

全国道路施設データベース トンネル
データ登録用 API 仕様書

Ver 1.32

令和6年9月4日

一般社団法人 日本建設機械施工協会
施工技術総合研究所

目次

目次.....	- 2 -
1. 本仕様書の概要.....	- 2 -
2. API 接続方式.....	- 4 -
2.1. 通信プロトコル	- 4 -
2.2. REST 対応.....	- 4 -
2.3. エンドポイント一覧（バージョン1）	- 4 -
2.4. 文字コード.....	- 4 -
2.5. リクエスト.....	- 4 -
2.5.1. データ登録 API	- 4 -
2.5.2. 処理状況監視 API	- 4 -
2.5.3. 付属ファイル登録 API	- 4 -
2.6. レスポンス.....	- 5 -
2.6.1. データ形式.....	- 5 -
2.7. エラーレスポンス.....	- 7 -
2.8. データ登録 API の処理フロー	- 8 -
2.9. API のパラメータ	- 9 -
2.9.1. トンネルデータ登録 API.....	- 9 -
2.9.2. 処理状況監視 API	- 19 -
2.9.3. 付属ファイルの登録 API.....	- 20 -
2.9.4. 付属ファイルの削除 API.....	- 22 -
2.9.5. 点検調書ファイルの登録 API	- 23 -
2.9.6. イメージファイルの登録 API	- 24 -
2.9.7. 日付表記.....	- 25 -
2.9.8. API 認証	- 25 -
2.10. 登録 Json 内容.....	- 25 -

1. 本仕様書の概要

本仕様書は、「全国道路施設データベース トンネル」のオンラインで提供する諸元や点検データ等をデータ登録するための API に関する仕様書です。

更新履歴

Ver1.00		新規作成
Ver1.10	2023/09/07	点検調書のファイル登録 API 追加
Ver1.20	2023/09/19	イメージファイル登録 API 追加
Ver1.30	2023/12/19	付属ファイル削除 API の記述を追加
Ver1.31	2024/01/12	URL の誤記を修正
Ver1.32	2024/09/04	トンネル図面、点検図面、点検写真の登録についてを追記

2. API 接続方式

2.1. 通信プロトコル

セキュアな HTTP プロトコルである HTTPS によって通信を行う。API を設置するサーバーは TLS(SSL)対応を必須とする。

2.2. REST 対応

本 API のデータ参照様式は、REST（記述方式）に則った仕様を基本とする。

2.3. エンドポイント一覧（バージョン1）

〇〇/xROAD/api/v1/tunnels/import	トンネルデータ登録更新削除
〇〇/xROAD/api/v1/tunnels/import/status/{処理 ID}	登録処理中の監視
〇〇/xROAD/api/v1/tunnels/upload/{施設 ID}/{年度}	付属ファイルをアップロード
〇〇/xROAD/api/v1/tunnels/otherFileDelete/{ファイル ID}	付属ファイルを削除
〇〇/xROAD/api/v1/tunnels/uploadreport77/{施設 ID}/{年度}	点検調書をアップロード
〇〇/xROAD/api/v1/tunnels/uploadimagefile/{ファイル ID}	イメージファイルの登録

※〇〇は「[https:// road-structures-db-tunnel.mlit.go.jp](https://road-structures-db-tunnel.mlit.go.jp)」

2.4. 文字コード

本 API を利用した通信上で、日本語を扱う場合は、文字コードは UTF-8 とする。

2.5. リクエスト

2.5.1. データ登録 API

REST の仕様に則って、POST メソッドを用いる。

登録する施設データは JSON 形式のファイルを用いる。

アップロードする施設データファイルの仕様

- ・ファイル形式は、JSON 形式とする。(RFC8259)
- ・文字コードは UTF-8 とする。
- ・項目名の省略、独自の項目名の利用は禁止とする。
- ・特殊記号、機種依存文字の利用は禁止とする。
- ・施設 ID+点検年度で一意のデータとする。
- ・施設アップロードファイルの項目名については、別紙「トンネル API パラメータ名一覧」を参照。

