

Enhanced Reporting Requirements

Overview of Web Service Examples

Contents

Latest Version History.....	3
Audience.....	3
Document context.....	3
Introduction	4
Overview.....	4
Example 1 – Full ERR Life Cycle	9
1.1 Lookup ERN Request Example	9
1.2 Lookup ERN Response Example.....	10
1.3 ERR Submission Request Example	12
1.4 ERR Submission Response Example	14
1.5 Check ERR Submission Request Example.....	16
1.6 Check ERR Submission Response Example	17
1.7 Check ERR Run Request Example.....	20
1.8 Check ERR Run Response Example	21
1.9 Lookup ERR Monthly Report Example	22
1.10Lookup ERR Monthly Report Response Example.....	24
Example 2 – Overpayment of an Expense/Benefit to an Employee.....	27
2.1 ERR Submission Overpayment Example	27
2.2 Correction of ERR Submission Overpayment Example	27
Example 3 – Underpayment of an Expense/Benefit to an Employee	28
3.1 ERR Submission Underpayment Example.....	28
3.2 Correction of ERR Submission Underpayment Example.....	28
Example 4 – Amendment of Incorrect ERR Submission	29

4.1 Invalid ERR Submission Example.....	29
4.2 Amendment using Previous Line Item ID Example	29
4.3 Amendment using Line Item IDs to Delete Example	30
Appendix A – Business Rules	32
Employment ID Business Rules	32
Paying expenses to an employee where the PPSN is not available.....	32
Appendix B – Line Item Correction rules.....	34
Correction rules.....	34
Note on correction rules for data items:	34

Version

1.0

Version Date

03/08/20223

Latest Version History			
Version	Change Date	Section	Change Description
1.0	09/08/2023	All	Document published.
1.1	16/08/2023	Examples 2, 3 & 4 added Appendix B	Examples 2, 3 & 4 added Line item number 9 updated

Audience

This document is for any software provider who has chosen to build or update their products to allow for Enhanced Reporting Requirements (ERR).

Document context

This document provides examples of business scenarios including technical messages in both SOAP/XML and REST/JSON. This document is designed to be read in conjunction with the SOAP/XML or REST/JSON example files as well as the rest of the Revenue Commissioners' Enhanced Reporting Requirements documentation suite including the relevant technical documents.

Introduction

This document describes the XML and JSON web service examples provided as part of the Enhanced Reporting Requirements Process. The examples are provided for ERR software developers to describe some of the new processes Revenue will be implementing as part of Enhanced Reporting Requirements.

Please note:

- There is an adjoining Zip file for each example outlined in this document. The Zip file contains the XML and JSON example files.
- Monetary figures in all examples are for illustrative purposes only.
- The tables below are for illustrative purposes only. They are not an exhaustive list of fields that are required in an ERR submission.

Overview

- Before preparing an ERR Submission an employer can start off by requesting the employee's employment ID from Revenue (if they do not already have this information) by submitting a Lookup Enhanced Reporting Notification (ERN) request.
- Revenue receives the employer's request and sends the ERN to the employer (Lookup ERN response).

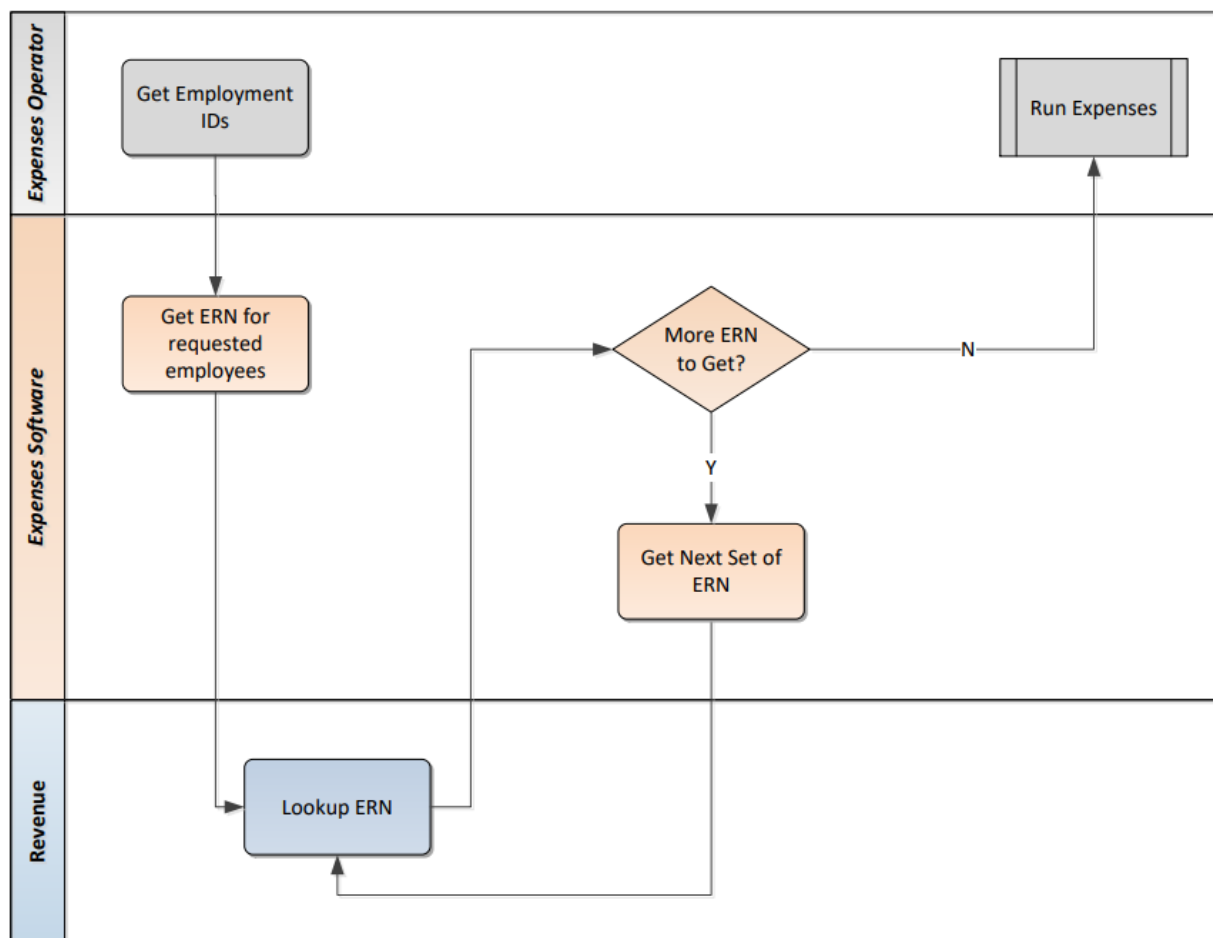


Figure 1 Get Enhanced Reporting Notification (ERN) process Flow

- When the employer has received the ERN they run their ERR Submission event and submit this to Revenue (ERR Submission Request).
- Once Revenue has verified that it is a valid submission an acknowledgement response is created and sent to the employer.

