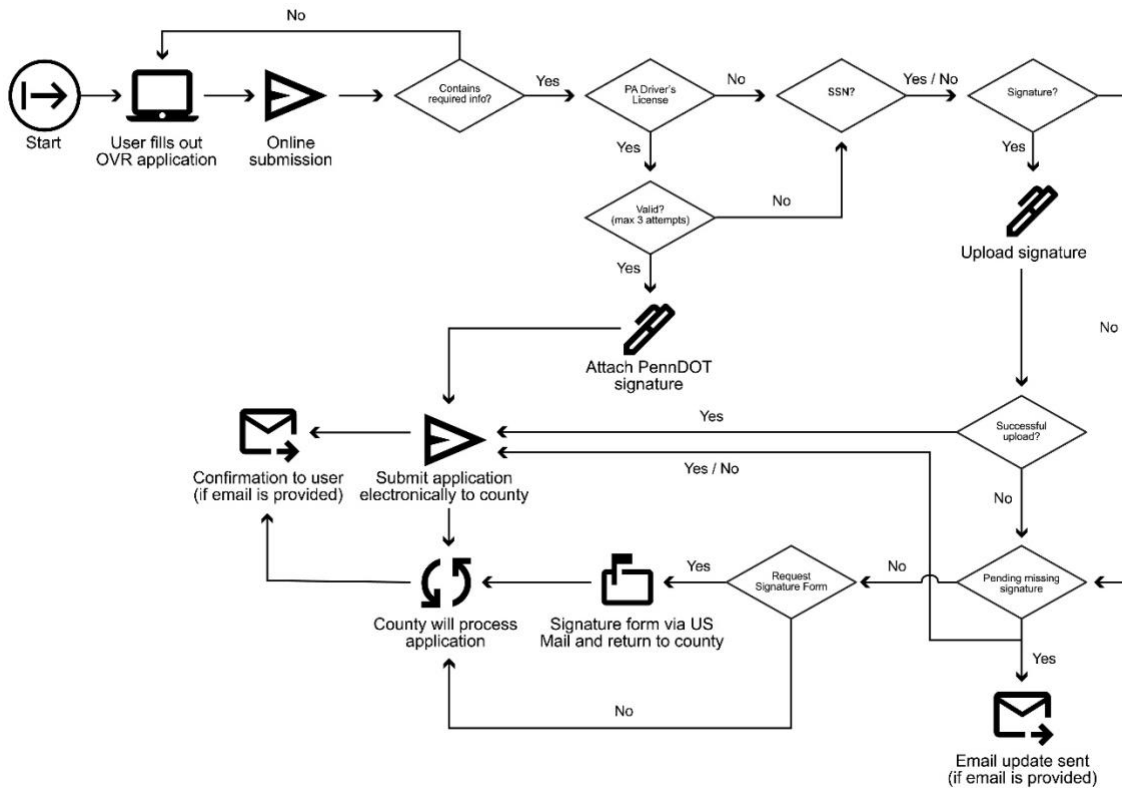


Figure 2: Simultaneous Online Voter Registration and Vote by Mail Process

This figure outlines the application workflow for simultaneous submissions of voter registration and mail-in ballot applications.

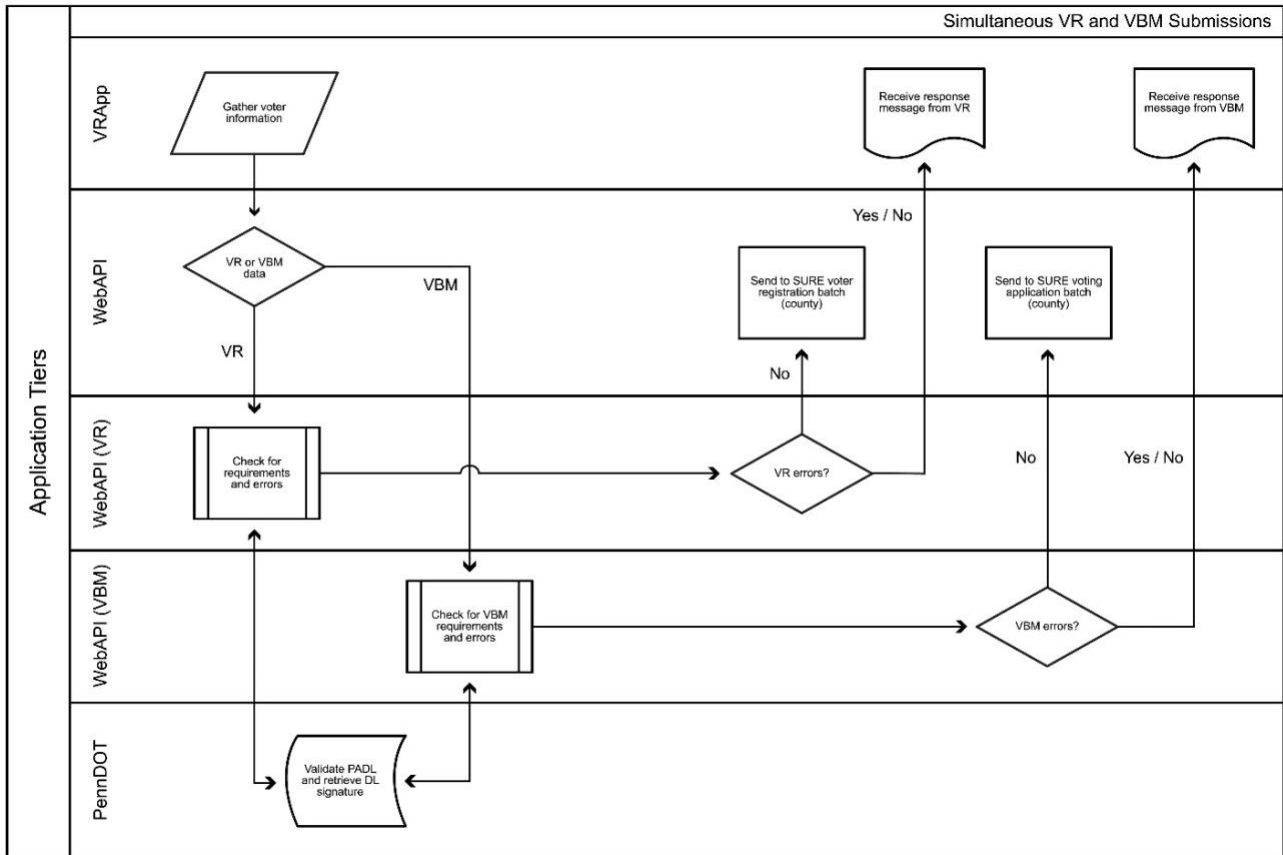
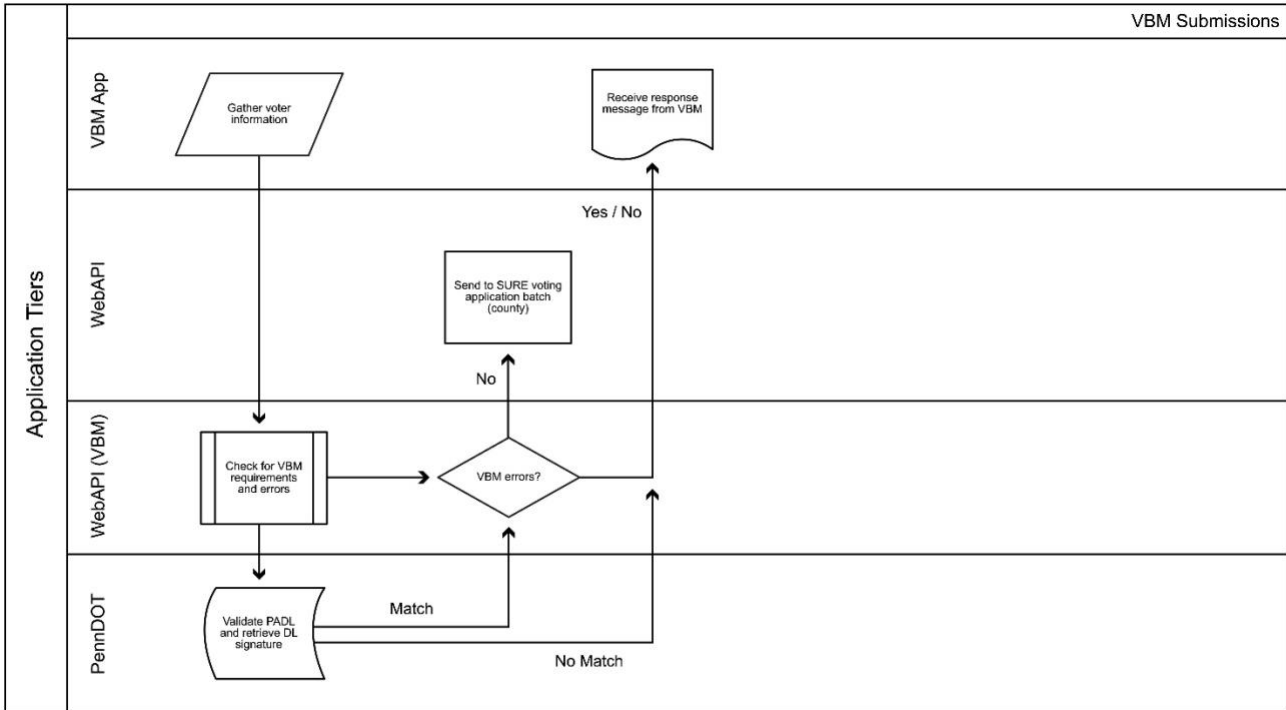


Figure 3: Vote by Mail Application Process

This figure outlines the workflow of applications that are for mail-in ballots only.