2.5.2. 処理状況監視 API

REST の仕様に則って、GET メソッドを用いる。

2.5.3. 付属ファイル登録 API

REST の仕様に則って、POST メソッドを用いる。

2.6. レスポンス

2.6.1. データ形式

ファイル以外は、JSON 形式の文字列として以下の形式で返す。

metadata

API の内容やリクエスト情報など

resultset

検索結果の件数や 1 ページのデータ数など

result

参照に成功した場合は参照したデータを、失敗した場合はエラーの内容を含んだ処理結果を返す。

No.	区分	パラメータ名	内容
1	metadata	title	データの内容を示すタイトル
2		detail	データの内容説明
3		parameter	リクエスト時に渡されたパラメータ ファイルバイナリは除く
4	resultset	is_error	処理にエラーがある場合は true。ない場合は false
5		error_title	エラーのタイトル。is_error が true の場合は必須
6		error_detail	エラーの詳細
7		count	対象データの件数
8		limit	一度に参照するデータ数の上限
9		offset	データ参照開始位置
10	result	—	結果データ

レスポンス例)

```
{
  "metadata": {
    "title": "施設データの登録",
    "detail": "施設データの登録",
    "parameter": {
      "type": 1
    }
  },
  "resultset": {
    "is_error": false
  },
  "result": {
    (処理結果)
  }
}
```

レスポンス例 エラー

```
{
  "metadata": {
    "title": "施設データの登録",
    "detail": "施設データの登録",
    "parameter": {
      "type": 1
    }
  },
  "resultset": {
    "is_error": true,
    "error_title": "file が指定されていません"
    "error_detail": "file が指定されていません"
  }
}
```

2.7. エラーレスポンス

処理中に致命的なエラーが発生した場合、適切な HTTP レスポンスコードとエラー内容を JSON 形式で返す。

No.	レスポンスコード	内容
1	400	リクエスト内容に問題有り
2	401	認証エラー
3	404	存在しない API にアクセスされた
4	429	一定時間内のアクセス数超過 ※現在は制限していないが、将来的に制限する可能性があることに留意する必要がある。
5	500	サーバ内部エラー
6	503	メンテナンス中

レスポンスボディ(本文)に JSON 形式でエラーな内容を記述する。

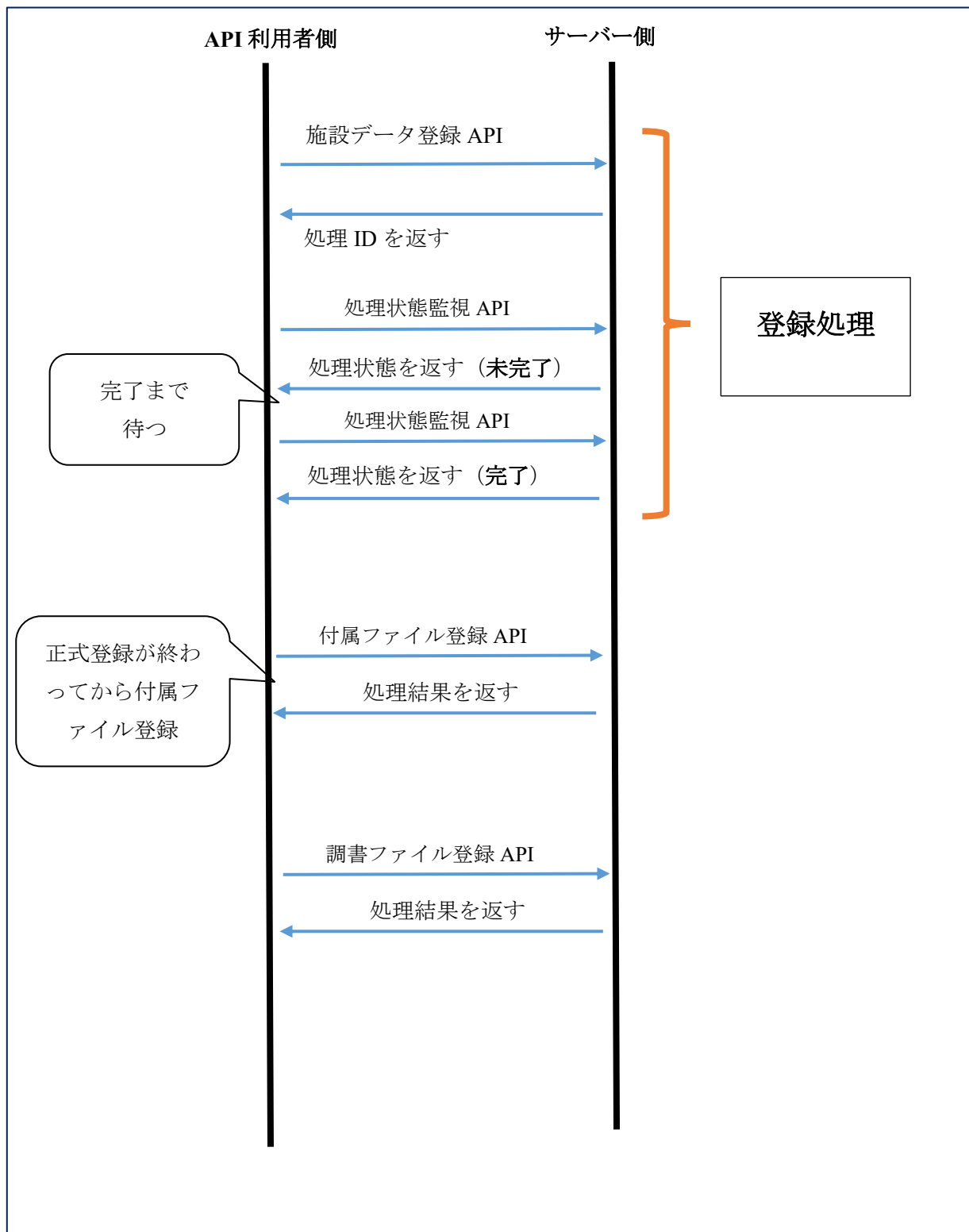
No.	パラメータ名	内容
1	code	HTTP レスポンスコード
2	message	エラーの詳細な内容

