

NRDR MIPS Data Exchange

Version 1.0

# Revision History

|  |  |  |
| --- | --- | --- |
| Date | Version | Description |
| 10/09/2023 | 1.0 | Draft |

# Background

This document describes the data model, data elements, data exchange architecture between the partner and the ACR NRDR MIPS, and the JSON schema for the MIPS data.

# Architecture

 

This document describes the NRDR MIPS data elements, JSON schema, APIs, and error messages. It also demonstrates how to use the authorization token when consuming the MIPS APIs.

The diagram below depicts the events that take place in the ACR’s Identity Provider (Okta) during the authentication and authorization processes. Please refer to the Okta documentationfor a detailed description of [ACR SSO Authentication service](https://nrdrsupport.acr.org/a/solutions/articles/11000090513?portalId=11000000200).

 

The MIPS APIs are a set of REST web services. They return an acknowledgement to the caller when submission is received. It is the responsibility of the caller to capture this acknowledgement to ensure the submission has reached the ACR successfully.

The frequency of the data submission is defined by the partner, however, the ACR prefers the submission to happen more frequently so that the facility/physician will have a faster turnaround time for review and be able to resubmit data before the CMS submission deadline if needed.

Partners shall be able to query the status of the submitted transactions by using the “Get MIPS Transaction Status” API.

# Data Model

The MIPS data to be submitted to the ACR shall be “wrapped” in a transaction. Each transaction has one or more Exams. Each exam can have several measures.

This approach provides the flexibility of submitting multiple exams for a facility within a single transaction. Note: Each transaction shall contain exams for one facility.

# JSON Schema

To successfully submit a transaction, you must provide the following minimum required data elements:

|  |  |  |
| --- | --- | --- |
| MIPS Transaction | Exam | Measure\_Data |
| Transaction\_IdTransaction\_DateTime Partner\_IdApp\_IdFacility\_IdNum\_Of\_Exam\_IncludedExam (At least 1 exam record is required) | Physician\_Group\_TINPhysician\_NPIExam\_Unique\_Id Exam\_DateTime Patient\_Id Patient\_Age Patient\_SexPatient\_Medicare\_Beneficiary **(optional)**Patient\_Medicare\_Advantage **(optional)**Num\_Of\_Measures\_IncludedMeasure\_Data**(At least 1 measure is required)** | Measure\_Num CPT\_code Diagnosis\_Code Numerator\_Code |

The following composite key shall be used to identify the MIPS exam record:

* Exam\_DateTime
* Physician\_Group\_TIN,
* Physician\_NPI,
* Patient\_Id,
* Measure\_Num,
* CPT\_code,
* Exam\_Unique\_Id

If this composite key already exists in the MIPS database, the API will update this existing record; otherwise, it will create a new exam record.

## **ACR MIPS JSON Mapping Specification**

The **ACR\_MIPS\_JSON\_Mapping.xlsx** document describes the validation rules and format of the JSON data elements. Most of the values of the data elements are mapped to either numeric or alpha-numeric codes. The data elements with a ‘Y’ under the “Required to submit a transaction” column is the minimum set of data elements that you need to provide to create a “MIPS” exam record in the database.

To have a “complete” status on an exam record, you must submit the data elements marked as “Required” or “Conditional Required” under the “Business Rules” column.

The data elements that are marked as “Optional” under the “Business Rules” column are not required for submission, that is, a record will not be rejected if an optional field is missing. However, if you do know them, we expect you to provide them. This expectation of complete data is specified in the participation agreement signed by all participating facilities.

## **MIPS 1.0 Upload JSON Schema**

This schema describes the data elements in a transaction for MIPS file version 1.0.

