



MFSA: JSON Schema Handbook for Non-Technical Users



V1.0

Contents



MFSA: JSON SCHEMA GUIDE	1
1 INTRODUCTION	4
1.1 WHO SHOULD USE THIS GUIDE?	4
1.2 WHAT IS COVERED IN THIS GUIDE?	4
1.3 WHAT IS NOT COVERED IN THIS GUIDE?	5
1.4 WHY DOES THE SCHEMA MATTER?	5
2 WHAT IS A JSON SCHEMA?	6
2.1 THE MAIN BUILDING BLOCKS OF AN HRRF SCHEMA	6
2.2 WHAT DOES A DATAPOINT PROPERTY LOOK LIKE?	8
2.2.1 UNDERSTANDING DATAPOINT IDS	10
2.3 UNDERSTANDING COMMON VALUE TYPES	11
2.4 ALLOWED VALUES AND DROPDOWN-STYLE QUESTIONS	12
2.5 SINGLE-VALUE FIELDS VS MULTI-VALUE FIELDS	12
2.6 REQUIRED FIELDS AND CONDITIONAL FIELDS	13
2.6.1 ALWAYS REQUIRED FIELDS	13
2.6.2 CONDITIONALLY REQUIRED FIELDS	13
2.6.3 COMMON CONDITIONAL SYMBOLS	14
2.6.4 UNDERSTANDING SUM CONDITIONS	15
3 CONNECTING THE SCHEMA TO THE EXCEL TEMPLATE	16
4 BUILDING A JSON SUBMISSION FROM THE SCHEMA	17
5 COMMON MISTAKES TO AVOID	19
6 GLOSSARY	21

Tables

Table 2.1. HRRF Schema Components	7
Table 2.2. Datapoint components.	9
Table 2.3. Datapoint ID components.	10
Table 2.4. Supported data types.	11
Table 2.5. Supported value formats.	11
Table 2.6. Expected field responses.....	12
Table 2.7. Most used conditionally symbols.	14
Table 2.8. Examples of SUM condition.	15
Table 3.1. Mapping of schema properties to Excel template.	16
Table 5.1. Common errors and their resolutions.	19

1 Introduction

This guide is designed for users who need to understand and work with HRRF JSON Schemas without requiring technical knowledge of JSON, programming, or schema development.

The HRRF JSON Schema is the official specification used to define what information can be reported, what values are allowed, which fields are mandatory, and when additional information may be required. While the schema itself is written in a technical format, the concepts behind it can be understood by business users, reviewers, reporting teams, and template owners.

This guide explains the key components of an HRRF schema in plain language and shows how they relate to the reporting process.

1.1 Who Should Use This Guide?

This guide is intended for anyone involved in preparing, reviewing, or understanding HRRF reporting requirements, including:

- Business users, who need to understand what information is being requested and how datapoints relate to one another.
- Template users, who need to map schema fields to Excel templates, reporting forms, or data collection tools.
- Reviewers and quality assurance teams, who need to understand why certain values are mandatory, restricted, or conditionally required.
- Technical users, who want a business-friendly companion to the detailed technical schema specification.

1.2 What Is Covered in This Guide?

This guide provides an overview of the key concepts needed to understand and use an HRRF JSON Schema. It explains how schemas are organised and structured, how datapoints are identified, and how information is represented within the schema. It also describes the purpose of commonly used schema attributes, such as titles, descriptions, and data types, and explains how validation rules are used to control the values that can be submitted.

In addition, the guide explains the difference between mandatory and optional fields, how dependency rules determine when additional information becomes required, and how the information contained within a schema can be used to prepare, review, or construct a valid submission. The aim is to help users

understand not only what information is required, but also why it is required and how the various components of the schema work together.

1.3 What Is Not Covered in This Guide?

The guide does not cover software development practices, programming techniques, or coding approaches used when implementing reporting solutions. It also excludes information relating to internal HRRF systems, databases, processing pipelines, and other platform-specific components that are not exposed through the published schema. Similarly, the technical processes used to generate schema artefacts, and the implementation details of system integrations are outside the scope of this guide.

1.4 Why Does the Schema Matter?

The schema acts as the rulebook for a submission. It defines the information that can be reported, identifies which datapoints are mandatory, specifies the values and formats that are permitted, and determines when additional information must be provided based on dependency conditions. In effect, the schema establishes the complete set of rules that a submission must satisfy before it can be accepted.

