



“The Complete EDI Solution”

**Application Programming Interface (API)
Patient Data Request version 4.0.0
April 19, 2018**

Office Ally, Inc.
1300 SE Cardinal Court
Suite #190
Vancouver, WA 98683
www.officeally.com

Table of Contents

INTRODUCTION	3
Overview	3
TERMS OF USE.....	3
Scope.....	3
Use	3
Attribution.....	3
Modification or False Representation of Content	3
Right to Limit.....	3
Service Termination	4
Changes.....	4
Disclaimer of Warranties	4
Limitations on Liability	4
General Representations	4
Indemnification.....	4
Miscellaneous	5
Disputes	5
No Waiver of Rights	5
REGISTRATION	6
AUTHENTICATION	7
JQUERY	7
PATIENT SEARCH.....	8
DATA REQUEST	9
Allergy	9
Assessment & Plan of Treatment	10
Care Team Member	11
Goal	12
Health Concern	13
Immunization	14
Implantable Device	15
Lab Test	17
Lab Result.....	18
Medication	19
Patient Demographics.....	20
Patient CCDA.....	21
Problem.....	22
Procedure.....	23
Vital Sign.....	24

INTRODUCTION

Overview

In the 2015 Edition of Health IT Certification Criteria, ONC established new criteria §170.315(g)(7), §170.315(g)(8) , and §170.315(g)(9) which requires a certified health IT to demonstrate the ability provide patient-facing application access to the Common Clinical Data Set via an API. Application Programming Interface (API) refers to technology which allows one software program to access the services provided by another software program.

TERMS OF USE

Office Ally, Inc. offers some of its data in machine readable format through an Application Programming Interface (API). This service is offered subject to your acceptance of the terms and conditions (Agreement) contained herein.

Scope

All of the content, documentation, code, and related materials made available to you through the API are subject to these terms. Access to use the API or its content constitutes acceptance of this Agreement.

Use

You may use any of the Office Ally, Inc. API's to develop a service or services to search, display, analyze, retrieve, view, and otherwise "get" information from Office Ally, Inc.

Attribution

You may use the Office Ally, Inc. name in order to identify the source of the API content, and are subject to these rules. You shall not use the names of the Office Ally, Inc. to imply endorsement, approval, sponsorship, association, or authorization of any product, service, or entity (not-for-profit, commercial, or otherwise).

Modification or False Representation of Content

You may not modify any content accessed through the API in any manner which could reasonably be interpreted or construed as conveying the false impression that Office Ally, Inc. is the source of such content.

Right to Limit

Your use of the API may be subject to certain limitations on access, calls, or use as set forth within this Agreement or otherwise provided by Office Ally, Inc. If Office Ally, Inc. reasonably believes that you have attempted to exceed or circumvent these limits, your ability to use the API may be permanently or temporarily blocked. Office Ally, Inc. may monitor your use of the API to improve the service or to ensure compliance with this Agreement.

Service Termination

If you wish to terminate this Agreement, you may do so by refraining from further use of the API. In its sole discretion, Office Ally, Inc. reserves the right to discontinue providing the API or data for any reason. Office Ally, Inc. may stop providing the API to you if, in our opinion, your use violates any Office Ally, Inc. policy. See our Privacy and Legal Notices and Website Terms and Conditions.

You agree that you are using this API “as is” and at your own risk. All provisions of this Agreement which by their nature should survive termination shall survive termination including, without limitation, warranty disclaimers, indemnity, and limitations of liability.

Changes

Office Ally, Inc. reserves the right, at its sole discretion, to modify or replace this Agreement, in whole or in part. Your continued use of or access to the API following posting of any changes to this Agreement constitutes acceptance of those modified terms. Office Ally, Inc. may, in the future, offer new services and/or features through the API. Such new features and/or services shall be subject to the terms and conditions of this Agreement. Office Ally, Inc. may change any aspect of the API or the data offered at any time for any reason.

Disclaimer of Warranties

The API and all data provided through the API is provided “as is” and on an “as-available” basis. All express or implied conditions, representations, and warranties, including without limitation the warranties of merchantability, fitness for a particular purpose, and non-infringement are hereby disclaimed. Office Ally, Inc. makes no warranty that the API data will be error-free or that access thereto will be continuous or uninterrupted.

