

EEP Integration Guide

Integration Specification

August 2026

Contents

1	EEP Integration Guide	3
1.1	Revision History	3
1.2	Introduction	3
1.2.1	Purpose	3
1.2.2	Scope	3
1.2.3	Related Documents	3
2	System Overview	3
2.1	Calculation Document	4
2.2	Business Key	4
2.3	Calculation Key	4
2.4	Calculation Lifecycle	5
3	Initial Request	5
3.1	Purpose	5
3.2	Processing	5
3.3	Multiple Requests	5
4	Optional Secondary Request	6
4.1	Purpose	6
4.2	Processing	6
4.3	Multiple Requests	6
5	Calculation Synchronization	6
5.1	Request	6
5.2	Sending the Synchronization Request	6
5.2.1	Windows	7
5.2.2	macOS / Linux	7
5.3	Response	7
5.4	Recommended Client Behaviour	7
6	Direct Calculation Recall	7
6.1	Direct Link	8
7	EEP Endpoints	8
8	Sending an EEP JSON File	8
8.1	Windows	8
8.2	macOS / Linux	9
8.3	Response	9
8.4	Test Mode	9
9	Server Version Reporting	9
10	Integration Recommendations	9

Field	Value
Version	1.2
Status	Official Release
Publication Date	August 2026

1 EEP Integration Guide

1.1 Revision History

Version	Date	Description
1.2	2026	Clarified Calculation Document uniqueness, introduced the Business Key definition, and documented replacement of an existing calculation by a new Initial Request with the same Business Key. This version is applicable to EEP Specification Version 1.2.
1.1	2026	Added synchronization workflow, direct calculation recall, server version reporting and integration recommendations. This version is also applicable to EEP Specification Version 1.2.
1.0	2026	Initial public release.

1.2 Introduction

1.2.1 Purpose

This guide explains how to integrate a Timing System with the EET Calculator using the Electronic Equivalent Time Exchange Protocol (EEP).

1.2.2 Scope

This guide describes:

- EET Calculator architecture;
- Calculation Document;
- Initial Request;
- Optional Secondary Request;
- EEP Endpoints;
- Integration Recommendations.
- Calculation synchronization;
- Direct calculation recall;
- Server version reporting.

Request and response bodies are defined in the EEP Specification.

1.2.3 Related Documents

- EEP Specification
- EET Calculator Administrator Guide

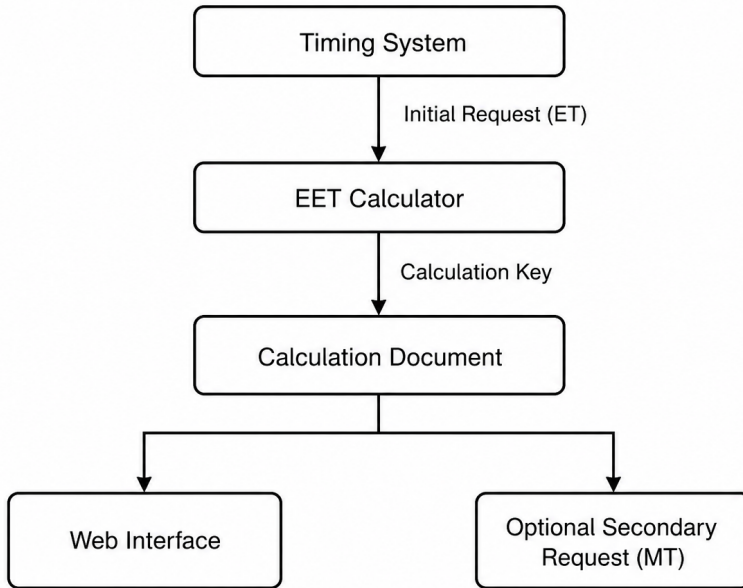
2 System Overview

An Initial Request creates a Calculation Document from the supplied Electronic Times (ET) and returns a unique Calculation Key.

Optional Secondary Requests enrich the same Calculation Document by providing Manual Times (MT) using its Calculation Key.