By bringing all these requirements together into a single, machine-readable specification, the schema helps ensure that submissions are consistent, complete, and aligned with the reporting requirements. It provides clarity for those preparing submissions, confidence for those reviewing them, and a common reference point for everyone involved in the reporting process.

Understanding the schema enables users to interpret reporting requirements correctly, identify mandatory and conditional datapoints, validate reported values, and understand the reasons behind validation rules. As a result, users can prepare, review, and assess submissions with greater confidence, even without detailed technical knowledge of JSON or schema design.

Note: Think of the HRRF schema as a digital version of a reporting form with built-in instructions and validation rules. It tells you what information is required, how it should be entered, and which additional questions need to be answered depending on earlier responses.

2 What is a JSON Schema?

A JSON Schema is a machine-readable set of rules that defines what information can be included in a JSON submission and how that information should be structured. It describes the fields that may appear in the submission, explains what each field represents, specifies the type of data expected, identifies which fields are mandatory, and defines any validation rules that must be satisfied.

The schema also contains dependency rules that determine when additional fields become applicable based on values provided elsewhere in the submission. In this way, the schema acts as both a data definition and a validation framework, ensuring that submissions are complete, consistent, and aligned with the reporting requirements.

Think of the JSON Schema as the instructions, guidance notes, and validation rules that accompany a reporting questionnaire. The schema tells you:

- What questions exist.
- Which questions must be answered.
- What type of answer is expected.
- Which answers are allowed.
- When additional questions become relevant.

The JSON submission is the completed questionnaire containing the answers.

In simple terms: The schema defines the rules, while the submission contains the data entered according to those rules.

A submission that follows all the rules defined in the schema is considered valid. A submission that omits mandatory information, provides invalid values, or ignores dependency requirements will fail validation.

2.1 The Main Building Blocks of an HRRF Schema

An HRRF schema is made up of several key components that work together to describe what information can be submitted, what values are allowed, and which reporting rules must be followed.

Most users will primarily interact with the datapoints defined within the schema, but understanding the purpose of the main schema components can make it easier to interpret reporting requirements and troubleshoot validation issues.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "$id": "https://.../<schema-file>.json",
```

```

"title": "HRRF Taxonomy Schema - ...",
"type": "object",
"properties": {
  "FINS_PR_2641_v1": {
    "type": "integer",
    "minimum": 0
  },
  "FINS_PR_2648_v1": {
    "type": "integer",
    "maximum": 100
  },
  "FINS_GI_11_v1": {
    "type": "string",
    "enum": [
      "Yes",
      "No",
    ]
  }
},
"required": [
  "FINS_GI_11_v1"
],
"additionalProperties": false,
"requiredIfConditionRefernce": [
  "[{[FINS_PR_2641_v1, FINS_PR_2648_v1]; (${FINS_GI_11_v1} <> 'Yes')}]"]
]
}

```

Table 2.1. HRRF Schema Components

Schema Component	What It Means in Simple Terms	Why Is It Needed?
\$schema	Identifies the version of the JSON Schema standard used to create the schema. This information is used by validation tools and does not affect how users complete a submission.	It tells the system how to interpret the schema rules.
\$id	A unique identifier for the schema file. Think of it as the schema's reference number or unique address.	It helps identify the schema version being used.
title	The name of the schema. This provides a human-readable description of the reporting package and helps users identify the correct schema.	It helps confirm the correct return is being submitted.
type	Defines the overall structure of the submission. HRRF submissions are always submitted as a collection of datapoints and values	It defines the root structure of the submission. All submissions are represented as JSON object, i.e. type : object.

	contained within a single JSON object.	
properties	The most important section of the schema. It contains the complete list of datapoints that may be reported, together with their descriptions, data types, and validation rules.	It defines the questions in the form.
required	Identifies the datapoints that must always be completed. If any of these datapoints are missing, the submission will fail validation.	It defines questions that must always be answered.
additionalProperties	Controls whether additional, undefined datapoints can be included. In most HRRF schemas this is set to false, which means only datapoints defined in the schema may be submitted.	It defines whether a user is allowed to submit additional unsupported questions or not.
requiredIfConditionReference	Contains dependency rules that make additional datapoints mandatory when certain conditions are met. This is an HRRF-specific feature and is used to implement conditional reporting requirements.	It defines whether additional questions become mandatory.

Together, these components provide everything needed to understand what information should be reported and how it should be submitted.

2.2 What Does a Datapoint Property Look Like?

Every field that can be reported in an HRRF submission is represented by a datapoint. Datapoints are stored within the schema's properties section and contain both business information and validation rules.

Each datapoint has a unique identifier that allows the system to recognise the information being reported. In addition to the identifier, the datapoint definition includes a title, description, expected data type, and other metadata that helps users understand where the field belongs and how it should be completed.