Limitations on Liability

In no event will Office Ally, Inc., or its officers, employees, contractors, or agents be liable for any lost revenue, profit, or data, or for special, indirect, consequential, incidental, or punitive damages, however caused and regardless of theory of liability arising out of the use or inability to use the API or data offered through the API, even if Office Ally, Inc. has been advised of the possibility of such damages. The foregoing limitations shall apply even if the above stated warranty disclaimer fails to its essential purpose.

General Representations

You hereby warrant that:

- (1) Your use of the API will be in strict accordance with the Office Ally, Inc. Privacy Policy, this Agreement, and all applicable laws and regulations; and
- (2) Your use of the API will not infringe or misappropriate the intellectual property rights of any third party.

Indemnification

You agree to indemnify and hold harmless the Office Ally, Inc., its officers, employees, contractors, agents, and the like from and against any and all liabilities, claims, and expenditures, including attorney’s fees, arising out of your use of the API, including but not limited to violation of this Agreement.

Miscellaneous

This Agreement constitutes the entire Agreement between Office Ally, Inc. and you concerning the subject matter hereof, and may be modified only by the posting of a revised version on this page by Office Ally, Inc.

Disputes

Any disputes arising out of this Agreement and access to or use of the API shall be governed by Federal law.

No Waiver of Rights

Office Ally, Inc.'s failure to exercise or enforce any right or provision of this Agreement shall not constitute waiver of such right or provision.

REGISTRATION

Office Ally, Inc. uses **OAuth2** tokens for Authentication to its API. To Authentication you will need 4 elements:

- A Client ID
- A Client Secret
- A Username
- A Password

A Client ID and Secret are provided by Office Ally, Inc. as a means to protect our users. Each application that wishes to have access to our data will have to request a Client ID and Secret from Office Ally, Inc.

REQUEST

URL: POST /clients/register

Query Parameters:

name (string): Vendor Name

Required Headers:

Origin: Should be automatically sent by the client.

RESPONSE

Type: application/json

```
{
  "client_id": "string",
  "client_secret": "string",
}
```

Exceptions:

400 - Bad Request: Invalid Search Elements

AUTHENTICATION

JQUERY

A Username and Password can be provided to the Patient by a provider. The provider can go to the patient record in the EHR system and generate an API token. The API token generation requires the patient email address to be on file. Once completed, the credentials can be printed and given to the API user.

USING JQUERY: To Request an OAuth2 token, the user will have to send a post request outlined in the example below:

```
$.ajax({
  type: 'POST',
  url: 'token',
  dataType: 'application/x-www-form-urlencoded',
  data: 'grant_type=password' +
        '&username=' + [Username] +
        '&password=' + [Password] +
        '&client_id=' + [Client ID] +
        '&client_secret=' + [Client Secret]
});
```

The response will be in the following format:

```
{
  "access_token": "[Bearer Token]",
  "token_type": "bearer",
  "expires_in": [Expiration in seconds],
  ".issued": "[Issued Date]",
  ".expires": "[Expiration Date]"
}
```

Note. The first time the request is made from a client for a particular user, the authentication will fail. The user will have to activate the new authentication through a link sent by email.

Once the authentication info is received, the 'Bearer Token' will have to be included in all future requests through the Authorization header. Fill the authorization header as follows:

```
$.ajax({
  type: 'GET',
  url: 'patients'
  headers: { 'Authorization': 'Bearer ' + [Bearer Token] }
});
```

PATIENT SEARCH

REQUEST

URL: GET/patients

Query Parameters:

firstName (string : optional): Patient First Name

lastName (string : optional): Patient Last Name

sex (string : optional): Patient Gender ('M' for Male, 'F' for Female, 'U' for Unknown)