{

 "name": "MIPS\_Transaction Request",

 "Properties": {

 "Transaction\_Id": {

 "type": "string",

 "required": true

 },

 "Transaction\_DateTime": {

 "type": "string",

 "required": true

 },

 "Num\_Of\_Exam\_Included": {

 "type": "number",

 "required": true

 },

 "Partner\_Id": {

 "type": "string",

 "required": true

 },

 "App\_Id": {

 "type": "string",

 "required": true

 },

 "Facility\_Id": {

 "type": "string",

 "required": true

 },

 "Exam": {

 "type": "array",

 "required": true,

 "properties": {

 "Physician\_Group\_TIN": {

 "type": "string",

 "required": true

 },

 "Physician\_NPI": {

 "type": "string",

 "required": true

 },

 },

 "Patient\_Id": {

 "type": "string",

 "required": true

 },

 "Patient\_Age": {

 "type": "string",

 "required": true

 },

 "Patient\_Sex": {

 "type": "string",

 "required": true

 },

 "Patient\_Medicare\_Beneficiary": {

 "type": "string",

 "required": false

 },

 "Patient\_Medicare\_Advantage": {

 "type": "string",

 "required": false

 },

 "Exam\_Unique\_Id": {

 "type": "string",

 "required": true

 },

 "Exam\_DateTime": {

 "type": "string",

 "required": true

 },

 "Num\_of\_Measures\_Included": {

 "type": "number",

 "required": true

 },

 "Measure\_Data": {

 "type": "array",

 "required": true,

 "properties": {

 "Measure\_Num": {

 "type": "string",

 "required": true

 },

 "CPT\_Code": {

 "type": "string",

 "required": true

 },

 "Diagnosis\_Code": {

 "type": "string",

 "required": true

 },

 "Numerator\_Code": {

 "type": "string",

 "required": true

 }

 }

 }

 }

 }

 }

}

{

 "name":"MIPS\_Transaction Response",

 "properties":{

 "Message":{

 "type":"string",

 },

 }

}

Note:

* For MIPS measures, ‘Diagnosis\_Code’ translates to ‘Denominator Diagnosis Code’.
* For GRID 2.0 measures, ‘Diagnosis\_Code’ translates to ‘Secondary Diagnosis Code’.

## **Get MIPS Transaction Status Schema**

The “**Exam\_Transaction\_Status**” schema describes the data elements that are included in the “Get MIPS Transaction Status” API call initiated by the partner. Partner will use this API to query the status of a previously submitted transaction..

**Note**: The value of the **Exam\_Transaction\_Status.Error\_Msg** attribute may consist ofmultiple error messages that occurred while validating the transaction. Each message contains the error code‐message pair. Messages are separated by the ‘|’ delimiter.

For example:

P2001:Missing Transaction\_ID|P2011:Missing Transaction\_DateTime

The value of the **Exam\_Transaction\_Status.Exam\_Error.Error\_Msg** attribute may consist ofmultiple error messages that occurred while validating an exam. Each message contains the error code-message pair. Messages are separated by the ‘|’ delimiter.

For example:

P3001:Missing Exam\_Unique\_id|P3091:Missing Patient\_id

The value of the **Exam\_Transaction\_Status.Exam\_Errors.Measure\_Data\_Error.Error\_Msg** attribute

may consist of multiple error messages that occurred while validating a measure. Each message contains the error code‐message pair. Messages are separated by the ‘|’ delimiter.

For example:

P4002:Invalid CPT code|P4032:Invalid numerator

MIPS Transaction Status Request Schema

{

 "name":"MIPS\_Transaction\_Status Request",

 "properties":{

 "Transaction\_Id":{

 "type":"string",

 "required":"true"

 },

 "Facility\_Id":{

 "type":"string",

 "required":"true"

 },

 "App\_Id":{

 "type":"string",

 "required":"true"

 },

 "Partner\_Id":{

 "type":"string",

 "required":"true"

 },

 }

}

MIPS Transaction Status Response Schema

{

 "name": "MIPS\_Transaction\_Status Response",

 "Properties": {

 "Transaction\_Id": {

 "type": "string",

 "required": true

 },

 "Transaction\_DateTime": {

 "type": "string",

 "required": true

 },

 "Num\_Of\_Exam\_Included": {

 "type": "number",

 "required": true

 },

 "Num\_Of\_Exam\_In\_Error": {

 "type": "number",

 "required": true

 },

 "Partner\_Id": {

 "type": "string",

 "required": true

 },

 "App\_Id": {

 "type": "string",

 "required": true

 },

 "Facility\_Id": {

 "type": "string",

 "required": true

 },

 "Error\_Msg": {

 "type": "string",

 "required": false

 },

 "Exam\_Errors": {

 "type": "array",

 "required": true,

 "properties": {

 "Exam\_Unique\_Id": {

 "type": "string",

 "required": true

 },

 "Error\_Msg": {

 "type": "string",

 "required": true

 },

 "Num\_Of\_Measures\_Included": {

 "type": "number",

 "required": true

 },

 "Num\_Of\_Measures\_In\_Error": {

 "type": "number",

 "required": true

 },

 "Measure\_Data\_Error": {

 "type": "array",

 "required": true,

 "properties": {

 "Measure\_Num": {

 "type": "string",

 "required": true

 },

 "Error\_Msg": {

 "type": "string",

 "required": false

 }

 }

 }

 }

 }

 }

}

**"MIPS\_WebService\_Error"** schema describes detailed error information which is returned by all API responses with a status code other than 200:

{

 "name": "MIPS\_WebService\_Error",

 "Properties": {

 "error\_code": {

 "type": "string",

 "required": true

 },

 "error\_description": {

 "type": "string",

 "required": true

 }

}

# API

## **Authentication**

To be able to call API you must add JWT token received from ACR SSO provider in the “Authorization” request header using the following format:

Bearer <jwt-token>

Responses with a status code 401 indicate errors related to the authentication. Following error codes are used to provide more detailed error:

* `invalid\_token`, token is invalid or expired
* `missing\_token`, token is not provided in the request header

## **Create Transaction**

POST

This request is used to upload a new set of MIPS exams data.

Acceptable request representations:

* application/json

Possible responses:

* 200, returned when transaction data is uploaded and transaction is created successfully,
* 400, content-type: application/json
	+ error\_code = ‘data\_error’

Transaction data is missing or doesn’t have valid JSON representation.

* 401, content-type: application/json
	+ error\_code = ‘expired\_token’

Indicates that authentication token may have expired.

* 403, content-type: application/json
	+ error\_code = ‘no\_permission’

Authenticated user doesn’t have permissions to upload transaction data for a given facility.

## **GET transaction status**

POST

This request is used to get the status of a single transaction.

Possible responses:

* 200,content-type: application/json

Response will contain JSON data according to **MIPS\_Exam\_Transaction\_Status** schema

* 400, content-type: application/json
	+ error\_code = ‘missing\_param’

Indicates that missing transaction ID query string parameter

* 401, content-type: application/json
	+ error\_code = ‘expired\_token’

Indicates that authentication token may have expired.

* 404, content-type: application/json
	+ error\_code = ‘invalid\_param’

Indicates that transaction with a given ID doesn’t exist

# Test environment settings

URL: <https://nrdr9x.acr.org/MIPSDataSubmissionService/api/ExamAPI/PostMIPSData>

App\_Id: 10006

Facility\_Id and Partner\_Id will be provided by the ACR IT team.

## **Examples**

Creating transaction:

Sample parameters:

Facility\_Id: "100000"

Partner\_Id: "30007"

App\_Id: "10006"

Authorization: “Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjcnNhLXNoYTEiLCJ4NXQiOiIzWndDNC15STBQQmNHeDAyWlNsQjdaa1NRLUUifQ.eyJpc3MiOiJodHRwczovL1NlY3VyZUF1dGgwMVZNLmFjci5vcmciLCJhdWQiOiJodHRwczovL1NlY3VyZUF1dGgwMVZNLmFjci5vcmciLCJuYmYiOjE0MjIyNzY3ODIsImV4cCI6MTQyMjM2MzE4MiwiY2xpZW50X2lkIjoiMWY1ZjM5NTI0ZjIyNGRmMDg0NTIwYTJmYWE5YTkyNzUiLCJzY29wZSI6WyJvcGVuaWQiLCJwcXJzIl0sInN1YiI6IjEwMDAwMF9udWFuY2UiLCJhbXIiOiJwYXNzd29yZCIsImF1dGhfdGltZSI6IjE0MjIyNzY3NTciLCJpZHAiOiJTZWN1cmVBdXRoNSJ9.AFsoPNtIT8N0N6w1y0OZLokdplfRM2Y8OAvquMXpkoI\_6b9ANEhXY72ubuylt6-4vdJ165aVUXPRDTF37UT9cqv21hRTpOfYYM0tV5gksUbvLzKaPFt6EWbB8ctQnHs1pxyDGSbKDM-dR6upj6x0eZ7O46Zwi8o8Zhkeu0bFZPNmLS7y96\_A9uyGBkbt69Ih0aCBNRvtabqkD3vJsWB4EIfiqJJ2RO8XPCGfSoYAnuosgH5Cm6kMNN7WiX4rgFps20MryBLtutKztcNhaj-l65gzxyrJpgmq8r9dBWiFI6nFc-rKkNr6Gdj2TH9EABloC6aO\_CQQFtgrXFLEchnSPw”