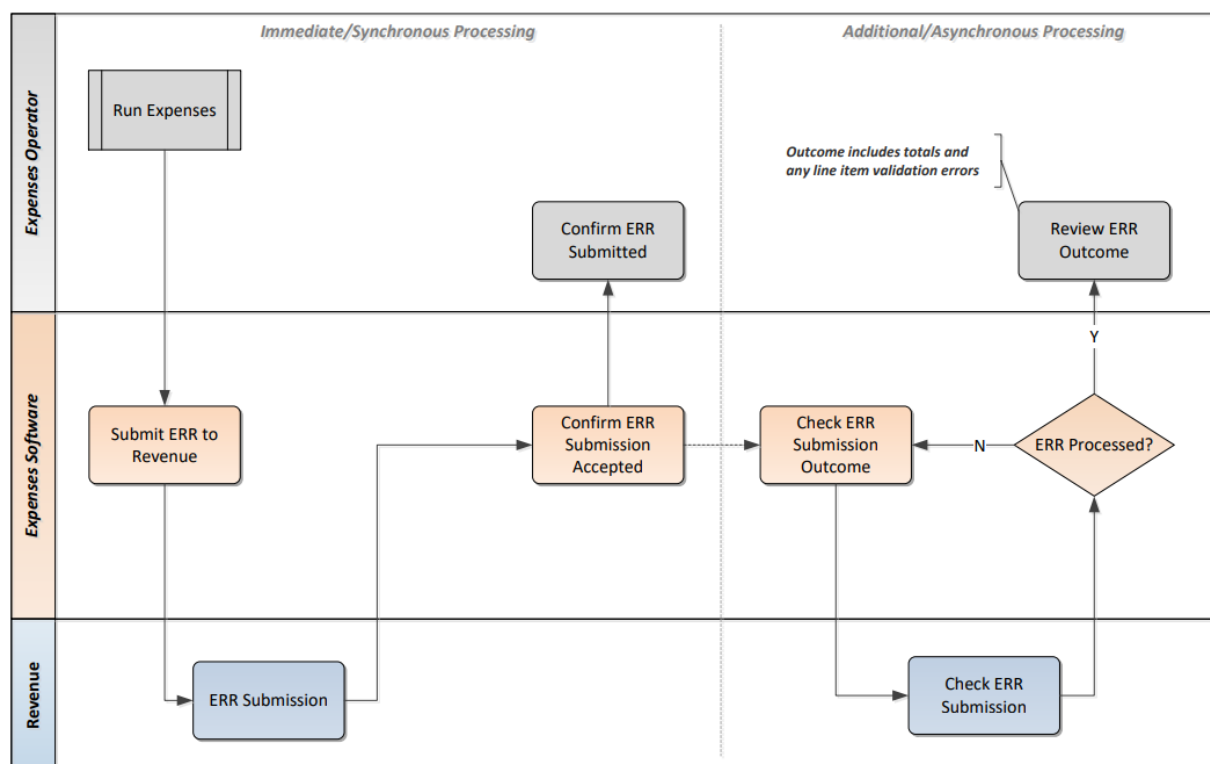


Figure 2 ERR Submission Process Flow

- This document also illustrates the processes involved in Check ERR Submission requests and Check ERR Run requests.
 - The Check ERR Submission request submitted by an employer is requesting the overall total of expenses/benefits for a specific ERR submission. The submission is identified by the Enhanced Reporting Submission Reference (generated by the expense/benefit software when employer runs an ERR event).

- The Check ERR Run request submitted by an employer is requesting the overall total of expenses/benefits associated with the Run. The response will also contain information on the individual submissions (Line Items) that made up the ERR run including the totals for the line items and the category (and sub-category if applicable) of expense/benefit. The ERR run the employer wishes to request information on is identified by the Enhanced Reporting Run Reference (generated by the expense/benefit software when employer runs an ERR event).

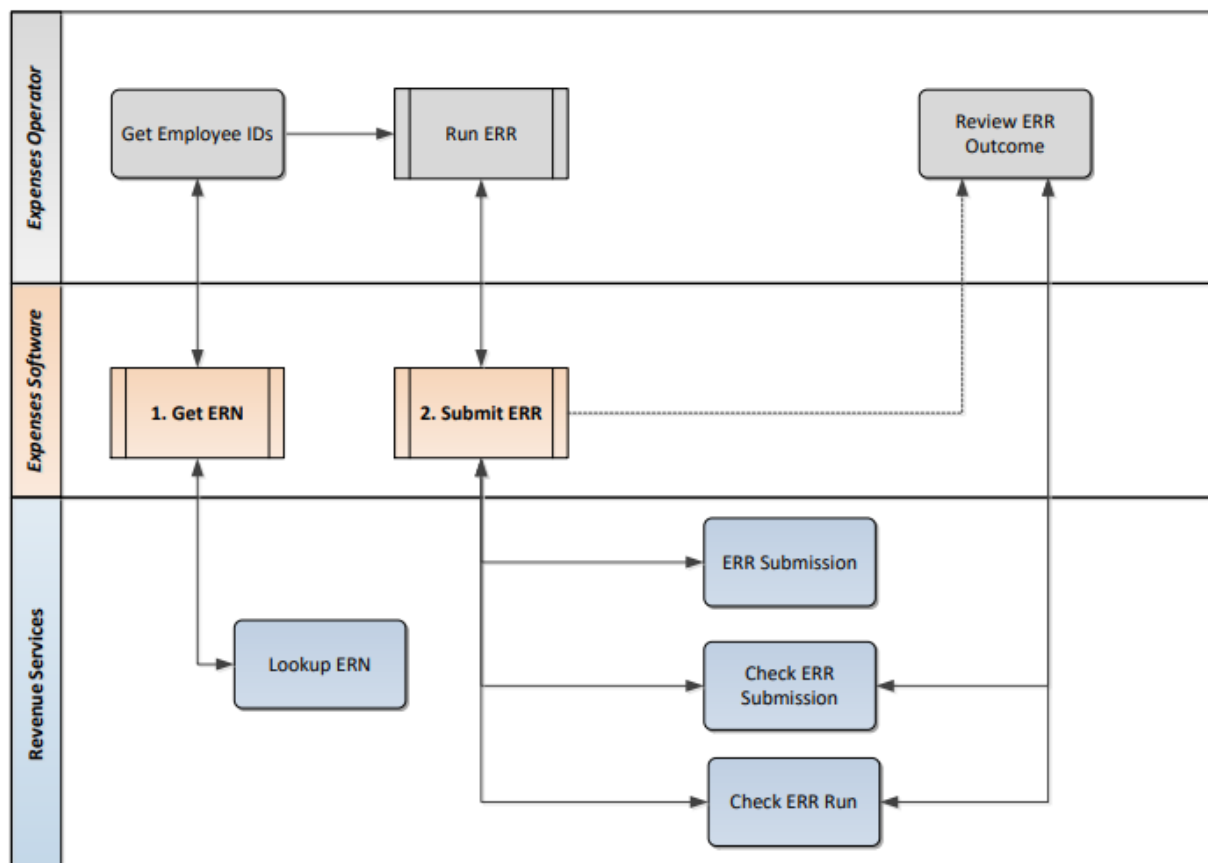


Figure 3 ERR Submission, Check ERR Run & Check ERR Submission Process Flow

- This document also illustrates the processes involved in the Lookup ERR Monthly Report webservice. The ERR Monthly Report for the requested month will contain a category breakdown which contains the overall totals for each category of expense, a sub-category breakdown which contains the overall totals for each sub-category of expense for the requested month. The figures that make up these totals are based on ERR submission data submitted to Revenue by the employer/agent. Any ERR Run that contains a line item with a pay date in the specific period will be detailed in the run reference breakdown section of the monthly report. Totals from line items that have failed validation and have been subsequently deleted will not be included in the Statutory Return. Totals from previously deleted line items will not be included in the Lookup ERR Return by Period response unless they have been resubmitted. Totals from invalid line items will not be included in the monthly report total.

