

API Reference

Catasto

API Version: 3.0.0

This API provides structured access to official Italian real estate registry information related to properties and subjects (individuals and legal entities).

****Dati Catastali****

The API allows consultation of the Italian Cadastre to retrieve detailed information about buildings and land plots and their registered holders. It is possible to:

****Visure Catastali****

****Visure Ipotecarie****

****Estratto Mappa****

****Elaborato Planimetrico****

- consult cadastral data for a specific property
- identify the registered holders of a property
- retrieve the list of properties owned by a physical or legal subject
- obtain detailed descriptions of properties associated with a subject
- determine the total number of land plots and buildings owned by a subject
- retrieve the list of properties located on a specific address
- The API includes the generation of the cadastral view PDF (Visura Catastale) for both properties and subjects.
- The API includes access to land registry searches (visure ipotecarie), which provide information about registry entries (notes/formalities) associated with properties or subjects, as recorded by the Italian Land Registry Offices (Conservatorie).
- Through land registry search services it is possible to:
- identify the Land Registry Offices where a subject has
- regist entries (notes/formalities)
- retrieve the list of registry entries for a subject or property
- within a specific Land Registry Office
- obtain the detailed information of a single registry entry
- Through this service you can get the PDF of the map extract from the land registry
- Through this service you can get the PDF of the cadastral building plan

CONTACT

NAME: Support

URL: <https://openapi.it/en/support>

Terms of service: <https://openapi.it/en/terms-and-conditions>

INDEX

1. ELABORATO PLANIMETRICO	4
1.1 GET /elaborato_planimetrico	4
1.2 POST /elaborato_planimetrico	5
1.3 GET /elaborato_planimetrico/{id}	8
1.4 GET /elaborato_planimetrico/{id}/documento	10
2. ESTRATTO MAPPA	12
2.1 GET /estratto_mappa	12
2.2 POST /estratto_mappa	13
2.3 GET /estratto_mappa/{id}	16
2.4 GET /estratto_mappa/{id}/documento	18
3. INDIRIZZO	19
3.1 GET /indirizzo	19
3.2 POST /indirizzo	20
3.3 GET /indirizzo/{id}	22
4. RICHIESTA	25
4.1 POST /richiesta/{endpoint}/	25
4.2 GET /richiesta	33
4.3 GET /richiesta/{id}	33
5. TERRITORIO	42
5.1 GET /territorio	42
5.2 GET /territorio/{provincia_or_nome_provincia_or_id}	42
5.3 GET /territorio/{provincia_or_nome_provincia_or_id}/{comune}	43
5.4 GET /territorio/conservatorie	44
5.5 GET /territorio/conservatorie/{conservatoria_or_id}	44
6. VISURA CATASTALE	46
6.1 GET /visura_catastale	46
6.2 POST /visura_catastale	46
6.3 GET /visura_catastale/{id}	52
6.4 GET /visura_catastale/{id}/documento	54
7. VISURA IPOTECARIA	56
7.1 POST /ipotecarie-ispezione_nazionale	56
7.2 POST /ipotecarie-elenco-note	58
7.3 POST /ipotecarie-dettaglio-nota	60
7.4 GET /ipotecarie	65
7.5 GET /ipotecarie/{id}	66
7.6 GET /ipotecarie/{id}/documento	70

Security and Authentication

SECURITY SCHEMES

KEY	TYPE	DESCRIPTION
bearerAuth	http, bearer	

API

1. ELABORATO PLANIMETRICO

Methods that allow you to obtain a cadastral building plan

1.1 GET /elaborato_planimetrico

Extracts all requests

This method allows you to see all the requests of cadastral building plan

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data [{
    id string
    endpoint string
    stato enum ALLOWED:in_erogazione,
    in_lavorazione, evasa
    callback
    ONE OF
    OPTION 1 {
      method enum DEFAULT:JSON
      ALLOWED:POST, JSON
      field string DEFAULT:data
      url* string
      retry integer between 0 and 5
      headers {
        key string
      }
      custom {
    }
```

Array of object:

method enum **DEFAULT:JSON**
ALLOWED:POST, JSON
Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.

field string **DEFAULT:data**
This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.

url* string
A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.

retry integer **between 0 and 5**
DEFAULT:0
The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

headers {
A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.
key string
}

custom {
This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant

information that you need for your application.

