526 / 4142 Reporting and Monitoring

(DRAFT)

Table of Contents

[Monitoring and Triage for 526/4142 3](#_Toc528312486)

[Summary 3](#_Toc528312487)

[New Reporting / Tracking 3](#_Toc528312488)

[Error Scenarios 3](#_Toc528312489)

[4142 Back End Processing 4](#_Toc528312490)

[PDF Processing via ICMHS 6](#_Toc528312491)

[ICMHS Flow 7](#_Toc528312492)

[526 / 4142 Submission Process 7](#_Toc528312493)

[Steps during Submission Process 7](#_Toc528312494)

[Scenario 1: Both 526 and 4142 submissions are successful 8](#_Toc528312495)

[Scenario 2: 526 submission successful, but 4142 fails 10](#_Toc528312496)

[Scenario 3: 526 submission fails, 4142 submit isn’t attempted 12](#_Toc528312497)

[Appendix 13](#_Toc528312498)

[JSON Implementation Details 13](#_Toc528312499)

# Monitoring and Triage for 526/4142

## Summary

The 4142 and 4142a processing have been integrated with the 526 claims submission processing. Some additional reporting capabilities have been added for the 526 and some additional monitoring available for the 4142. These are highlighted below. Below are some processing that should be tracked.

* PDF submissions via DocumentsServices
  + Includes any supporting evidence or forms uploaded via vets.gov website
* 526 submissions
* 4142 submissions
  + Initial ICMHS call
  + Subsequent status calls – manual

## New Reporting / Tracking

* New tracker and reporting classes have been written and made available to the ancillary forms for reporting their status. Details of the implementation are included with this GitHub EPIC
  + <https://github.com/department-of-veterans-affairs/vets.gov-team/issues/13659>
* This implementation will allow the system to pull a daily report on the status of 526 submissions and check jobs / logs to verify the status of ancillary items along with the main form submission

## Error Scenarios

Below are some error scenarios for 526/4142 and some additional triage and reporting techniques. These are in addition to existing tools such as Sentry for log files.

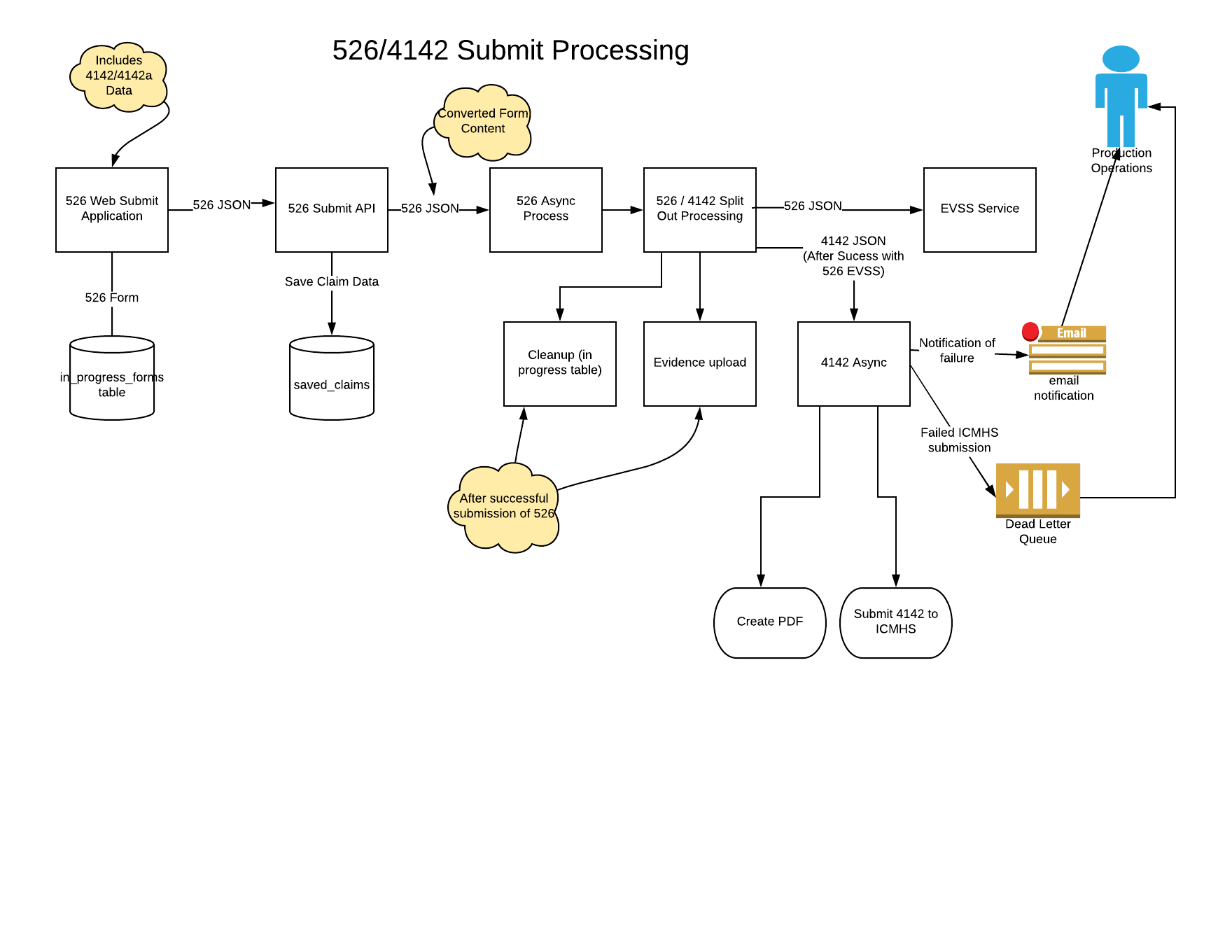
**Question regarding notifications via Job Queue failure**

