



CAITEL

Catasto Impianti Termici — Liguria

Specifiche Funzionali

API Accatastamento — Nuovo Impianto Termico

Versione API: 1.0.6

Versione documento: 1.0.6

Regione Liguria — Liguria Digitale

11 Settembre 2026

PM: Priano Francesca | **Autore:** Ferrando Fabrizio

Supporto: caitel-api-assistenza@liguriadigitale.it

Sommario

1. Introduzione	5
1.1 Scopo del documento	5
1.2 Panoramica funzionale	5
1.3 Ambienti	5
2. Flusso funzionale — Accatastamento nuovo impianto	6
2.1 Fase 1 — Caricamento tipologiche	6
2.2 Fase 1 — Caricamento metadati campi (opzionale ma consigliato)	7
2.3 Fase 2 — Chiamate dinamiche per dati variabili	7
2.3.1 Selezione provincia → caricamento comuni	7
2.3.2 Digitazione indirizzo — autocomplete	7
2.4 Fase 2 — Regole condizionali sezione Generatore	8
2.5 Fase 2 — Regole obbligatorietà sezione Responsabile	9
2.6 Fase 3 — Verifica preventiva dei dati	9
2.7 Fase 4 — Inserimento impianto	9
2.8 Struttura multi-generatore / multi-macchina	10
3. Dettaglio endpoint	12
3.1 POST /nuovo-impianto/verifica-dati	12
3.2 POST /nuovo-impianto/inserisci	12
3.3 GET /tipologiche/tipi-intervento	14
3.4 GET /tipologiche/tipi-generatori-per-nuovo-impianto	14
3.5 GET /tipologiche/tipi-fluido-vettore	14
3.6 GET /tipologiche/tipi-macchina-termica	14
3.7 GET /tipologiche/tipi-generatore-rendimento	14
3.8 GET /tipologiche/tipi-responsabilita	14
3.9 GET /tipologiche/tipi-colore-civico	14
3.10 GET /tipologiche/is-macchine-a-biomassa	14
3.11 GET /tipologiche/province-liguria	15
3.12 GET /tipologiche/province	15
3.13 GET /tipologiche/comuni-provincia/{identificativo-provincia}	15
3.14 GET /tipologiche/indirizzi-provincia-comune-nuovo-o-modifica-impianto/{identificativo-provincia}/{identificativo-comune}/{parte}	15
3.15 GET /proprieta/accatastamento/{sezione}	15
3.16 GET /status	15

4. Strutture dati — Request body	16
4.1 AccatastamentoRichiesta	16
4.2 ImpiantoDaAccatastare	16
4.3 GruppoGeneratore	16
4.4 MacchinaGeneratore	17
4.5 ForniturePodPdr	18
4.6 ResponsabileImpianto	18
5. Strutture dati — Response	20
5.1 Struttura comune di tutte le response	20
5.2 ImpiantoAccatastato — payload di elemento per POST /nuovo-impianto/inserisci	20
5.3 ResponseMessage — struttura del messaggio di esito	20
5.4 ResponseMessageDetails — dettaglio errori di validazione (risposta 400)	21
5.5 ProprietaResp — metadati campi (risposta GET /proprieta/accatastamento/*)	21
5.6 ListaValori / ListaValoriWithRequest — risposta tipologiche	22
5.7 BasePairString — coppia identificativo/valore	22
6. Gestione errori	23
6.1 Tabella codici HTTP	23
6.2 Struttura risposta 400 — errori di validazione	23
6.3 Struttura risposta 200 — inserimento riuscito	24
6.4 Diagramma di stato — POST /nuovo-impianto/inserisci	24
7. Note tecniche per l'integrazione	26
7.1 Sequenza di caricamento consigliata	26
7.2 Formati data	26
7.3 Identificativi tipologiche — regola fondamentale	26
7.4 Gestione POD e PDR	26
7.5 Autocomplete indirizzi — implementazione consigliata	26
7.6 Transazionalità dell'inserimento	27
7.7 Campo request nelle response — correlazione e debug	27
7.8 Libretto dell'Impianto — stato post-accatastamento	27
7.9 Verifica stato sistema (Health check del sistema)	27
7.10 Rappresentare un impianto con un solo generatore/macchina	27
8. Changelog	28
8.1 v1.0.5 → v1.0.6	28
8.1.1 Rename degli schemi (stesso significato, nome nuovo)	28

8.1.2 Struttura multi-generatore / multi-macchina	28
8.1.3 Schema ImpiantoDaAccatastare (ex NuovImpiantoDatImpianto)	29
8.1.4 Schema GruppoGeneratore / MacchinaGeneratore (ex NuovImpiantoDatiGeneratore)	29
8.1.5 Schema ResponsabileImpianto (ex NuovImpiantoDatiResponsabile)	29
8.1.6 Endpoint	30
8.1.7 Schema ResponseMessage	30
8.1.8 Schema ResponseMessageDetails	30
8.1.9 Schema ProprietaResp	31
8.1.10 Parametro CodiceFiscaleOperatore (header)	31
8.1.11 Sicurezza — schema BearerAuth	31
8.1.12 Schema aggiunto: ImpiantoAccatastato	31
8.1.13 Schema AccatastamentoRisposta (ex NuovImpiantoRespWithReq) — correzione.....	31
8.1.14 Eco della request (campo "request.corpo") — correzione	31
8.1.15 Campo "request" nelle risposte di errore (400/403/404/500) — correzione.....	32

1. Introduzione

1.1 Scopo del documento

Il presente documento descrive le specifiche funzionali dell'API CAITEL Accatastamento (versione 1.0.6), pubblicata da Regione Liguria — Liguria Digitale. L'API consente agli operatori abilitati l'accatastamento di impianti termici nel sistema regionale del Libretto dell'Impianto Termico (CAITEL).

Il documento è rivolto agli sviluppatori di sistemi client (gestionali di impresa, portali web, app mobile) che necessitano di integrare la funzionalità di accatastamento nei propri sistemi, senza accedere direttamente all'interfaccia web CAITEL.

I dettagli relativi al flusso che devono seguire le ditte accreditate su Caitel per poter svolgere le attività legate all'integrazione dell'utilizzo delle API nei loro sistemi gestionali e i dettagli relativi alla componente di autenticazione per ogni chiamata API sono descritti nel documento `API_Caitel_istruzioni_abilitazione_ditte_1.0.0.pdf`

1.2 Panoramica funzionale

L'API espone i seguenti servizi REST:

- Accatastamento di un nuovo impianto termico con contestuale creazione della bozza del libretto dell'Impianto
- Verifica preventiva dei dati prima dell'inserimento definitivo
- Recupero delle liste di valori (tipologiche) necessarie a popolare i campi del request body
- Recupero dei metadati di validazione dei campi per sezione del request body
- Autocomplete degli indirizzi per comune e provincia
- Health check del sistema

Il risultato dell'accatastamento è la creazione del Libretto dell'Impianto in stato bozza sul database CAITEL, con assegnazione di un codice catastale univoco all'impianto. Per poter compilare il Rapporto di Efficienza Energetica (RCEE) è necessario validare il libretto.

1.3 Ambienti

Ambiente	Base URL	Sandbox
Produzione	<code>https://services-caitel.regione.liguria.it/api/accatastamento</code>	No
Test	<code>https://services-caitel-test.regione.liguria.it/api/accatastamento</code>	No

Nota: Tutti gli esempi di URL in questo documento omettono la base URL per brevità.

2. Flusso funzionale — Accatastamento nuovo impianto

Il processo di accatastamento si articola in quattro fasi sequenziali. Le fasi 1 e 2 possono essere eseguite prima di raccogliere i dati utente; le fasi 3 e 4 sono sequenziali e richiedono che i dati siano stati raccolti dal sistema client.

Fase	Descrizione	Endpoint coinvolti
1 — Caricamento dati di supporto	Recupero tipologiche e metadati campi. Eseguita prima di raccogliere i dati.	GET /tipologiche/* e GET /proprietà/accatastamento/*
2 — Raccolta dati dinamica	Chiamate dinamiche per comuni e indirizzi, eseguite al variare dei dati selezionati.	GET /tipologiche/comuni-provincia/{id} GET /tipologiche/indirizzi-provincia-comune-*/...
3 — Verifica preventiva	Validazione completa lato server prima dell'invio definitivo. Facoltativa ma consigliata.	POST /nuovo-impianto/verifica-dati
4 — Inserimento	Creazione definitiva dell'impianto sul catasto regionale.	POST /nuovo-impianto/inserisci

2.1 Fase 1 — Caricamento tipologiche

Prima di raccogliere i dati da inviare, il client recupera i valori di tutte le liste di selezione. Tutte le chiamate sono GET senza parametri nel path, ad eccezione dell'header di autenticazione.