The Calculation Document is then available in the web interface using its Calculation Key.

System Overview



2.1 Calculation Document

The Calculation Document is the persistent representation of an EET calculation.

Its structure and lifecycle are defined in the EEP Specification.

A Calculation Document:

- is created by an Initial Request;
- may be progressively enriched by Optional Secondary Requests;
- is then processed through the EET Calculator web interface.

Each Calculation Document is uniquely identified by a Calculation Key throughout its lifetime.

At any time, only one Calculation Document exists on the server for a given Business Key.

2.2 Business Key

The Business Key identifies the EET calculation independently of the Calculation Key assigned by the server.

It is composed of the following five race members, all represented as strings:

`(season, codex, run, eet_bib, missing_impulse)`

Two Calculation Documents have the same Business Key when all five string values are identical.

When an Initial Request is accepted, if a Calculation Document already exists with the same Business Key, the existing document is removed and the newly created document is persisted in its place.

The new Calculation Document receives a new Calculation Key. Consequently, at most one Calculation Document persists on the server for a given Business Key.

2.3 Calculation Key

The Calculation Key (`calculation_id`) is assigned by the server when the Calculation Document is created.

It uniquely identifies the stored Calculation Document and is returned in the Initial Response.

The Calculation Key is distinct from the Business Key. It shall be used in all Optional Secondary Requests related to that Calculation Document.

If a new Initial Request replaces a Calculation Document having the same Business Key, the previous Calculation Key no longer identifies the current Calculation Document.

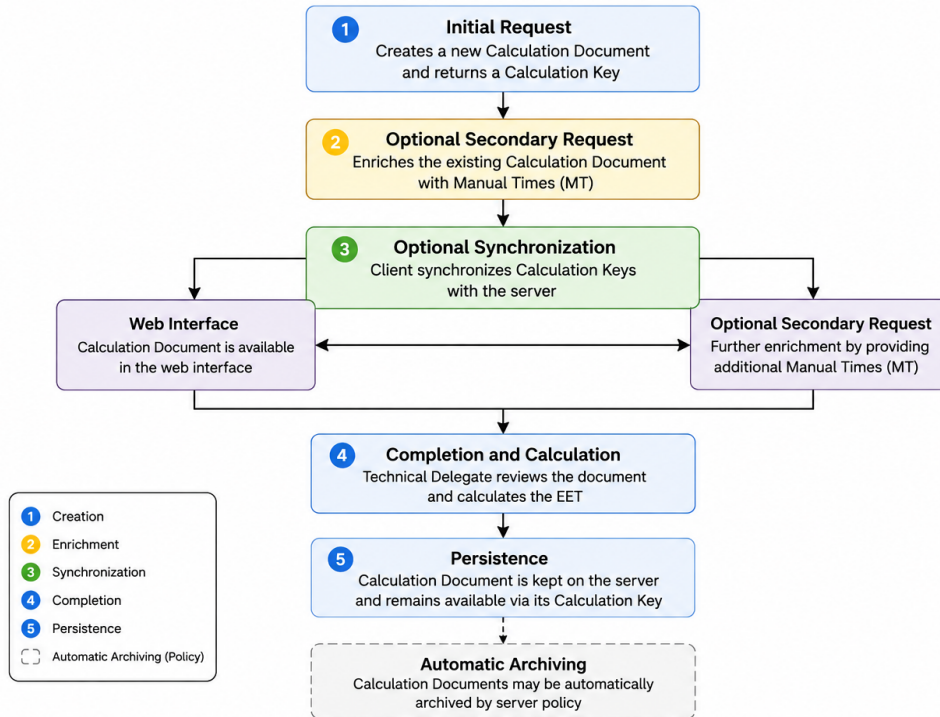
2.4 Calculation Lifecycle

1. Creation by an Initial Request.
2. Optional enrichment by one or more Secondary Requests.
3. Optional synchronization with the server.
4. Completion and calculation through the web interface.

Calculation Documents may subsequently be deleted automatically according to the server retention policy.