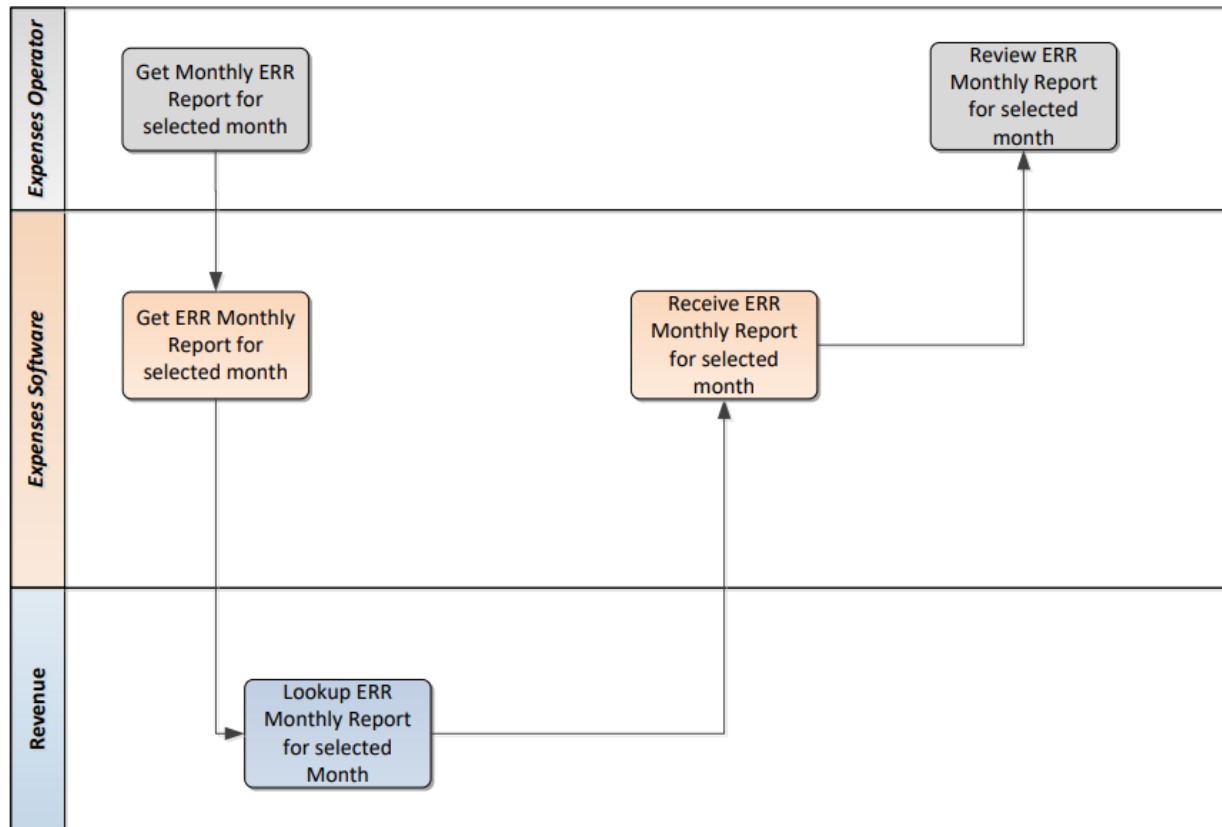


Figure 4 Returns Reconciliation Service – Lookup ERR Return by Period Process Flow

Example 1 – Full ERR Life Cycle

Example 1 demonstrates the full life cycle of the new Enhanced Reporting Requirements processes for an employer (Employer1).

1.1 Lookup ERN Request Example

Employer1 has an employee that they are submitting expenses/benefits for. Employer1 does not know the Employment ID of the employee. Employer1 can request the Employment ID for employees by submitting a Lookup ERN request to Revenue before an ERR submission.

Figure 5 below illustrates the XML structure of a Lookup ERN request.

Figure 7 below illustrates the JSON structure used when calling the Lookup ERN endpoint.



Figure 5 Lookup ERN Request - XML Structure



Figure 6 The Elements that make up Employee ID - XML Structure

Look Up ERN Request

GET /ern/{employerRegistrationNumber}/{taxYear}

Request to get an ERN by PPSN.

PARAMETERS

Path Parameters ?

employerRegistrationNumber	string Required Employer's PAYE Registration Number.
taxYear	integer <int32> Required The tax year to which the ERN lookup relates (YYYY).

Query Parameters ?

agentTain	string Tax Advisor Identification Number. Required if ERN is requested by an agent on behalf of the employer.
softwareUsed	string Required Name of third party software product.
softwareVersion	string Required Version of third party software product.
ppsns	Multiple query params of string Required Look up ERN by PPSN. Required parameter

Figure 7 Request to get ERN – JSON Structure

1.2 Lookup ERN Response Example

The different sources that trigger an updated ERN to be made available to an employer are:

- The employer submits a Lookup ERN Request to Revenue.

In this example the employee's ERN details are sent to the employer.

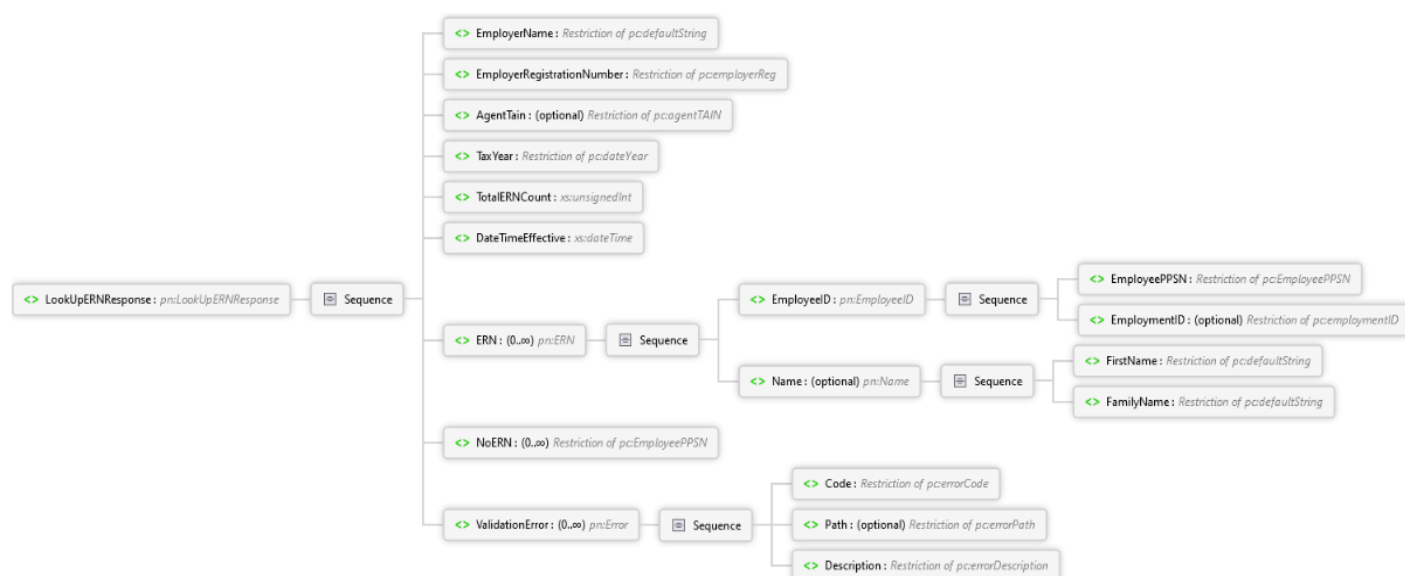


Figure 8 Lookup ERN Response - XML Structure

RESPONSE SAMPLES

200 successful operation

```
{
  "employerName": "string",
  "employerRegistrationNumber": "string",
  "taxYear": 2000,
  "agentTain": "string",
  "totalERNCount": 0,
  "dateTimeEffective": "2023-07-13T11:29:22Z",
  - "erns": [
    - {
      - "employeeID": {
        "employeePpsn": "stringst",
        "employmentID": "string"
      },
      - "name": {
        "firstName": "string",
        "familyName": "string"
      }
    }
  ],
  - "noERNS": [
    "string"
  ],
  - "validationErrors": [
    - {
      "code": "string",
      "path": "string",
      "description": "string"
    }
  ]
}
```

Figure 9 Response to a Request to get ERN - JSON Structure

1.3 ERR Submission Request Example

In this example Employer1 uses the Employment ID for employee Ann that was received in the ERN Response to run the ERR Submission event. Ann is being reimbursed for Travel and Subsistence with the sub-category Travel Vouched.

Employer1 runs the expenses/benefits and submits the ERR Submission request to Revenue.