**Retries how does this happen**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Notes** | **Triage** |
| 526 and 4142 Successful | 526 is successfully submitted via EVSS. 4142 is successfully submitted through ICMHS | Statistics can be found in DB table disability\_compensation\_job\_statuses |
| 526 Successful, 4142 failed | 526 is successfully submitted via EVSS. 4142 fails at call to ICMHS API | Statistics can be found in DB table disability\_compensation\_job\_statuses  Failed job will be in Sidekiq Dead Job Queue |
| 526 Successful, 4142 successful via ICMHS but fails in downstream system | 526 is successfully submitted via EVSS. 4142 is successfully submitted through ICMHS API. 4142 encounters error in downstream processing | Statistics can be found in DB table disability\_compensation\_job\_statuses  ICMHS can be polled to determine status   * curl -v -F token=$CENTRALMAILTOKEN -F uuid='<FROM disability\_compensation\_job\_statuses ' <ICMHS status URL>   GENERAL STATUS FOR ALL SUBMISION |
| 526 failed, 4142 not attempted | 526 fails via EVSS. 4142 is not attempted. | Statistics can be found in DB table disability\_compensation\_job\_statuses  Failed job will be in Sidekiq Dead Job Queue |
| Supporting Evidence uploads | User can upload supporting evidence documents. The uploads leverage the EVSS DocumentsService API. | Statistics can be found in DB table disability\_compensation\_job\_statuses |
|  |  |  |
|  |  |  |

# 4142 Back End Processing

The diagram below depicts the process flow for 4142 data and the integration with the 526 flow.



The 526 Async process will receive JSON content that contains both the 526 and 4142/4142a data. The process will extract the 526 data and submit to EVSS for processing. Upon successful processing (acknowledgement from EVSS), the 526 async process will invoke the 4142 async process with the extracted 4142 data and veteran profile data. The 526 async process will need to translate the extract JSON to a format for the 4142 process.