- 7 days for a TEST calculation
- Otherwise, calculations from season N-2 and earlier may be deleted.

Calculation Document Lifecycle



3 Initial Request

3.1 Purpose

Creates a new Calculation Document from the supplied Electronic Times (ET).

3.2 Processing

Upon successful validation:

- a new Calculation Document is created;
- a new Calculation Key is assigned;
- the Calculation Document is persisted on the server.

The Calculation Key is returned to the client application and should be stored locally together with the Calculation Document for future synchronization, direct recall and Optional Secondary Requests.

3.3 Multiple Requests

Each Initial Request is processed as a request to create the Calculation Document corresponding to its Business Key.

If no Calculation Document with the same Business Key exists, a new Calculation Document is created and persisted.

If a Calculation Document with the same Business Key already exists, the existing document is removed and replaced by the newly created document.

In both cases, the new Calculation Document receives a new Calculation Key.

For example:

Initial Request #1

```
Business Key = ("2026", "FRA1234", "1", "25", "start")
-> Calculation Key = "Ab12Cd"
```

Initial Request #2

```
Business Key = ("2026", "FRA1234", "1", "25", "start")
-> replaces the document identified by "Ab12Cd"
-> Calculation Key = "Xy34Zk"
```

After Initial Request #2 has been accepted, only the new Calculation Document persists for that Business Key.

4 Optional Secondary Request

4.1 Purpose

Enriches an existing Calculation Document by providing Manual Times (MT).

4.2 Processing

Upon successful validation:

- the supplied Manual Times are incorporated into the Calculation Document identified by its Calculation Key;
- a new version of the Calculation Document is created;
- the previous version is removed from the server;
- the new version is persisted using the same Calculation Key.

4.3 Multiple Requests

Multiple Optional Secondary Requests may be submitted for the same Calculation Key.

Each accepted request updates the Calculation Document identified by its Calculation Key.

5 Calculation Synchronization

Timing software may periodically synchronize any locally stored Calculation Keys with the EET Calculator server.

The synchronization request allows the client application to determine whether each locally stored calculation still exists on the server.

5.1 Request

POST /api/eep/synchronization

Request body:

```
{
  "calculation_ids": [
    "1fpH6G",
    "1fpZPH",
    "1fqkBF",
    "1fq1Pg"
  ]
}
```

The `calculation_ids` array contains the Calculation Keys currently stored by the client application.

5.2 Sending the Synchronization Request

The JSON document can be submitted to the synchronization endpoint using an HTTP POST request.

The request must specify the content type as:

Content-Type: application/json

5.2.1 Windows

Using `curl.exe` from Command Prompt or PowerShell:

```
curl.exe -X POST https://pg-chrono.fr/api/eep/synchronization ^  
-H "Content-Type: application/json" ^  
--data-binary "@c:/path_to_your_file.json"
```

The `@` character tells `curl` to read the JSON document from the specified file rather than sending the file name itself.

5.2.2 macOS / Linux

Using `curl` from a terminal:

```
curl -X POST https://pg-chrono.fr/api/eep/synchronization -H "Content-Type: application/json" --data-b
```

The path may of course be replaced by the actual location of the JSON file, for example:

```
--data-binary "@/home/user/eet/synchronization.json"
```

5.3 Response

Example:

```
{  
  "calculations": {  
    "1fpH6G": {  
      "exists": true  
    },  
    "1fpZPH": {  
      "exists": false  
    },  
    "1fqkBF": {  
      "exists": true  
    }  
  }  
}
```

For each submitted Calculation Key:

- `exists = true` indicates that the calculation is still available on the server.
- `exists = false` indicates that the calculation has been removed from the server.

5.4 Recommended Client Behaviour

For each synchronized Calculation Key:

- if `exists` is `true`, keep the local calculation;
- if `exists` is `false`, the client application may inform the user that the calculation no longer exists on the server and optionally delete the obsolete local file after user confirmation.

To keep the local repository consistent with the server, it is recommended to perform a synchronization immediately after a successful Initial Request has been accepted by the server. This allows newly created calculations to be registered locally while identifying obsolete calculations that may no longer exist on the server.