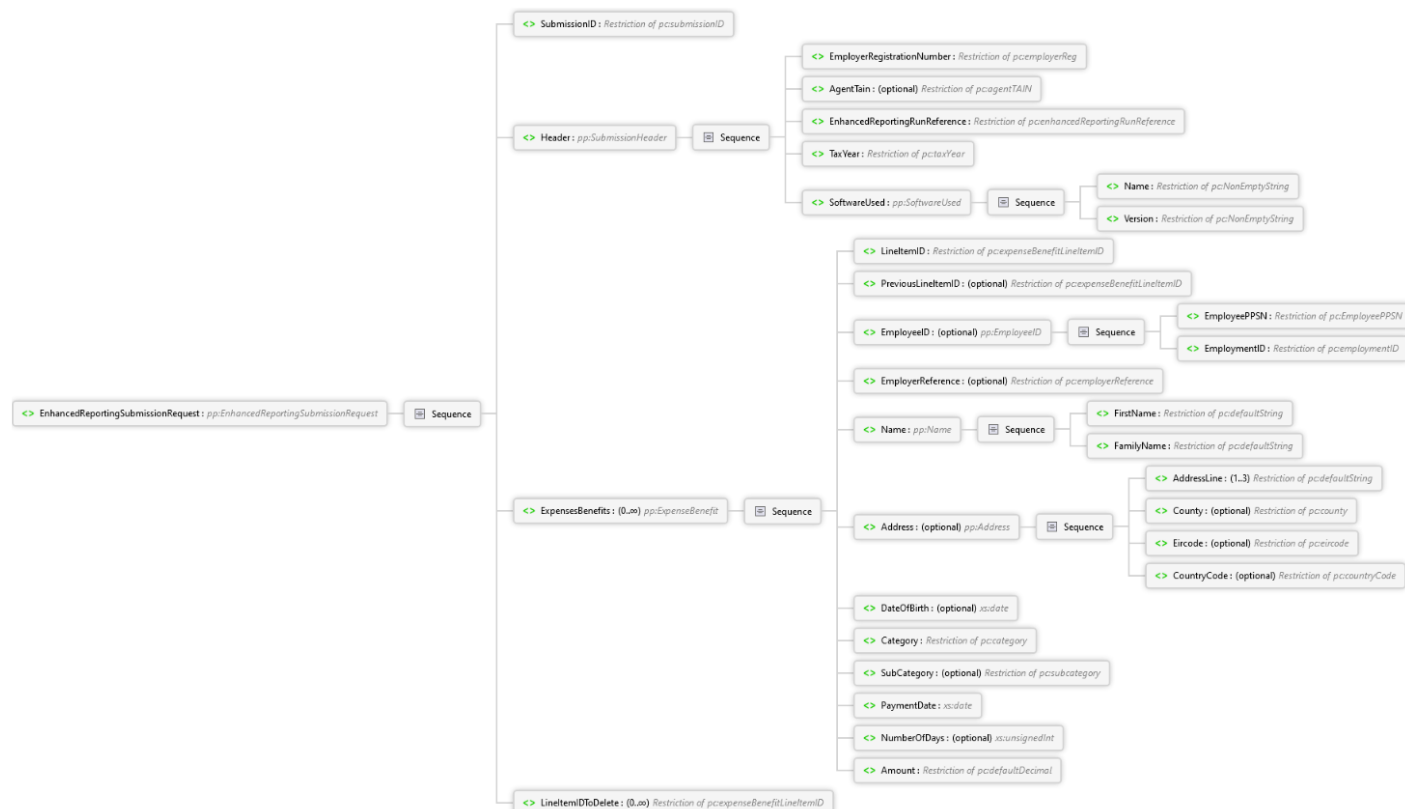


Figure 10 ERR Submission Request – XML Structure

Enhanced Reporting Submission Request

POST /enhanced_reporting/{employerRegistrationNumber}/{taxYear}/{enhancedReportingRunReference}/{submissionID}

Employer's PAYE Enhanced Reporting Submission Request.

PARAMETERS

Path Parameters

employerRegistrationNumber	string Required Data Item Ref: 1. Employer's PAYE Registration Number.
taxYear	integer <int32> Required Data Item Ref: 6. Used to identify the tax year to which the submission relates (YYYY).
enhancedReportingRunReference	string Required Data Item Ref: 4. Used to identify the Enhanced Reporting event to which the submission refers.
submissionID	string Required Data Item Ref: 3. Unique submission identifier. Must be unique for submissions under a given employer's PAYE registration number.

Query Parameters

agentTain	string Data Item Ref: 2. Tax Advisor Identification Number. Required if submission filed by agent on behalf of employer.
softwareUsed	string Required Name of third party software product.
softwareVersion	string Required Version of third party software product.

expensesBenefits ^	Array of ExpenseBenefit List of Expense Benefits that make up the Employer Enhanced Reporting Submission.
Array [
lineltemID	string [0..50] characters Required / [A-Za-z0-9_\-]* / Data Item Ref: 5. Used to uniquely identify individual line items.
previousLineltemID	string [0..50] characters / [A-Za-z0-9_\-]* / Data Item Ref: N/A. Used to identify a previous version of this line item when corrections are made. Can only reference line items within the same Enhanced Reporting Run.
employeeID v	EnhancedReportingEmployeeID Unique references to an employment for an employee. Includes PPSN and Employment ID.
employerReference	string [0..50] characters / [A-Za-z0-9áéíóúÁÉÍÓÚ _.,~!/@;£\$%&"<>\\.*()\[\]{}+~?]* / Data Item Ref: 14. Employee's internal staff identifier/reference. This will be mandatory if the PPSN isn't available and must remain unchanged in all submissions received until the PPSN is available. Optional if PPSN is available.
name v	EnhancedReportingName Required Employee Name.
address v	EnhancedReportingAddress The address of the employee. This will be mandatory if the PPSN isn't available.
dateOfBirth	string <date> Data Item Ref: 13. Employee's Date of Birth - format DD/MM/YYYY. This will be Mandatory if the PPSN isn't available.
category	string Required Valid values: "TRAVEL_AND_SUBSISTENCE" "REMOTE_WORKING_DAILY_ALLOWANCE" "SMALL_BENEFITS_EXEMPTION" Data Item Ref: 18. Category of Expense Benefit. Possible values are Travel and Subsistence, Remote Working Daily Allowance, Small Benefits Exemption.
subCategory	string Valid values: "TRAVEL_VOUCHED" "TRAVEL_UNVOUCHED" "SUBSISTENCE_VOUCHED" "SUBSISTENCE_UNVOUCHED" "SITE_BASED_EMPLOYEES" "EMERGENCY_TRAVEL" "EATING_ON_SITE" Data Item Ref: 19. Subcategory of Expense Benefit. Will be mandatory if relevant category is selected. Possible values for Travel and Subsistence category are Travel Vouched, Travel Unvouched, Subsistence Vouched, Subsistence Unvouched, Site Based Employees, Emergency Travel, Eating on Site.
paymentDate	string <date> Required Data Item Ref: 16. Date employee is paid Expense Benefit (YYYY-MM-DD).
numberOfDays	integer <int32> Data Item Ref: 20. Number of Days. The total number of days for the Expense Benefit.
amount	number [-999999999..999999999.99] Required Data Item Ref: 17. Expense Benefit Amount. The total amount of the Expense Benefit.
]	
lineltemIDsToDelete ^	Array of LineltemID For correction submissions, this should include any line item IDs across the run to be deleted.
Array [
lineltem	string [0..50] characters / [A-Za-z0-9_\-]* /
]	

Figure 11 Employers ERR Submission Request - JSON Structure

1.4 ERR Submission Response Example

Once Revenue has verified that the ERR Submission request is a valid submission an acknowledgement response is created and sent to the employer.

This informs the employer that the ERR submission has been acknowledged by Revenue.

If the ERR submission is not valid an error response is created and sent to the employer.



Figure 12 ERR Submission Response - XML Structure

Responses

^ 200 successful operation

RESPONSE SCHEMA

acknowledgementStatus	string Required Valid values: "ACKNOWLEDGED" "REJECTED" Acknowledgement status for Employer's PAYE Enhanced Reporting Submission Request. Possible values are Acknowledged or Rejected.
acknowledgementID	string [0..50] characters / [A-Za-z0-9_]* / Unique ID for acknowledged requests - to be used for technical support if required.
validationErrors ^	Array of EnhancedReportingError. List of Submission Validation Errors (if any).
Array [
code	string [0..50] characters Required / [A-Za-z0-9_]* / Technical Error Code - to be used for mapping to error messages.
path	string [0..500] characters / [A-Za-z0-9áéíóúÁÉÍÓÚ=_^~!/@;€\$%#&*~\.*\N\{}+?]* / Path to error in schema, if available.
description	string [0..500] characters Required / [A-Za-z0-9áéíóúÁÉÍÓÚ=_^~!/@;€\$%#&*~\.*\N\{}+?]* / Technical description of error.
]	
validationWarnings ^	Array of Warning List of Submission Validation Warnings (if any).
Array [
code	string [0..50] characters Required / [A-Za-z0-9_]* / Technical Code for warning - to be used for mapping to warning messages.
path	string [0..500] characters / [A-Za-z0-9áéíóúÁÉÍÓÚ=_^~!/@;€\$%#&*~\.*\N\{}+?]* / Path to warning in schema, if available.
description	string [0..500] characters Required / [A-Za-z0-9áéíóúÁÉÍÓÚ=_^~!/@;€\$%#&*~\.*\N\{}+?]* / Technical description of warning.
]	
— 400 Bad request. Invalid path, parameter or value supplied	
— 401 Authentication is required or has failed	
— 403 Insufficient permissions to process request	
— 404 Resource not found	
— 405 Request method is not supported for the requested resource	

Figure 13 Response to the ERR Submission Request - JSON Structure

1.5 Check ERR Submission Request Example

In this example Employer1 submits a Check ERR Submission request to Revenue.