# PDF Processing via ICMHS

The 4142 Async job will take the form data that has been collected and create a single PDF consisting of the 4142 and 4142a data. The PDF creation will leverage the platform capability including the application of a digital signature to the document. It is assumed that the 4142 processing would not be invoked unless the Vet certified the data via the certification checkbox. The 4142 job will create the necessary meta-data and leverage the ICMHS API to submit the PMR PDF for processing by the PMR vendor.

### ICMHS Flow



# 526 / 4142 Submission Process

The section documents several conditions, system behavior, track and reporting status during 526 and 4142 submission process.

## Steps during Submission Process

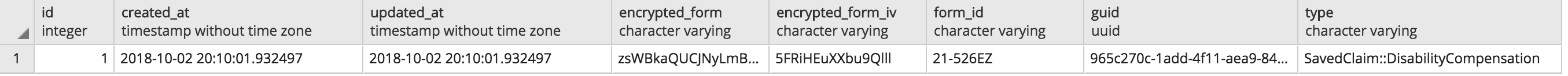
1. Veteran submits 526 application with the included private medical records (PMR) form 4142 data
2. 526 controller [*V0::DisabilityCompensationFormController*] submit method parses the entire form as JSON object
3. If form4142 is present in 526 form, then the class [*EVSS::DisabilityCompensationForm::Form4142*] translates it to the expected JSON format for 4142 background sidekiq job to process it
4. Form 526 data translation is completed by the class [*EVSS::DisabilityCompensationForm::DataTranslation*]
5. 526 controller then asynchronously invokes submit form 526 background job [*EVSS::DisabilityCompensationForm::SubmitForm526*] passing multiple parameters, including the translated form 4142
6. 526 background job then submits form526 to an upstream submission service (currently EVSS) and sets Claim ID in response
7. 526 background job then asynchronously invokes 4142 background job [*CentralMail::SubmitForm4142Job*]
8. 4142 background job generates, stamps 4142/4142A PDF document. Job then uploads the document to Central Mail using Benefits Intake API (ICMHS)
9. 526 background job then asynchronously invokes 526 clean up background job (*EVSS::DisabilityCompensationForm::SubmitForm526Cleanup*] for cleaning up 21-526EZ form data from InProgress table and REDIS cache
10. If uploads are available, 526 background job then asynchronously invokes background batch job [*EVSS::DisabilityCompensationForm::SubmitUploads*] to upload evidence attachments into VBMS via EVSS document service

During submit application, listed below are the expected possibilities:

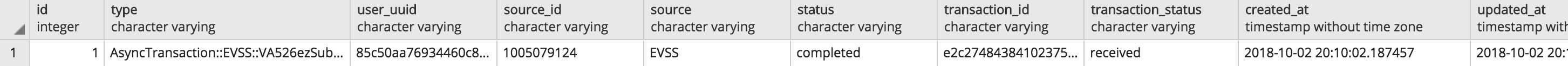
1. Both 526 and 4142 submissions are successful
2. 526 submission successful, but 4142 fails
3. 526 submission fails, and hence the process doesn’t attempt to submit 4142

## Scenario 1: Both 526 and 4142 submissions are successful

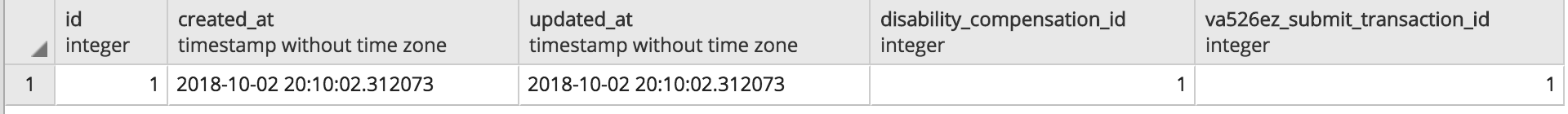
* In this scenario, it can be concluded that Steps 6 and 8 were successfully completed
* Post submit for the form with id **21-526EZ** (Claims for Increase), PostgreSQL tables would contain records as shown below (example from developer local testing)
  + saved\_claims