```
"FINS_PR_2641_v1": {
  "title": "Total Amount",
  "description": "Enter the total reported amount.",
```

```

    "type": "number",
    "propertyName": "FINS_PR_2641_v1",
    "section": "Prudential Return",
    "sequence": 1,
    "table_header": "Own funds requirement",
    "row_value": "Total exposure amount",
    "examples": [
      1000
    ]
  }

```

The example above represents a datapoint that collects the email address of a contact person. The schema not only identified the datapoint but also explains what should be entered and how the value should be validated.

Table 2.2. Datapoint components.

Datapoint Component	Description	Example Explained
FINS_PR_2641_v1	The datapoint identifier. This is the exact field name used in the JSON submission and is often referenced in reporting templates.	FINS_PR_2641_v1 represents the unique identifier for the datapoint.
title	The question or label that should be shown to users completing the submission.	"Total Amount" is shown to the user to show what is the expected input.
description	Additional guidance explaining what information should be provided.	Explains that the user is expected to enter the total reported amount.
type	The kind of value expected, such as text, number, whole number, yes/no value, or a list of values.	The value must be of type number.
format	Additional validation applied to the value. For example, percentage.	N/A
propertyName	A repeat of the datapoint identifier to make the schema easier to read and understand.	FINS_PR_2641_v1 represents the unique identifier for the datapoint.
section	The reporting section or questionnaire area to which the field belongs.	The field belongs to the Contact Details section.
sequence	The recommended order in which the field should appear within the template.	This datapoint appears in the 1 st position relative to the entire submission.
table_header	The table heading that provides additional context for the field.	Explains that the datapoint belongs under a table heading

		called "Own funds requirement".
row_value	The row label used when reconstructing table or matrix-style reporting structures.	This explains that the datapoint belongs to row with label "Total exposure amount".
examples	This provides an example of a valid value.	1000 would be a valid value for this datapoint.

Together, these properties provide everything needed to understand what information is being requested and how it should be reported.

2.2.1 Understanding Datapoint IDs

Every datapoint in an HRRF schema is assigned a unique identifier, commonly referred to as a datapoint ID. This identifier is the primary reference used to match information across the schema, Excel templates, guidance documents, validation rules, and JSON submissions.

A datapoint ID remains consistent throughout the reporting process, making it the most reliable way to identify a field regardless of where it appears.

The identifier is made up of several components, each providing information about the datapoint. A typical datapoint identifier follows the following structured naming convention:

```
FINS_PR_2641_v1
|   |   |   |
|   |   |   +-- version suffix
|   |   +----- datapoint/base sequence
|   +----- module abbreviation
+----- licence / reporting prefix
```

In this example, the datapoint belongs to the FINS reporting framework, it forms part of the Prudential (PR) module, its unique question number is 2641 and the current version of the datapoint is version 1 (v1).

Table 2.3. Datapoint ID components.

Datapoint ID Component	Example	Description
Licence / Reporting prefix	FINS	Identifies the applicable licence type, reporting framework, return family.
Module abbreviation	PR	Identifies the reporting module or section to which the datapoint belongs.

Datapoint/ Base sequence	2641	Uniquely identifies the datapoint within the module.
Version suffix	v1	Indicates the version of the datapoint currently in use.

Datapoint IDs are used throughout the reporting lifecycle to ensure that information can be matched accurately between different artefacts. Although the title and description of a datapoint may change over time, the datapoint ID remains the primary method of identifying the field.

Note: Always use the full datapoint ID, including the version suffix, when matching, reviewing, or reporting information. The datapoint ID is the single most important identifier within the HRRF reporting framework.

2.3 Understanding Common Value Types

Every datapoint in an HRRF schema is assigned a type, and in some cases a more specific format. The type determines what kind of value can be reported, while the format provides additional validation rules for how that value should be structured.

Understanding these value types helps users determine what information is expected and how it should be entered.

Table 2.4. Supported data types.

Type	Definition	Example
string	Text value	"Yes"
number	Integer or decimal number	123.45
integer	Whole number only	42
boolean	true or false	true
array	Ordered list of values	["A", "B"]

Table 2.5. Supported value formats.

Format	Applicable Data Type	Description	Example
date	string	ISO date	"2026-07-01"
date-time	string	ISO timestamp	"2026-07-01T22:00:00Z"
time	string	Time value	"22:00:00"
email	string	Email address	"mfsa@mfsa.mt"
uri	string	Web address or URI	"https://www.mfsa.mt"
currency	number	A monetary value expressed in a specific currency.	25000, 150000.25

percentage	number, integer	A value expressed as a percentage.	100, 25.5
------------	-----------------	------------------------------------	-----------

Note: When reviewing a datapoint, the type explains what kind of information is expected while the format provides additional rules about how that information should be entered. Together, it is ensured that the submissions are consistent, accurate and easy to validate.

2.4 Allowed Values and Dropdown-Style Questions

Some datapoints are restricted to a predefined list of allowed answers. In JSON Schema, this restriction is represented using an enum (short for enumeration). An enumeration defines all the values that are acceptable for a particular field.

In practical terms, an enum is equivalent to a dropdown list in an Excel template or online form. Rather than entering a free-text response, the user must select one of the available options.

Restricting a field to a predefined list helps ensure consistent reporting across all submissions, fewer data entry errors, easier analysis and comparison of reported information and clear and unambiguous answers.

Note: Always use the values exactly as they appear in the schema or template. Avoid changing spelling, spacing, punctuation, or casing unless specifically instructed to do so.

2.5 Single-Value Fields vs Multi-Value Fields

Most datapoints expect a single answer as a response; however, some datapoints allow multiple answers. In HRRF schemas, these multi-select fields are typically represented as arrays.

Table 2.6. Expected field responses.

Field Style	Schema Clue	Description
Single value	"type": "string", "type": "number", "type": "integer", "type": "boolean"	Provide one value only.
Multi-value/Array	"type": "array", "items": {...}	Provide a list of values for the datapoint.
Limited array	"maxItems": number	Provide no more than the allowed number of values.

Unique array	"uniqueItems": true	Each value in the list must be listed once.
--------------	---------------------	---

Note: Single value fields require only one answer, while arrays allow multiple values. The schema clearly indicates which approach should be used and may also restrict how many values can be provided and whether duplicate values are allowed.

2.6 Required Fields and Conditional Fields

Not all datapoints follow the same reporting rules. Some datapoints must always be submitted while others only become relevant in specific circumstances. In HRRF schemas, there are two main kinds of mandatory fields:

- Always Required Fields: Fields that must be completed in every submission.
- Conditionally Required Fields: Fields that become mandatory only when certain conditions are met.

Understanding the distinction is important because a field that appears optional may become mandatory depending on other values provided in the submission.

2.6.1 Always Required Fields

These fields are defined in the schema's required field. These fields must be included in every submission regardless of any other values reported.

```
"required": "FINS_PR_2641_v1"
```

The example above implies that each submission must include question "FINS_PR_2641_v1" submitted.

2.6.2 Conditionally Required Fields

These fields are controlled by dependency rules. These fields only become required when a predefined condition evaluates to true. The condition may depend on a specific answer being selected, a numeric threshold being reached, a value being present or a combination of multiple conditions. These are follow-up questions that only apply in certain situations.

2.6.2.1 Simple Conditional Requirements

```
[{[CCCSP_GV_251_v1];
  (${CCCSP_GI_293_v1} = 'Legal Person')
}]
```

In the example above, CCSP_GV_251_v1 becomes required when CCSP_GI_293_v1 is equal to "Legal Person". If any other value than "Legal Person" is submitted, then CCSP_GV_251_v1 is not mandatory.

2.6.2.2 Multiple Fields Triggered by One Condition