```
}
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST
  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
}
parametri {
  foglio
    ONE OF
    prop0
    string
    prop1
    integer
  particella
    ONE OF
    prop0
    string
    prop1
    integer
  sezione_urbana
    ONE OF
    prop0
    string
    prop1
    integer
  comune      string
  sezione    string
  provincia    string
}
documento      string
esito          string
timestamp      integer
owner          string
}]
success        boolean
message        string
error          integer
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

1.2 POST /elaborato_planimetrico

Create a cadastral building plan request

Creates a cadastral building plan request.

Check status with GET /elaborato_planimetrico/{id} and download the PDF with GET /elaborato_planimetrico/{id}/documento when available.

REQUEST

REQUEST BODY - application/json

```
{
  provincia*           string
  comune*             string
  sezione            string
  sezione_urbana     string
  foglio*
    ONE OF
    prop0
      string
    prop1
      integer
  particella*
    ONE OF
    prop0
      string
    prop1
      integer
  callback
    ONE OF
    OPTION 1 {
      method  enum    DEFAULT:JSON
                        ALLOWED:POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                        Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                        value pair named data (customizable using the field parameter) within the request body. If the method is set to
                        'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw
                        JSON data will be directly inserted into the request body without any encoding.

      field    string  DEFAULT:data
                        This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it
                        determines the name of the parameter that will be sent in the POST request.

      url*     string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring
                        authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum
                        timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.

      retry    integer  between 0 and 5
                        DEFAULT:0
                        The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is
                        reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval
                        is often introduced between attempts to avoid overloading the server. This interval may increase exponentially
                        with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the
                        request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or
                        network issues.

      headers {
        A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various
        purposes, including authentication, content negotiation, and custom metadata.

        key string
      }
      custom {
        This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the
        'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
        application.
      }
    }
}
OPTION 2 {
  method  enum    ALLOWED:POST, JSON
                        Indicate through which method you want to receive data with the callback, by default it is POST

  field    string  Name of the field containing the callback return data
  url      string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
```

```

    }
  }
}

```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```

{
  data {
    id          string
    endpoint    string
    stato       enum      ALLOWED:in_erogazione,
                           in_lavorazione, evasa

    callback
      ONE OF
      OPTION 1{
        method  enum      DEFAULT:JSON
                           ALLOWED:POST, JSON
                           Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated
                           with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be
                           encoded into a key-value pair named data (customizable using the field parameter) within the
                           request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the
                           Content-Type header set to application/json. The raw JSON data will be directly inserted into the
                           request body without any encoding.

        field   string    DEFAULT:data
                           This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                           string, it determines the name of the parameter that will be sent in the POST request.

        url*    string    A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                           requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'.
                           The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a
                           200 status code is received.

        retry   integer    between 0 and 5
                           DEFAULT:0
                           The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                           attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                           failure. A time interval is often introduced between attempts to avoid overloading the server. This
                           interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                           The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                           those indicating temporary errors (e.g., 502, 503) or network issues.

        headers {
          A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
          various purposes, including authentication, content negotiation, and custom metadata.

          key string
        }
        custom {
          This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
          within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
          need for your application.
        }
      }
    }
  }
  OPTION 2{
    method  enum      ALLOWED:POST, JSON
                           Indicate through which method you want to receive data with the callback, by default it is POST

    field   string    Name of the field containing the callback return data
    url     string    Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
}
parametri {

```