Modes of Operation

To accommodate various development techniques, the Online Voter Registration and Vote by Mail API was developed to support two error response modes, interactive (verbose error responses) and batch (first error only response). The key differentiator between these modes is the level of error responses associated with a submission (POST call).

Batch and Interactive modes both require that all XML tags are present with at least the required fields containing data. The expected response with associated errors for interactive mode is that the error response will contain all errors encountered in the API attempting to process the submission. Batch mode will only respond with the first error encountered in the API attempting to process the submission.

Key Mode Differentiators

Mode	POST Calls	Expected Response
Batch	Submit all fields for an application within a single POST call (XML)	All responses to each field will be received in the order submitted/processed, but all at once.
Interactive	Submit all fields for an application within a single POST call (XML)	Single error or success response received per POST call/XML submission. This is the first error encountered or final successful submission.

Applications can be submitted using either the interactive or batch mode. The level of error response is determined by the <BATCH> field and a response from the API is required prior to submitting the next application.

With Valid PennDOT ID or Driver's License

Voter registration or mail-in ballot applications that are submitted interactively with a valid PennDOT ID number or PA driver's license number rely on the API and other services to attach a digitized copy of the applicant's PennDOT signature to the voter registration application. Once a response and signature are received by PennDOT, the subsequent posting of application data will be committed to the SURE system as an appropriate response generated to the posting organization's application via the API per POST call.

With Social Security Number and Signature

If the applicant does not have a PennDOT ID or driver's license for voter registration or mail-in ballot requests, the applicant can submit the last four digits of their social security number and/or upload a digital file of their signature. This functionality should be incorporated into the posting organization's application logic. The signature submission will be processed by an imaging service within the SURE architecture.

If the signature file does not meet the API standards or fails to properly process, the applicant may, at the discretion of the posting organization, have the option to print the blank registration form. The applicant can then sign it and mail it to their county voter registration office. The posting organization can indirectly request a signature card be sent to the applicant's address by leveraging the continueAppSubmit XML tag. This option can be built into the posting organization's workflow. It will leverage the existing SURE system workflows to generate and mail the missing signature correspondence. There is no limit to the number of retries supported when submitting validation for PennDOT driver's license and signature uploads for the Online Voter Registration and Vote by Mail API.

Once successfully submitted, the API will provide a message back to the applicant that includes the appropriate application ID number(s). All further correspondence will be managed by the SURE system. Applicants may receive communications via email if an email address was provided with their application.

In batch mode, all information is submitted within one POST call. Applications that are submitted in a batch process should allow the applicant to submit both a PennDOT ID or driver's license number and a signature image file.

Flow of Information

The following UML sequence diagrams show the flow of information throughout the process and how the various systems interact with each other.

Figure 4: UML Sequence Diagram for Interactive Mode

This figure outlines the process flow for interactive mode.

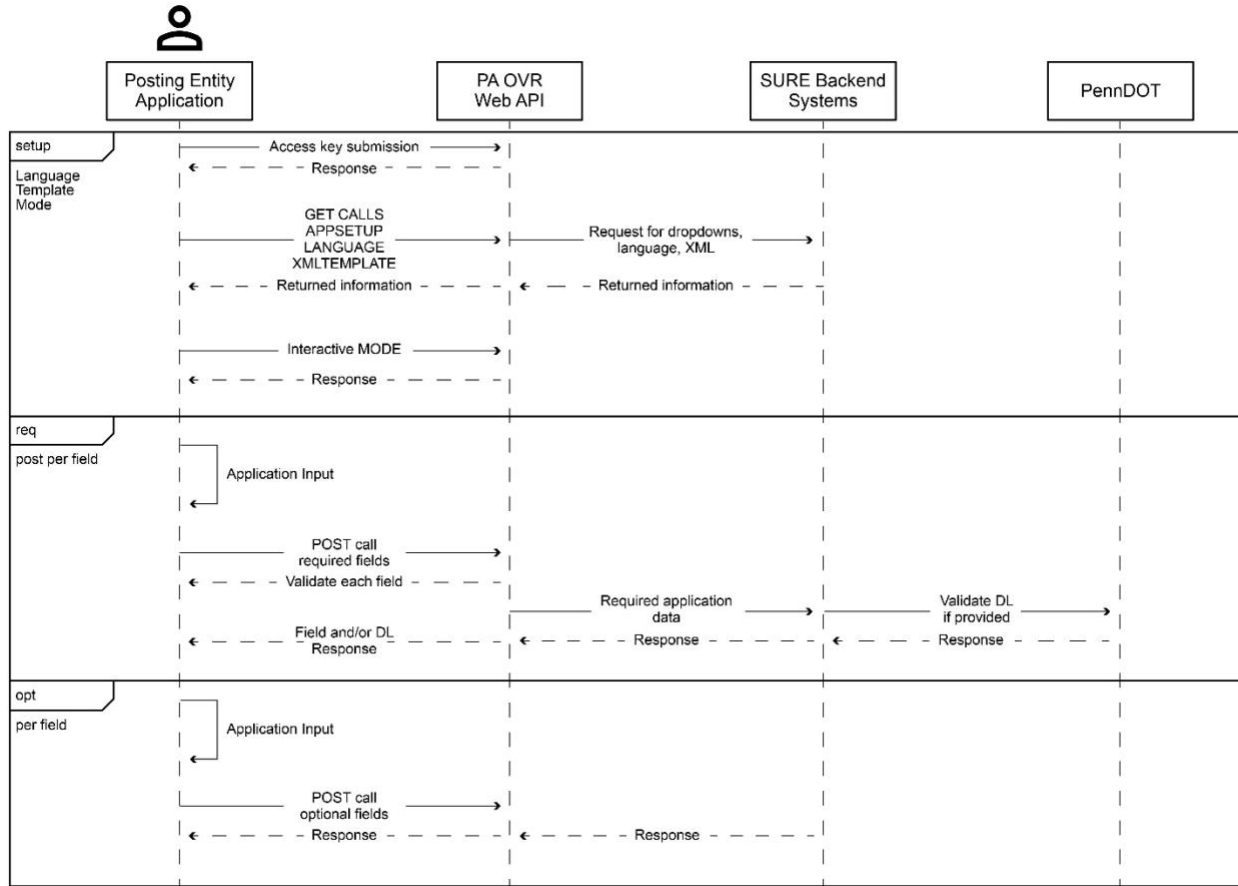
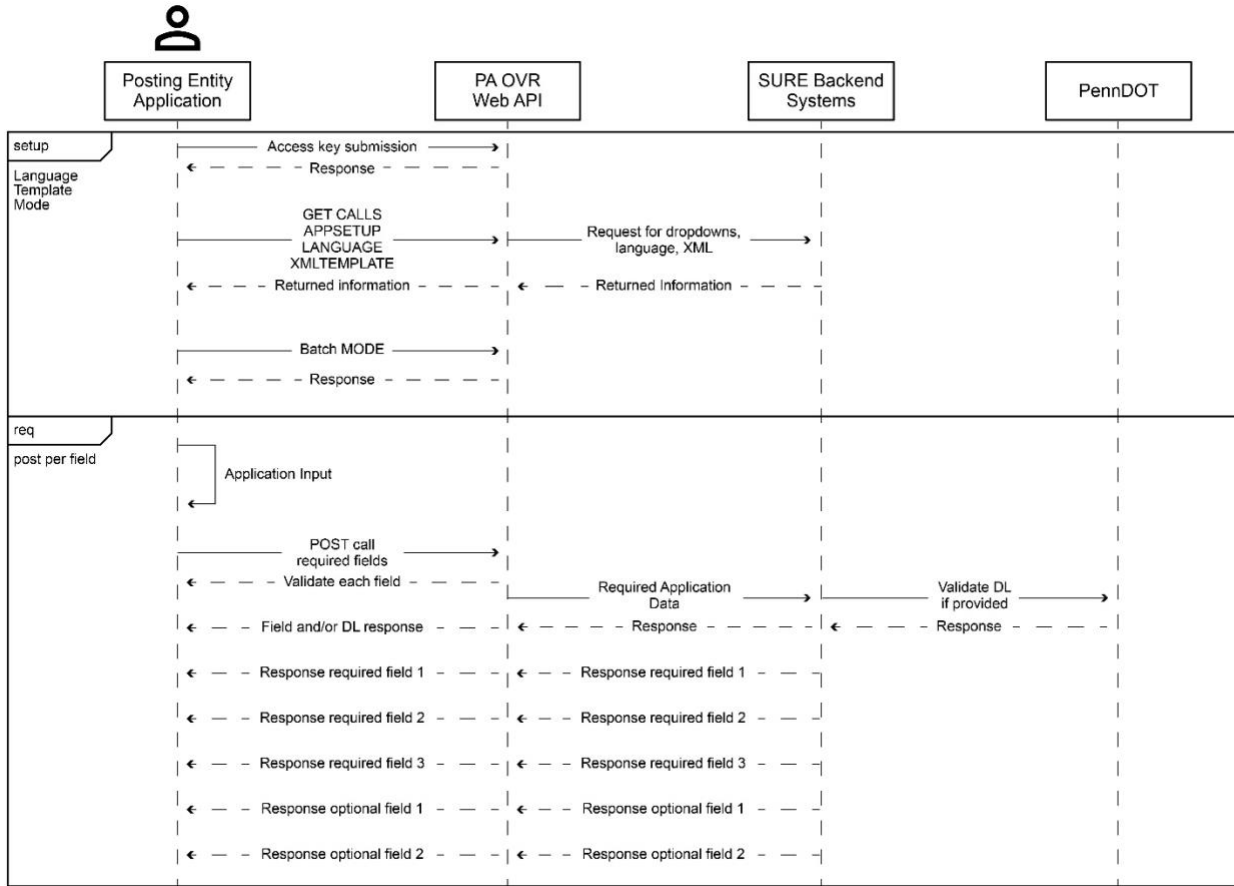