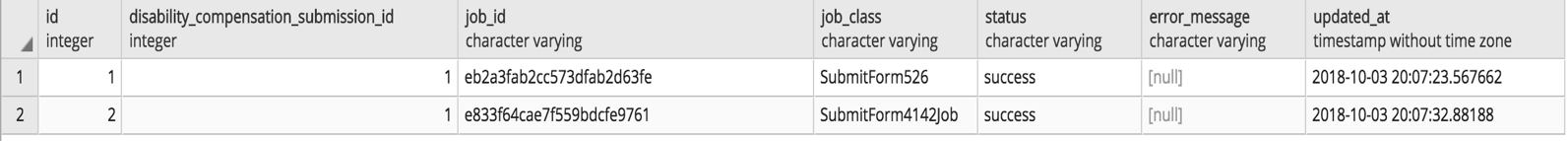
* + async\_transactions



* + disability\_compensation\_submissions



* + disability\_compensation\_job\_statuses



* 4142 submission job status is tracked in the newly created table “disability\_compensation\_job\_statuses”.
  + PR from Ad Hoc for the new table creation and related code changes. BAH also did few fixes and updates for 4142 integration work within the same PR

🡪 <https://github.com/department-of-veterans-affairs/vets-api/pull/2326>

* Rails application log with the entries as shown below for successful submissions of 526 and 4142

2018-10-04 16:04:11.461125 D [13:70316550366300] Rails -- [upsert]

Selector: {"job\_id"=>"4bb8e6d557d4fe0d4696a67a"}

Setter: {"disability\_compensation\_submission\_id"=>1, "job\_class"=>"SubmitForm526", "job\_id"=>"4bb8e6d557d4fe0d4696a67a", "status"=>"success", "updated\_at"=>2018-10-04 16:04:11 UTC}

2018-10-04 16:04:11.522771 I [13:70316550366300] Rails -- Form526 Submission -- { "saved\_claim\_id" => 1, "submission\_id" => 1, "job\_id" => "4bb8e6d557d4fe0d4696a67a", "event" => "success" }

2018-10-04 16:04:11.522836 I [13:70316550366300] Rails -- Form526 Submission -- { "saved\_claim\_id" => 1, "submission\_id" => 1, "job\_id" => "4bb8e6d557d4fe0d4696a67a", "event" => "try" }

2018-10-04 16:04:15.569678 D [13:86655540] Rails -- [upsert]

Selector: {"job\_id"=>"122c7b160b859915fb3749dc"}

Setter: {"disability\_compensation\_submission\_id"=>1, "job\_class"=>"SubmitForm4142Job", "job\_id"=>"122c7b160b859915fb3749dc", "status"=>"success", "updated\_at"=>2018-10-04 16:04:15 UTC}

2018-10-04 16:04:15.575856 I [13:86655540] Rails -- Form4142 Submission -- { "saved\_claim\_id" => 1, "submission\_id" => 1, "job\_id" => "122c7b160b859915fb3749dc", "event" => "success" }

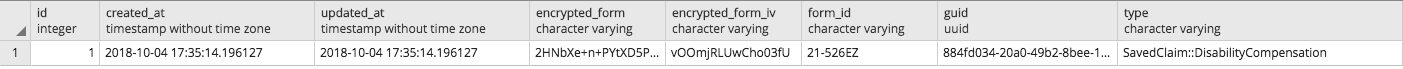
2018-10-04 16:04:15.575915 I [13:86655540] Rails -- Form4142 Submission -- { "saved\_claim\_id" => 1, "submission\_id" => 1, "job\_id" => "122c7b160b859915fb3749dc", "event" => "try" }