```
[{[FINS_PR_2641_v1, FINS_PR_2648_v1];
  {$FINS_GI_11_v1 = 'Yes'}
}]
```

In the example above, if FINS_GI_11_v1 is equal to "Yes" then both FINS_PR_2641_v1 and FINS_PR_2648_v1 become required. Alternatively, if any value other than "Yes" is submitted then FINS_PR_2641_v1 and FINS_PR_2648_v1 are not mandatory.

Note: Datapoints listed in the required section must always be submitted. Datapoints references in the dependency rules become mandatory if the specific condition is satisfied. Together, these define the complete set of datapoints that may need to be reported.

2.6.3 Common Conditional Symbols

Dependency rules use condition symbols to determine whether a particular business rule is true or false. These symbols allow the schema to compare answers, evaluate thresholds, check whether information has been provided, and decide when additional fields become required.

The table below explains the most used symbols and how they should be interpreted.

Table 2.7. Most used conditionally symbols.

Symbol	Description	Example
=	Equal to	#{A} = 'YES'
<>	Not equal to	#{B} <> 'NO'
>	Greater than	#{A} > 10
<	Less than	#{B} < 10
>=	Greater than or equal to	#{A} >= 10
<=	Less than or equal to	#{B} <= 10
AND	Both conditions must be satisfied.	A = YES AND B > 0
OR	At least one condition must be satisfied.	A = YES OR B > 0
NOT	The reverse condition must be satisfied.	NOT A = YES
IN (...)	Value is one of a list	A IN ("Other", "Both")

NOT IN (...)	Value is not one of a list	A NOT IN ("Other", "Both")
IS NULL	Value is absent or null	A IS NULL
IS NOT NULL	Value is present and non-null.	B IS NOT NULL

Note: Condition symbols allow the schema to make decisions based on the answers provided. They are used to determine when additional information becomes required and help ensure that reporting requirements adapt to the circumstances of each submission.

2.6.4 Understanding SUM Conditions

Some dependency rules are based on the combined value of multiple fields rather than the value of a single field. For example, a reporting requirement may only become relevant when the total of several amounts exceeds a particular threshold. To make these calculations easier for the validation engine to evaluate, SUM conditions are represented as standard addition expressions within the schema, i.e. these appear in the schema as a calculation using the + symbol.

Table 2.8. Examples of SUM condition.

Example	Expression
The total of A, B, and C is greater than zero.	$\{A\} + \{B\} + \{C\} > 0$
The total of A and B is greater than the total of C and D.	$\{A\} + \{B\} > (\{C\} + \{D\})$

Note: The + symbol within a dependency condition implies that the schema is calculating a total across multiple fields before evaluating the business rule. The condition becomes true when the resulting calculation satisfied the specified comparison.

3 Connecting the Schema to the Excel Template

Although the schema and the Excel template look quite different, they describe the same reporting requirements and use the same underlying datapoint identifiers. This allows users to move easily between the schema, the data collection template, and the accompanying guidance material.

The datapoint ID acts as the common link between all artefacts. By matching the datapoint ID, users can identify the corresponding field, understand its purpose, review any validation requirements, and locate additional guidance.

The table below provides a guide on how to navigate between the schema and Excel template.

Table 3.1. Mapping of schema properties to Excel template.

Schema	Excel Sheet	Excel Column	Guide
propertyName	2-[Module]	root_id	Use this as the link between the two.
title	2-[Module]	label	Shows the user-friendly question name.
description	3-Guidance	description	Explains what to provide.
section	2-[Module]	description	Groups related questions.
enum	2-[Module]	Dropdown in Value columns	Provides allowed answers.
required	3-Guidance	required	Specifies whether the field is mandatory.

4 Building a JSON Submission from the Schema

The schema contains all the information needed to create a valid JSON submission. It identifies the datapoints that can be reported, defines the values that are allowed, specifies which fields are mandatory, and describes when additional fields become required.

The process below provides a simple approach for constructing a submission using the schema alone.

1. Choose the correct schema.
 - a. Select the schema that matches the relevant licence, submission version, and reporting module.
 - b. If preparing a complete return, use the ALL Schema, as it contains all applicable datapoints and dependency rules.
2. Identify available datapoints.
 - a. Review the schema's properties section.
 - b. Each entry in the properties section represents a datapoint that may be included in the submission.
 - c. The datapoint definition explains what information is expected and how it should be validated.
3. Use the datapoint ID as the field name.
 - a. The datapoint identifier becomes the JSON field name.
 - b. Always use the identifier exactly as it appears in the schema.
4. Provide valid values.
 - a. Ensure that each value matches the expected:
 - i. Type (e.g. string, number, boolean)
 - ii. Format (e.g. email, uri, date)
 - iii. Allowed values (enum)
 - iv. Constraints (e.g. minimum, maximum, pattern)
5. Complete all mandatory fields.
 - a. Review the schema's required section.
 - b. Every datapoint listed in required must be included in the submission.
6. Review conditional requirements.
 - a. Check any dependency rules defined in `requiredIfConditionReference`.
 - b. Additional datapoints may become required depending on the answers provided elsewhere in the submission.
7. Do not add undefined fields.
 - a. If the schema contains: `"additionalProperties": false`, then only datapoints defined in the schema may be included.
 - b. Extra fields will cause validation to fail.