Figure 5: UML Sequence Diagram for Batch Mode

This figure outlines the process flow for batch mode.



Required Fields in Application

All API calls must include every XML tag, though every field isn't required for submission. Applications built using the Voter Registration and Vote by Mail API must capture all the data required to submit an application. The Field Definition Library denotes which fields are required and optional to collect. The [Field Definition Library](#) denotes which fields are required and optional as you develop your application. The Department of State recommends that you reference this document throughout your development process.

Signature Requirements

If the PennDOT ID is validated, the API will link the PennDOT signature to the application and discard the submitted signature file. If the PennDOT ID cannot be validated, the system will then attempt to process the uploaded signature image file (if available) to determine if it meets signature quality standards. If neither process can capture the applicant's signature, the posting organization should leverage the continueAppSubmit XML tag. In using this tag, the application will still be submitted into the SURE system but be placed into a 14-day "hold" status, pending signature receipt.

If a signature is not received after 14 days, the application will be placed into a "pending" state and the applicant will either receive a signature capture card from their county voter registration office or missing signature correspondence via the SURE system. Ultimately, the applicant must supply a signature to their county voter registration office to successfully complete their registration. Once an application is successfully submitted, the

API will provide a message back to the posting organization's custom application that includes an application ID number. If a valid email address is provided, the applicant will receive an email with their application ID number.

When collecting signatures for voter registration applications, signatures should be submitted with the following specifications to ensure the quality of the images. These guidelines are meant to increase the performance of the API and acceptance of the digital signatures into the SURE system.

Signatures should be:

- Formatted as TIF, TIFF, JPG, BMP, or PNG
- No larger than 60 pixels high and 180 pixels wide
- At least 75 dpi
- Captured in black and white only (no grey tones or color) and meet the following thresholds:
 - <98% white pixels
 - <90% black pixels
- Datatype transmitted should be Base64

Signatures should not be rotated.

[How Long Does the Process Take](#)

From applying to use the API to launching your application, expect the process to take roughly four months from start to finish.

[Sample Timeline](#)

This timeline includes estimated calendar days for the different phases of the process. Timelines will vary on a project-by-project basis.

- **Kickoff Activities:** 8 days
 - This phase of the process includes an introduction to the API. Your organization will also receive the resources to begin development.
- **Development Activities:** 34 days
 - Your organization may begin development of the application.
- **Review Activities:** 20 days
 - The Department of State will review and provide feedback on the work completed.
- **Application Testing Activities:** 30 days
 - This phase includes testing of the application and continued support from the Department of State.
- **Testing Activities:** 12 days
 - The Department of State will test the API for function and performance.
- **Approval Checklist:** 37 days
 - The completed application will be reviewed by the Department of State.
- **Final Approval:** 8 days
 - All checklists and remediation plans have been completed.
- **Production Activities:** 26 days
 - Your organization will receive PROD access and will prepare a deployment action plan.
- **Testing:** 5 days
 - This phase involves the testing of the function and performance of the application.
- **Post-Production Activities:** 12 days

- The Department of State will conduct a post-production check-in to monitor the status of your application. Quarterly reviews will also be scheduled to continue to monitor the upkeep and use of the application.

[More details on the process are available in the project plan overview.](#)

Process Status

Access to the staging and production environments will be granted at specific stages of the process. This chart documents at what phase you will receive access to the various environments. Status levels are granted and tracked separately for the online voter registration and vote by mail processes. If your organization has requested access for both functions, you may have different status levels for each.

Status Level	Milestone by Posting Organization	Staging/UAT Access	Production Access
Pending	Submitted registration	None	None
Pre-Approved	Signed and returned terms of use	Read Only	None
Approved	Voter registration or vote by mail application ready for development and testing	Read/Write	None
Certified	Voter registration or vote by mail application passed Department of State review and structured testing	None	Read/Write
Certified Testing	Continued development and testing of voter registration or vote by mail application enhancements for organizations looking to build upon their current application	Read/Write	Read/Write
Declined	Did not return terms of use or declined for other reasons	None	None
Removed	At request of posting organization	None	None

How to Make Calls

The Online Voter Registration and Vote by Mail API uses HTTP methods with a specific set of call parameters. Request payloads to the API should be formatted with a Content Type of JSON ("Content-Type: application/json") which passes XML within the body of the message. The API's responses are returned as a Content Type of JSON with XML body tags.

The Department of State's system processes online voter registration as separate requests from vote by mail applications. This table outlines the shared versus separate technical aspects of each.

Feature	Is the feature shared or specific to the application function (Online Voter Registration with optional Vote by Mail or Vote by Mail Only Application)?
Registration Process	Specific to each function
Terms of Use	Specific to each function
Security	Specific to each function
Organization with Affiliates	Specific to each function
Status (Use Same Values)	Specific to each function
Languages	Shared
XML Schema (Get Calls for Application Setup and Schema)	Specific to each function
Required Fields	Specific to each function
Error Responses (GETERRORVALUES)	Shared
County and Municipality Structure (GETMUNICIPALITIES)	Shared
Precedence and Processing Order	Specific to each function

Base URLs

All API calls will be sent through one of the two URLs below depending on if you are making the request to the STAGING or PROD servers. You will add on query parameters to all GET calls, and you can add in a body request to all POST calls.

Staging: <https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr>

Production: <https://PAOVRWebAPI.vote.pa.gov/SureOVRWebAPI/api/ovr>

Return Responses

Responses returned from API calls are JSON strings that include XML inside of them. The response will have quotation marks around it. The XML may be formatted on one line and may have a variable amount of spacing between tags but will surround strings with no extra spacing.

For example, for the GETLANGUAGES call, you might get a response like:

Headers:

```
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: X-AspNet-Version: 4.0.30319
Request-Context: appId=cid-v1:abcdefga-1111-2222-3333-123456789012
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 1; mode=block
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: Origin, X-Requested-With, Content-Type, Accept
Access-Control-Allow-Methods: GET,PUT,POST,DELETE,OPTIONS
```

Date: Fri, 22 Mar 2024 21:41:28 GMT
Content-Length: 330

Body:

```
"<OVRLookupData> <Languages> <LanguageCode>LANGENG</LanguageCode>  
<Language>English</Language> </Languages> <Languages>  
<LanguageCode>LANGSPN</LanguageCode> <Language>Spanish</Language> </Languages>  
<Languages> <LanguageCode>LANGTCN</LanguageCode> <Language>Chinese</Language>  
</Languages></OVRLookupData>"
```

For post call submissions with applications the responses will have an application id, submission date and the signature type applied to the application.

Error Responses

You might send an API call that is missing information, has an improper authorization key, or doesn't match the expected formatting. In that case, a response will be sent showing an error code. Responses are also JSON strings that include XML inside of them.

To get a full error description, use the [GETERRORVALUES API call](#).

Example error response from an API call with an invalid access key:

```
"<RESPONSE><APPLICATIONID></APPLICATIONID><APPLICATIONDATE></APPLICATIONDATE><SIGNATURE></SIGNATURE><ERROR>VR_WAPI_InvalidAccessKey</ERROR></RESPONSE>"
```

API Calls

[GET] GETAPPLICATIONSETUP

Description: Used to get the possible values for a voter registration application and a mail-in ballot application, if desired.