```

    foglio
      ONE OF
      prop0
        string
      prop1
        integer
    particella
      ONE OF
      prop0
        string
      prop1
        integer
    sezione_urbana
      ONE OF
      prop0
        string
      prop1
        integer
    comune      string
    sezione     string
    provincia     string
  }
  documento      string
  esito           string
  timestamp       integer
  owner           string
}
success          boolean
message          string
error            integer
}

```

STATUS CODE - 402: Payment Required

RESPONSE MODEL - application/json

STATUS CODE - 406: Not Acceptable

RESPONSE MODEL - application/json

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

1.3 GET /elaborato_planimetrico/{id}

Detail of the cadastral building plan

Allows you to see the detail of the cadastral building plan identified by the id

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	cadastral building plan id

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    id          string
    endpoint    string
    stato       enum          ALLOWED:in_erogazione,
                              in_lavorazione, evasa

    callback
      ONE OF
      OPTION 1{
        method  enum          DEFAULT:JSON
                              ALLOWED:POST, JSON
                              Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with
                              the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded
                              into a key-value pair named data (customizable using the field parameter) within the request body. If the
                              method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to
                              application/json. The raw JSON data will be directly inserted into the request body without any
                              encoding.

        field    string        DEFAULT:data
                              This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                              string, it determines the name of the parameter that will be sent in the POST request.

        url*     string        A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                              requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The
                              maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200
                              status code is received.

        retry    integer        between 0 and 5
                              DEFAULT:0
                              The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                              attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                              failure. A time interval is often introduced between attempts to avoid overloading the server. This
                              interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                              The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                              those indicating temporary errors (e.g., 502, 503) or network issues.

        headers {
          A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
          various purposes, including authentication, content negotiation, and custom metadata.

          key  string
        }
        custom {
          This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
          within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
          need for your application.
        }
      }
    }
  }
  OPTION 2{
    method  enum          ALLOWED:POST, JSON
                              Indicate through which method you want to receive data with the callback, by default it is POST

    field    string        Name of the field containing the callback return data
    url      string        Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
  parametri {
    foglio
      ONE OF
      prop0
      string
      prop1
      integer
  }
```

```

    particella
      ONE OF
      prop0
        string
      prop1
        integer
    sezione_urbana
      ONE OF
      prop0
        string
      prop1
        integer
    comune      string
    sezione     string
    provincia     string
  }
  documento      string
  esito           string
  timestamp       integer
  owner           string
}
success          boolean
message          string
error            integer
}

```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

1.4 GET /elaborato_planimetrico/{id}/documento

Download cadastral building plan

This method allows you to directly download the document

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	cadastral building plan id

RESPONSE

STATUS CODE - 200: Content of the file

RESPONSE MODEL - application/PDF

string

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 422: Unprocessable Entity

2. ESTRATTO MAPPA

Methods that allow you to obtain a map extract

2.1 GET /estratto_mappa

Extracts all requests

This method allows you to see all the requests of extract map you have made

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data [{
    id                string
    endpoint          string
    stato             enum          ALLOWED:in_erogazione,
                                in_lavorazione, evasa

    callback
      ONE OF
      OPTION 1 {
        method enum    DEFAULT:JSON
                                ALLOWED:POST, JSON
                                Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with
                                the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded
                                into a key-value pair named data (customizable using the field parameter) within the request body. If the
                                method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to
                                application/json. The raw JSON data will be directly inserted into the request body without any
                                encoding.

        field string    DEFAULT:data
                                This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                                string, it determines the name of the parameter that will be sent in the POST request.

        url* string     A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                                requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The
                                maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200
                                status code is received.

        retry integer   between 0 and 5
                                DEFAULT:0
                                The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                                attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                                failure. A time interval is often introduced between attempts to avoid overloading the server. This
                                interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                                The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                                those indicating temporary errors (e.g., 502, 503) or network issues.

        headers {
          A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
          various purposes, including authentication, content negotiation, and custom metadata.

          key string
        }
        custom {
          This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
          within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
          need for your application.
        }
      }
    }
  ]
}
```