## Scenario 2: 526 submission successful, but 4142 fails

* For this scenario, it can be concluded that Steps 6 was successful, however there were failures during Step 8
* For 4142 PDF upload, ICMHS API call will immediately return one of the following codes to vets.gov

|  |  |
| --- | --- |
| Error Code | Description |
| 200 | Success – ICMHS received successfully, but not processed and at VBMS |
| 400 | Duplicate – document already uploaded with that uuid |
| 401 | Unauthorized – missing or incorrect token |
| 409 | Conflict - mismatched attachments and numbers |
| 412 | Precondition Failed – for meta data failures |
| 415 | Unsupported File type |
| 422 | Unprocessable Request– no file, or issue with PDF, missing attachments |
| 500 | Server Error |

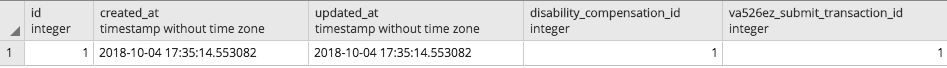
* 4142 background job will retry submissions for all the conditions listed below
  + ICMHS returns error code in the range of 201-600
  + ICMHS connection times out (Error of type “*Common::Exceptions::GatewayTimeout”*)
  + Any other standard error (Error of type “StandardError”) raised during processing
* Post submit for the form with id **21-526EZ** (Claims for Increase), PostgreSQL tables would contain records as shown below (example from developer local testing)
  + saved\_claims



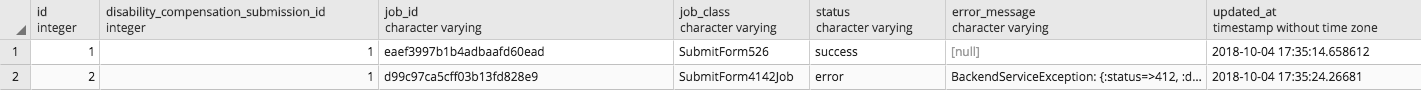
* + async\_transactions

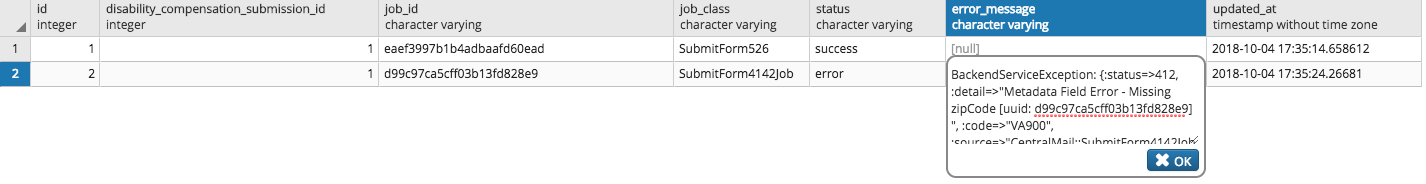


* + disability\_compensation\_submissions



* + disability\_compensation\_job\_statuses





* Rails application log with the entries as shown below for the successful 526 submission, but failed 4142

2018-10-04 17:35:14.718440 D [13:43130340] Rails -- [upsert]

Selector: {"job\_id"=>"eaef3997b1b4adbaafd60ead"}

Setter: {"disability\_compensation\_submission\_id"=>1, "job\_class"=>"SubmitForm526", "job\_id"=>"eaef3997b1b4adbaafd60ead", "status"=>"success", "updated\_at"=>2018-10-04 17:35:14 UTC}

2018-10-04 17:35:14.767671 I [13:43130340] Rails -- Form526 Submission -- { "saved\_claim\_id" => 1, "submission\_id" => 1, "job\_id" => "eaef3997b1b4adbaafd60ead", "event" => "success" }

2018-10-04 17:35:14.767760 I [13:43130340] Rails -- Form526 Submission -- { "saved\_claim\_id" => 1, "submission\_id" => 1, "job\_id" => "eaef3997b1b4adbaafd60ead", "event" => "try" }

2018-10-04 17:35:24.266094 E [13:43127280 submit\_form4142\_job.rb:144] Rails -- { "Error Message" => "BackendServiceException: {:status=>412, :detail=>\"Metadata Field Error - Missing zipCode [uuid: d99c97ca5cff03b13fd828e9] \", :code=>\"VA900\", :source=>\"CentralMail::SubmitForm4142Job\"}" }

2018-10-04 17:35:24.266266 E [13:43127280 sentry\_logging.rb:42] Rails -- BackendServiceException: {:status=>412, :detail=>"Metadata Field Error - Missing zipCode [uuid: d99c97ca5cff03b13fd828e9] ", :code=>"VA900", :source=>"CentralMail::SubmitForm4142Job"}.