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API
sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "GETAPPLICATIONSETUP"	The command to request the values. Must be the value "GETAPPLICATIONSETUP".
sysparm_Language	0, 1, or 2	The language to provide the values in: 0 – English 1 – Spanish 2 – Traditional Chinese

Response:

Tag	Description	Tags Nested Inside
<NewDataSet>	The object to describe the application setup data. It is the single, outermost tag.	<Suffix> <Race> <UnitTypes> <AssistantType> <Gender> <PoliticalParty> <Text_OVRAppInDeclaration> <Text_OVRAppInAssistanceDeclaration> <NextVRDeadline> <NextElection> <Text_OVRMailInAppInDeclaration> <Text_OVRMailInAppInComplDate> <Text_OVRMailInBallotRecvdDate> <MailinAddressTypes> <Text_OVRMailInElectionName> <Text_OVRMailInAppInComplTime> <Text_OVRMailInBallotRecvdTime> <County> <States>
<Suffix>	An object describing one type of the suffix of a person's name. Will be one or more <Suffix> tag for each suffix that's available.	<NameSuffixCode> <NameSuffixDescription>
<NameSuffixCode>	The code for a single suffix (e.g., "Jr"), allowing the organization's application to build a drop down.	
<NameSuffixDescription>	The description for a single suffix (e.g., "Junior")	
<Race>	An object describing one type of a person's race. Will be one or more <Race> tag for each race that's available.	<RaceCode> <RaceDescription>
<RaceCode>	The code for a single race (e.g. "1"), allowing the organization's application to build a drop down.	

<RaceDescription>	The description for a single race (e.g. "Native American or Alaskan Native")	
<UnitTypes>	An object describing one type of a person's address. Will be one or more <UnitType> tag for each unit type that's available.	<UnitTypesCode> <UnitTypesDescription>
<UnitTypesCode>	The code for a single unit type (e.g. "APT"), allowing the organization's application to build a drop down.	
<UnitTypesDescription>	The description for a single unit type (e.g. "Apartment")	
<AssistanceType>	An object describing one type of a person's assistance type. Will be one or more <AssistanceType> tag for each assistance type that's available.	<AssistanceTypeCode> <AssistanceTypeDescription>
<AssistanceTypeCode>	The code for a single assistance type (e.g. "WC"), allowing the organization's application to build a drop down.	
<AssistanceTypeDescription>	The description for a single assistance type (e.g. "Wheelchair")	
<Gender>	An object describing one type of gender a person can have. Will be one or more <Gender> tag for each gender that's available.	<GenderCode> <GenderDescription>
<GenderCode>	The code for a single gender (e.g. "F"), allowing the organization's application to build a drop down.	

<GenderDescription>	The description for a single gender (e.g. "Female")	
<PoliticalParty>	An object describing one political party a person can be affiliated with. Will be one or more <PoliticalParty> tag for each party that's available.	< PoliticalPartyCode > <PoliticalPartyDescription>
<PoliticalPartyCode>	The code for a political party (e.g. "LN"), allowing the organization's application to build a drop down.	
<PoliticalPartyDescription>	The description for a political party (e.g. "Libertarian")	
<Text_OVRAppInDeclaration>	The text for the declaration an applicant needs to agree to. Will be only one tag.	<Text>
<Text>	The text for the single declaration.	
<Text_OVRAppInAssistanceDeclaration>	The text for the declaration for assistance an applicant needs to agree to. Will be only one tag.	<Text>
<Text>	The text for the single declaration.	
<NextVRDeadline>	The next voter registration deadline. Will only be one tag.	<NextVRDeadline>
<NextVRDeadline>	The text form for the date for the deadline.	
<NextElection>	The next election date. Will only be one tag.	<NextElection>
<NextElection>	The text form for the date of the election.	
<Text_OVRMailInApplnDeclaration>	The text for the declaration for a mail in ballot an applicant needs to agree to. Will only be one tag.	<Text>
<Text>	The text for the single declaration.	

<Text_OVRMailInApplnComplDate>	The date the mail in application needs to be completed by. Will only be one tag.	<Text_OVRMailInApplnComplDate>
<Text_OVRMailInApplnComplDate>	The text for the date for the deadline.	
<Text_OVRMailInBallotRecvdDate>	The date the mail in ballot needs to be received by. Will only be one tag.	<Text_OVRMailInBallotRecvdDate>
<Text_OVRMailInBallotRecvdDate>	The text for the date of the deadline.	
<MailinAddressTypes>	An object describing a mail-in ballot address type. Will be one or more <MailinAddressTypes> for each type that's available.	<MailinAddressTypesCode> <MailinAddressTypesDescription.
<MailinAddressTypesCode>	The code for a single mail-in address type (e.g. "R"), allowing the organization's application to build a drop down.	
<MailinAddressTypesDescription>	The description for a single mail-in address type (e.g. "Residential")	
<Text_OVRMailInElectionName>	An object describing the next election name. Will only be one tag.	<ElectionName>
<ElectionName>	The text name for the next election.	
<Text_OVRMailInApplnComplTime>	The time the mail-in ballot needs to be completed by. Will only be one tag.	<Time>
<Time>	The text for the time the mail-in ballot is due.	
<Text_OVRMailInBallotRecvdTime>	The time the mail-in ballot needs to be received by the county. Will only be one tag.	<RecvdTime>
<RecvdTime>	The text for the time the mail-in ballot is due.	
<County>	An object describing one county. Will be one or	<countyID> <Countyname>

	more <County> tag for each county that's available.	
<countyID>	The code for a single county (e.g. "2311")	
<Countyname>	The description for a single county (e.g. "Dauphin")	
<States>	An object describing one state. Will be one or more <States> tag for each state that's available.	<Code> <CodesDescription>
<Code>	The code for a single state (e.g. "PA"), allowing the organization's application to build a drop down.	
<CodesDescription>	The description for a single state (e.g. "Pennsylvania")	

Examples:

- Request the values in English on the staging server:

Request:

https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=GETAPPLICATIONSETUP&sysparm_Language=0

Response (example truncated to reduce space):

```
"<NewDataSet> <Suffix> <NameSuffixCode>II</NameSuffixCode>
<NameSuffixDescription>II</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>III</NameSuffixCode>
<NameSuffixDescription>III</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>IV</NameSuffixCode>
<NameSuffixDescription>IV</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>JR</NameSuffixCode>
<NameSuffixDescription>JR</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>SR</NameSuffixCode>
<NameSuffixDescription>SR</NameSuffixDescription> </Suffix>
```

- Request the values in Traditional Chinese on the production server:

Request:

https://PAOVRWebAPI.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=GETAPPLICATIONSETUP&sysparm_Language=2

Response (example truncated to reduce space):

```
"<NewDataSet> <Suffix> <NameSuffixCode>II</NameSuffixCode>
<NameSuffixDescription>II</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>III</NameSuffixCode>
<NameSuffixDescription>III</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>IV</NameSuffixCode>
<NameSuffixDescription>IV</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>JR</NameSuffixCode>
<NameSuffixDescription>JR</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>SR</NameSuffixCode>
<NameSuffixDescription>SR</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>V</NameSuffixCode>
<NameSuffixDescription>V</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>VI</NameSuffixCode>
<NameSuffixDescription>VI</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>VII</NameSuffixCode>
<NameSuffixDescription>VII</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>I</NameSuffixCode>
<NameSuffixDescription>I</NameSuffixDescription> </Suffix> <Race>
<RaceCode>A</RaceCode> <RaceDescription>亞裔</RaceDescription> </Race> <Race>
<RaceCode>T</RaceCode> <RaceDescription>兩種以上種族</RaceDescription> </Race>
<Race> <RaceCode>O</RaceCode> <RaceDescription>其他</RaceDescription> </Race>
<Race> <RaceCode>P</RaceCode> <RaceDescription>夏威夷島原住民或其他太平洋島原住民
</RaceDescription> </Race> <Race> <RaceCode>W</RaceCode> <RaceDescription>白
人</RaceDescription> </Race> <Race> <RaceCode>I</RaceCode> <RaceDescription>
美洲原住民或阿拉斯加原住民</RaceDescription> </Race> <Race> <RaceCode>H</RaceCode>
<RaceDescription>西班牙裔或拉丁裔</RaceDescription> </Race> <Race>
<RaceCode>B</RaceCode> <RaceDescription>黑人或非裔美國人</RaceDescription> </Race>
```

[GET] GETBALLOTAPPLICATIONSETUP

Description: Used to get the possible values for a mail-in ballot application.