```

    }
    OPTION 2{
        method enum    ALLOWED:POST, JSON
                        Indicate through which method you want to receive data with the callback, by default it is POST
        field  string   Name of the field containing the callback return data
        url    string   Url where the callback call will be made
        data {
            Object to be completed at your convenience with additional data that you can use back with the callback
        }
    }
}
parametri {
    foglio
    ONE OF
    prop0
    string
    prop1
    integer
    particella
    ONE OF
    prop0
    string
    prop1
    integer
    comune    string
    sezione  string
    provincia  string
}
documento    string
esito        string
timestamp    integer
owner        string
}]
success      boolean
message      string
error        integer
}

```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

2.2 POST /estratto_mappa

Create a map extract request

Create a request to obtain a map extract for the land registry (Catasto Terreni).

The section parameter is mandatory for municipalities that require it. You can verify if a section is required using the endpoint: GET /territorio/{provincia}/{comune}?tipo_catasto=t.

The request returns a request id. Use this ID to monitor the status via GET /estratto_mappa/{id} and to download the PDF document via GET /estratto_mappa/{id}/documento once the status is "evasa".

REQUEST

REQUEST BODY - application/json

```

{
    provincia*          string

```

```

comune*                string
sezione                string
foglio*
  ONE OF
  prop0
    string
  prop1
    integer
particella*
  ONE OF
  prop0
    string
  prop1
    integer
callback
  ONE OF
  OPTION 1{
    method  enum    DEFAULT:JSON
                  ALLOWED:POST,  JSON
                  Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                  Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                  value pair named data (customizable using the field parameter) within the request body. If the method is set to
                  'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw
                  JSON data will be directly inserted into the request body without any encoding.

    field    string  DEFAULT:data
                  This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it
                  determines the name of the parameter that will be sent in the POST request.

    url*     string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring
                  authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum
                  timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.

    retry    integer  between 0 and 5
                  DEFAULT:0
                  The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is
                  reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval
                  is often introduced between attempts to avoid overloading the server. This interval may increase exponentially
                  with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the
                  request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or
                  network issues.

  headers {
    A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various
    purposes, including authentication, content negotiation, and custom metadata.

    key  string
  }
  custom {
    This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the
    'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
    application.
  }
}
OPTION 2{
  method  enum    ALLOWED:POST,  JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST

  field    string  Name of the field containing the callback return data
  url      string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
}

```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    id          string
    endpoint    string
    stato       enum          ALLOWED:in_erogazione,
                                in_lavorazione, evasa

    callback
      ONE OF
      OPTION 1{
        method  enum          DEFAULT:JSON
                                ALLOWED:POST, JSON
                                Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated
                                with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be
                                encoded into a key-value pair named data (customizable using the field parameter) within the
                                request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the
                                Content-Type header set to application/json. The raw JSON data will be directly inserted into the
                                request body without any encoding.

        field   string        DEFAULT:data
                                This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                                string, it determines the name of the parameter that will be sent in the POST request.

        url*    string        A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                                requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'.
                                The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a
                                200 status code is received.

        retry   integer        between 0 and 5
                                DEFAULT:0
                                The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                                attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                                failure. A time interval is often introduced between attempts to avoid overloading the server. This
                                interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                                The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                                those indicating temporary errors (e.g., 502, 503) or network issues.

        headers {
          A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
          various purposes, including authentication, content negotiation, and custom metadata.

          key string
        }
        custom {
          This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
          within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
          need for your application.
        }
      }
    }
  }
  OPTION 2{
    method  enum          ALLOWED:POST, JSON
                                Indicate through which method you want to receive data with the callback, by default it is POST

    field   string        Name of the field containing the callback return data
    url     string        Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
  parametri {
    foglio
      ONE OF
      prop0
      string
      prop1
      integer
  }
```

```

    particella
      ONE OF
      prop0
        string
      prop1
        integer
    comune      string
    sezione    string
    provincia    string
  }
  documento      string
  esito           string
  timestamp      integer
  owner          string
}
success          boolean
message          string
error            integer
}

```

STATUS CODE - 402: Payment Required

RESPONSE MODEL - application/json

STATUS CODE - 406: Not Acceptable

RESPONSE MODEL - application/json

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

2.3 GET /estratto_mappa/{id}

Detail of the extract map

Allows you to see the detail of the extract map identified by the id

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	Extract Map id

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```

{
  data {
    id          string
    endpoint    string
    stato       enum
                ALLOWED:in_erogazione,
                in_lavorazione, evasa
    callback
      ONE OF
      OPTION 1 {

```