Endpoint	Campo nel request body	Note
GET /tipologiche/tipi-intervento	identificativoTipoIntervento	Es. Nuova installazione, Sostituzione generatore
GET /tipologiche/tipi-generatori-per-nuovo-impianto	identificativoTipoGeneratore	Solo i tipi compatibili con accatastamento nuovo impianto. Determina le regole sui campi condizionali (vedi §2.3)
GET /tipologiche/tipi-fluido-vettore	identificativoTipoFluidoVettore	Es. acqua, aria, altro. Se selezionato "altro" si attiva il campo altroFluidoVettore
GET /tipologiche/tipi-macchina-termica	identificativoTipoMacchinaTermica	Abilitato solo per tipologie generatore 1 e 2
GET /tipologiche/tipi-generatore-rendimento	identificativoTipoGeneratore di Calore	Es. Generatore aria calda. Abilitato solo per tipologie 1 e 2
GET /tipologiche/tipi-responsabilita	identificativoTipoResponsabilita	Es. Manutentore, Installatore
GET /tipologiche/tipi-colore-civico	identificativoColoreCivico	N = nero (default), R = rosso
GET /tipologiche/is-macchine-a-biomassa	identificativoIsMacchinaaBiomassa	Abilitato solo per tipologia generatore 1
GET /tipologiche/province-liguria	identificativoProvincia (sezione impianto)	Solo le 4 province della Liguria (SP, SV, GE, IM)
GET /tipologiche/province	identificativoProvincia (sezione responsabile)	Tutte le province italiane

Nota: Gli identificativi restituiti (es. "1", "2") sono opachi: non assumere significato posizionale e non usare valori hardcoded. Chiamare sempre i servizi tipologiche per popolare le liste. I valori possono variare tra ambiente di test e produzione.

2.2 Fase 1 — Caricamento metadati campi (opzionale ma consigliato)

Il client può richiedere i metadati di validazione per ciascuna sezione del request body. I metadati descrivono, per ogni campo: etichetta, placeholder, obbligatorietà, tipo di controllo e tutti i vincoli di validazione (regex, lunghezze, range, date). Permettono di replicare lato client la stessa logica che il backend applica lato server.

Endpoint	Sezione del request body
GET /proprietà/accatastamento/nuovoImpianto	Dati generali dell'impianto (localizzazione, tipo intervento)
GET /proprietà/accatastamento/nuovoImpiantoGeneratore	Dati tecnici del generatore
GET /proprietà/accatastamento/nuovoImpiantoForniture	Dati forniture (POD/PDR)
GET /proprietà/accatastamento/nuovoImpiantoResponsabile	Dati del responsabile dell'impianto

La risposta è una mappa campo→ProprietàResp. La chiave corrisponde esattamente al nome del campo nel request body (es. "pod", "civico", "dataIntervento"). Vedere §5.5 per la struttura dettagliata di ProprietàResp.

2.3 Fase 2 — Chiamate dinamiche per dati variabili

2.3.1 Selezione provincia → caricamento comuni

Alla selezione della provincia (sia nella sezione impianto che nella sezione responsabile) il client deve caricare i comuni corrispondenti:

```
GET /tipologiche/comuni-provincia/{identificativo-provincia}
```

Il parametro {identificativo-provincia} è il codice ISTAT della provincia restituito dalla tipologica province (es. "010" per Genova). La risposta è una lista di BasePairString con identificativo (codice ISTAT comune) e valore (nome comune).

Nota: Alla selezione di una nuova provincia occorre resettare il comune e l'indirizzo selezionati, poiché non appartengono più alla nuova provincia.

2.3.2 Digitazione indirizzo — autocomplete

Per il campo indirizzo dell'impianto il client implementa un autocomplete: ad ogni modifica del campo di testo (consigliato debounce di 300ms, minimo 1 carattere) viene chiamato il servizio:

```
GET /tipologiche/indirizzi-provincia-comune-nuovo-o-modifica-impianto/{identificativo-provincia}/{identificativo-comune}/{parte}
```

dove {parte} è il testo parziale digitato dall'utente. La risposta è una lista di indirizzi che contengono la stringa cercata, con identificativo (codice univoco dell'indirizzo) e valore (stringa descrittiva, es. "VIA ROMA").

Il client deve salvare entrambi i valori dell'indirizzo selezionato: l'identificativoIndirizzo (codice, da inserire nel request body) e il valoreIndirizzo (stringa, campo informativo opzionale nel request body).

Nota: La chiamata indirizzi richiede che provincia e comune siano già selezionati. Se l'utente modifica la provincia o il comune occorre resettare anche il campo indirizzo.

2.4 Fase 2 — Regole condizionali sezione Generatore

I campi della sezione Generatore hanno obbligatorietà e disponibilità condizionali in base alla tipologia selezionata (identificativoTipoGeneratore). La tabella seguente riassume le regole applicate sia dal frontend che dal backend.

Nota: La tabella si applica ai campi di ciascun elemento dell'array "macchine" di ogni generatore (vedi §2.8) — la tipologia resta un dato di gruppo (identificativoTipoGeneratore sul generatore, non sulla singola macchina).

Tipologia	Campi aggiuntivi obbligatori	Campi non applicabili (inviare null o omettere)
1 — Caldaia / moduli con biomassa	identificativoTipoMacchinaTermica identificativoTipoGeneratoreDiCalore numeroModuli identificativoIsMacchinaaBiomassa	numeroCircuiti i tre booleani frigo (*)
2 — Caldaia / moduli senza biomassa	identificativoTipoMacchinaTermica identificativoTipoGeneratoreDiCalore numeroModuli	numeroCircuiti identificativoIsMacchinaaBiomassa i tre booleani frigo (*)
3 — Frigorifero ad assorbimento (recupero calore)	numeroCircuiti almeno uno dei tre booleani frigo (*)	identificativoTipoMacchinaTermica identificativoTipoGeneratoreDiCalore numeroModuli identificativoIsMacchinaaBiomassa
4 — Frigorifero ad assorbimento (fiamma diretta)	numeroCircuiti almeno uno dei tre booleani frigo (*)	identificativoTipoMacchinaTermica identificativoTipoGeneratoreDiCalore numeroModuli identificativoIsMacchinaaBiomassa
5 — Frigorifero a ciclo di compressione	numeroCircuiti almeno uno dei tre booleani frigo (*)	identificativoTipoMacchinaTermica identificativoTipoGeneratoreDiCalore numeroModuli identificativoIsMacchinaaBiomassa
6 — Altro	(nessuno aggiuntivo)	identificativoTipoMacchinaTermica identificativoTipoGeneratoreDiCalore numeroModuli numeroCircuiti identificativoIsMacchinaaBiomassa i tre booleani frigo (*)
7 — Altro	(nessuno aggiuntivo)	identificativoTipoMacchinaTermica identificativoTipoGeneratoreDiCalore numeroModuli numeroCircuiti identificativoIsMacchinaaBiomassa i tre booleani frigo (*)

(*) I tre campi booleani frigo sono: isAdAssorbimentoRecuperoCaloreFrigo, isAdAssorbimentoFiammaFrigo, isACicloDiCompressione. Sono mutuamente esclusivi: al più uno può essere true. Per le tipologie 3, 4 e 5 almeno uno deve essere true.

Nota: Il campo altroFluidoVettore diventa obbligatorio quando identificativoTipoFluidoVettore corrisponde al valore "altro" (valore da verificare tramite tipologica). In tutti gli altri casi deve essere null o omissso.

2.5 Fase 2 — Regole obbligatorietà sezione Responsabile

I campi anagrafici del responsabile hanno obbligatorietà condizionale. È obbligatoria esattamente una delle tre combinazioni seguenti:

Combinazione	Campi obbligatori	Caso d'uso
1 — Persona fisica	cognome + nome + codiceFiscale	Il responsabile è una persona fisica identificata con CF
2 — Azienda con Partita IVA	ragioneSociale + partitalva	Il responsabile è un'azienda identificata con P.IVA (11 cifre)
3 — Azienda con Codice Fiscale	ragioneSociale + codiceFiscale	Il responsabile è un'azienda il cui referente è identificato tramite CF persona fisica (16 caratteri). Il campo codiceFiscale accetta esclusivamente CF a 16 caratteri nel formato italiano standard.

Sono sempre obbligatori, indipendentemente dalla combinazione: identificativoTipoResponsabilita, identificativoComune, indirizzo, civico, email, telefono.

2.6 Fase 3 — Verifica preventiva dei dati

Prima dell'inserimento definitivo è fortemente consigliato chiamare il servizio di verifica. Questo servizio esegue una validazione completa lato server equivalente a quella dell'inserimento, senza però creare alcun record sul database.

```
POST /nuovo-impianto/verifica-dati
Content-Type: application/json

{ <AccatastamentoRichiesta> }
```

La verifica esegue i seguenti controlli:

- Validazione formale del body (campi obbligatori, formati, pattern regexp)
- Validazione semantica (coerenza tra i valori, regole condizionali per tipologia)
- Verifica esistenza dei valori di riferimento (provincia, comune, indirizzo, tipologiche)
- Verifica che l'impianto non sia già presente a catasto con lo stesso indirizzo e POD

Esiti possibili:

- HTTP 200 — dati validi. elemento = true (sempre). Procedere con l'inserimento.
- HTTP 400 — dati non validi. Dettagli per campo in informazioni.ulterioriMessaggiConDettagli.
- HTTP 404 — identificativi di riferimento non trovati (provincia, comune, indirizzo inesistenti).