The employer would like to know the current status of a specific ERR submission. The Submission ID is used to identify the ERR submission that the employer is requesting.

Figure 14 below illustrates the XML structure of a Check ERR Submission request.

Figure 15 below illustrates the JSON structure used when calling the Check ERR Submission endpoint.



Figure 14 Check ERR Submission Request - XML Structure

Check ERR Submission Request

GET /enhanced_reporting/{employerRegistrationNumber}/{taxYear}/{enhancedReportingRunReference}/{submissionID}

Employer's PAYE Check ERR Submission Request.

PARAMETERS

Path Parameters ?

employerRegistrationNumber	string Required Data Item Ref: 1. Employer's PAYE Registration Number.
taxYear	integer <int32> Required Data Item Ref: 6. Used to identify the tax year to which the submission relates (YYYY).
enhancedReportingRunReference	string Required Data Item Ref: 4. Used to identify the Enhanced Reporting event to which the submission refers.
submissionID	string Required Data Item Ref: 3. Unique Submission ID of the Employer Enhanced Reporting Submission. Used to identify individual submissions.

Query Parameters ?

agentTain	string Data Item Ref: 2. Tax Advisor Identification Number. Required if submission is filed by agent on behalf of employer.
softwareUsed	string Required Name of third party software product.
softwareVersion	string Required Version of third party software product.

Figure 15 Request to Check the Current Status of an Employer's ERR Submission - JSON Structure

1.6 Check ERR Submission Response Example

In this example the Check ERR Submission response shows Employer1 that the ERR submission has a status of 'Completed'.

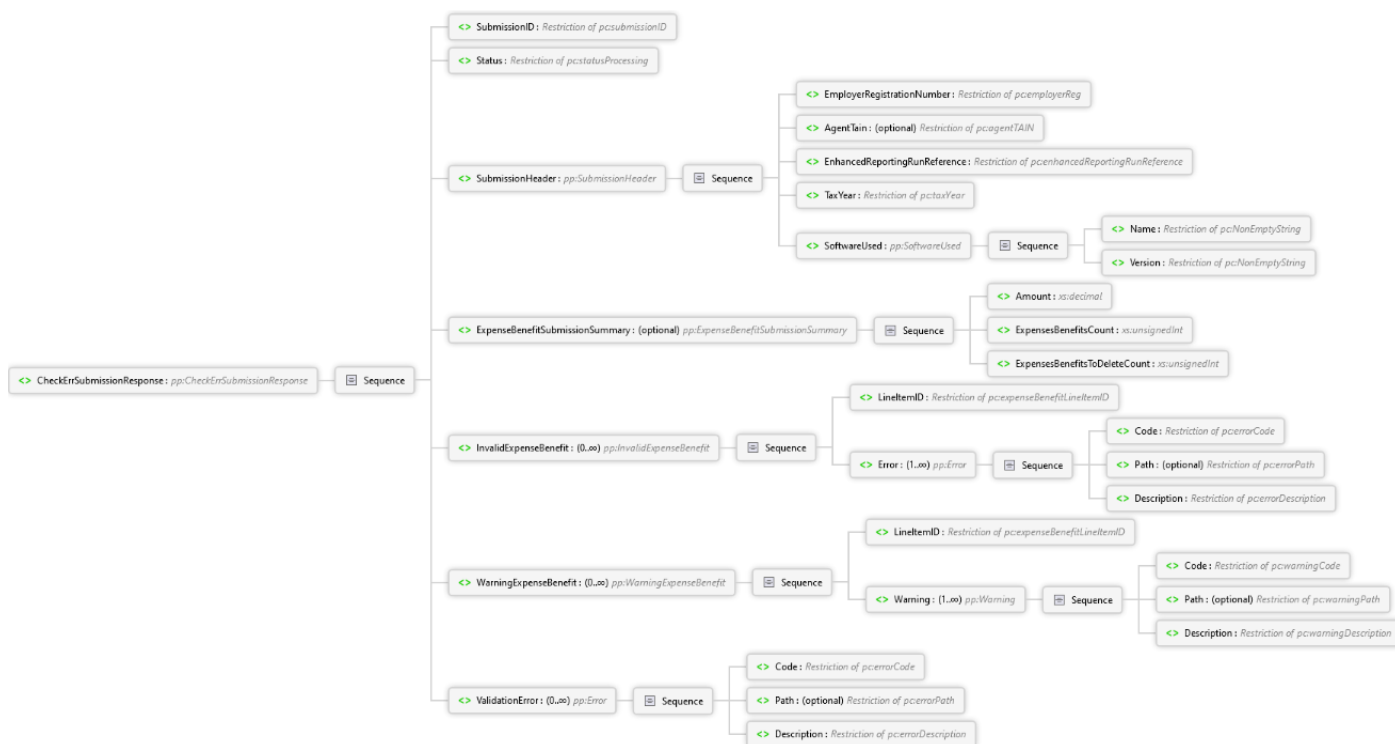


Figure 16 Check ERR Submission Response - XML Structure

^ 200 successful operation

submissionID	string [0..50] characters Required // [A-Za-z0-9_~]* Data Item Ref: 3. Unique Submission ID of the Employer Enhanced Reporting Submission. Used to identify individual submissions.
status	string Valid values: "PENDING" "PROCESSED" "COMPLETED" Data Item Ref: N/A. Current Status of Enhanced Reporting Submission. Possible values are Pending or Completed.
expenseBenefitSubmissionSummary	ExpenseBenefitSubmissionSummary Summary totals for Expense Benefit Submission. Included when submission has completed processing.
amount	number Required Data Item Ref: N/A. Total amount across valid line items on submission.
expensesBenefitsCount	Integer <int64> Required Data Item Ref: N/A. Total number of line items included on submission.
expensesBenefitsToDeleteCount	Integer <int64> Required Data Item Ref: N/A. Total number of Expense Benefit line item IDs for deletion included on submission.
invalidExpensesBenefits	Array of InvalidExpenseBenefit List of Invalid Expenses Benefits (if any). Any in this list will not have been saved.
InvalidItemID	string [0..50] characters Required // [A-Za-z0-9_~]* LineItemID of the Expense Benefits containing validation errors.
errors	Array of EnhancedReportingError Required List of validation errors related to the Expense Benefits.
expenseBenefitWarnings	Array of WarningExpenseBenefit List of Expense Benefit warnings (if any) associated with valid item. Any in this list will have been saved.
InvalidItemID	string [0..50] characters Required // [A-Za-z0-9_~]* LineItemID of the Expense Benefit containing validation warnings.
warnings	Array of Warning Required Details of validation check warning.
validationErrors	Array of EnhancedReportingError List of Errors (if any) on the CheckErrSubmissionRequest.
code	string [0..50] characters Required // [A-Za-z0-9_~]* Technical Error Code - to be used for mapping to error messages.
path	string [0..500] characters // [A-Za-z0-940&E100 _~\~/@:€\$%&'()*+,-./:;<=>?`{ }~]* Path to error in schema, if available.
description	string [0..500] characters Required // [A-Za-z0-940&E100 _~\~/@:€\$%&'()*+,-./:;<=>?`{ }~]* Technical description of error.

- 400 Bad request. Invalid path, parameter or value supplied
- 401 Authentication is required or has failed
- 403 Insufficient permissions to process request
- 404 Resource not found
- 405 Request method is not supported for the requested resource

1.7 Check ERR Run Request Example

In this example Employer1 submits a Check ERR Run request to Revenue. The employer would like to know the overall amounts (expenses and benefits) that are reported to Revenue for this ERR Run. The Run Reference is used to identify the ERR event that this submission refers to.

Figure 18 below illustrates the XML structure of a Check ERR Run request.

Figure 19 below illustrates the JSON structure used when calling the Check ERR Run endpoint.