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API
sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "GETBALLOTAPPLICATIONSETUP"	The command to request the values. Must be the value "GETBALLOTAPPLICATIONSETUP".

sysparm_Language	Number	The language to provide the values in: 0 – English 1 – Spanish 2 – Traditional Chinese
------------------	--------	---

Response:

Tag	Description	Tags Nested Inside
<NewDataSet>	The object to describe the application setup data. It is the single, outermost tag.	<Suffix> <UnitTypes> <NextElection> <Text_OVRMailInApplnDeclaration> <Text_OVRMailInApplnComplDate> <Text_OVRMailInBallotRecvdDate> <MailinAddressTypes> <Text_OVRMailInElectionName> <Text_OVRMailInApplnComplTime> <Text_OVRMailInBallotRecvdTime> <County> <States>
<Suffix>	An object describing one type of the suffix of a person's name. Will be one or more <Suffix> tag for each suffix that's available.	<NameSuffixCode> <NameSuffixDescription>
<NameSuffixCode>	The code for a single suffix (e.g., "Jr"), allowing the organization's application to build a drop down.	
<NameSuffixDescription>	The description for a single suffix (e.g., "Junior")	
<UnitTypes>	An object describing one type of a person's address. Will be one or more <UnitType> tag for each unit type that's available.	<UnitTypesCode> <UnitTypesDescription>
<UnitTypesCode>	The code for a single unit type (e.g. "APT"), allowing the organization's application to build a drop down.	
<UnitTypesDescription>	The description for a single unit type (e.g. "Apartment")	
<NextElection>	The next election date. Will only be one tag.	<NextElection>
<NextElection>	The text form for the date of the election.	
<Text_OVRMailInApplnDeclaration>	The text for the declaration for a mail in ballot an applicant needs to agree to. Will only be one tag.	<Text>
<Text>	The text for the single declaration.	

<Text_OVRMailInApplnComplDate>	The date the mail in application needs to be completed by. Will only be one tag.	<Text_OVRMailInApplnComplDate>
<Text_OVRMailInApplnComplDate>	The text for the date for the deadline.	
<Text_OVRMailInBallotRecvdDate>	The date the mail-in ballot needs to be received by. Will only be one tag.	<Text_OVRMailInBallotRecvdDate>
<Text_OVRMailInBallotRecvdDate>	The text for the date of the deadline.	
<MailinAddressTypes>	An object describing a mail-in ballot address type. Will be one or more <MailinAddressTypes> for each type that's available.	<MailinAddressTypesCode> <MailinAddressTypesDescription>
<MailinAddressTypesCode>	The code for a single mail-in address type (e.g. "R"), allowing the organization's application to build a drop down.	
<MailinAddressTypesDescription>	The description for a single mail-in address type (e.g. "Residential")	
<Text_OVRMailInElectionName>	An object describing the next election name. Will only be one tag.	<ElectionName>
<ElectionName>	The text name for the next election.	
<Text_OVRMailInApplnComplTime>	The time the mail-in ballot needs to be completed and received by the county. Will only be one tag.	<Time>
<Time>	The text for the time of the deadline.	
<Text_OVRMailInBallotRecvdTime>	The time the mail in ballot needs to be received by the county. Will only be one tag.	<RecvdTime>
<RecvdTime>	The text for the time for the deadline.	
<County>	An object describing one county. Will be one or more <County> tag for each county that's available.	<countyID> <Countyname>
<countyID>	The code for a single county (e.g. "2311")	
<Countyname>	The description for a single county (e.g. "Dauphin")	
<States>	An object describing one state. Will be one or more <States> tag for each state that's available.	<Code> <CodesDescription>

<Code>	The code for a single state (e.g. "PA")	
<CodesDescription>	The description for a single state (e.g. "Pennsylvania")	

Example:

- Request the values in English on the production server:

Request:

https://PAOVRWebAPI.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X5F6G7H8I9J0K&sysparm_action=GETBALLOTAPPLICATIONSETUP&sysparm_Language=0

Response (truncated to keep example short):

```
"<NewDataSet> <Suffix> <NameSuffixCode>II</NameSuffixCode>
<NameSuffixDescription>II</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>III</NameSuffixCode> <NameSuffixDescription>III</NameSuffixDescription>
</Suffix> <Suffix> <NameSuffixCode>IV</NameSuffixCode>
<NameSuffixDescription>IV</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>JR</NameSuffixCode> <NameSuffixDescription>JR</NameSuffixDescription>
</Suffix> <Suffix> <NameSuffixCode>SR</NameSuffixCode>
<NameSuffixDescription>SR</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>V</NameSuffixCode> <NameSuffixDescription>V</NameSuffixDescription>
</Suffix> <Suffix> <NameSuffixCode>VI</NameSuffixCode>
<NameSuffixDescription>VI</NameSuffixDescription> </Suffix> <Suffix>
<NameSuffixCode>VII</NameSuffixCode> <NameSuffixDescription>VII</NameSuffixDescription>
</Suffix> <Suffix> <NameSuffixCode>I</NameSuffixCode>
<NameSuffixDescription>I</NameSuffixDescription> </Suffix> <UnitTypes>
<UnitTypesCode>APT</UnitTypesCode> <UnitTypesDescription>APARTMENT</UnitTypesDescription>
</UnitTypes> <UnitTypes> <UnitTypesCode>BSM</UnitTypesCode>
<UnitTypesDescription>BASEMENT</UnitTypesDescription> </UnitTypes> <UnitTypes>
<UnitTypesCode>BOX</UnitTypesCode> <UnitTypesDescription>BOX #</UnitTypesDescription>
</UnitTypes> <UnitTypes> <UnitTypesCode>BLD</UnitTypesCode>
```

[GET] GETLANGUAGES

Description: Used to determine the languages offered by the voter registration and mail-in ballot system.

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API

sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "GETLANGUAGES"	The command to request the values. Must be the value "GETLANGUAGES".

Response:

Tag	Description	Tags Nested Inside
<OVRLookupData>	An object representing the collection of language data. The root object of the result.	<Languages>
<Languages>	An object representing a code and the name of a language	<LanguageCode> <Language>
<LanguageCode>	A code to represent the language	
<Language>	The name of the language	

Examples:

- Request the languages available on the staging server:

Request:

https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABC-D-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=GETLANGUAGES

Response:

```
"<OVRLookupData> <Languages> <LanguageCode>LANGENG</LanguageCode>
<Language>English</Language> </Languages> <Languages>
<LanguageCode>LANGSPN</LanguageCode> <Language>Spanish</Language>
</Languages> <Languages> <LanguageCode>LANGTCN</LanguageCode>
<Language>Chinese</Language> </Languages></OVRLookupData>"
```

[GET] GETXMLTEMPLATE

Description: Used to identify all the XML tags and format of the submission payload for the voter registration and optional combined vote by mail application..

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API
sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "GETXMLTEMPLATE"	The command to request the tags and format. Must be the value "GETXMLTEMPLATE".

Response:

Tag	Description	Tags Nested Inside
<APIOnlineApplicationData xmlns='OVRExternaldata'>	An object representing a template to fill with data and submit. Should also include the namespace.	<record>
<record>	An object representing a single registration record.	<batch> <FirstName> <MiddleName> <LastName> <TitleSuffix> <united-states-citizen> <eighteen-on-election-day> <isnewregistration> <name-update> <address-update> <ispartychange> <isfederalvoter> <DateOfBirth> <Gender> <Ethnicity> <Phone> <Email> <streetaddress> <streetaddress2> <unittype> <unitnumber> <city> <zipcode> <donthavePermtOrResAddresses> <county> <municipality> <mailingaddress> <mailingcity> <mailingstate> <mailingzipcode> <drivers-license> <ssn4> <signatureimage> <continueAppSubmit> <donthavebothDLandSSN> <politicalparty> <otherpoliticalparty> <needhelptovote> <typeofassistance> <preferredlanguage> <voterregnumber> <previousreglastname> <previousregfirstname>

		<previousregmiddlename> <previousregaddress> <previousregcity> <previousregstate> <previousregzip> <previousregcounty> <previousregyear> <declaration1> <assistedpersonname> <assistedpersonAddress> <assistedpersonphone> <assistancedeclaration2> <ispollworker> <bilingualinterpreter> <pollworkerspeaklang> <secondEmail> <ismailin> <istransferpermanent> <mailinaddresstype> <mailinballotaddr> <mailincity> <mailinstate> <mailinzipcode> <mailinward> <mailinlivesince> <mailindeclaration>
<batch>	A numerical value that tells the system whether this is a batch (1) or an interactive (0) filing.	
<FirstName>	The voter's first name.	
<MiddleName>	The voter's middle name or initial.	
<LastName>	The voter's last name	
<TitleSuffix>	The suffix of the voter's name (from available values from the GETAPPLICATIONSETUP call).	
<united-states-citizen>	A numerical value that represents if the voter is a citizen of the United States (1 for yes, 0 for no).	
<eighteen-on-election-day>	A numerical value that represents if the voter is 18 years old on or before election day (1 for yes, 0 for no).	
<isnewregistration>	A numerical value that represents if this is a new application for the voter (1 for yes, 0 for no).	
<name-update>	A numerical value that represents if this is a change of name on an existing registration (1 for yes, 0 for no).	