2018-10-04 17:35:24.276371 D [13:43127280] Rails -- [upsert]

Selector: {"job\_id"=>"d99c97ca5cff03b13fd828e9"}

Setter: {"disability\_compensation\_submission\_id"=>1, "error\_message"=>"BackendServiceException: {:status=>412, :detail=>\"Metadata Field Error - Missing zipCode [uuid: d99c97ca5cff03b13fd828e9] \", :code=>\"VA900\", :source=>\"CentralMail::SubmitForm4142Job\"}", "job\_class"=>"SubmitForm4142Job", "job\_id"=>"d99c97ca5cff03b13fd828e9", "status"=>"error", "updated\_at"=>2018-10-04 17:35:24 UTC}

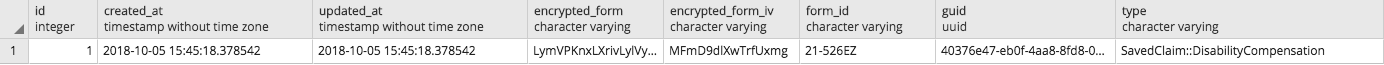
2018-10-04 17:35:24.282209 E [13:43127280 job\_status.rb:53] Rails -- Form4142 Submission -- { "saved\_claim\_id" => 1, "submission\_id" => 1, "job\_id" => "d99c97ca5cff03b13fd828e9", "event" => "error", "error\_message" => "BackendServiceException: {:status=>412, :detail=>\"Metadata Field Error - Missing zipCode [uuid: d99c97ca5cff03b13fd828e9] \", :code=>\"VA900\", :source=>\"CentralMail::SubmitForm4142Job\"}" }

2018-10-04 17:35:24.282476 I [13:43127280] Rails -- Form4142 Submission -- { "saved\_claim\_id" => 1, "submission\_id" => 1, "job\_id" => "d99c97ca5cff03b13fd828e9", "event" => "try" }

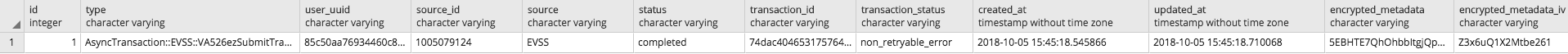
* As seen from the Rails logs, submit of 4142 failed with error code 412 is returned from ICMHS. Error details are logged and recorded in the “disability\_compensation\_job\_statuses” table
* For 4142 background job, a maximum retries of 10 is configured which will likely result in a runtime of ~8 hours
* **For 526 or 4142 submissions, there will LESS likely be a scenario where 526 form and 4142 PDF will be submitted with some missing data that went un-caught during schema validation. However, for this less likely scenario the ONLY way submissions can later be re-tried if the following ACTIONS are POSSIBLE (Ad Hoc and DevOps to confirm the feasibility of the below actions)**
  + **Job is available to be retried. For job to be retries, it has to be in the Dead or Retries Queue**
  + **Arguments passed to the job (form, headers, user info etc.) can be fixed if required for certain cases before manual / configured retry happens in sidekiq.**

## Scenario 3: 526 submission fails, 4142 submit isn’t attempted

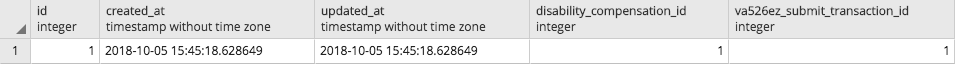
* For this scenario, it can be concluded that Step 6 was failed and hence Step 8 didn’t execute
* 526 submission to EVSS can fail for various reasons of application error, backend EVSS system error and timeouts
* Post submit for the form with id **21-526EZ** (Claims for Increase), PostgreSQL tables would contain records as shown below (example from developer local testing)
  + saved\_claims