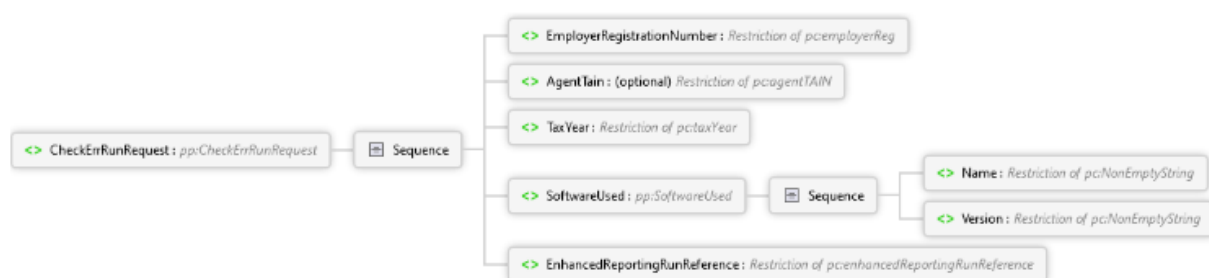


Figure 18 Check ERR Run Request – The Run Reference is used to identify the ERR Run - XML Structure

Check ERR Run Request

GET /enhanced_reporting/{employerRegistrationNumber}/{taxYear}/{enhancedReportingRunReference}

Employer's PAYE Check ERR Run Request.

PARAMETERS

Path Parameters ?

employerRegistrationNumber	string Required Data Item Ref: 1. Employer's PAYE Registration Number.
taxYear	integer <int32> Required Data Item Ref: 6. Used to identify the tax year to which the submission relates (YYYY).
enhancedReportingRunReference	string Required Data Item Ref: 4. Used to identify the Enhanced Reporting event to which the submission refers.

Query Parameters ?

agentTain	string Data Item Ref: 2. Tax Advisor Identification Number. Required if submission is filed by agent on behalf of employer.
softwareUsed	string Required Name of third party software product.
softwareVersion	string Required Version of third party software product.

Figure 19 Request to Check the Current Status of an Employer’s PAYE ERR Run - JSON Structure

1.8 Check ERR Run Response Example

A Check ERR Run Response can either have a status of:

- ‘Pending’ – Will be marked as ‘Pending’ if any submissions making up the ERR Run are at a status of Pending.

Or

- ‘Processed’ – Will be marked as ‘Processed’ when all submissions that make up the ERR run have finished processing.

Please note: The term ‘Completed’ is not used in relation to the Check ERR Run Response, as the process is not final. Unlike, the Check ERR Submission Operation, where the submission is sent directly to Revenue and a Correction Submission must be sent for any amendments; the Check ERR Run Response is just a confirmation that an ERR run has been submitted to Revenue.

In this example the Check ERR Run response shows Employer1 that the ERR submission has a status of ‘Processed’. The overall totals for expenses/benefits for the ERR run will not be shown while the ERR Submission request has a status of ‘Pending’.

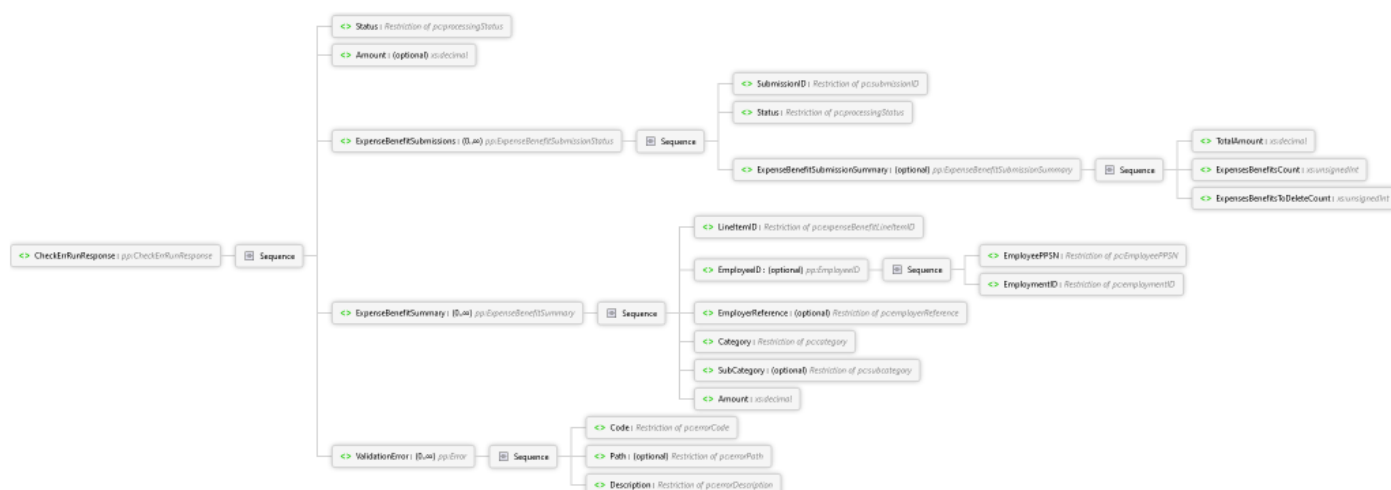


Figure 20 Check ERR Run Response

```
{
  "status": "PENDING",
  "amount": 0,
  - "expenseBenefitSubmissions": [
    - {
      "submissionID": "string",
      "status": "PENDING",
      - "expenseBenefitSubmissionSummary": {
        "totalAmount": 0,
        "expensesBenefitsCount": 0,
        "expensesBenefitsToDeleteCount": 0
      }
    }
  ],
  - "expenseBenefitSummaries": [
    - {
      "lineItemID": "string",
      - "employeeID": {
        "employeePpsn": "stringst",
        "employmentID": "string"
      },
      "employerReference": "string",
      "category": "TRAVEL_AND_SUBSISTENCE",
      "subCategory": "TRAVEL_VOUCHERED",
      "amount": 0
    }
  ],
  - "validationErrors": [
    - {
      "code": "string",
      "path": "string",
      "description": "string",
      "id": "string"
    }
  ]
}
```

Figure 21 Response to the Request to Check the Current Status of an Employer's ERR Run - RESTful web service

1.9 Lookup ERR Monthly Report Example

Example 1.9 demonstrates the employer/agent requesting an ERR Monthly Report for January 2024. The ERR Monthly Report contains the overall totals for the categories of expenses/benefits, the overall totals for the sub-categories and the run reference breakdown for the specified month.

The category summary overall totals will be based on the ERR submission data with pay dates in the selected month i.e. only figures from line items with pay dates in the selected month will be included in the return.

The category summary overall totals include the sum of each category in the ERR submissions for the selected period.

The sub-category summary overall totals include the sum of each category in the ERR submissions for the selected period.

Each ERR Run that contains line items with pay dates in the selected month will be detailed in the Lookup ERR Monthly Report response.

The ERR Monthly Report will be for completed months only and not available for the current month.

In this example Employer1 submits a Lookup ERR Monthly Report request to Revenue, with the month being January 2024. The employer would like to know the overall totals (categories and sub-categories of expenses/benefits) that make up the ERR Monthly Report for January 2024. The month along with the Employer Registration Number are used to identify the Month and the employer that this request refers to.

Figure 34 below illustrates the XML structure of a Return request.

Figure 35 below illustrates the JSON structure used when calling the Return Request endpoint.



Figure 22 Lookup ERR Monthly Report Request – XML Structure

Request Monthly ERR Report

GET /enhanced-reporting/reports/monthly/{employerRegistrationNumber}/{taxYear}/{month}

Employer request to get a report of all expenses/benefits within the month provided.

PARAMETERS

Path Parameters ?

employerRegistrationNumber	string Required Data Item Ref: 301. Employer's PAYE Registration Number.. Used to identify employer to which the period details relate.
taxYear	integer <int32> Required Data Item Ref: 305. Tax year that the expenses/benefits report is being requested for.
month	string Required Valid values: "JANUARY" "FEBRUARY" "MARCH" "APRIL" "MAY" "JUNE" "JULY" "AUGUST" "SEPTEMBER" "OCTOBER" "NOVEMBER" "DECEMBER" Data Item Ref: 306. Month to process for report breakdown.

Query Parameters ?

agentTain	string Data Item Ref: 302. Used to identify the agent requesting on behalf of the employer and to ensure that an agent link exists for this employer agent relationship for the period that the request relates to.
softwareUsed	string Required Data Item Ref: 303. Third party software product identifier: name.
softwareVersion	string Required Data Item Ref: 304. Third party software product identifier: version.

Figure 23 Lookup ERR Monthly Report Request – JSON Structure

1.10 Lookup ERR Monthly Report Response Example

In this example the Lookup ERR Monthly Report response is sent to the employer/agent.