レスポンス例

```
{
  "code": 400,
  "message": "検索範囲が日本国内を超えています"
}
```

2.8. データ登録 API の処理フロー

施設データの登録処理における、処理フローを以下に示す。



2.9. APIのパラメータ

2.9.1. トンネルデータ登録 API

URL : ○○/xROAD/api/v1/tunnels/import

メソッド: POST

指定した施設データファイル (*.json) をアップロードし、取込処理を開始する。

取込処理開始後は、処理完了を待たずにレスポンスとして処理 ID を返す。

データの送信には RFC7578 に則って「multipart/form-data」形式を利用する。

No.	パラメータ	内容	省略時
1	File	施設データファイル (*.json)	省略不可
2	Type	処理タイプ (1:追加更新json 2:削除json)	省略不可

レスポンス

No.	パラメータ	内容	例
1	status	処理状況(0: 未実行 1:実行中 2:完了 3:エラー)	2
2	message	メッセージ	登録が完了しました
3	processid	処理ID	654321

使用例

トンネルデータの登録

POST https://○○○○/xROAD/api/v1/tunnels/import

HTTP ヘッダ

Content-Type: multipart/form-data; boundary=----BoundaryO5quBRiT4G7Vm3R7

リクエストボディ

-----BoundaryO5quBRiT4G7Vm3R7

Content-Disposition: form-data; name="type"

1

-----BoundaryO5quBRiT4G7Vm3R7

Content-Disposition: form-data; name="file"; filename="ファイル名"

Content-Type: text/plain

(ファイルの内容)

-----BoundaryO5quBRiT4G7Vm3R7--

トンネル図面、点検図面、点検写真の登録について

図面および写真ファイルをアップロードするためには、事前にトンネルデータ登録 API を使用してファイル名を登録する必要があります。

ファイル名を登録すると、file_id が割り当てられるため、その file_id を取得してから「**イメージファイルの登録**」API を使用し別途、図面や写真ファイルをアップロードします。

トンネルデータ登録 API は、追加、更新、削除の操作が可能です。

以下に具体的な条件を示します。

図面

追加	file_id が未指定の場合、新規追加となります。
更新	既存の file_id を指定すると更新されます。 存在しない file_id を指定した場合、エラーとなります。
全件削除	図面要素に空配列を指定すると、該当する図面タイプのデータがすべて削除されます。null を指定した場合は無視されます。
部分削除	登録済みの file_id を JSON 要素に含めないと、そのファイルは削除されます。