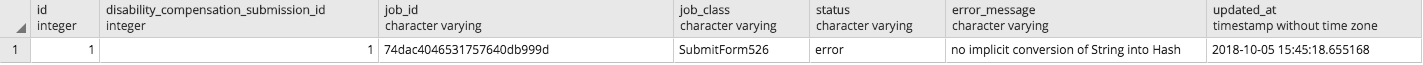
* + async\_transactions



* + disability\_compensation\_submissions



* + disability\_compensation\_job\_statuses



* For 526 background job, a maximum retry of 13 is configured which will likely result in a runtime of ~25 hours (This is as per the comment in EVSS::DisabilityCompensationForm::SubmitForm526 )
* 526 Background job follows the below conditions for retries
  + EVSS timeout errors are **re-tried** for submissions
  + Any other service exceptions or standard errors are logged into sentry, but **NOT** **retried**
* For timeout errors, if 526 is submitted successfully within the window of 13 retries, then 4142 will be attempted and follow the path mentioned in Scenario 2
* For any other service or standard errors, the errors and details will be logged into sentry and no retries will be attempted. In this case, manual intervention is needed. **UNCLEAR about this process yet, need a discussion with AdHoc and DevOps**. The only identified option for now is getting on a box, then via rails console use the sidekiq api to fix the data. UNCLEAR to me yet if this process even works or has been tried before.
* 4142 submission will be complete **ONLY** when 526 submission job is available for retry and is completed without errors.

# Appendix

## JSON Implementation Details

For 4142 and 4142A forms, PDF is generated using a JSON form content. As a part of 526 integration, there was an agreement that Ad Hoc will own completing the code changes to extract 4142 form content from 526 form, translate it into the JSON format, save it and pass it as expected by the background job of 4142.

Expected sample JSON file for 4142 PDF generation is available on: <https://github.com/department-of-veterans-affairs/vets-api/blob/4142-api-async/spec/fixtures/pdf_fill/21-4142/kitchen_sink.json> and included below.

4142 Background Job Initial Version is available on: <https://github.com/department-of-veterans-affairs/vets-api/blob/4142-api-async/app/workers/central_mail/submit_form4142_job.rb>

Details of the JSON fields are as below:

* Fields populated from pre-fill information as below:  
  - "veteranFullName", "vaFileNumber", "veteranSocialSecurityNumber", "veteranServiceNumber" , "veteranDateOfBirth", "veteranAddress { "city","country","postalCode", "street","street2","state"}, "email", "veteranPhone"
* Fields available from 526 integrated form as captured from 4142 UI as below:
  + "providerFacility" [{"providerFacilityName","treatmentDateRange"[{"from","to"}],"providerFacilityAddress" {"street","street2","city","state","country","postalCode"}}, "privacyAgreementAccepted", "limitedConsent"

**4142 definition to be integrated into 526 Schema is shown below:**

*const form4142 = {type: 'object',properties: {limitedConsent: {type: 'string'},providerFacility: {type: 'array',required: ['providerFacilityName','treatmentDateRange','providerFacilityAddress'],items: {type: 'object',properties: {providerFacilityName: {type: 'string'},treatmentDateRange: {type: 'array',properties: {from: {$ref: '#/definitions/date'},to: {$ref: '#/definitions/date'}}},providerFacilityAddress: {$ref: '#/definitions/centralMailAddress'}}}},*