The most up to date version of the report is sent to the employer/agent. The category breakdown details show the overall amounts for each category for the month. The sub-category breakdown details show the overall amounts for each sub-category for the month.

The ERR Run Reference breakdown section lists all the ERR runs that contain line items with a pay date in January 2024.

Table 1 Details of the employers Return Response for January 2024¹

Employer Registration Number	Agent TAIN	Month	Year	Number of Expenses/benefits	Overall Total
1234567T	N/A	January	2024	2	€165
Category Breakdown					
Small Benefits Exemption			Travel and Subsistence		
€65			€100		
Sub-Category Breakdown					
Travel Vouched					
€100					
ERR Run Details					
ERR Run Reference	Submission Date		Total Amount		
Run1	2024-07-25		€100		
Run2	2024-07-25		€65		

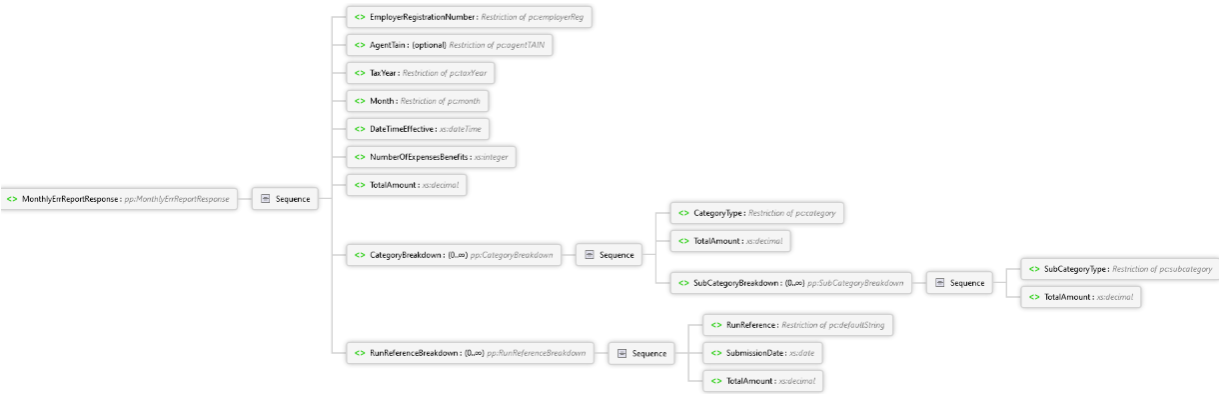


Figure 24 Lookup ERR Return by Period Response – XML Structure

¹ Totals from previously deleted line items will not be included in the Lookup ERR Return by Period response unless they have been resubmitted.

RESPONSE SAMPLES

- 200 successful operation

```
{
  "employerRegistrationNumber": "string",
  "agentTain": "string",
  "taxYear": 2000,
  "month": "JANUARY",
  "dateTimeEffective": "2023-08-01T08:31:42Z",
  "numberOfExpensesBenefits": 0,
  "totalAmount": 0,
  - "categoryBreakdowns": {
    - "property1": {
      "totalAmount": 0,
      - "subCategoryBreakdown": {
        "property1": 0,
        "property2": 0
      }
    },
    - "property2": {
      "totalAmount": 0,
      + "subCategoryBreakdown": { - }
    }
  },
  - "runReferenceBreakdowns": {
    - "property1": {
      "submissionDate": "2023-08-01",
      "totalAmount": 0
    },
    - "property2": {
      "submissionDate": "2023-08-01",
      "totalAmount": 0
    }
  }
}
```

Figure 25 Lookup ERR Return by Period Response – ERR Run Details – JSON Structure

Example 2 – Overpayment of an Expense/Benefit to an Employee

If an Employee has been overpaid in an ERR submission e.g. if he was paid for an expense/benefit they weren't entitled to and the Employee actually received this overpayment, the correction to recoup this overpayment should be fixed in the next ERR event.

This example demonstrates how an employer can correct an overpayment to an employee in the next ERR event.

If there is an overpayment of an expense/benefit, and a recoupment does not take place then the overpayment will become taxable income.

2.1 ERR Submission Overpayment Example

John gets paid remote working relief of €16 a week. In Week 1 John was out sick one day. His employer doesn't pay him Remote Working Relief for sick leave so John should only have been paid €12.80 that week. However, the ERR operator wasn't informed on time and so he was paid the full €16.

Table 2 Details of the Week 1 ERR Submission for John

ERR Run Reference	Remote Working Relief	Number of days
Week 1	€16	5

2.2 Correction of ERR Submission Overpayment Example

An ERR operator should arrange to recoup an overpayment when they are informed of an error in the amount of Expenses/Benefits paid to John.

In this new ERR submission the ERR operator reports a negative figure of 3.20 for Remote Working Relief and a negative day in the Number of Days field. The date of this submission should reflect the date of the recoupment and not the original pay date.

If there is an overpayment of an expense/benefit, and a recoupment does not take place then the overpayment will become taxable income.

Table 3 Details of the ERR Submission Dealing with the Recoupment Submission for John

ERR Run Reference	Remote Working Relief	Number of days
Week1	€16	5
Week1-v2R	– €3.20	– 1

Example 3 – Underpayment of an Expense/Benefit to an Employee

This example demonstrates how an employer can correct an underpayment to an employee in the next ERR event.

If an employee has been underpaid in a ERR event e.g. if they weren't paid their full amount of Expense/Benefit, the correction of the underpayment should be fixed in the next ERR event.

3.1 ERR Submission Underpayment Example

Mary submitted receipts for Travel and Subsistence (sub-category Travel Vouched) for €150. The Expenses/Benefits Operator mistakenly paid Mary €100 as the amount for the expense/benefit and this was reported to revenue in the ERR Submission on 01/02/2024.

3.2 Correction of ERR Submission Underpayment Example

Mary's employer makes a separate payment of €50 to Mary on 02/01/2024 to correct the underpayment. An ERR Submission is made to Revenue on 02/01/2024 detailing this payment of €50. These ERR submissions reflect the actual amount of Expenses/Benefits Mary received. In this way Mary's Revenue record of Expenses/Benefits will match what was in the ERR Submissions.

Example 4 – Amendment of Incorrect ERR Submission

A reporting error can be an error in a financial field that didn't actually happen. In this example the reporting error was that the employee actually got paid the correct amount of expenses/benefits but the amount of expenses/benefits on the report that Revenue received was different.

These examples demonstrate how an employer can amend an incorrect submission item on an ERR submission using either the previous line item ID field or line item IDs to delete.

The first correction method is to amend the incorrect submission item using the previous line item ID field. The employer submits a new ERR submission, using the original ERR Run reference, with the corrected submission item(s) included. The corrected submission item(s) has its "Previous Line Item ID" field set with the value of Line Item ID of original incorrect submission item(s). By setting the Previous Line ID field, Revenue is instructed to replace the submission item referenced in this field with the new submission item. This is the preferred method as it creates a link between the correct submission item(s) and the original incorrect submission item(s) that is being replaced.

The second method involves deleting the incorrect submission item(s) and submitting a new submission item(s) with the corrected values. The employer submits a new ERR submission, using the original ERR Run reference, with the corrected submission item(s) included however the Previous Line ID field is **not** set in these submission items. The employer also enters the Line Item IDs* of the original incorrect submission items in the "Line Item IDs to Delete" section of this ERR submission. By setting the "Line Item IDs to Delete" section, Revenue is instructed to delete the submission item(s) referenced in this section**. This method allows the Employer to provide corrected submission items and delete incorrect submission items but does not create a link between the two

*Unique reference given to each line items across submissions for a given ERR run, that are generated by the ERR software when running an ERR event.

** Please note that deleted line item IDs cannot be reused in a ERR submission under the same ERR run reference

4.1 Invalid ERR Submission Example

Patrick was correctly paid €50 in Week 1 for Travel and Subsistence. However, in the ERR Submission request that the employer submitted to Revenue Patrick's pay was actually reported as €500.

4.2 Amendment using Previous Line Item ID Example

The **Error! Reference source not found.** ERR event needs to be replaced. This can be done by submitting a new ERR event for the employee for Week 1 with the correct pay and a new Line Item ID (e.g. "E3-v2"). The ERR event to be amended is identified in previous line item ID field the by the unique Line Item ID (e.g. E3-v1).

Now Patrick’s Revenue record of Expenses/Benefits will match what was in his submission item for Week 1.