2.7 Fase 4 — Inserimento impianto

L'inserimento definitivo dell'impianto viene eseguito con la stessa struttura dati della verifica preventiva. Il servizio esegue internamente l'intera sequenza di validazione prima di procedere con la scrittura sul database, all'interno di una transazione atomica.

```
POST /nuovo-impianto/inserisci
Content-Type: application/json
```

```
{ <AccatastamentoRichiesta> }
```

In caso di successo (HTTP 200) il backend esegue nell'ordine:

- Validazione completa dei dati (stessa logica di verifica-dati)
- Verifica che l'impianto non esista già a catasto
- Inserimento o aggiornamento del Responsabile
- Inserimento dell'Impianto
- Collegamento Impianto-Responsabile
- Per ciascun generatore (uno o più per chiamata, array "generatori"): inserimento del Gruppo e collegamento Gruppo-Azienda operatrice
- Per ciascuna macchina del generatore (una o più per generatore, array "macchine"): inserimento della Macchina termica
- Creazione del Libretto dell'Impianto in stato bozza

La risposta 200 contiene in elemento un oggetto `ImpiantoAccatastato` con, tra gli altri campi, `codiceImpianto` (il codice catastale assegnato all'impianto, es. "07D9690034567890") e `descrizioniGruppiOmogenei` (una voce per ciascun gruppo effettivamente presente sull'impianto, quindi una per ogni generatore inserito nel flusso multi-generatore) — vedi §5.2 per la struttura completa. Il campo `informazioni.messaggio` riporta la conferma dell'avvenuto inserimento.

Nota: Il Libretto dell'Impianto viene creato in stato bozza. Per poter compilare il Rapporto di Efficienza Energetica (RCEE) è necessario validare il libretto tramite l'interfaccia web CAITEL.

2.8 Struttura multi-generatore / multi-macchina

Il campo "generatori" è un array in cui ogni elemento è un gruppo omogeneo, e ogni gruppo può a sua volta contenere più macchine tramite il proprio campo "macchine" (schema `MacchinaGeneratore`, vedi §4.4). Ogni elemento di "generatori" porta solo i dati di gruppo (`identificativoTipoGeneratore`, `identificativoTipoFluidoVettore`, `altroFluidoVettore`); i dati tecnici di ciascuna macchina (`dataInstallazione`, `potenza`, `fabbricante`, `modello`, `matricola`, campi condizionali per tipologia) vanno su ogni elemento di "macchine". Un impianto con un solo generatore e una sola macchina si rappresenta con una lista di un elemento in "generatori", il cui unico elemento ha a sua volta una lista di un elemento in "macchine". Questa struttura permette anche di accatastare in un'unica chiamata un impianto con più gruppi di generatori e più macchine per gruppo.

Esempio di payload con 2 generatori (tipologia 2 e tipologia 3) e 2 macchine sul primo generatore:

```
{
  "impianto": { ... },
  "forniture": { "pod": "IT001E12345678", "pdr": "00112233445566" },
  "responsabile": { ... },
  "generatori": [
    {
      "identificativoTipoGeneratore": "2",
      "identificativoTipoFluidoVettore": "1",
      "macchine": [
        { "dataInstallazione": "2024-11-10", "identificativoTipoMacchinaTermica": "3",
          "identificativoTipoGeneratoreIdiCalore": "4", "numeroModuli": "1",
          "potenzaTermicaUtileNominalekW": "24.0", "fabbricante": "Vaillant",

```



```
        "modello": "ecoTEC plus VMW 266/5-5", "matricola": "1234567890" },
    { "dataInstallazione": "2024-11-10", "identificativoTipoMacchinaTermica": "3",
      "identificativoTipoGeneratorediCalore": "4", "numeroModuli": "1",
      "potenzaTermicaUtileNominalekW": "35.5", "fabbricante": "Vaillant",
      "modello": "ecoTEC plus VMW 356/5-5", "matricola": "1234567891" }
  ]
},
{
  "identificativoTipoGeneratore": "3",
  "identificativoTipoFluidoVettore": "2",
  "macchine": [
    { "dataInstallazione": "2025-03-20", "numeroCircuiti": "2",
      "isAdAssorbimentoRecuperoCaloreFrigo": true,
      "potenzaTermicaUtileNominalekW": "18.5", "fabbricante": "Daikin",
      "modello": "EWYQ016ADVP", "matricola": "9876543210" }
  ]
}
]
```

Nota: Esempio completo nel file `NuovoImpiantoMultiGeneratoreMultiMacchina.json` allegato allo YAML.

3. Dettaglio endpoint

3.1 POST /nuovo-impianto/verifica-dati

Proprietà	Valore
Method	POST
Path	/nuovo-impianto/verifica-dati
Content-Type	application/json
Request body	AccatastamentoRichiesta (obbligatorio)
Risposta 200	EsitoVerificaDatiAccatastamentoResponse — elemento sempre true se dati validi
Risposta 400	BadRequestCaitelResponse — informazioni.ulterioriMessaggiConDettagli con dettaglio per campo
Risposta 403	ForbiddenCaitelResponse — header mancante o operatore non autorizzato
Risposta 404	NotFoundCaitelResponse — identificativi di riferimento non trovati
Risposta 500	InternalServerErrorCaitelResponse

Nota: Questo endpoint non scrive nulla sul database. Può essere chiamato più volte senza effetti collaterali. Tuttavia esegue anche la verifica di esistenza impianto: se un impianto con lo stesso indirizzo e POD esiste già restituisce errore 400.

3.2 POST /nuovo-impianto/inserisci

Proprietà	Valore
Method	POST
Path	/nuovo-impianto/inserisci
Content-Type	application/json
Request body	AccatastamentoRichiesta (obbligatorio)
Risposta 200	AccatastamentoRisposta — elemento = oggetto ImpiantoAccatastato (vedi §5.2), con codiceImpianto il codice catastale assegnato
Risposta 400	BadRequestCaitelResponse
Risposta 403	ForbiddenCaitelResponse
Risposta 404	NotFoundCaitelResponse — provincia, comune o indirizzo non trovati
Risposta 500	InternalServerErrorCaitelResponse

Nota: L'operazione è transazionale: in caso di errore in qualsiasi fase dell'inserimento l'intera operazione viene annullata (rollback). Nessun dato parziale viene scritto sul database.

Esempio di request body:

```
{
  "impianto": {
    "identificativoTipoIntervento": "1",
    "dataIntervento": "2026-05-11",
    "identificativoProvincia": "008",
    "identificativoComune": "DIANO MARINA",
    "identificativoIndirizzo": "13753",
    "valoreIndirizzo": "GENERALE ARDOINO",
    "civico": "10",
    "identificativoColoreCivico": "N",
    "letteraOpalazzo": null,
    "scala": null,
    "interno": null,
    "internoLettera": null
  },
  "generatori": [
```



```
{
  "identificativoTipoGeneratore": "2",
  "identificativoTipoFluidoVettore": "1",
  "altroFluidoVettore": null,
  "macchine": [
    {
      "dataInstallazione": "2000-01-01",
      "identificativoTipoMacchinaTermica": null,
      "identificativoTipoGeneratoreDiCalore": "1",
      "isAdAssorbimentoRecuperoCaloreFrigo": false,
      "isAdAssorbimentoFiammaFrigo": false,
      "isACicloDiCompressione": false,
      "potenzaTermicaUtileNominalekW": "24.55",
      "numeroModuli": null,
      "numeroCircuiti": null,
      "identificativoIsMacchinaaBiomassa": null,
      "fabbricante": "CHAFFOTEAUX",
      "modello": "PIGMA 25 CF EU (OK)",
      "matricola": "1234667788"
    }
  ]
},
"forniture": {
  "pod": "IT123E000000000",
  "pdr": null
},
"responsabile": {
  "identificativoTipoResponsabilita": "1",
  "cognome": "PAPERINO",
  "nome": "PAOLINO",
  "codiceFiscale": null,
  "ragioneSociale": null,
  "partitaIva": null,
  "identificativoProvincia": "009",
  "identificativoComune": "SAVONA",
  "indirizzo": "PIETRO PALEOCAPA",
  "civico": "12",
  "email": "Info@itglobalservice.it",
  "telefono": "0183752270"
}
}
```

Nota: In questo esempio il responsabile è una persona fisica (combinazione 1: cognome + nome); codiceFiscale è null perchè non fornito, ragioneSociale e partitaIva sono null. Il generatore è di tipologia 2 (caldaia senza biomassa): identificativoTipoMacchinaTermica è null, numeroModuli è null, i booleani frigo sono false.

Esempio di risposta (HTTP 200):

```
{
  "elemento": {
    "codiceImpianto": "07D9690034567890",
    "identificativoImpianto": "aGVsbG8...",
    "identificativoResponsabile": "d29ybGQ...",
    "identificativoIndirizzo": "13753",
    "esistente": false,
  }
}
```



```
{
  "descrizioniGruppiOmogenei": ["1 - anno scadenza bollino: 2027"]
},
"responseStatus": "200 OK",
"informazioni": {
  "tipo": "SUCCESS",
  "messaggio": "Impianto 07D9690034567890 inserito con successo!"
}
}
```

Nota: *identificativoImpianto e identificativoResponsabile sono valori cifrati (vedi @EncryptedId): trattarli come opachi, da ripassare così come sono alle altre API CAITEL che li richiedono.*

3.3 GET /tipologiche/tipi-intervento

Restituisce la lista dei tipi di intervento disponibili per un nuovo impianto (es. Nuova installazione, Sostituzione generatore, ecc.). Utilizzata per popolare il dropdown "Tipo intervento" nella sezione impianto.