<address-update>	A numerical value that represents if this is a change of address on an existing registration (1 for yes, 0 for no).	
<ispartychange>	A numerical value that represents if this is a change of party on an existing registration (1 for yes, 0 for no).	
<isfederalvoter>	A numerical value that represents if this is a federal or state employee registering in the county of last residence (1 for yes, 0 for no).	
<DateOfBirth>	The date of birth of the voter (in “YYYY-MM-DD” format).	
<Gender>	The gender of the voter (from available values in the GETAPPLICATIONSETUP call).	
<Ethnicity>	The race of the voter (from available values in the GETAPPLICATIONSETUP call).	
<Phone>	The phone number of the voter (in the format “###-###-####”).	
<Email>	The email address of the voter.	
<streetaddress>	The voter’s first line of their address.	
<streetaddress2>	The voter’s second line of their address (if applicable).	
<unittype>	The voter’s residence unit type (from available values in the GETAPPLICATIONSETUP call).	
<unitnumber>	The voter’s residence unit number (as a string).	
<city>	The voter’s residence city.	
<zipcode>	The voter’s residence ZIP code (as a five character string of numbers).	
<donthavePermtOrResAddress>	A numerical value that represents if the voter has no street or permanent address (1 for yes, 0 for no).	
<county>	The voter’s county of residence (from available values in the GETAPPLICATIONSETUP call).	
<municipality>	The voter’s municipality of residence (from available values in the GETAPPLICATIONSETUP call).	
<mailingaddress>	The voter’s mailing street address.	
<mailingcity>	The voter’s mailing address city.	
<mailingstate>	The voter’s mailing address state.	
<mailingzipcode>	The voter’s mailing address ZIP code (as a string of either format “#####” or “#####-####”).	
<drivers-license>	The voter’s 8-digit driver’s license number (as a string with no spaces or other characters). See the section about	

	conditional requirements for the voter to validate their information.	
<ssn4>	The voter's last 4 digits of their social security number (as a string of digits). See the section about conditional requirements for the voter to validate their information.	
<signatureimage>	A TIFF, JPG, BMP, or PNG image encoded in Base64. See the section about conditional requirements for the voter to validate their information.	
<continueAppSubmit>	A numerical value that represents if the voter would like to proceed with the application if the driver's license fails to validate (1 for yes, 0 for no).	
<donthavebothDLandSSN>	A numerical value that represents if the voter does not have or want to provide a driver's license, PennDOT ID card, or a Social Security number (1 for yes, 0 for no).	
<politicalparty>	The voter's political party code (from the available values in the GETAPPLICATIONSETUP call).	
<otherpoliticalparty>	If the voter's political party is "Other", this is the value of that other party (up to 50 characters).	
<needhelptovote>	A numerical value that represents if the voter needs assistance when they go to vote (1 for yes, 0 for no).	
<typeofassistance>	The type of assistance the voter needs at the polling location (from the available values in the GETAPPLICATIONSETUP call).	
<preferredlanguage>	The voter's preferred language for assistance (as a string).	
<voterregnumber>	The voter's previous voter registration number.	
<previousreglastname>	If updating the voter's registration, this was their previous last name.	
<previousregfirstname>	If updating the voter's registration, this was their previous first name.	
<previousregmiddlename>	If updating the voter's registration, this was their previous middle name or initial.	
<previousregaddress>	If updating the voter's registration, this was their previous address.	
<previousregcity>	If updating the voter's registration, this was their previous city.	
<previousregstate>	If updating the voter's registration, this was their previous state.	

<previousregzip>	If updating the voter's registration, this was their previous ZIP code.	
<previousregcounty>	If updating the voter's registration, this was their previous county (from available values from the GETAPPLICATIONSETUP call).	
<previousregyear>	If updating the voter's registration, this was their previous registration year.	
<declaration1>	A numerical value representing if the voter accepts the contents of the declaration (1 for yes, 0 for no).	
<assistedpersonname>	If the person had assistance filling this out, this is the assistant's name.	
<assistedpersonAddress>	If the person had assistance filling this out, this is the assistant's address.	
<assistedpersonphone>	If the person had assistance filling this out, this is the assistant's phone number.	
<assisteddeclaration2>	If the person had assistance filling this out, a numerical value representing if the assistant accepts the contents of the declaration (1 for yes, 0 for no).	
<ispollworker>	A numerical value representing if the voter wishes to enroll as a poll worker (1 for yes, 0 for no).	
<bilingualinterpreter>	A numerical value representing if the voter wishes to enroll as a bilingual interpreter on election day (1 for yes, 0 for no).	
<pollworkerspeaklang>	The voter's languages they can speak (up to 50 characters).	
<secondEmail>	The voter's alternate email address (if applicable).	
<ismailin>	A numerical value representing if the mail-in ballot fields are required (1 for yes, 0 for no).	
<istransferpermanent>	A numerical value representing if the voter would like their "permanent status" transferred to their new county of residence (1 for yes, 0 for no).	
<mailinaddresstype>	A character value representing if the mail-in ballot field is a residential ("R"), mailing ("M"), or alternative ("A") address.	
<mailinballotaddr>	The street address of where a mail-in ballot should be mailed.	
<mailincity>	The city of where a mail-in ballot should be mailed.	
<mailinstate>	The state of where a mail-in ballot should be mailed.	

<mailinzipcode>	The ZIP code of where a mail-in ballot should be mailed.	
<mailinward>	The ward of where a mail-in ballot should be mailed (from available values from the GETAPPLICATIONSETUP call).	
<mailinlivedsince>	The year the voter has lived at this residence (as a four-digit numerical value).	
<mailindeclaration>	A numerical value representing if the voter accepts the mail-in ballot declaration (1 for yes, 0 for no).	

Examples:

- Request the tags and format available on the staging server:

Request:

https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABC D-12AB-34DE-0X0X-5F6G7H8I9JOK&sysparm_action=GETXMLTEMPLATE

Response (truncated to shorten the example):

```
"<APIOnlineApplicationData xmlns='OVRexternaldata'> <record> <batch></batch>
<FirstName></FirstName> <MiddleName></MiddleName> <LastName></LastName>
<TitleSuffix></TitleSuffix> <united-states-citizen></united-states-citizen> <eighteen-on-election-
day></eighteen-on-election-day> <isnewregistration></isnewregistration> <name-update></name-
update> <address-update></address-update> <ispartychange></ispartychange>
<isfederalvoter></isfederalvoter> <DateOfBirth></DateOfBirth> <Gender></Gender>
<Ethnicity></Ethnicity> <Phone></Phone> <Email></Email>
```

[GET] GETBALLOTXMLTEMPLATE

Description: Used to identify all the XML tags and format of the submission payload for the mail-in ballot system.

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API
sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "GETBALLOTXMLTEMPLATE"	The command to request the tags and format. Must be the value "GETBALLOTXMLTEMPLATE".

Response:

Tag	Description	Tags Nested Inside
-----	-------------	--------------------

<APIBallotApplicationData xmlns='Ballotexternaldata'>	An object representing a template to be filled out and submitted. Will include the namespace.	<record>
<record>	An object representing one application record.	<batch> <FirstName> <MiddleName> <LastName> <TitleSuffix> <DateOfBirth> <Phone> <Email> <drivers-license> <county> <streetaddress> <streetaddress2> <unittype> <unitnumber> <city> <zipcode> <mailinlivesince> <ElectionDistrict> <Mail-Ballot-to-Different-Address> <Is-Mailing-Country-USA> <mailingcountry> <mailingaddress> <mailingaddress2> <mailingunittype> <mailingunitnumber> <mailingcity> <mailingstate> <mailingzipcode> <Address-Is-My> <Annual-MailIn-Request> <mailindeclaration>
<batch>	A numerical value that tells the system whether this is a batch (1) or an interactive (0) submission.	
<FirstName>	The voter's first name.	
<MiddleName>	The voter's middle name or initial.	
<LastName>	The voter's last name	
<TitleSuffix>	The suffix of the voter's name (from available values from the GETAPPLICATIONSETUP call).	
<DateOfBirth>	The date of birth of the voter (in "YYYY-MM-DD" format).	
<Phone>	The phone number of the voter (in the format "###-###-####").	
<Email>	The email address of the voter.	