8. Validate the completed submission.
 - a. Before submission, verify that:
 - i. All mandatory fields are present.

- ii. All dependency rules have been satisfied.
- iii. All values comply with the schema definitions.
- iv. No undefined datapoints have been included.

5 Common Mistakes to Avoid

Most submission and validation issues arise from minor mistakes such as using the wrong field name, entering an invalid value, overlooking a dependency rule, or misunderstanding what a field expects.

Understanding these common errors can significantly reduce validation failures and make the submission process smoother.

Table 5.1. Common errors and their resolutions.

Common issue	Why it matters	How to avoid it
Using the label instead of the datapoint ID (root_id)	Labels may change over time, but datapoint IDs remain fixed.	Always use the full datapoint ID as the JSON field name.
Modifying a dropdown value	Allowed (enumeration) values must match exactly.	Select values from the dropdown or copy them exactly as shown.
Entering an invalid website address	URI fields require a complete address.	Use https://www.company.com or http://www.company.com instead of www.company.com .
Entering multiple values into a single field	Arrays expect separate values rather than combined values.	Provide a list of values rather than one combined value.
Ignoring conditional rules	Additional fields may become mandatory based on previous responses.	Review all dependency rules before finalising the submission.
Providing the wrong data type	The schema expects specific types of values.	Ensure that the value entered satisfies the data type criteria.
Treating example values as schema values	Schemas contain example values to help users understand what kind of information is required.	Use example values as a guide. Provide the relevant value instead.
Assuming example values are mandatory	An example illustrates the expected value type or format should a value be submitted.	Submit a value only if datapoint is required, directly or through a condition rule.
Confusing empty values with missing values	A blank field "" is different from an omitted field.	If a value is not available/applicable, omit field rather than submitting a blank value.
Adding extra fields that do not exist in the schema	Undefined fields are typically rejected	Only use datapoints defined in the schema.

Glossary

This glossary explains common terms used throughout the HRRF Schema User Guide. Definitions are intentionally written in plain language to support business users, reviewers, template owners, and other non-technical stakeholders.

Term	Definition
ALL Schema	A schema that contains all reporting modules and datapoints for a particular reporting package. It is typically used when preparing or validating a complete submission.
Allowed Values	A predefined list of values that can be selected or submitted for a field. These are usually defined using the enum field.
Array	A field that allows multiple values to be reported for the same datapoint like a multi-select question.
Boolean	A field that can only contain true or false. Often used for Yes/No style questions.
Conditional Field	A field that only becomes required when a specified condition is met.
Datapoint	A single piece of information collected as part of a submission. Each datapoint is represented by a unique datapoint ID.
Datapoint ID	The unique identifier assigned to a datapoint. It is used across schemas, templates, guidance documents, and submissions.
Dependency Rule	A rule that determines when a datapoint becomes required based on the value of another datapoint.
Description	Additional guidance explaining what information should be provided for a datapoint.
Draft-07	The version of the JSON Schema standard used by HRRF schemas.
Enum (Enumeration)	A fixed list of allowed values for a datapoint like a dropdown list in Excel.
Excel Template	The workbook used to collect and structure reporting information before submission.
Format	Additional validation applied to a field. Examples include email, date, and uri.
JSON	JavaScript Object Notation, the data format used to represent HRRF submissions.
JSON Schema	A machine-readable rulebook that defines what information can be submitted, what values are allowed, and which fields are required.

Label	The user-friendly name displayed for a datapoint in a template or questionnaire.
Module	A logical grouping of related datapoints within a return, such as General Information, Prudential, or Governance.
Property	A datapoint definition within a schema. Properties define the field, its type, and its validation rules.
Property Name	The datapoint ID used as the field name within the JSON submission.
Required Field	A field that must always be completed regardless of other answers.
requiredIfConditionRefernce	The HRRF dependency mechanism used to define conditionally required fields.
root_id	The datapoint identifier used in Excel templates. It corresponds directly to the datapoint ID in the schema.
Schema	The published specification that defines how a submission should be structured and validated.
Section	A grouping used to organise related datapoints within a reporting return.
Sequence	The intended ordering of datapoints within a section. Used to reconstruct questionnaires and templates.
String	A text value such as a name, address, identifier, or comment.
Submission	The completed JSON payload containing reported datapoint values.
Title	The short descriptive name of a datapoint shown to users.
Type	The kind of value expected for a field, such as string, number, integer, boolean, or array.
URI	A complete website or internet address, typically beginning with https:// or http://.
Validation	The process of checking whether a submission follows the rules defined in the schema.
Validation Rule	A rule that determines whether a submitted value is acceptable.
Version Suffix	The final part of a datapoint ID, such as v1, indicating the version of the datapoint.