Table 4 Details of the Corrected Week 1 ERR Submission for Patrick

ERR Run Reference	Line Item ID	Amount	Category	Sub-Category	Previous Line item ID
Week 1	E3-v2	€50	Travel & Subsistence	Travel Vouched	E3-v1

4.3 Amendment using Line Item IDs to Delete Example

The above ERR event could also be amended in two stages, firstly by deleting the original submission and then secondly by submitting a new ERR event.

This is done by deleting the original ERR event for the employee for Week 1 and submitting a new ERR event for the employee with the correct submission items and a new Line Item ID (e.g. “E3- v2”). The incorrect ERR event that is to be deleted is identified in the line item IDs to delete field by the unique Line Item ID (e.g. E3-v1).

Now Patrick’s Revenue record of Expenses/Benefits will match what was in his submission item for Week 1.

Table 5 Details of the Corrected Week 1 ERR Submission for Patrick

ERR Run Reference	Line Item ID	Amount	Category	Sub-Category	Line item to Delete
Week 1	E3-v2	€50	Travel & Subsistence	Travel Vouched	E3-v1

<pre> subCategory paymentDate numberOfDays amount] </pre>	<p>Data Item Ref: 18. Category of Expense Benefit. Possible values are Travel and Subsistence, Remote Working Daily Allowance, Small Benefits Exemption.</p> <p>string</p> <p>Valid values: "TRAVEL_VOUCHED" "TRAVEL_UNVOUCHED" "SUBSISTENCE_VOUCHED" "SUBSISTENCE_UNVOUCHED" "SITE_BASED_EMPLOYEES" "EMERGENCY_TRAVEL" "EATING_ON_SITE"</p> <p>Data Item Ref: 19. Subcategory of Expense Benefit. Will be mandatory if relevant category is selected. Possible values for Travel and Subsistence category are Travel Vouched, Travel Unvouched, Subsistence Vouched, Subsistence Unvouched, Site Based Employees, Emergency Travel, Eating on Site.</p> <p>string <date> Required</p> <p>Data Item Ref: 16. Date employee is paid Expense Benefit (YYYY-MM-DD).</p> <p>integer <int32></p> <p>Data Item Ref: 20. Number of Days. The total number of days for the Expense Benefit.</p> <p>number [-999999999..999999999.99] Required</p> <p>Data Item Ref: 17. Expense Benefit Amount. The total amount of the Expense Benefit.</p>
<p>lineItemIDsToDelete ^</p>	<p>Array of LineItemID</p> <p>For correction submissions, this should include any line item IDs across the run to be deleted.</p>
<pre> Array [lineItem] </pre>	<p>string [0..50] characters / [A-Za-z0-9_\.]* /</p>

Figure 26 ERR Submission Request – Line Item ID to Delete for Correction Submissions

Appendix A – Business Rules

Employment ID Business Rules

The ‘*Employment ID*’ field has a unique value for each separate employment that an employee has with each employer. It will be set by the employer. Revenue will use it, along with the *Employer Number*, the *PPSN*, when creating an employment for an employee. If the *PPSN* is available, the *Employment ID* field is mandatory, except for when doing a specific employee RPN lookup.

The *Employment ID* will be particularly useful where:

- An employee has multiple employments with the same employer, as it can uniquely identify each employment.
- An employee ceases and re-commences employments with the same employer.

Paying expenses to an employee where the PPSN is not available

Jane is starting a job with employer 1234567T for the first time on 1 November 2024. Jane has only recently moved to Ireland, and has no PPSN. In the absence of a PPSN, the employer assigns **Employer Reference** of 7654 to Jane and in addition must record Jane’s address and date of birth in the Expense Submission. The *Employment ID* is not assigned. Revenue cannot create an employment for Jane until the PPSN becomes available and she registers for PAYE.

Any Expense Submissions made for Jane without the PPSN are held on a holding table until the PPSN becomes available.

Table 6 Submitting where the PPSN is not available

Employer number	Run Reference	Employee Name	Employee PPSN	Employer Reference	Employment Id	Address	Date of Birth
1234567T	Week 33	Jane		7654		Apartment A, Main Street, Co. Galway	01/01/1975

Jane has been assigned a PPSN, 1234567X, in time for it to be included in the ERR run on 1 December 2024. An *Employment ID* of ‘1’ is allocated to Jane by the employer. The *Employer Reference* is also included in the ERR Submission which will link the PPSN to any previous ERR submissions for Jane.

Please note: the Employer Reference must be the same on all submissions. When the PPSN becomes available the next ERR Submission should contain the Employer Reference, Employment ID and PPSN in order for the submissions to link.

Please note: an ERR submission will not create an employment on revenue records.

Table 7 Including the PPSN once it becomes available

Employer number	Run Reference	Employee Name	Employee PPSN	Employer Reference Id	Employment	Address	Date of Birth
1234567T	Week 37	Jane	0000004H	7654	1	Apartment A, Main Street, Co. Galway	01/01/1975

Appendix B – Line Item Correction rules

Correction rules

Correction types will be as follows –

Correction Type	Correction Description
1	Overpayment to the Employee that can be fixed in the next Expense/Benefit Event 1. The correct information can be input in the next Event/Benefit submission
2	Underpayment to the Employee that can be fixed in the next Expense/Benefit Event 2. The correct information can be input in the next Event/Benefit submission
3	Reporting Error that cannot be fixed in the next Expense/Benefit Event 3. The incorrect line item would need to be deleted from the original Submission
4	Reporting Error that should be fixed in the next Expense/Benefit Event 4. The correct information can be input in the next Event/Benefit submission

Note on correction rules for data items:

If an Employee has been overpaid in an expenses event e.g. if they were paid for Expense/Benefit they were not entitled to and the Employee actually received this overpayment, the correction to recoup this overpayment should be done in the next Expense/Benefit submission.

Reporting errors could include an employee being omitted from the Expense/Benefit Report even though they were actually paid. The missing line item would need to be submitted to Revenue.

Reporting errors could also include an employee being included in the Expense/Benefit Report even though they weren't actually paid. This line item would need to be deleted.

Submission Request Header Data Items:

Item line number	Data Item	Correction Type	Context
1	Employer Registration Number	N/A	
2	Agent Tain	N/A	
3	Enhanced Reporting Submission Reference	N/A	
4	Enhanced Reporting Run Reference	N/A	
5	Line Item ID	N/A	
6	Tax Year	N/A	
7	Software Used	N/A	
8	Software Version	N/A	

Submission Request Line Items:

Item line number	Data Item	Correction Type	Context
9	Employee PPSN	3 or 4	<ol style="list-style-type: none"> Employers should take reasonable steps to verify that the PPSN used is correct. The PPSN checker is available to the employer in ROS The employer should delete the incorrect Expense/Benefit (Correction Type 3) or the original submission should be corrected in the next Event/Benefit submission using the Previous Line Item ID to identify the Expense/Benefit to be corrected.

Item line number	Data Item	Correction Type	Context
			3. The employer is responsible for dealing with any data protection issues arising within their systems.
10	First Name	4	These fields are used to match submissions without PPSNs to the correct Revenue record
11	Family Name	4	These fields are used to match submissions without PPSNs to the correct Revenue record
12	Address	4	These fields are used to match submissions without PPSNs to the correct Revenue record
13	Date of Birth	4	These fields are used to match submissions without PPSNs to the correct Revenue record
14	Employer Reference	3	The original submission must be corrected.
15	Employment ID	3	The original submission must be corrected.
16	Date of Payment/Benefit	3	The original submission must be corrected.
17	Amount / Value (€)	1, 2 or 3	If the error meant that the Employee was overpaid (Correction Type 1) or underpaid (Correction Type 2), it should be fixed in the next Expense/Benefit submission/s. If it was a Reporting error (Correction Type 3), the original submission must be corrected.
18	Category	1, 2 or 3	If the error meant that the Employee was overpaid (Correction Type 1) or underpaid (Correction Type 2), it should be fixed in the next Expense/Benefit submission/s. If it was a Reporting error (Correction Type 3), the original submission must be corrected.
19	Sub-Category	1, 2 or 3	If the error meant that the Employee was overpaid (Correction Type 1) or underpaid (Correction Type 2), it should be fixed in the next Expense/Benefit submission/s. If it was a Reporting error (Correction Type 3), the original submission must be corrected.
20	Number of Days	4	