写真

追加	file_name を指定し、かつ file_id が未指定の場合、新規追加となります。
更新	既存の file_id を指定すると更新されます。存在しない file_id を指定した場合、エラーとなります。
削除	登録済みの file_id を JSON 要素に含めないと、そのファイルは削除されません。

図面 file_id のパラメータ

- zumen[n].file_id
- tenkenichizu[n].file_id
- tenkentenkaizu[n].file_id
- tenkenkikitenkaizu[n].file_id
- tenkenspantenkaizu[n].file_id

写真 file_id のパラメータ

- tenkenhontai[n].file_id
- tenkenfuzoku[n].file_id

図面（追加）例

file_id が未指定の場合、新規追加となります。

使用例

file_id が空文字、null、要素が存在しない場合は新規追加です。

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    ... (省略)

    "tenkenichizu": [
      {
        "file_id": "",
        "file_name": "file_name1"
      },
      {
        "file_id": null,
        "file_name": "file_name2"
      },
      {
        "file_name": "file_name2"
      }
    ]
  }
]
```

図面（更新）例

既存の file_id を指定すると更新されます。

存在しない file_id を指定した場合、エラーとなります。

使用例

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    ... (省略)

    "tenkenichizu": [
      {
        "file_id": "TZ_0001",
        "file_name": "file_name1(EDIT)"
      },
      {
        "file_id": "TZ_0002",
        "file_name": "file_name2(EDIT)"
      }
    ]
  }
]
```

図面（全件削除）例

図面要素に空配列を指定すると、該当する図面タイプのデータがすべて削除されます。
null を指定した場合は無視されます。

使用例

以下の指定は点検に紐づく **tenkenichizu** をすべて削除します。

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    ... (省略)

    "tenkenichizu": []
  }
]
```

以下の指定は処理しません。（無視されます）

▼パターン1 : null

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    "tenkenichizu": null
  }
]
```

▼パターン2 : 要素が存在しない

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
  }
]
```

図面（部分削除）例

登録済みの file_id を JSON 要素に含めないと、そのファイルは削除されます。

使用例

file_id が TZ_0001, TZ_0002, TZ_0003 が存在しているとき
TZ_0002 を削除したい場合は以下のようなになる。

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    ... (省略)

    "tenkenichizu": [
      {
        "file_id": "TZ_0001
        "file_name": "file_name1(EDIT)"
      },
      {
        "file_id": " TZ_0003",
        "file_name": "file_name2(EDIT)"
      }
    ]
  }
]
```

図面（追加、更新、削除が混在）例

登録済みの file_id を JSON 要素に含めないと、そのファイルは削除されます。

使用例

file_id が TZ_0001, TZ_0002, TZ_0003 が存在している。

このとき TZ_0001 を更新、TZ_0002 を削除、

TZ_0004 を追加したい場合は以下ようになる。

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    ... (省略)

    "tenkenichizu": [
      {
        "file_id": "TZ_0001
        "file_name": "file_name"
      },
      {
        "file_id": " TZ_0004",
        "file_name": "file_name"
      }
    ]
  }
]
```

写真（追加）例

file_name を指定し、かつ file_id が未指定の場合、新規追加となります。

使用例

file_id が空文字、null、要素が存在しない場合は新規追加です。

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    ... (省略)
    "tenkenhontai": [
      {
        "span_number": "txt439",
        "deformation_number": "440",
        ... (省略)
        "file_id": "",
        "file_name": "file_name13"
      },
      {
        "span_number": "txt439",
        "deformation_number": "440",
        ... (省略)
        "file_id": null,
        "file_name": "file_name13"
      },
      {
        "span_number": "txt439",
        "deformation_number": "440",
        ... (省略)
        "file_name": "file_name13"
      }
    ],
    ... (省略)
  }
]
```


写真（更新）例

既存の file_id を指定すると更新されます。

存在しない file_id を指定した場合、エラーとなります。

使用例

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    ... (省略)
    "tenkenhontai": [
      {
        "span_number": "txt439",
        "deformation_number": "440",
        ... (省略)
        "file_id": "TP_0001",
        "file_name": "file_name13"
      }
    ],
    ... (省略)
  }
]
```

写真（削除）例

登録済みの file_id を JSON 要素に含めないと、そのファイルは削除されます。

使用例