```

method enum    DEFAULT:JSON
                ALLOWED:POST, JSON
                Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with
                the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded
                into a key-value pair named data (customizable using the field parameter) within the request body. If the
                method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to
                application/json. The raw JSON data will be directly inserted into the request body without any
                encoding.

field string   DEFAULT:data
                This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                string, it determines the name of the parameter that will be sent in the POST request.

url* string    A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The
                maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200
                status code is received.

retry integer  between 0 and 5
                DEFAULT:0
                The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                failure. A time interval is often introduced between attempts to avoid overloading the server. This
                interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                those indicating temporary errors (e.g., 502, 503) or network issues.

headers {
  A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
  various purposes, including authentication, content negotiation, and custom metadata.
  key string
}
custom {
  This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
  within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
  need for your application.
}
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST

  field string   Name of the field containing the callback return data
  url string     Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
}
parametri {
  foglio
    ONE OF
    prop0
    string
    prop1
    integer
  particella
    ONE OF
    prop0
    string
    prop1
    integer
  comune string
  sezione string
  provincia string
}
documento string
esito string

```

```

    timestamp
    owner
  }
  success
  message
  error
}
integer
string
boolean
string
integer

```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

2.4 GET /estratto_mappa/{id}/documento

Download map

This method allows you to directly download the document

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	Extract map id

RESPONSE

STATUS CODE - 200: Content of the file

RESPONSE MODEL - application/PDF

```
string
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 422: Unprocessable Entity

RESPONSE MODEL - application/json

3. INDIRIZZO

Method that allows you to search for an address

3.1 GET /indirizzo

Search for an address

This endpoint allows two functions:

- Synchronous search: Provide the query parameters (province, municipality, section, address) to obtain the corresponding id_indirizzo, which can be used in POST/richiesta/indirizzo.
- Request list: If you do not provide any query parameters, you will get a list of id_indirizzo from your previous POST/indirizzo requests.

REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
provincia	string	Province (**Required for synchronous search**)
comune	string	City (**Required for synchronous search**)
sezione	string	Section of the city if it is present
indirizzo	string	Address searched; enter only the address and not the toponym (**Required for synchronous search**)

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

ONE OF

OPTION 1 {

data {

indirizzi [{

Array of object:

id_indirizzo string

indirizzo string

}]

}

success boolean

message string

error integer

}

OPTION 2 {

data [{

Array of object:

id string

endpoint string

indirizzo string

stato string

timestamp integer

owner string

}]

success boolean

```

    message    string
    error      integer
}

```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

STATUS CODE - 406: Not Acceptable

RESPONSE MODEL - application/json

STATUS CODE - 412: Precondition Failed

RESPONSE MODEL - application/json

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

3.2 POST /indirizzo

Create address search request

This method allows you to create a request to obtain an id_indirizzo, which will subsequently be used in the POST /richiesta/indirizzo endpoint.

REQUEST

REQUEST BODY - application/json

```

{
  provincia*      string
  comune*        string
  sezione       string
  indirizzo*      string
  callback

```

ONE OF

OPTION 1 {

```

  method  enum    DEFAULT:JSON
              ALLOWED:POST, JSON

```

Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.

```

  field    string  DEFAULT:data

```

This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.

```

  url*     string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.

```

```

  retry    integer  between 0 and 5

```

DEFAULT:0

The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

```

  headers {

```

A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.

```

    key string
  }
  custom {