dateOfBirth (date-time : optional): Patient Date of Birth (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "FirstName": "string",
    "LastName": "string",
    "Sex": "string",
    "DateOfBirth": "date-time",
    "Race": [{
      "HL7Code": "string",
      "RollDownID": "string",
      "Description": "string"
    },...],
    "Ethnicity": [{
      "HL7Code": "string",
      "RollDownID": "string",
      "Description": "string",
    },...],
    "LanguageCode": "string",
    "SmokingStatus": [{
      "StartDate": "date-time",
      "EndDate": "date-time",
      "Description": "string",
      "Frequency": "string",
    },...],
  },
  ...
]
```

Exceptions:

400 - Bad Request: Invalid Search Elements

401 - Unauthorized: Invalid token, or unauthorized record.

DATA REQUEST

Allergy

REQUEST

URL: GET/patients/{patientId}/allergies

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "ID": "int",
    "DrugID": "int",
    "CodeSystem": "string",
    "DrugName": "string",
    "EncounterDate": "date-time",
    "ReactionDate": "date-time",
    "Reaction": "string",
    "Status": "string",
    "Severity": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Assessment & Plan of Treatment

REQUEST

URL: GET/patients/{patientId}/assessments

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "EncounterID": "int",
    "EncounterDate": "date-time",
    "PlanOfCare": "string",
    ...
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Care Team Member

REQUEST

URL: GET/patients/{patientId}/careteammembers

Path:

patientId (int): Patient Identifier

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "FirstName": "string",
    "LastName": "string",
    "NameSuffix": "string",
    "Taxonomy": "string",
    "TaxonomyClassification": "string",
    "OfficeName": "string",
    "AddressLine1": "string",
    "City": "string",
    "State": "string",
    "Zip": "string",
    "Phone": "string",
    "NPI": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Goal

REQUEST

URL: GET/patients/{patientId}/goals

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "EncounterID": "int",
    "EncounterDate": "date-time",
    "GoalCode": "string",
    "Description": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Health Concern

REQUEST

URL: GET/patients/{patientId}/healthconcerns

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "EncounterID": "int",
    "ChiefComplaint": "string",
    "EncounterTypeDesc": "string",
    "OfficeName": "string",
    "DiagnosisCode": "string",
    "DiagnosisDescription": "string",
    "EncounterDate": "date-time",
    "CodeSystem": "string"
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Immunization

REQUEST

URL: GET/patients/{patientId}/immunizations

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "ID": "int",
    "EncounterDate": "date-time",
    "CVXCode": "string",
    "DateGiven": "date-time",
    "VaccineType": "string",
    "Vaccine": "string",
    "CodeSystem": "string",
    "AdministeredAmount": "string",
    "AdministeredUnit": "string",
    "Manufacturer": "string",
    "LotNo": "string",
    "RouteOfAdministration": "string",
    "StaffFirstName": "string",
    "StaffLastName": "string",
    "OfficeName": "string",
    "AddressLine1": "string",
    "City": "string",
    "State": "string",
    "Zip": "string",
    "Reaction": "string",
    "Taxonomy": "string",
    "TaxonomyClassification": "string",
    "FirstName": "string",
    "LastName": "string",
    "NameSuffix": "string"
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record

Implantable Device

REQUEST

URL: GET/patients/{patientId}/implantabledevices

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "MedicalDeviceID": "int",
    "StatusDesc": "string",
    "ModifiedByDate": "date-time",
    "Comments": "string",
    "DeviceID": "string",
    "Brand": "string",
    "CompanyName": "string",
    "Catalog": "string",
    "IAgency": "string",
    "Version": "string",
    "DeviceCount": "string",
    "Description": "string",
    "IsLotNumber": "bool",
    "ExpirationDate": "date-time",
    "IsSerialNumber": "bool",
    "IsManufactured": "bool",
    "SafetyInfo": "string",
    "IsContainsNRL": "bool",
    "IsContainsNoNRL": "bool",
    "IsSingleUse": "bool",
    "IsRxUse": "bool",
    "IsOTC": "bool",
    "IsKit": "bool",
    "IsComboProduct": "bool",
    "IsDeviceHCTP": "bool",
    "DeviceUDIProdInfo": "string",
    "ManufacturedDateProdInfo": "date-time",
    "ExpirationDateProdInfo": "date-time",
    "LotBatchNumberProdInfo": "string",
    "SerialNumberProdInfo": "string",