<drivers-license>	The voter's 8-digit driver's license number (as a string with no spaces or other characters). See the section about conditional requirements for the voter to validate their information.	
<county>	The voter's county of residence (from available values in the GETBALLOTAPPLICATIONSETUP call).	
<streetaddress>	The voter's first line of their address.	
<streetaddress2>	The voter's second line of their address (if applicable).	
<unittype>	The voter's residence unit type (from available values in the GETBALLOTAPPLICATIONSETUP call).	
<unitnumber>	The voter's residence unit number (as a string).	
<city>	The voter's residence city.	
<zipcode>	The voter's residence ZIP code (as a five-character string of numbers).	
<mailinlivesince>	The year the voter has lived at this residence (as a four-digit numerical value).	
<ElectionDistrict>	The voter's residence municipality code (from available values in the GETMUNICIPALITIES call).	
<Mail-Ballot-to-Different-Address>	A numerical value representing if the ballot should be sent to a different address than the residential address (1 for yes, 0 for no).	
<Is-Mailing-Country-USA>	A numerical value representing if the mailing address is within the US (1 for yes, 0 for no).	
<mailingcountry>	The country of where a mail-in ballot should be mailed.	
<mailingaddress>	The street address of where a mail-in ballot should be mailed.	
<mailingaddress2>	The street address second line of where a mail-in ballot should be mailed (if applicable).	
<mailingunittype>	The unit type of where a mail-in ballot should be mailed (from available values in the GETBALLOTAPPLICATIONSETUP call).	
<mailingunitnumber>	The unit number of where a mail-in ballot should be mailed (as a string).	
<mailingcity>	The city of where a mail-in ballot should be mailed.	
<mailingstate>	The state of where a mail-in ballot should be mailed.	
<mailingzipcode>	The ZIP code of where a mail-in ballot should be mailed.	
<Address-Is-My>	A string indicating what type of address the mailing address is.	
<Annual-MailIn-Request>	A numerical value representing if the voter would like to be put on the "permanent mail-in list" on their voter registration record (1 for yes, 0 for no).	
<mailindeclaration>	A numerical value representing if the voter accepts the contents of the declaration (1 for yes, 0 for no).	

Examples:

- Request the tags and format available on the staging server:

Request:

https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=GETBALLOTXMLTEMPLATE

Response:

```
"<APIBallotApplicationData xmlns='Ballotexternaldata'> <record>
<batch></batch> <!-- APPLICANT INFORMATION --> <FirstName></FirstName>
<MiddleName></MiddleName> <LastName></LastName>
<TitleSuffix></TitleSuffix> <DateOfBirth></DateOfBirth> <Phone></Phone>
<Email></Email> <!-- IDENTIFICATION --> <drivers-license></drivers-
license> <!-- CURRENT ADDRESS --> <county></county>
<streetaddress></streetaddress> <streetaddress2></streetaddress2>
<unittype></unittype> <unitnumber></unitnumber> <city></city>
<zipcode></zipcode> <mailinlivesince></mailinlivesince>
<ElectionDistrict></ElectionDistrict> <Mail-Ballot-to-Different-
Address></Mail-Ballot-to-Different-Address> <!-- MAILING ADDRESS -->
<Is-Mailing-Country-USA></Is-Mailing-Country-USA>
<mailingcountry></mailingcountry> <mailingaddress></mailingaddress>
<mailingaddress2></mailingaddress2> <mailingunittype></mailingunittype>
<mailingunitnumber></mailingunitnumber> <mailingcity></mailingcity>
<mailingstate></mailingstate> <mailingzipcode></mailingzipcode>
<Address-Is-My></Address-Is-My> <!-- ANNUAL MAIL-IN REQUEST -->
<Annual-MailIn-Request></Annual-MailIn-Request> <!-- DECLARATION -->
<mailindeclaration></mailindeclaration>
</record></APIBallotApplicationData>"
```

- Request the tags and format available on the production server:

Request:

https://PAOVRWebAPI.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=GETBALLOTXMLTEMPLATE

Response:

(same response)

[GET] GETERRORVALUES

Description: Used to retrieve the error code and message pair for the for the voter registration and mail-in ballot systems.

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API
sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "GETERRORVALUES"	The command to request the tags and format. Must be the value "GETERRORVALUES".
sysparm_Language	Number	The language to provide the error in: 0 – English 1 – Spanish 2 – Traditional Chinese

Response:

Tag	Description	Tags Nested Inside
<OVRLookupData>	An object representing the collection of error messages	<MessageText>
<MessageText>	An object representing one error code and text	<ErrorCode> <ErrorText>
<ErrorCode>	An error code	
<ErrorText>	The text of what the error code represents in the language selected in the sysparm_Language parameter.	

Examples:

- Request the error code and message in English from the staging server:

Request:

https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9JOK&sysparm_action=GETERRORVALUES&sysparm_Language=0

Response (example truncated to save space):

```
"<OVRLookupData> <MessageText> <ErrorCode>VR_WAPI_InvalidAccessKey</ErrorCode>  
<ErrorText>Access Key is Invalid.</ErrorText> </MessageText> <MessageText>  
<ErrorCode>VR_WAPI_InvalidAction</ErrorCode> <ErrorText>Action not  
found.</ErrorText> </MessageText> <MessageText>  
<ErrorCode>VR_WAPI_InvalidAPIbatch</ErrorCode> <ErrorText>Batch value is  
Invalid.</ErrorText> </MessageText> <MessageText>  
<ErrorCode>VR_WAPI_InvalidOVRCounty</ErrorCode> <ErrorText>Your county of  
residence is required.</ErrorText> </MessageText> <MessageText>  
<ErrorCode>VR_WAPI_InvalidOVRDL</ErrorCode> <ErrorText> please provide valid DL or  
pick continuesubmit checkbox.</ErrorText> </MessageText> <MessageText>
```

- Request the error code and message in Traditional Chinese from the production server:

Request:

https://PAOVRWebAPI.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=GETERRORVALUES&sysparm_Language=2

Response:

```
"<OVRLookupData> <MessageText> <ErrorCode>VR_WAPI_InvalidAccessKey</ErrorCode>
<ErrorText>訪問密鑰無效。</ErrorText> </MessageText> <MessageText>
<ErrorCode>VR_WAPI_InvalidAction</ErrorCode> <ErrorText>未找到操作。</ErrorText>
</MessageText> <MessageText> <ErrorCode>VR_WAPI_InvalidAPIbatch</ErrorCode>
<ErrorText>批次值無效。</ErrorText> </MessageText> <MessageText>
<ErrorCode>VR_WAPI_InvalidOVRCounty</ErrorCode> <ErrorText>您居住的縣為必填。
</ErrorText> </MessageText> <MessageText>
<ErrorCode>VR_WAPI_InvalidOVRDL</ErrorCode> <ErrorText>請提供有效的 DL 或選擇
continuesubmit 複選框。</ErrorText> </MessageText> <MessageText>
<ErrorCode>VR_WAPI_InvalidOVRDLformat</ErrorCode> <ErrorText>請輸入有效的 8 位 PA
駕駛執照或 PennDOT 身份證號碼。</ErrorText> </MessageText> <MessageText>
<ErrorCode>VR_WAPI_InvalidOVRDOB</ErrorCode> <ErrorText>請輸入一個有效的出生日期。
</ErrorText> </MessageText> <MessageText>
<ErrorCode>VR_WAPI_InvalidOVRemail</ErrorCode> <ErrorText>電子郵件格式不正確。請更正
並再試一次。</ErrorText> </MessageText> <MessageText>
<ErrorCode>VR_WAPI_InvalidOVRmailingzipcode</ErrorCode> <ErrorText>郵遞區號須為 5
位數或 9 位數 (郵遞區號 +4) </ErrorText> </MessageText> <MessageText>
```

[GET] GETMUNICIPALITIES

Description: Used to return the municipalities in a given county (in English only) for the for the voter registration and mail-in ballot systems.

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API
sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "GETMUNICIPALITIES"	The command to request the tags and format. Must be the value "GETMUNICIPALITIES"
sysparm_Language	Number	The language to provide the results in: 0 – English 1 – Spanish 2 – Traditional Chinese Note: County and municipality names will be in English
sysparm_County	String	Name of the county in all caps (ex: "ADAMS", "ALLEGHENY", etc.)