**Note**: the authentication/authorization mechanism in the MIPS Exam API adopts the ACR Connect security model which requires an authorization token after authenticated successfully. The token in the sample parameters above is an example.

function createTransaction(transactionData, token) {

    $.ajax({

        type: 'POST',

        url: "https://nrdr9x.acr.org/MIPSDataSubmissionService/api/ExamAPI/PostMIPSData”,

        data: JSON.stringify(transactionData),

        dataType: "json",

        contentType:"application/json; charset=utf-8",

        headers: { "Authorization": "Bearer " + token },

        success: function (data) {

            // handle data load success

        }

    });

}

This function can be used in the following way:

Example (For CMSYear 2023):

 [

 {

 "Transaction\_Id": "Test01082023\_0",

 "Transaction\_DateTime": "01/08/2023",

 "App\_Id": "12340",

 "Partner\_Id": "1000242",

 "Facility\_Id": "100210",

 "Num\_Of\_Exam\_Included": "1",

 "Exam": [

 {

 "Physician\_Group\_TIN": "232323232",

 "Physician\_NPI": "1962690453",

 "Exam\_Unique\_Id": "TestExam01082023\_1",

 "Exam\_DateTime": "01/08/2023",

 "Patient\_Id": "TestPatient01082023\_1",

 "Patient\_Age": "40",

 "Patient\_Sex": "M",

 "Patient\_Medicare\_Beneficiary": null,

 "Patient\_Medicare\_Advantage": null,

 "Num\_Of\_Measures\_Included": "1",

 "Measure\_Data": [

 {

 "Measure\_Num": "MSN16",

 "CPT\_Code": "76706",

 "Diagnosis\_Code": "EE014",

 "Numerator\_Code": "PM"

 }

 ]

 }

 ]

 }

]

var token = GetAuthorizationToken();

createTransaction(transactionData, token);

Getting transaction status:

function getTransaction(transactionId, token, partnerId, appId, facilityId) {

    $.ajax({

        type: POST,

        url: "https://nrdr9x.acr.org/DataSubmissionService/api/ExamAPI/GetMIPSTransactionStatus ",

        data: { "Transaction\_Id": transactionId, "App\_Id": appId, "Partner\_Id":

partnerId,"Facility\_Id": facilityId },

        headers: { "Authorization": "Bearer " + token },

        success: function (data) {

            // handle JSON with transaction data

        }

    });

}

For UAT use the following values:

https://nrdr9x.acr.org/DataSubmissionService/api/ExamAPI/GetMIPSTransactionStatus?Transaction\_id=<transaction id>

&Facility\_Id=<facility id>&Partner\_Id=<partner id>&App\_Id=<app id>

Sample parameters:

                Facility\_Id: "100000"

                Partner\_Id: "10000"

                App\_Id: "10006"

Transaction\_Id : “Test\_Transaction\_1”

**Note**: the authentication/authorization mechanism in the MIPS Exam API adopts the ACR Connect security model which requires an authorization token after authenticated successfully.

# Validation & Error Messages

The Web service will validate the following:

1. JSON syntax
2. Transactions Limit (max 10 transactions) and size (max 20 MB JSON data file).
3. Required fields
4. Response value of the data elements based on the business rules stated in the file **ACR\_MIPS\_JSON\_Mapping.xlsx**

The table below lists the possible messages generated during the validation process for a transaction:

|  |  |
| --- | --- |
| **Status Code** | **Message**  |
| **Web Service Data Upload Error Codes** |
| P1001 | Data received successfully. |
| P1002 | Unable to import data submitted. Data does not match schema or Data submitted is empty or null. |
| P1003 | MIPS does not allow exams related measure data as the count of your transaction is more than <TransactionsMaxCount> or file size is more than 20MB. |
| P1004 | Error in validating token <exception message> |
| P1005 | MIPS Data submission through web service is not open yet. |
| P1006 | Error in saving data <exception message>. |
| **Transaction level** |
| P2001 | Missing Transaction\_ID |
| P2002 | Transaction\_ID must be unique. <transaction\_id> has been submitted in a previous transaction. |
| P2004 | Transaction\_ID must be unique. Transaction\_ID is submitted multiple times in same file. |
| P2011 | Missing Transaction\_DateTime |
| P2012 | Transaction\_DateTime is not a valid date time format |
| P2013 | Transaction\_DateTime is a future date |
| P2021 | Missing Num\_of\_Exam\_Included |
| P2022 | Num\_of\_Exam\_Included is not an integer |
| P2031 | Missing PartnerID |
| P2032 | PartnerID is not valid, it must be integer. |
| P2041 | Missing AppID |
| P2042 | AppID is not valid, it must be integer. |
| P2051 | Missing Facility\_Id |
| P2052 | Facility\_Id is not valid, it must be integer. |
| P2053 | Facility has not registered for MIPS |
| P2071 | Missing exam object information |
| P2072 | Data in tag Num\_Of\_Exams\_Included <num\_of\_exams\_included> does not match with exams received |
| **Exam level** |
| P3001 | Missing Unique\_Exam\_ID |
| P3002 | Exam\_Unique\_ID cannot contain special Character '@'. |
| P3004 | Exam\_Unique\_ID must be unique. Exam\_Unique\_ID [<exam id>] is submitted multiple times in Transaction. |
| P3011 | Missing Exam\_Date. |
| P3012 | Exam\_DateTime is not a valid date time in format mm/dd/yyyy |
| P3013 | Exam\_Date is a future date.  |
| P3041 | Missing Physician\_NPI |
| P3042 | Physician (<NPI>) has not registered for MIPS. Physician is required to register in order to participate. |
| P3043 |  Physician <NPI> attestation pending. |
| P3044 | PhysicianNPI( <NPI>) you have submitted does not belong to facilityID( <Facility\_ID>). |
| P3051 | Missing Physician\_Group\_TIN in submitted Data |
| P3052 | Physician\_Group\_TIN (<TIN>) must be a 9‐digit code |
| P3053 | Physician\_Group\_TIN (<TIN>) must be numeric |
| P3054 | Physician Group TIN/NPI (<TIN>/<NPI>) mapping not found for facility in NRDR |
| P3061 | Missing Patient\_ID |
| P3062 |  Patient\_Id cannot contain special Character @. |
| P3071 | Missing Patient\_Age |
| P3072 | Patient age must be a 1‐3-digit number |
| P3073 | Patient age is out of range. Valid range is 1 to 130 |
| P3081 | Missing Patient Sex |
| P3082 | Entered Patient\_Sex must be \"M\" or \"F\" or \"U\" or \"O\" |
| P3121 | Missing Num\_of\_Measures\_Included |
| P3122 | Num\_of\_Measures\_Included is not an integer. |
| P3123 | Data in Num\_of\_Measures\_Included (<Num\_of\_Measures\_Included>) does not match with Measures received. |
| P3131 | Missing object Measure\_Data information |
| P3137 | Exam Date is not active for CMS submission |
| **Measure Data level** |
| P4003 | Patient must be [<AgeRestrictionFrom>] and older to be eligible for measure (<Measure Num>). Submitted patient age value is (<Patient\_Age>). |
| P4005 | Entered Measure\_Num (<Measure Num>) is not active for CMS Year (<CMSYear>). |
| P4006 | Invalid Sex (Patient\_Sex) entered for Measure Number (<Measure Num>). |
| P4011 | Missing CPT\_code |
| P4012 | Invalid CPT\_code <cpt code> entered for measure number <measure num> |
| P4013 | Exam Date is not in Acceptable Date range for CPT CODE <cpt code>. |
| P4021 | Missing Diagnosis\_Code |
| P4022 | Invalid Diagnosis\_Code <diagnosis code> entered for measure number <measure num> |
| P4023 | Diagnosis code is excluded. This exam is excluded from initial population for Diagnosis\_Code[ <diagnosis code>] and Measure Number [ <measure\_num>]. |
| P4031 | The numerator response value <numerator code> is missing for this Measure Number <measure num> but will be included in the eligible population. |
| P4032 | The numerator response value <numerator code> is incorrect for this Measure Number <measure num> but will be included in the eligible population. |
| P4033 | Missing measure#. Unable to determine measure#. Found More than one measures#[<measure\_num>] based CPT code, CMSYear and Numerator\_Response value. |
| P4035 | Missing measure#. Unable to determine measure# based on CPT code, CMSYear and Numerator\_Response value. |
| P4036 | The cpt code [{0}] matches the denominator exclusion code. |
| P4037 | This physician record shows an Inactive on your facility's Manage Physicians page. |
| P4038 | Measure Number [Measure\_num] not in acceptable age ranges (40-70) for the CRITERIA3. |
| P4039 | Invalid CPT code and Numerator code criteria combination for Measure Number [Measure\_num]. |