*privacyAgreementAccepted: {$ref: '#/definitions/privacyAgreementAccepted'}}};*

**Sample 4142 JSON Payload:**

|  |
| --- |
| { |
|  | "privacyAgreementAccepted": true, |
|  | "vaFileNumber": "12345678", |
|  | "veteranSocialSecurityNumber": "123456789", |
|  | "veteranServiceNumber": "987654321", |
|  | "veteranFullName": { |
|  | "first": "Testy", |
|  | "middle": "Tester", |
|  | "last": "Testerson" |
|  | }, |
|  | "veteranDateOfBirth": "1981-11-05", |
|  | "veteranAddress": { |
|  | "city": "Baltimore", |
|  | "country": "USA", |
|  | "postalCode": "21231-1234", |
|  | "street": "street", |
|  | "street2": "1B", |
|  | "state": "MD" |
|  | }, |
|  | "email": "testy.testerson@gmail.com", |
|  | "veteranPhone": "2123456789", |
|  | "limitedConsent": "true", |
|  | "providerFacility": [ |
|  | { |
|  | "providerFacilityName": "provider 1", |
|  | "treatmentDateRange": [ |
|  | { |
|  | "from": "1980-1-1", |
|  | "to": "1985-1-1" |
|  | }, |
|  | { |
|  | "from": "1986-1-1", |
|  | "to": "1987-1-1" |
|  | } |
|  | ], |
|  | "providerFacilityAddress": { |
|  | "street": "123 Main Street", |
|  | "street2": "1B", |
|  | "city": "Baltimore", |
|  | "state": "MD", |
|  | "country": "USA", |
|  | "postalCode": "21200-1111" |
|  | } |
|  | }, |
|  | { |
|  | "providerFacilityName": "provider 2", |
|  | "treatmentDateRange": [ |
|  | { |
|  | "from": "1980-2-1", |
|  | "to": "1985-2-1" |
|  | }, |
|  | { |
|  | "from": "1986-2-1", |
|  | "to": "1987-2-1" |
|  | } |
|  | ], |
|  | "providerFacilityAddress": { |
|  | "street": "456 Main Street", |
|  | "street2": "1B", |
|  | "city": "Baltimore", |
|  | "state": "MD", |
|  | "country": "USA", |
|  | "postalCode": "21200-1111" |
|  | } |
|  | }, |
|  | { |
|  | "providerFacilityName": "provider 3", |
|  | "treatmentDateRange": [ |
|  | { |
|  | "from": "1980-3-1", |
|  | "to": "1985-3-1" |
|  | }, |
|  | { |
|  | "from": "1986-3-1", |
|  | "to": "1987-3-1" |
|  | } |
|  | ], |
|  | "providerFacilityAddress": { |
|  | "street": "789 Main Street", |
|  | "street2": "1B", |
|  | "city": "Baltimore", |
|  | "state": "MD", |
|  | "country": "USA", |
|  | "postalCode": "21200-1111" |
|  | } |
|  | }, |
|  | { |
|  | "providerFacilityName": "provider 4", |
|  | "treatmentDateRange": [ |
|  | { |
|  | "from": "1980-4-1", |
|  | "to": "1985-4-1" |
|  | }, |
|  | { |
|  | "from": "1986-4-1", |
|  | "to": "1987-4-1" |
|  | } |
|  | ], |
|  | "providerFacilityAddress": { |
|  | "street": "101 Main Street", |
|  | "street2": "1B", |
|  | "city": "Baltimore", |
|  | "state": "MD", |
|  | "country": "USA", |
|  | "postalCode": "21200-1111" |
|  | } |
|  | }, |
|  | { |
|  | "providerFacilityName": "provider 5", |
|  | "treatmentDateRange": [ |
|  | { |
|  | "from": "1980-5-1", |
|  | "to": "1985-5-1" |
|  | }, |
|  | { |
|  | "from": "1986-5-1", |
|  | "to": "1987-5-1" |
|  | } |
|  | ], |
|  | "providerFacilityAddress": { |
|  | "street": "102 Main Street", |
|  | "street2": "1B", |
|  | "city": "Baltimore", |
|  | "state": "MD", |
|  | "country": "USA", |
|  | "postalCode": "21200-1111" |
|  | } |
|  | } |
|  | ] |
|  | } |