Response:

Tag	Description	Tags Nested Inside
<OVRLookupData>	An object representing the collection of municipalities	<Municipality>
<Municipality>	An object representing a single municipality	<MunicipalityType> <MunicipalityID> <MunicipalityIDname> <CountyID> <CountyName>
<MunicipalityType>	The numerical type of municipality	
<MunicipalityID>	An ID code representing the municipality	
<MunicipalityIDname>	The name of the municipality	
<CountyID>	The numerical ID number of the county	
<CountyName>	The county name	

Examples:

- Request the municipalities for Jefferson County in English from the staging server:

Request:

<https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm AuthKey=1234ABC D-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm action=GETMUNICIPALITIES&sysparm County=JEFFERSON&sysparm Language=0>

Response (example truncated to save space):

```
"<OVRLookupData> <Municipality> <MunicipalityType>0</MunicipalityType>
<MunicipalityID /> <MunicipalityIDname /> <CountyID>0</CountyID> <CountyName
/> </Municipality> <Municipality> <MunicipalityType>4</MunicipalityType>
<MunicipalityID>MN01</MunicipalityID> <MunicipalityIDname>BARNETT
TOWNSHIP</MunicipalityIDname> <CountyID>2322</CountyID>
<CountyName>JEFFERSON</CountyName> </Municipality> <Municipality>
<MunicipalityType>4</MunicipalityType> <MunicipalityID>MN02</MunicipalityID>
<MunicipalityIDname>BEAVER TOWNSHIP</MunicipalityIDname> <CountyID>2322</CountyID>
<CountyName>JEFFERSON</CountyName> </Municipality>
```

- Request the municipalities for Dauphin County in Spanish from the production server:

Request:

<https://PAOVRWebAPI.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm action=GETMUNICIPALITIES&sysparm County=DAUPHIN&sysparm Language=1>

Response (example truncated to save space):

```
"<OVRLookupData> <Municipality> <MunicipalityType>0</MunicipalityType>
<MunicipalityID /> <MunicipalityIDname /> <CountyID>0</CountyID> <CountyName
/> </Municipality> <Municipality> <MunicipalityType>4</MunicipalityType>
<MunicipalityID>MN011</MunicipalityID> <MunicipalityIDname>BERRYSBURG
BOROUGH</MunicipalityIDname> <CountyID>2311</CountyID>
<CountyName>DAUPHIN</CountyName> </Municipality> <Municipality>
<MunicipalityType>4</MunicipalityType> <MunicipalityID>MN041</MunicipalityID>
<MunicipalityIDname>CONEWAGO TOWNSHIP</MunicipalityIDname>
```

```

<CountyID>2311</CountyID>    <CountyName>DAUPHIN</CountyName>  </Municipality>
<Municipality>    <MunicipalityType>4</MunicipalityType>
<MunicipalityID>MN012</MunicipalityID>    <MunicipalityIDname>DAUPHIN
BOROUGH</MunicipalityIDname>    <CountyID>2311</CountyID>
<CountyName>DAUPHIN</CountyName>  </Municipality>

```

[POST] SETAPPLICATION

Description: Used to submit a voter registration application and optional mail-in ballot.

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API
sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "SETAPPLICATION"	The command to submit an application. Must be the value "SETAPPLICATION".
sysparm_Language	Number	The language to provide the results in: 0 – English 1 – Spanish 2 – Traditional Chinese Note: County and municipality names will be in English

Body Params:

The body for this will include the contents received from the GETXMLTEMPLATE API call filled in with the applicant’s data. See the fields above or the [Field Definition Document](#) to confirm the types and formats of fields that should be sent.

Response:

Tag	Description	Tags Nested Inside
<RESPONSE>	An object representing the response from the server about the application.	<APPLICATIONID> <APPLICATIONDATE> <SIGNATURE> <ERROR> <MAILINAPPLICATIONDATE>
<APPLICATIONID>	An application ID number (or empty if error).	
<APPLICATIONDATE>	The date and time the application was accepted (or empty if error).	
<SIGNATURE>	A string that indicates the status of the submitted signature (for acceptance) or empty (for error). Examples: "Submitted Without Signature"	

	"Submitted With PennDOT Signature" "Submitted With Uploaded Signature"	
<ERROR>	An error code if there was a problem with the application (from the available values from the GETERRORVALUES call).	
<MAILINAPPLICATIONDATE>	The date and time the mail-in ballot application was accepted (only available if the mail-in ballot field was set to 1 and there was no error).	

Examples:

- Submit an application in English to the staging server without signature using a value of 1 for the ContinueAppSubmit tag :

Request:

https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=SETAPPLICATION&sysparm_Language=0

Response:

"<RESPONSE><APPLICATIONID>31863</APPLICATIONID><APPLICATIONDATE>Jun 5 2017 1:51PM</APPLICATIONDATE><SIGNATURE>Submitted Without Signature</SIGNATURE><ERROR></ERROR></RESPONSE>"

- Submit an application in English to the production server with PennDOT ID signature:

Request:

https://PAOVRWebAPI.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=SETAPPLICATION&sysparm_Language=0

Response:

"<RESPONSE><APPLICATIONID>31863</APPLICATIONID><APPLICATIONDATE>Jun 5 2017 1:51PM</APPLICATIONDATE><SIGNATURE>Submitted With PennDOT Signature</SIGNATURE><ERROR></ERROR></RESPONSE>"

[POST] SETBALLOTAPPLICATION

Description: Used to submit a mail-in ballot application.

Query Params:

Key	Value	Description
JSONv2		Required parameter for the API
sysparm_AuthKey	GUID	The authorization key provided to you by the Department of State
sysparm_action	String "SETBALLOTAPPLICATION"	The command to submit an application. Must be the value "SETBALLOTAPPLICATION".

sysparm_Language	Number	The language to provide the results in: 0 – English 1 – Spanish 2 – Traditional Chinese Note: County and municipality names will be in English
------------------	--------	--

Body Params:

The body for this will include the contents received from the GETBALLOTXMLTEMPLATE API call filled in with the applicant’s data.

Response:

Tag	Description	Tags Nested Inside
<RESPONSE>	An object representing the response from the server about the application.	<APPLICATIONID> <APPLICATIONDATE> <SIGNATURE> <ERROR>
<APPLICATIONID>	An application ID number (or empty if error).	
<APPLICATIONDATE>	The date and time the application was accepted (or empty if error).	
<SIGNATURE>	An empty object. (Only used for voter registration, not mail-in ballots.)	
<ERROR>	An error code if there was a problem with the application (from the available values from the GETERRORVALUES call).	

Examples:

- Submit an application in English to the staging server:
Request:
https://PAOVRWebAPI.beta.vote.pa.gov/SureOVRWebAPI/api/ovr?JSONv2&sysparm_AuthKey=1234ABCD-12AB-34DE-0X0X-5F6G7H8I9J0K&sysparm_action=SETBALLOTAPPLICATION&sysparm_Language=0
Response:
"<RESPONSE><APPLICATIONID>407271</APPLICATIONID><APPLICATIONDATE>May 09, 2024 4:37:09 PM</APPLICATIONDATE><SIGNATURE></SIGNATURE><ERROR></ERROR></RESPONSE>"

References and Resources

These resources are referenced throughout the document and provide additional information to support your organization in using the API.

Register to Use the API

- [Register on the Online Voter Registration and Vote by Mail API website](#)
- [Register and have a Keystone Login Account](#)

Terms of use

- [Online Voter Registration API Terms of Use](#)
- [Vote by Mail API Terms of Use](#)

Project Plan

- [Sample Timeline and Milestones](#)

Field Definition Library

- [Online Voter Registration and Vote by Mail API Field Dictionary](#)
- [XML Guidelines](#)
 - You must complete registration to use the API to access this.

Voter Registration Drives

- [PA Voter Registration Guide](#)

Contact the Department of State

If you have additional questions about the Online Voter Registration and Vote by Mail API, please contact the Department of State at ra-voterreg@pa.gov.

Revision History

Date	Version	Description
7/26/2016	0.0	Initial Publication
4/16/2017	0.1	Ready for Review
4/21/2017	0.2	Feedback Incorporated, Added Release Notes and Project Timeline
5/3/2017	0.3	Added batch vs interactive differentiation, field processing order, expected response codes, clarifications to technical description
5/23/2017	1.0	Clarification in multi-language support, interactive mode, and batch mode with continueAppSubmit=1 and miscellaneous formatting and clarity issues.
6/22/2017	1.1	<ul style="list-style-type: none">• Added responses for ID verification on successful and error submissions. Updated Release Notes.• Field Description spreadsheet updated with new parties (Green and Libertarian) and GetApplicationSetup
10/2/2017	1.2	<ul style="list-style-type: none">• Appendix added for full GetApplicationSetup output• Format Changes
11/16/2017	1.3	<ul style="list-style-type: none">• Corrected eSignature size and defined contrast thresholds• Removed duplicate Trailer UNITTYPE in GETAPPLICATIONSETUP example

6/1/2018	1.4	<ul style="list-style-type: none"> • Added error response messaging for eSignatures not meeting the 96 dpi threshold. • Clarification on processing in Batch with ID scenarios • Update GETAPPLICATIONSETUP to update Political Party, Race and UNITTYPE • Updated Field Spec reference • Updated Release Notes
2/20/2020	1.5	<ul style="list-style-type: none"> • Added ACT 77 Mail-In Ballot information, details, links to updated documents and changes to GET calls
5/19/2020	1.6	<ul style="list-style-type: none"> • Added Transfer Permanent Status Flag (Act 12) and subsequent changes to GET calls • Added Mail-In Address Mapping Guidelines
10/26/2022, updated on 11/2/2023	1.7	<ul style="list-style-type: none"> • Updated URLs • Added Traditional Chinese • Updated Release Notes Section • Electronic Signature Specifications Section • Questions in Mail-In Ballot Address Section and Links not working.
2/20/2024	1.8	<ul style="list-style-type: none"> • Added Mail-In Ballot Application Features of Vote by Mail (VBM)
5/30/2024	1.9	<ul style="list-style-type: none"> • Revised to incorporate mail-in ballot only API and other technical language adjustments