3.4 GET /tipologiche/tipi-generatori-per-nuovo-impianto

Restituisce i soli tipi di generatore compatibili con l'accatastamento di un nuovo impianto. L'identificativo restituito determina le regole sui campi condizionali (vedere §2.4). Non chiamare /tipologiche/tipi-generatori generici — la lista potrebbe contenere tipologie non applicabili al nuovo impianto.

3.5 GET /tipologiche/tipi-fluido-vettore

Lista dei tipi di fluido vettore (es. acqua, aria, altro). Se l'utente seleziona "altro" (valore da verificare tramite la lista), il campo altroFluidoVettore nel request body diventa obbligatorio.

3.6 GET /tipologiche/tipi-macchina-termica

Lista dei tipi di macchina termica. Utilizzata per il campo identificativoTipoMacchinaTermica, obbligatorio solo per le tipologie generatore 1 e 2.

3.7 GET /tipologiche/tipi-generatore-rendimento

Lista dei tipi di generatore di calore (es. Generatore aria calda, Caldaia a condensazione). Utilizzata per identificativoTipoGeneratorediCalore, obbligatorio solo per tipologie 1 e 2.

3.8 GET /tipologiche/tipi-responsabilita

Lista dei tipi di responsabilità dell'impianto (es. Manutentore, Installatore, Terzo Responsabile). Sempre obbligatorio nella sezione Responsabile.

3.9 GET /tipologiche/tipi-colore-civico

Restituisce i due valori ammessi per il colore del civico: N (nero) e R (rosso). Il default suggerito è N. Il valore va inserito in identificativoColoreCivico.

3.10 GET /tipologiche/is-macchine-a-biomassa

Lista dei valori SI/NO per indicare se la macchina è a biomassa. Obbligatorio solo per la tipologia generatore 1.

3.11 GET /tipologiche/province-liguria

Le 4 province della Liguria (Genova, La Spezia, Savona, Imperia) con codice ISTAT. Utilizzata per il dropdown "Provincia" della sezione impianto. L'impianto deve necessariamente trovarsi in Liguria.

3.12 GET /tipologiche/province

Tutte le province italiane con codice ISTAT. Utilizzata per il dropdown "Provincia" della sezione responsabile. Il responsabile può risiedere in qualsiasi provincia italiana.

3.13 GET /tipologiche/comuni-provincia/{identificativo-provincia}

Lista dei comuni della provincia specificata (codice ISTAT). Chiamata dinamicamente alla selezione della provincia. Usata sia per la sezione impianto che per la sezione responsabile.

Parametro	In	Tipo	Esempio	Descrizione
id-provincia	path	string	010	Codice ISTAT della provincia (da tipologica province)

3.14 GET /tipologiche/indirizzi-provincia-comune-nuovo-o-modifica-impianto/{identificativo-provincia}/{identificativo-comune}/{parte}

Autocomplete degli indirizzi filtrati per provincia, comune e testo parziale. Chiamata ad ogni digitazione nel campo indirizzo dell'impianto (debounce 300ms consigliato). Richiede che provincia e comune siano già stati selezionati.

Parametro	In	Tipo	Esempio	Descrizione
id-provincia	path	string	010	Codice ISTAT della provincia
id-comune	path	string	025	Codice ISTAT del comune
parte	path	string	VIA G	Testo parziale dell'indirizzo (min 1 carattere)

3.15 GET /proprieta/accatastamento/{sezione}

Restituisce i metadati di validazione dei campi per la sezione richiesta. La risposta è una mappa campo→ProprietaResp (vedere §5.5).

Valore sezione	Contenuto
nuovoImpianto	Dati generali dell'impianto (localizzazione, tipo intervento, civico)
nuovoImpiantoGeneratore	Dati tecnici del generatore (tipologia, fluido, potenza, date, componenti)
nuovoImpiantoForniture	Dati forniture elettriche e gas (POD, PDR)
nuovoImpiantoResponsabile	Dati anagrafici del responsabile dell'impianto

3.16 GET /status

Health check del sistema. Restituisce HTTP 200 con elemento = true se il servizio è raggiungibile e il database è disponibile. Consigliato prima di sessioni massive di accatastamento.

4. Strutture dati — Request body

4.1 AccatastamentoRichiesta

Corpo della richiesta per entrambi gli endpoint POST. impianto, forniture, responsabile e generatori sono sempre obbligatori (vedi §2.8 per la struttura di "generatori"):

Campo	Tipo	Obbligatorio	Descrizione
impianto	ImpiantoDaAccatastare	SI	Localizzazione dell'impianto e tipo di intervento
generatori	GruppoGeneratore[]	SI	Uno o più gruppi, ciascuno con una o più macchine (§2.8, §4.4)
forniture	ForniturePodPdr	SI	Punto di prelievo energia elettrica e gas
responsabile	ResponsabileImpianto	SI	Anagrafica del responsabile dell'impianto

4.2 ImpiantoDaAccatastare

Dati di localizzazione dell'impianto e tipo di intervento.

Campo JSON	Obbl.	Tipo	Vincoli	Descrizione
identificativoTipoIntervento	SI	string	Da tipologica	GET /tipologiche/tipi-intervento
dataIntervento	SI	string	AAAA-MM-GG ^d{4}-d{2}-d{2}\$	Data dell'intervento
identificativoProvincia	SI	string	Codice ISTAT	GET /tipologiche/province-liguria (solo Liguria)
identificativoComune	SI	string	Codice ISTAT	GET /tipologiche/comuni-provincia/{id}
identificativoIndirizzo	SI	string	Codice univoco	Dalla risposta autocomplete indirizzi
valoreIndirizzo	NO	string		Stringa descrittiva es. "VIA ROMA" (informativo)
civico	SI	string	^d{1,4}\$ min 1, max 9999	Numero civico (1-4 cifre)
identificativoColoreCivico	SI	string	N oppure R	N = nero (default), R = rosso
letteraOpalazzo	NO	string	Max 20 car.	Lettera o numero di palazzo es. "A", "12"
scala	NO	string	Max 20 car.	Scala (es. A, 1, Destra)
interno	NO	string	Max 20 car.	Numero interno
internoLettera	NO	string	Max 1 car.	Lettera associata all'interno

4.3 GruppoGeneratore

Dati del gruppo omogeneo generatore: tipologia, fluido vettore e la lista delle macchine del gruppo. I dati tecnici delle macchine (dataInstallazione, potenza, fabbricante, modello, matricola e i campi condizionali per tipologia) sono su ogni elemento di "macchine" — vedi §4.4.

Campo JSON	Obbl.	Tipo	Vincoli	Descrizione
------------	-------	------	---------	-------------

identificativoTipoGeneratore	SI	string	Da tipologica	GET /tipologiche/tipi-generatori-per-nuovo-impianto
identificativoTipoFluidoVettore	SI	string	Da tipologica	GET /tipologiche/tipi-fluido-vettore
altroFluidoVettore	Cond.	string		Obbligatorio se tipoFluidoVettore = "altro"
macchine	SI	MacchinaGeneratore[]		Lista delle macchine del gruppo (§4.4), una o più per generatore

4.4 MacchinaGeneratore

Dati di una singola macchina all'interno di un elemento dell'array "generatori" (vedi §2.8): un gruppo generatore può avere più macchine. Le regole di obbligatorietà condizionale per tipologia (§2.4) si applicano a questi campi usando identificativoTipoGeneratore del generatore a cui la macchina appartiene, non un campo di questo oggetto.

Campo JSON	Obbl.	Tipo	Vincoli	Descrizione
dataInstallazione	SI	string	AAAA-MM-GG $\wedge\{4\}-\{2\}-\{2\}$	Data di installazione della macchina
identificativoTipoMacchinaTermica	Cond.	string	Da tipologica	Obbligatorio per tipologia generatore 1 e 2. GET /tipologiche/tipi-macchina-termica
identificativoTipoGeneratore di Calore	Cond.	string	Da tipologica	Obbligatorio per tipologia generatore 1 e 2. GET /tipologiche/tipi-generatore-rendimento
potenzaTermicaUtileNominalekW	SI	string	$\wedge d+(\.d+)?$$ ≥ 0	Potenza termica nominale in kW es. "24.0"
numeroModuli	Cond.	string	$\wedge d+$$	Intero positivo. Obbligatorio per tipologia generatore 1 e 2
numeroCircuiti	Cond.	string	$\wedge d+$$	Intero positivo. Obbligatorio per tipologie generatore 3, 4, 5
identificativoIsMacchinaaBiomassa	Cond.	string	Da tipologica	Obbligatorio per tipologia generatore 1. GET /tipologiche/is-macchine-a-biomassa
isAdAssorbimentoRecuperoCaloreFrigo	Cond.	boolean	true false	Mutualmente esclusivo con gli altri due boolean frigo. Min uno true per tipologie generatore 3/4/5
isAdAssorbimentoFiammaFrigo	Cond.	boolean	true false	Mutualmente esclusivo con gli altri due boolean frigo
isACicloDiCompressione	Cond.	boolean	true false	Mutualmente esclusivo con gli altri due boolean frigo
fabbricante	NO	string		Nome del fabbricante es. "Daikin"
modello	SI	string		Modello della macchina

matricola	SI	string		Numero di serie / matricola della macchina
-----------	----	--------	--	--

4.5 ForniturePodPdr

Dati dei punti di prelievo energia. Solo POD è obbligatorio.