```
[
  {
    "shisetsu_id": "42.97037,141.17514",
    ... (省略)
    "tenkenhontai": [
      {
        "span_number": "txt439",
        "deformation_number": "440",
        ... (省略)
        "file_id": null,
        "file_name": "file_name13"
      },
      {
        "span_number": "txt439",
        "deformation_number": "440",
        ... (省略)
        "file_id": "",
        "file_name": "file_name13"
      },
      {
        "span_number": "txt439",
        "deformation_number": "440",
        ... (省略)
        "file_name": "file_name13"
      },
    ],
    ... (省略)
  }
]
```

2.9.2. 処理状況監視 API

指定処理 ID の施設データファイルの取込処理状況を取得する。

URL : ○○/xROAD/api/v1/tunnels/import/status/{処理 ID}

メソッド: GET

レスポンス

No.	パラメータ	内容	例
1	status	処理状況(0: 未実行 1:実行中 2:完了 3:エラー)	2
2	message	メッセージ	登録が完了しました

使用例

処理 ID 「9999」の処理状況を取得する

GET https://○○○○/xROAD/api/v1/tunnels/import/status/9999

トンネルデータの取込処理の完了待ちに使用する。

エラー内容 例

エラー内容	対応
【shisetsu_id】は必須です。	施設 ID を設定してください。
【shisetsu_id】 (***)は、文字数をオーバーしています。最大文字数:18	施設 ID は18文字で設定してください。
【管理者コード】(***)を編集する権限がありません。	***の管理者のトンネルの編集する権限がありません。管理者コードに誤りが無いか確認してください。
【shisetsu_kubun】は必須です。	施設区分に「2」を設定してください。
【shisetsu_kubun】 (***)は不正な値です。	施設区分に「2」を設定してください。
【kanrisya_code】は必須です。	管理者コードを設定してください。
【kanrisya_code】 (***)、管理者コードは見つかりませんでした。	正しい、管理者コードを設定してください。
【syogen.rosen.douro_syubetsu】 (***)、道路種別コードが見つかりませんでした。	正しい、道路種別を設定してください。
【###】 (***)は、数値に変換できませんでした。	項目###に指定された***は数値に変換できないため、値を確認してください。

エラーメッセージを元に修正し再度、登録してください。

2.9.3. 付属ファイルの登録 API

指定した施設 ID・点検年度の施設データに、付属ファイルをアップロードする。

URL : OO/xROAD/api/v1/tunnels/upload/{施設 ID}/{年度}

メソッド: POST

データの送信には RFC7578 に則った「multipart/form-data」形式を利用する。

No.	パラメータ	内容	省略時
1	file	アップロード対象ファイル	省略不可
2	total	送信するファイルの総数	1
3	count	何番目のファイルかを表す、1 から始まる数字	1
4	processid	最初の施設データ送信時に発行された処理 ID	新規処理として処理 ID を新たに発行する

レスポンス

No.	パラメータ	内容	例
1	message	メッセージ	“ファイル送信が完了しました”
2	processid	リクエスト時に受け取った処理 ID をそのまま返す。ただし、リクエスト時に処理 ID が指定されていなかった場合は、新規に処理 ID を発行されて返	654321

使用例

施設 ID 「35.2134,139.112」、年度 「2020」 の施設データに付属ファイルの登録

POST

https://○○○○/xROAD/api/v1/tunnels/upload/35.2134,139.112/2020

HTTP ヘッダ

Content-Type: multipart/form-data; boundary=----BoundaryO5quBRiT4G7Vm3R7

リクエストボディ

----- BoundaryO5quBRiT4G7Vm3R7

Content-Disposition: form-data; name="total"

10

----- BoundaryO5quBRiT4G7Vm3R7

Content-Disposition: form-data; name="count"

7

----- BoundaryO5quBRiT4G7Vm3R7

Content-Disposition: form-data; name="processed"

7654321

----- BoundaryO5quBRiT4G7Vm3R7

Content-Disposition: form-data; name="file"; filename="ファイル名"

Content-Type: text/plain

(ファイルの内容)

-----BoundaryO5quBRiT4G7Vm3R7--

2.9.4. 付属ファイルの削除 API

公開用 API の「付属ファイルリスト取得 API」でファイル ID を識別し、ファイルを削除する。

URL : ○○/xROAD/api/v1/tunnels/otherFileDelete/{fileID}

メソッド: POST

No.	パラメータ	内容	例
1	fileID	ファイルID	

レスポンス

No.	パラメータ	内容	例
		なし	

※is_error=false で、正常終了。

レスポンス例)