Synchronization helps maintain consistency between the local calculation repository and the calculations currently available on the EET Calculator server.

6 Direct Calculation Recall

A Calculation Document may be recalled directly through its Calculation Key.

This mechanism allows:

- review of an existing calculation;
- manual completion of missing Manual Times;
- PDF generation;

- further processing by the Technical Delegate.

The recalled Calculation Key remains visible in the user interface.

6.1 Direct Link

A Calculation Document may be opened directly by constructing a URL containing its Calculation Key.

Format:

`https://pg-chrono.fr/api/calculation/<CalculationKey>`

Example:

`https://pg-chrono.fr/api/calculation/1fpH6G`

Opening this URL in the user's default web browser recalls the corresponding Calculation Document.

If the Calculation Key is unknown or no longer exists on the server, the application displays an appropriate error message.

7 EEP Endpoints

Endpoint	HTTP Method	Purpose
Initial Request	POST	Creates a Calculation Document
Optional Secondary Request	POST	Enriches a Calculation Document
Calculation Synchronization	POST	Verifies Calculation Keys
Calculation Recall	GET	Opens a Calculation Document

Request and response bodies are defined in the EEP Specification.

Every successful EEP request returns the associated Calculation Key.

Error responses are defined in the EEP Specification.

Calculation Synchronization returns the existence status of each supplied Calculation Key.

Calculation Recall opens the corresponding Calculation Document in the web interface.

8 Sending an EEP JSON File

An EEP JSON document can be submitted to the EET Calculator using an HTTP POST request to the EEP endpoint:

`https://pg-chrono.fr/api/eeep`

The request must specify the content type as:

`Content-Type: application/json`

8.1 Windows

Using `curl.exe` from Command Prompt or PowerShell:

```
curl.exe -X POST https://pg-chrono.fr/api/eeep ^
-H "Content-Type: application/json" ^
--data-binary "@c:/path_to_your_file.json"
```

The @ character tells `curl` to read the JSON document from the specified file rather than sending the file name itself.

8.2 macOS / Linux

Using `curl` from a terminal:

```
curl -X POST https://pg-chrono.fr/api/eep \  
  -H "Content-Type: application/json" \  
  --data-binary "@./path_to_your_file.json"
```

The path may of course be replaced by the actual location of the JSON file, for example:

```
--data-binary "@/home/user/eet/temp_EET.json"
```

8.3 Response

If the request is successfully accepted, the server returns the EEP response containing the **Calculation Key** associated with the newly created Calculation Document. The Calculation Key should be preserved by the client application for subsequent operations.

8.4 Test Mode

During the implementation and integration testing period, the use of the **TEST** mode is strongly recommended.

In this mode, JSON files submitted to the EET Calculator are automatically deleted after the configured retention period of 7 days. This allows integration testing without unnecessarily accumulating test Calculation Documents on the server.

The **TEST** setting should therefore be enabled during development and integration testing and disabled when the Timing System is put into production.

9 Server Version Reporting

Every successful EEP response includes informational messages indicating:

- the implemented EEP protocol version;
- the implemented EET Calculator version.

These messages are intended for diagnostic purposes and allow client software to verify compatibility with the server implementation.

Client software should treat these messages as informational only.

10 Integration Recommendations

- Preserve the Calculation Key returned by a successful Initial Request.
- Use the Calculation Key only for Optional Secondary Requests referring to the same Calculation Document.
- Treat (`season`, `codex`, `run`, `eet_bib`, `missing_impulse`) as the Business Key identifying the calculation independently of its Calculation Key.
- If a new Initial Request is submitted with the same Business Key, replace the locally stored Calculation Key with the new Calculation Key returned by the server.
- If the Electronic Times of the Initial Request are incorrect and a new calculation must be started, submit a new Initial Request without a Calculation Key.
- Periodically synchronize locally stored Calculation Keys.
- Use server version information to detect implementation differences.
- Ignore informational server messages not required by the application.