Campo JSON	Obbl.	Tipo	Pattern	Lunghezza	Descrizione
pod	SI	string	^IT[0-9]{3}E[0-9]{8,9}\$	14-15 car.	Point of Delivery — punto prelievo energia elettrica. Es. IT001E12345678
pdr	NO	string	^d{14}\$	14 cifre	Punto di Raccolta — punto prelievo gas. Es. 00112233445566

Nota: Se l'impianto non ha allacciamento al gas, pdr può essere omesso o null. Il POD deve essere quello relativo all'edificio in cui è installato l'impianto.

4.6 ResponsabileImpianto

Dati anagrafici del responsabile dell'impianto. Vedere §2.5 per le regole di obbligatorietà condizionale dei campi anagrafici.

Campo JSON	Obbl.	Tipo	Vincoli	Descrizione
identificativoTipoResponsabilita	SI	string	Da tipologica	GET /tipologiche/tipi-responsabilita
cognome	Comb. 1	string		Persona fisica — obbligatorio con nome + codiceFiscale
nome	Comb. 1	string		Persona fisica — obbligatorio con cognome + codiceFiscale
codiceFiscale	Comb. 1	string	^[A-Za-z]{6}[0-9]{2}[A-EHLMPRSTa-ehlmprst][0-9]{2}[A-Za-z][0-9]{3}[A-Za-z]\$ 16 car. esatti	CF persona fisica (16 caratteri). La lettera di mese deve appartenere a [A-EHLMPRSTa-ehlmprst]. Usato solo nella combinazione 1.
ragioneSociale	Comb. 2/3	string		Azienda — obbligatorio con partitalva o codiceFiscale
partitalva	Comb. 2	string	^[0-9]{11}\$ 11 cifre esatte	Partita IVA (11 cifre numeriche) — obbligatorio con ragioneSociale
identificativoProvincia	NO	string	Codice ISTAT	GET /tipologiche/province (tutte le province)
identificativoComune	SI	string	Codice ISTAT	Comune di residenza/sede. GET /tipologiche/comuni-provincia/{id}
indirizzo	SI	string		Via, piazza, ecc. (testo libero per il responsabile)
civico	SI	string		Numero civico
email	SI	string	^[a-zA-Z0-9._%+~]+@[a-zA-Z0-9.-]+.[a-zA-Z]{2,}\$	Indirizzo email di contatto



telefono	SI	string	^(?=[0-9])[0-9._eE+\\-\\/{8,250}\$ min 8, max 250 car.	Numero di telefono. Deve contenere almeno una cifra. Ammessi: cifre, spazi, . _ + \\ - /;
----------	----	--------	--	---

5. Strutture dati — Response

5.1 Struttura comune di tutte le response

Tutte le response (sia 2xx che 4xx/5xx) condividono la stessa struttura di base:

Campo	Tipo	Sempre presente	Descrizione
elemento / lista / mappa	vario	SI	Il payload effettivo della risposta. elemento per le POST, lista per le tipologiche, mappa per le proprietà
responseStatus	string	SI	Rappresentazione testuale del codice HTTP es. "200 OK", "400 BAD_REQUEST"
informazioni	ResponseMessage	SI	Messaggio di esito con tipo (SUCCESS/ERROR/WARNING), testo e dettagli
request	RequestPayload*	SI	Echo della request originale (body e/o parametri). Utile per debug e correlazione log

5.2 ImpiantoAccatastato — payload di elemento per POST /nuovo-impianto/inserisci

Per l'endpoint POST /nuovo-impianto/inserisci, il campo "elemento" della risposta non è una semplice stringa ma un oggetto ImpiantoAccatastato con la struttura seguente (per POST /nuovo-impianto/verifica-dati, invece, "elemento" è un boolean, sempre true — vedi §2.6).

Campo	Tipo	Descrizione
codiceImpianto	string	Codice catastale assegnato all'impianto, es. "07D9690034567890"
identificativoImpianto	string	Identificativo numerico univoco dell'impianto (valore cifrato)
identificativoResponsabile	string	Identificativo univoco del responsabile associato (valore cifrato)
identificativoIndirizzo	string	Identificativo univoco dell'indirizzo dell'impianto
esistente	boolean	true se l'impianto era già presente allo stesso indirizzo (non è stato creato ex novo)
descrizioniGruppiOmogenei	string[]	Una voce per ciascun gruppo omogeneo presente sull'impianto (esistente o appena creato). Con il flusso multi-generatore (§2.8) contiene tipicamente una voce per ogni elemento di "generatori" inviato

5.3 ResponseMessage — struttura del messaggio di esito

Campo	Tipo	Descrizione
tipo	enum	UNDEF INFO WARNING ERROR SUCCESS
messaggio	string	Messaggio principale leggibile dall'utente
systemMessageCode	string	Codice interno per correlare log e risposta. Presente solo in caso di errore. Riportare a supporto
stringaUnicaHtml	string	Aggregato HTML di tutti i messaggi (principale + ulteriori). Pronto per rendering nel frontend
ulterioriMessaggi	string[]	Lista di messaggi aggiuntivi testuali. Nella risposta 200 di inserisci contiene nota sul CF operatore e codice impianto

ulterioriMessaggiConDettagli	ResponseMessageDetails[]	Dettagli per campo. Presente nelle risposte 400 di validazione (vedere §5.4)
------------------------------	--------------------------	--

5.4 ResponseMessageDetails — dettaglio errori di validazione (risposta 400)

Ogni elemento di informazioni.ulterioriMessaggiConDettagli descrive un campo in errore. Da leggere in sequenza per presentare all'utente i campi da correggere.

Campo	Tipo	Descrizione
tipo	enum	ERROR, WARNING, INFO — severità del problema
dettaglio	string	Messaggio di errore per il campo (es. "Campo obbligatorio", "Formato non valido")
ulterioriDettagli	BasePairString[]	Dettagli aggiuntivi: nome del campo, etichetta visualizzata, valore ricevuto
conclusione	string	Testo conclusivo / azione suggerita all'utente
domanda	string	Domanda per l'utente in operazioni interattive (conferma sovrascrittura, ecc.)
possibiliRisposteDomanda	BasePairString[]	Opzioni di risposta alla domanda, come coppie identificativo/valore

5.5 ProprietaResp — metadati campi (risposta GET /proprieta/accatastamento/*)

Struttura che descrive le proprietà di visibilità e validazione di un singolo campo del request body, come configurato nel database CAITEL (vista V_LIB_CAMPI_FE). Il backend applica le stesse regole anche in fase di validazione server-side.

Campo	Tipo	Descrizione
campo	string	Nome del campo nel request body (coincide con la chiave della mappa)
etichetta	string	Etichetta descrittiva del campo (informativa per il client)
placeholder	string	Testo placeholder del campo input
ulterioreTesto	string	Testo di supporto da mostrare sotto il campo
unitaMisura	string	Unità di misura del campo es. "kW"
parent	string	Nome dell'oggetto padre nel request body es. "impianto", "responsabile"
parentDotCampo	string	Percorso completo padre.campo quando il campo è in un sotto-oggetto
proprieta	enum	OBBLIGATORIO — campo visibile e obbligatorio EDITABILE — campo visibile e opzionale SOLA-LETTURA — campo visibile non modificabile NON-VISIBILE — campo non visibile e non accettato nel body
tipoCampo	enum	TESTUALE INTERO DECIMALE CENTESIMALE MILLESIMALE BOOLEANO DATA TIPOLOGICA AREADITESTO OGGETTO
condizioneDescrizione	string	Messaggio di errore da mostrare all'utente se la validazione fallisce

condizioneRegex1/2	string	Espressioni regolari di validazione (regex1 OR regex2 se entrambe presenti)
condizioneLengthMin/Max	integer	Lunghezza minima/massima della stringa
condizioneValueMin/Max	number	Valore numerico minimo/massimo ammesso
condizioneStepSpin	number	Incremento per i campi di tipo SPIN
condizioneDateMinAbs/MaxAbs	date	Data minima/massima assoluta (formato yyyy-MM-dd)
condizioneDateMinDayRelToNow/MaxDayRelToNow	integer	Offset in giorni da oggi (negativo = passato, es. -365 = un anno fa; 0 = oggi)
condizioneSetValues1/OrSetValues2/3	string	Insiemi di valori ammessi separati da virgola. I tre insiemi sono in OR tra loro
condizioneDipendenteDaAltriCampi	boolean	true se la visibilità o obbligatorietà dipende dal valore di altri campi nel body
testoAlert	string	Testo di alert aggiuntivo (opzionale)
testoWarning	string	Testo di warning aggiuntivo (opzionale)

Nota: *proprietà e tipoCampo usano trattino (es. "NON-VISIBILE", "SOLA-LETTURA"), non underscore. tipoCampo non è più limitato a NUMERICO/TENDINA/SPIN: nella versione corrente della vista V_LIB_CAMPI_FE i valori numerici sono distinti in INTERO/DECIMALE/CENTESIMALE/MILLESIMALE e sono comparsi TIPOLOGICA, AREADITESTO, OGGETTO. Questo schema è condiviso con tutte le altre API CAITEL e non è specifico di Accatastamento.*

5.6 ListaValori / ListaValoriWithRequest — risposta tipologiche

Le due strutture hanno esattamente gli stessi campi (lista, numeroElementiLista, numeroTotaleElementi, paginaAttuale, pagineTotali, responseStatus, informazioni, request) e differiscono solo nel nome dello schema in base all'endpoint: ListaValori è usata dagli 8 GET /tipologiche/* senza parametri di percorso, ListaValoriWithRequest dai 2 GET con parametri dinamici in path (comuni-provincia e indirizzi-provincia-comune-*, vedi §3.13 e §3.14).