```
{
  "metadata": {
    "title": "施設点検データのファイル削除 API" ,
    "detail": "施設点検データをファイル削除する API です。" ,
    "parameter": {
      " fileID ": ○○○○
    }
  },
  "resultset": {
    "is_error": false
  },
  "result": {
    なし
  }
}
```

2.9.5. 点検調書ファイルの登録 API

指定した施設 ID・点検年度の施設データに、付属ファイルをアップロードする。

URL : ○○/xROAD/api/v1/tunnels/uploadreport77/{施設 ID}/{年度}

メソッド: POST

77条調査や直轄点検調書の Excel ファイルを登録します。

複数ファイルの場合は、Zip にし 1 ファイルとして登録してください。

データの送信には RFC7578 に則った「multipart/form-data」形式を利用する。

No.	パラメータ	内容	省略時
1	file	アップロード対象ファイル	省略不可

レスポンス

No.	パラメータ	内容	例
1	message	メッセージ	“ファイル送信が完了しました”
2	processid	データ送信時に発行された処理 ID	処理 ID を新たに発行する

※処理状況監視 API にて監視してください。

使用例

施設 ID 「35.2134,139.112」、年度 「2020」 の施設データに点検調書ファイルの登録

POST

https://○○○○/xROAD/api/v1/tunnels/uploadreport77/35.2134,139.112/2020

HTTP ヘッダ

Content-Type: multipart/form-data; boundary=----BoundaryO5quBRiT4G7Vm3R7

リクエストボディ

----- BoundaryO5quBRiT4G7Vm3R7

Content-Disposition: form-data; name="file"; filename="ファイル名"

Content-Type: text/plain

(ファイルの内容)

-----BoundaryO5quBRiT4G7Vm3R7--

2.9.6. イメージファイルの登録 API

指定したファイル ID の、イメージファイルをアップロードする。

URL : ○○/xROAD/api/v1/tunnels/uploadimagefile/{file_id}

メソッド: POST

事前に、「トンネルデータ登録 API」にて、図面類などの情報を保存すると自動的に file_id が採番されます。

データ取得用 API にて自動採番された file_id を取得し、本 API でイメージファイルを登録する。

データの送信には RFC7578 に則った「multipart/form-data」形式を利用する。

No.	パラメータ	内容	省略時
1	file	アップロード対象ファイル	省略不可
2	file_id	データ取得APIで取得した、ファイルID 図面類・点検写真等のファイルIDです。	省略不可

レスポンス

No.	パラメータ	内容	例
1	status	処理状況(2:完了 3:エラー)	2
2	message	メッセージ	“ファイル送信が完了しました”

使用例

ファイル ID 「ZU0398472」 のイメージファイルの登録

POST

https://○○○○/xROAD/api/v1/tunnels/uploadimagefile/ZU0398472

HTTP ヘッダ

Content-Type: multipart/form-data; boundary=----BoundaryO5quBRiT4G7Vm3R7

リクエストボディ

----- BoundaryO5quBRiT4G7Vm3R7

Content-Disposition: form-data; name="file"; filename="ファイル名"

Content-Type: text/plain

(ファイルの内容)

-----BoundaryO5quBRiT4G7Vm3R7--

2.9.7. 日付表記

- ・年データは西暦とする。フォーマットは YYYY

例) 2021

- ・年月フォーマットは YYYYMM

例) 202109

- ・年月日フォーマットは YYYYMMDD

例) 20210901

2.9.8. API 認証

API キー認証方式

API を利用するアプリごとに API キーを発行し、API アクセス時にその API キーを HTTP ヘッダに記述することによって認証する。

発行時に登録したドメイン名や IP アドレスからのアクセスは拒否する。

- ・ API キー記述ヘッダ

API-key

- ・ API キーフォーマット

半角英数字 40 文字

2.10. 登録 Json 内容

別紙パラメータ一覧を参照してください。