    "DistinctIDCode": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Lab Test

REQUEST

URL: GET/patients/{patientId}/labs/tests

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "OrderID": "int",
    "TestCode": "string",
    "TestName": "string",
    "OrderDate": "date-time",
    "FutureDescription": "string",
    "PendingDescription": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Lab Result

REQUEST

URL: GET/patients/{patientId}/labs/results

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "ID": "int",
    "Description": "string",
    "LoincTestCode": "string",
    "LoincTestName": "string",
    "ObservationValue": "string",
    "Units": "string",
    "Low": "string",
    "High": "string",
    "Flag": "string",
    "ReferencesRange": "string",
    "LineNumber": "string",
    "DateProcessed": "date-time",
    "ValueTypeID": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Medication

REQUEST

URL: GET/patients/{patientId}/medications

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "AdministeredMedicationID": "string",
    "DateGiven": "string",
    "Manufacturer": "string",
    "AdministeredAmount": "string",
    "AdministeredUnits": "string",
    "SigInformation": "string",
    "EncounterDate": "date-time",
    "ID": "string",
    "TimeLow": "date-time",
    "TimeHigh": "date-time",
    "DrugID": "string",
    "CodeSystem": "string",
    "DrugName": "string",
    "Status": "string",
    "MedicationDosage": "string",
    "DosageUnit": "string",
    "Route": "string",
    "Frequency": "string",
    "Sig": "string",
    "SigDosageFormDescription": "string",
    "SigDosageFormTypeID": "string",
    "Refill": "string",
    "Qty": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Patient Demographics

REQUEST

URL: GET/patients/{patientId}

Path:

patientId (int): Patient Identifier

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "FirstName": "string",
    "LastName": "string",
    "Sex": "string",
    "DateOfBirth": "date-time",
    "Race": [{
      "HL7Code": "string",
      "RollDownID": "string",
      "Description": "string"
    },...],
    "Ethnicity": [{
      "HL7Code": "string",
      "RollDownID": "string",
      "Description": "string",
    },...],
    "LanguageCode": "string",
    "SmokingStatus": [{
      "StartDate": "date-time",
      "EndDate": "date-time",
      "Description": "string",
      "Frequency": "string",
    },...],
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Patient CCDA

REQUEST

URL: GET/patients/{patientId}/ccda

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/XML

Clinical Document CCDA XML

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Problem

REQUEST

URL: GET/patients/{patientId}/problems

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "EncounterDate": "date-time",
    "DiagnosisCode": "string",
    "DiagnosisDescription": "string",
    "Status": "string",
    "CodeSystem": "string",
    "CodeSystemName": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Procedure

REQUEST

URL: GET/patients/{patientId}/procedures

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "EncounterCPTID": "string",
    "EncounterDate": "date-time",
    "CPT": "string",
    "Description": "string",
    "NPI": "string",
    "ProviderFirstName": "string",
    "ProviderLastName": "string",
    "ProviderSuffix": "string",
    "OfficeName": "string",
    "AddressLine1": "string",
    "City": "string",
    "State": "string",
    "Zip": "string",
    "Phone": "string",
    "Taxonomy": "string",
    "TaxonomyClassification": "string"
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Vital Sign

REQUEST

URL: GET/patients/{patientId}/vitalsigns

Path:

patientId (int): Patient Identifier

Query Parameters:

startDate (date-time : optional): Start Date (MM/DD/YYYY)

endDate (date-time : optional): End Date (MM/DD/YYYY)

Additional Headers:

Authorization: See [Authentication](#)

RESPONSE

Type: application/json

```
[
  {
    "VitalSignSystolic": "string",
    "VitalSignDiastolic": "string",
    "VitalSignHeight": "string",
    "VitalSignWeight": "string",
    "VitalSignBMI": "string",
    "VitalSignPulse": "string",
    "VitalSignTemperature": "string",
    "VitalSignGlucose": "string",
    "VitalSignWaist": "string",
    "VitalSignHeadCircum": "string",
    "EncounterDate": "date-time",
    "Units": "string",
  },
  ...
]
```

Exceptions:

401 - Unauthorized: Invalid token, or unauthorized record.

Revision Date	Changes	Approved By
08/03/2017	Original Document	Mike Kaneshiro/CIO
04/19/2018	Revised to 170.315(g)(7-9), added change log, updated version number	Mike Kaneshiro/CIO