Campo	Tipo	Descrizione
lista	BasePairString[]	Lista dei valori della tipologica. Ogni elemento ha identificativo (da inserire nel request body) e valore (stringa descrittiva)
numeroElementiLista	integer	Numero di elementi nella lista restituita
numeroTotaleElementi	integer	Numero totale di elementi (utile se paginata)
request	RequestPayloadBase	Echo dei parametri path/query della chiamata (vedi §5.1)

5.7 BasePairString — coppia identificativo/valore

Campo	Tipo	Descrizione
identificativo	string	Codice tecnico da usare nel request body
valore	string	Stringa descrittiva del valore (informativa per il client)
altroValore	string	Valore aggiuntivo (opzionale, dipende dal contesto)
selezionato	boolean	Flag di selezione (opzionale, usato in alcuni contesti)

6. Gestione errori

Tutte le risposte di errore usano Content-Type application/problem+json. La struttura è uniforme per tutti i codici di errore: `responseStatus`, informazioni (con tipo `ERROR` e messaggio), `request` (echo della request originale, vedi §5.1 e §8.1.15), `title`, `status`, `detail`.

6.1 Tabella codici HTTP

Codice HTTP	Titolo	Quando si verifica	Cosa verificare
200 OK		Operazione completata con successo	elemento per il payload, <code>informazioni.messaggio</code> per il testo di conferma
400 Bad Request	BAD_REQUEST	Body mancante, campi obbligatori assenti, formato errato, regexp non rispettata, regole condizionali violate, impianto già presente	<code>informazioni.ulterioriMessaggiConDettagli</code> per la lista dei campi in errore con dettaglio per ciascuno
403 Forbidden	FORBIDDEN	Header codice-fiscale-operatore assente, malformato o operatore non abilitato al servizio	Verificare presenza e formato header. Pattern: <code>^[A-Z]{6}[0-9]{2}[A-Z][0-9]{2}[A-Z][0-9]{3}[A-Z]\$</code>
404 Not Found	NOT_FOUND	Identificativi non trovati nel database: provincia, comune, indirizzo, tipologica inesistente	Verificare che gli identificativi siano stati ottenuti dai servizi tipologiche e non siano stati modificati
500 Internal Server Error	INTERNAL_SERVER_ERROR	Errore interno del sistema (DB non raggiungibile, eccezione non gestita)	Riportare <code>systemMessageCode</code> a supporto per correlazione con i log applicativi

6.2 Struttura risposta 400 — errori di validazione

Quando il body è presente ma contiene errori di validazione, la risposta 400 include la lista dettagliata dei campi in errore:

```
{
  "responseStatus": "400 BAD_REQUEST",
  "informazioni": {
    "tipo": "ERROR",
    "messaggio": "Dati non validi",
    "ulterioriMessaggiConDettagli": [
      {
        "tipo": "ERROR",
        "dettaglio": "Campo obbligatorio",
        "ulterioriDettagli": [
          { "identificativo": "pod", "valore": "POD - Point of Delivery" }
        ]
      }
    ]
  }
}
```



```
    ]
  },
  "request": {
    "corpo": { "impianto": { }, "forniture": { }, "responsabile": { }, "generatori": [
  ] }
  }
}
```

Nota: Il campo *request* (qui abbreviato per brevità) echeggia il body inviato in *request.corpo*: vedi §5.1, §7.7 e §8.1.15.

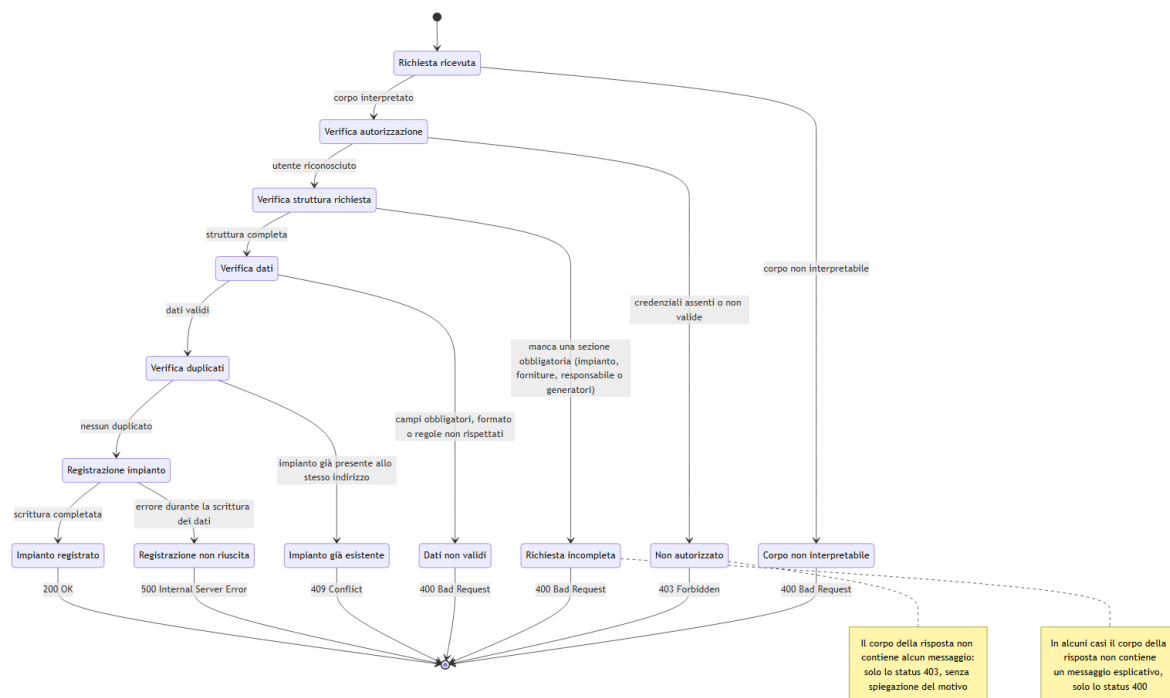
6.3 Struttura risposta 200 — inserimento riuscito

elemento è un oggetto *ImpiantoAccatastato* (vedi §5.2), non una stringa: il codice catastale è nel suo campo *codiceImpianto*.

```
{
  "elemento": {
    "codiceImpianto": "07D9690034567890",
    "identificativoImpianto": "aGVsbG8...",
    "identificativoResponsabile": "d29ybGQ...",
    "identificativoIndirizzo": "13753",
    "esistente": false,
    "descrizioniGruppiOmogenei": ["1 - anno scadenza bollino: 2027"]
  },
  "responseStatus": "200 OK",
  "informazioni": {
    "tipo": "SUCCESS",
    "messaggio": "Impianto 07D9690034567890 inserito con successo!\nAttenzione: per poter compilare il Rapporto Di Efficienza Energetica e necessario validare il libretto.",
    "ulterioriMessaggi": [
      "Il cf RSSMRA80A01F205X ha inserito correttamente l'impianto.",
      "Impianto con codice 07D9690034567890."
    ]
  }
}
```

6.4 Diagramma di stato — POST /nuovo-impianto/inserisci

Il diagramma seguente riassume tutti gli esiti possibili della chiamata di inserimento, dal ricevimento della richiesta alla risposta finale. Ogni transizione verso uno stato terminale corrisponde a uno status HTTP effettivamente restituito.



Stato	Condizione	Esito
Richiesta ricevuta	Corpo non interpretabile (JSON malformato o campo di tipo errato)	400 Bad Request
Verifica autorizzazione	Header codice-fiscale-operatore assente, malformato o non abilitato	403 Forbidden
Verifica struttura richiesta	Manca una sezione obbligatoria (impianto, forniture, responsabile o generatori)	400 Bad Request (*)
Verifica dati	Campi obbligatori, formato o regole condizionali non rispettati	400 Bad Request
Verifica duplicati	Impianto già presente allo stesso indirizzo	409 Conflict
Registrazione impianto	Errore durante la scrittura dei dati	500 Internal Server Error
Impianto registrato	Tutte le verifiche e le scritture completate	200 OK

Nota: (*) In alcuni casi il corpo della risposta di questo specifico stato non contiene un messaggio esplicativo: solo lo status 400, senza dettaglio del campo mancante. Stesso comportamento sullo stato "Verifica autorizzazione" (403). Fix pianificato lato backend.

7. Note tecniche per l'integrazione

7.1 Sequenza di caricamento consigliata

Per minimizzare la latenza percepita dall'utente, si consiglia di eseguire le chiamate di caricamento iniziale prima di raccogliere i dati da inviare:

- tutti i GET /tipologiche/* (10 chiamate)
- tutti i GET /proprietà/accatastamento/* (4 chiamate, opzionali)

7.2 Formati data

Tutti i campi data nel request body usano il formato ISO 8601: AAAA-MM-GG (es. 2025-04-30). Il pattern di validazione è `^\d{4}-\d{2}-\d{2}$`. Il backend potrebbe applicare vincoli aggiuntivi sulle date (min/max) configurati nella vista V_LIB_CAMPI_FE e restituiti da GET /proprietà/accatastamento/*.

7.3 Identificativi tipologiche — regola fondamentale

Gli identificativi restituiti dalle tipologiche sono stringhe numeriche opache (es. "1", "2"). Regole da rispettare obbligatoriamente:

- Non assumere alcun significato posizionale o semantico degli identificativi
- Non usare valori hardcoded: chiamare sempre i servizi /tipologiche/* per popolare le liste
- I valori possono differire tra ambiente di test e ambiente di produzione
- Salvare l'identificativo (non il valore testuale) per il request body

7.4 Gestione POD e PDR

Campo	Obbligatorio	Pattern	Lunghezza	Esempio	Note
pod	SI	<code>^IT[0-9]{3}E[0-9]{8,9}\$</code>	14-15 car.	IT001E12345678	Identificativo punto prelievo elettrico. Presente in bolletta elettrica
pdr	NO	<code>^\d{14}\$</code>	14 cifre	00112233445566	Identificativo punto prelievo gas. Presente in bolletta gas. Omettere se non applicabile

7.5 Autocomplete indirizzi — implementazione consigliata

Per l'autocomplete del campo indirizzo dell'impianto, si consiglia la seguente implementazione:

- Attivare la ricerca con minimo 1 carattere digitato
- Mostrare la lista dei suggerimenti (valore) come dropdown list
- Al completamento salvare sia identificativoIndirizzo (codice) che valoreIndirizzo (stringa)
- Validare che il valore digitato corrisponda a un identificativoIndirizzo valido prima dell'invio



Nota: Il campo di testo per l'indirizzo deve permettere all'utente di vedere e selezionare solo indirizzi presenti nel database. Un indirizzo digitato manualmente senza corrispondenza nel database causerà errore di validazione 400.

7.6 Transazionalità dell'inserimento

Il servizio POST /nuovo-impianto/inserisci è completamente transazionale. Se si verifica un errore in qualsiasi fase dell'inserimento (es. errore DB, violazione vincolo, timeout), l'intera operazione viene annullata e nessun dato parziale rimane sul database. In questo caso la risposta sarà HTTP 500 con `systemMessageCode` per la correlazione con i log server.

7.7 Campo request nelle response — correlazione e debug

Ogni risposta include il campo `request` che rispecchia la chiamata originale. Per le POST (inserisci, verifica-dati): `request.corpo` contiene il body inviato. Per le GET (tipologiche, proprietà): `request.parametri` contiene i path e query parameter come lista di coppie {identificativo, valore}. Utile per debug, correlazione con i log e verifica di cosa è stato effettivamente ricevuto dal server.

7.8 Libretto dell'Impianto — stato post-accatastamento

A seguito di un accatastamento riuscito il sistema crea automaticamente il Libretto dell'Impianto in stato bozza sul database CAITEL. Lo stato bozza implica le seguenti limitazioni:

- Il libretto è visibile agli operatori CAITEL ma non è ancora validato
- Non è possibile compilare il Rapporto di Controllo di Efficienza Energetica (RCEE) finché il libretto è in bozza
- Solo dopo la validazione del libretto si può compilare il RCEE

Il codice impianto restituito nella risposta 200 (campo `elemento.codiceImpianto`) identifica univocamente l'impianto nel sistema CAITEL e deve essere conservato dal sistema client per eventuali future operazioni sull'impianto.

7.9 Verifica stato sistema (Health check del sistema)

Prima di sessioni massive di accatastamento è consigliabile verificare lo stato del sistema con GET /status. Una risposta 200 con `elemento = true` indica che il servizio è operativo e il database è raggiungibile.

7.10 Rappresentare un impianto con un solo generatore/macchina

Il campo "generatori" (§2.8) è sempre una lista, anche quando l'impianto ha un solo gruppo omogeneo con una sola macchina: in quel caso si invia una lista di un solo elemento in "generatori", il cui unico elemento ha a sua volta una lista di un solo elemento in "macchine". La stessa struttura serve anche quando in una singola chiamata occorre accatastare più di un gruppo omogeneo e/o più di una macchina per gruppo — tipicamente per client che importano dati già strutturati per più generatori/macchine da un gestionale terzo, evitando di dover spezzare l'operazione in più chiamate separate.

Nota: Le regole di validazione (obbligatorietà condizionale per tipologia, checkbox frigo, altroFluidoVettore) vengono applicate a ogni generatore e a ogni sua macchina, con il percorso del campo in errore riportato come "generatori[i].identificativoTipoFluidoVettore" oppure "generatori[i].macchine[j].numeroCircuiti" (indici a base 0) in `informazioni.ulterioriMessaggiConDettagli`.

8. Changelog

8.1 v1.0.5 → v1.0.6

Questo changelog confronta lo YAML v1.0.5 inviato alle ditte esterne (Caitel_Accatastamento_v1.0.5.yaml) con lo YAML v1.0.6 finale. La maggior parte degli schemi ha cambiato nome (§8.1.1) mantenendo perlopiù gli stessi campi con lo stesso significato: i nomi nuovi sono più parlanti (es. AccatastamentoRichiesta al posto di NuovImpiantoReqBody) ma NON indicano un nuovo schema, salvo dove esplicitamente segnalato come tale nelle sezioni successive.

8.1.1 Rename degli schemi (stesso significato, nome nuovo)

Nome in v1.0.5	Nome in v1.0.6
NuovImpiantoReqBody	AccatastamentoRichiesta
NuovImpiantoDatiImpianto	ImpiantoDaAccatastare
NuovImpiantoDatiForniture	ForniturePodPdr
NuovImpiantoDatiResponsabile	ResponsabileImpianto
NuovImpiantoDatiGeneratore	GruppoGeneratore (vedi anche §8.1.2: cambia anche il contenuto)
NuovImpiantoRespWithReq	AccatastamentoRisposta
BooleanResp	EsitoVerificaDatiAccatastamentoResponse
BadRequestResp	BadRequestCaitelResponse
ForbiddenResp	ForbiddenCaitelResponse
NotFoundResp	NotFoundCaitelResponse
InternalServerErrorResp	InternalServerErrorCaitelResponse
RequestPayloadNuovImpiantoReqBody	RequestPayloadAccatastamentoRichiesta
ListaValoriWithReq	ListaValoriWithRequest (ora usata solo da comuni-provincia e indirizzi-*, vedi §8.1.6)

8.1.2 Struttura multi-generatore / multi-macchina

In v1.0.5 il campo "generatore" era un oggetto singolo obbligatorio, con i dati tecnici della (unica) macchina diretti sullo schema NuovImpiantoDatiGeneratore e nessun vincolo di formato. In v1.0.6 diventa possibile accatastare più gruppi omogenei e più macchine per gruppo in una sola chiamata (§2.8). Non è un semplice rename: la struttura dati cambia.

Campo / Aspetto	v1.0.5	v1.0.6
generatore	Campo obbligatorio (NuovImpiantoDatiGeneratore): dati tecnici dell'unico generatore/macchina, diretti sullo schema (dataInstallazione, potenza, fabbricante, modello, matricola, campi condizionali per tipologia — nessun pattern di formato)	Rimosso
generatori	Assente	Nuovo campo obbligatorio (array di GruppoGeneratore): sostituisce "generatore" — solo dati di gruppo (identificativoTipoGeneratore, identificativoTipoFluidoVettore, altroFluidoVettore). Un impianto con un

		solo generatore/macchina si rappresenta con una lista di un elemento (§2.8, §7.10)
macchine (su ogni elemento di generatori)	Assente	Nuovo campo (array di MacchinaGeneratore, nuovo schema §4.4): i dati tecnici che in v1.0.5 erano diretti su NuovImpiantoDatiGeneratore (dataInstallazione, potenza, fabbricante, modello, matricola, campi condizionali) si valorizzano ora per ciascuna macchina, con in aggiunta i pattern di formato — dataInstallazione: <code>^\d{4}-\d{2}-\d{2}\$</code> , potenzaTermicaUtileNominalekW: <code>^\d+([.]\d+)?\$</code> , numeroModuli e numeroCircuiti nessun pattern rigido
description / x-form-rules (schema GruppoGeneratore)	Assente	Aggiunta tabella e relativa estensione x-form-rules con le regole condizionali per tipologia (1–7): campi obbligatori e non applicabili, applicate ai campi di ciascun elemento di "macchine"

8.1.3 Schema ImpiantoDaAccatastare (ex NuovImpiantoDatiImpianto)

Campo	v1.0.5	v1.0.6
identificativoTipoIntervento	Nessun enum	Aggiunto enum: [1, 2, 3, 4] + valori in description
dataIntervento	type: string	Aggiunto pattern: <code>^\d{4}-\d{2}-\d{2}\$</code>
identificativoProvincia	Nessun enum (qualunque provincia italiana)	Aggiunto enum: [008, 009, 010, 011] — solo le 4 province liguri
identificativoColoreCivico	type: string libero	Aggiunto enum: [N, R]
letteraOpalazzo	maxLength: 4	minLength: 1, maxLength: 20
scala	Nessun vincolo lunghezza	maxLength: 20
interno	Nessun vincolo lunghezza	maxLength: 20
internoLettera	Nessun vincolo lunghezza	maxLength: 1

8.1.4 Schema GruppoGeneratore / MacchinaGeneratore (ex NuovImpiantoDatiGeneratore)

Campo	v1.0.5	v1.0.6
identificativoTipoGeneratore (GruppoGeneratore)	Nessun enum	Aggiunto enum: [1..7] — solo i tipi compatibili con nuovo impianto
identificativoTipoFluidoVettore (GruppoGeneratore)	Nessun enum	Aggiunto enum: [1, 2, 3]
identificativoTipoMacchinaTermica (MacchinaGeneratore)	Nessun enum	Aggiunto enum: [1, 2, 3, 4]
identificativoTipoGeneratore di Calore (MacchinaGeneratore)	Nessun enum	Aggiunto enum: [1, 2, 3, 4]
identificativoIsMacchina a Biomassa (MacchinaGeneratore)	Nessun enum, example errato "5"	Aggiunto enum: [SI, NO], example corretto in "SI"

8.1.5 Schema ResponsabileImpianto (ex NuovImpiantoDatiResponsabile)

Campo / Aspetto	v1.0.5	v1.0.6
required	cognome, nome, codiceFiscale sempre inclusi	cognome, nome, codiceFiscale rimossi dai required (obbligatorietà condizionale per combinazione, vedi §2.5) — restano sempre obbligatori solo

		identificativoTipoResponsabilita, identificativoComune, indirizzo, civico, email, telefono
codiceFiscale	Nessun vincolo di lunghezza o pattern	minLength: 11 maxLength: 16 pattern union persona fisica (16 caratteri) / ente (11 cifre): <code>^([A-Za-z]{6}[0-9]{2}[A-Za-z][0-9]{2}[A- Za-z][0-9]{3}[A-Za-z] [0-9]{11})\$</code>
partitalva	Nessun vincolo di lunghezza o pattern example: IT01234567890	pattern: <code>^((IT)?\d{11})?\$</code> — 11 cifre con prefisso IT opzionale
telefono	Nessun vincolo di lunghezza o pattern	minLength: 8 maxLength: 250 pattern: <code>^(?=[0-9])[0-9._e+\\-\\/\\,;]{8,250}\$</code>

8.1.6 Endpoint

Endpoint	v1.0.5	v1.0.6
POST /nuovo-impianto/verifica- dati	Body dichiarato come parametro query (in: query, required: true) — non conforme alla semantica OpenAPI per un body JSON	Corretto: body spostato in requestBody (application/json), allineato all'endpoint inserisci
GET /tipologiche/comuni- provincia/{...} e /tipologiche/indirizzi-provincia- comune-*/...	Nome del path param dichiarato (es. "id-provincia") non corrispondeva al placeholder nel path stesso ("{identificativo-provincia}") — mismatch OpenAPI	Corretto: nome del parametro allineato al path, aggiunte description ed example per ogni parametro
GET /tipologiche/* (esempi)	Nessun example nelle response	Aggiunti esempi di risposta per tipi- responsabilita, tipi-colore-civico, province-liguria

8.1.7 Schema ResponseMessage

Campo	v1.0.5	v1.0.6
systemMessageCode	Assente	Nuovo campo: codice interno per correlazione log/risposta
stringaUnicaHtml	Assente	Nuovo campo: aggregato HTML di tutti i messaggi
stringaUnicaHtmlCompleta	Assente	Nuovo campo: variante completa dell'aggregato HTML
confirmationToken	Assente	Nuovo campo (schema condiviso con tutte le API CAITEL): mai valorizzato nelle risposte di Accatastamento
tipo (enum)	UNDEF, INFO, WARNING, ERROR, SUCCESS	Aggiunto valore QUESTION

8.1.8 Schema ResponseMessageDetails

Campo	v1.0.5	v1.0.6
tipo	Assente	Nuovo campo: enum ERROR / WARNING / INFO / SUCCESS / UNDEF / QUESTION
conclusione	Assente	Nuovo campo: testo conclusivo del dettaglio
domanda	Assente	Nuovo campo: domanda per conferma operativa
possibiliRisposteDomanda	Assente	Nuovo campo: lista BasePairString — opzioni di risposta

8.1.9 Schema ProprietaResp

Schema condiviso con tutte le API CAITEL (non specifico di Accatastamento), che nel frattempo è evoluto indipendentemente da questa API.

Campo	v1.0.5	v1.0.6
proprieta	type: string (libero)	Aggiunto enum: OBBLIGATORIO, EDITABILE, SOLA-LETTURA, NON-VISIBILE (trattino, non underscore)
tipoCampo	type: string (libero)	Aggiunto enum: TESTUALE, INTERO, DECIMALE, CENTESIMALE, MILLESIMALE, BOOLEANO, DATA, TIPOLOGICA, AREADITESTO, OGGETTO
parentDotCampo, testoAlert, testoWarning	Assenti	Nuovi campi
condizioneValueMin/Max	type: integer	type: number
condizioneDateMinAbs condizioneDateMaxAbs	format: date-time	format: date

8.1.10 Parametro CodiceFiscaleOperatore (header)

Attributo	v1.0.5	v1.0.6
pattern	^[a-zA-Z0-9]{16,}\$ (generico, qualunque alfanumerico di almeno 16 caratteri)	^[A-Za-z]{6}[0-9LMNPQRSTUVWXYZlmnpqrstuv]{2}[A-EHLMPRStA-ehlmprst][0-9LMNPQRSTUVWXYZlmnpqrstuv]{2}[A-Za-z][0-9LMNPQRSTUVWXYZlmnpqrstuv]{3}[A-Za-z]\$ (vero formato CF persona fisica italiano, incluse le lettere sostitutive per omocodia)

8.1.11 Sicurezza — schema BearerAuth

Formalizzato lo schema di sicurezza già richiesto nei fatti dal gateway WSO2 (§1.5): security globale con BearerAuth (http, scheme bearer, bearerFormat JWT), assente in v1.0.5. Aggiunto anche il campo x-summary nel blocco info, assente in v1.0.5.

8.1.12 Schema aggiunto: ImpiantoAccatastato

Nuovo schema (§5.2) che descrive correttamente il campo "elemento" della risposta di POST /nuovo-impianto/inserisci: codicImpianto, identificativoImpianto, identificativoResponsabile, identificativoIndirizzo, esistente, descrizioniGruppiOmogenei.

Nota: Il campo booleano è "esistente", non "isEsistente": il bean espone il getter isEsistente(), che Jackson serializza togliendo il prefisso "is" (comportamento standard per i getter booleani). Chi integra deve leggere la chiave JSON "esistente".

8.1.13 Schema AccatastamentoRisposta (ex NuovImpiantoRespWithReq) — correzione

Campo	v1.0.5	v1.0.6
elemento	type: string — non corrispondeva al comportamento reale del backend: la risposta effettiva è sempre stata un oggetto, mai una stringa nuda	\$ref: ImpiantoAccatastato (vedi §5.2 e §8.1.12). Correzione di un difetto pre-esistente dello YAML, non legata al nuovo flusso multi-generatore

8.1.14 Eco della request (campo "request.corpo") — correzione

In v1.0.5 il campo request.corpo di ogni risposta con eco del body (es. AccatastamentoRisposta.request.corpo, EsitoVerificaDatiAccatastamentoResponse.request.corpo) è documentato con lo schema reale della request inviata

(es. `NuovoImpiantoReqBody`). In una revisione intermedia di questa API lo schema generico condiviso da tutte le API CAITEL con eco della request aveva perso questa capacità (il campo risultava sempre tipizzato genericamente, senza elencare i campi reali). Il difetto è stato corretto per la versione 1.0.6 finale, ripristinando l'eco tipizzata corretta (es. `RequestPayloadAccatastamentoRichiesta.corpo` → `AccatastamentoRichiesta`) su questa e su tutte le altre API CAITEL che condividono lo stesso meccanismo.

8.1.15 Campo "request" nelle risposte di errore (400/403/404/500) — correzione

Come per tutte le risposte di successo (§5.1), anche le quattro risposte di errore standard — `BadRequestCaitelResponse`, `ForbiddenCaitelResponse`, `NotFoundCaitelResponse`, `InternalServerErrorCaitelResponse` — includono ora il campo `request` (echo della request originale: corpo per le POST, parametri per le GET — vedi §5.1, §6.2 e §7.7). Il campo era assente in una revisione intermedia di questa API (correzione condivisa con tutte le API CAITEL, analoga a quella di §8.1.14) ed è stato ripristinato per allineare il comportamento delle risposte di errore a quello delle risposte di successo.