```

This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your application.

```
}
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
                                Indicate through which method you want to receive data with the callback, by default it is POST
  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
}
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    endpoint string
    stato    string
    callback
    ONE OF
    OPTION 1{
      method enum    DEFAULT:JSON
                                ALLOWED:POST, JSON
                                Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated
                                with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be
                                encoded into a key-value pair named data (customizable using the field parameter) within the
                                request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the
                                Content-Type header set to application/json. The raw JSON data will be directly inserted into the
                                request body without any encoding.
      field  string  DEFAULT:data
                                This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                                string, it determines the name of the parameter that will be sent in the POST request.
      url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                                requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'.
                                The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a
                                200 status code is received.
      retry  integer  between 0 and 5
                                DEFAULT:0
                                The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                                attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                                failure. A time interval is often introduced between attempts to avoid overloading the server. This
                                interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                                The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                                those indicating temporary errors (e.g., 502, 503) or network issues.
    }
    headers {
      A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
      various purposes, including authentication, content negotiation, and custom metadata.
      key string
    }
    custom {
      This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
      within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
      need for your application.
    }
  }
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
```

```

    field      string      Indicate through which method you want to receive data with the callback, by default it is POST
    url        string      Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
  parametri {
    provincia string
    comune    string
    sezione  string
    indirizzo  string
  }
  risultato {
    Always null in POST responses.
  }
  esito              string
  timestamp          integer
  owner              string
  id                  string
}
success              boolean
message              string
error                integer
}
```

STATUS CODE - 406: Not Acceptable

RESPONSE MODEL - application/json

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

3.3 GET /indirizzo/{id}

Detail of the address

Allows you to see the detail of the id_indirizzo identified by the id

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	Request id

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    data {
      endpoint      string
      stato         string
      callback
    }
  }
}
```

```

ONE OF
OPTION 1 {
    method enum    DEFAULT:JSON
                  ALLOWED:POST, JSON
                  Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated
                  with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be
                  encoded into a key-value pair named data (customizable using the field parameter) within the
                  request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the
                  Content-Type header set to application/json. The raw JSON data will be directly inserted into the
                  request body without any encoding.

    field string   DEFAULT:data
                  This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as
                  a string, it determines the name of the parameter that will be sent in the POST request.

    url* string    A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                  requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer
                  auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times
                  until a 200 status code is received.

    retry integer  between 0 and 5
                  DEFAULT:0
                  The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                  attempts is reached without a successful status code (e.g., 200 OK), the request is considered a
                  final failure. A time interval is often introduced between attempts to avoid overloading the server.
                  This interval may increase exponentially with each attempt to reduce the impact on an overloaded
                  server. The 'retry' parameter is configured to retry the request only in case of specific status codes,
                  typically those indicating temporary errors (e.g., 502, 503) or network issues.

    headers {
        A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
        various purposes, including authentication, content negotiation, and custom metadata.

        key string
    }
    custom {
        This object can be populated with any additional data you wish to include in the callback response. This data will be
        accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant
        information that you need for your application.
    }
}
OPTION 2 {
    method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST

    field string   Name of the field containing the callback return data
    url string     Url where the callback call will be made
    data {
        Object to be completed at your convenience with additional data that you can use back with the callback
    }
}
parametri {
    provincia string
    comune string
    sezione string
    indirizzo string
}
risultato {
    indirizzi [{
        Array of object:
        id_indirizzo string
        indirizzo string
    }]
}
esito string
timestamp integer
owner string
id string
}

```

success	boolean
message	string
error	integer
}	
success	boolean
message	string
error	integer
}	

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

4. RICHIESTA

Methods that allow you to create and view a request for information regarding buildings or land to the land registry

4.1 POST /richiesta/{endpoint}/

Creates a request

This method allows you to create a request to receive information on subjects and properties in the Italian land registry.

- **elenco_immobili** – returns detailed information for each property (immobile) within the specified foglio and particella. Note: the **sezione** field is required only in municipalities that have sections; **sezione_urbana** is never required.

- **prospetto_catastale** – returns detailed information for each property (immobile) within the specified foglio, particella, and subalterno, including ownership (intestatari) details. Note: the **sezione** field is required only in municipalities that have sections; **sezione_urbana** is never required.

- **ricerca_persona** – returns basic information about the owner, along with the list of properties owned by that person or entity within the specified province.

- **ricerca_nazionale** – Researches the properties owned by a subject across the entire national territory. The result returns an aggregated overview by province and municipality, showing the number of buildings and land owned, without returning full property details.

- **indirizzo** – returns the list of properties (with full details) associated with a given address identifier (**id_indirizzo**) obtained via POST/**indirizzo** (asynchronous) or GET /**indirizzo** (synchronous).
In some cases, for the address to return valid results, it is necessary to also specify the street number (civic number) together with the address. If the civic number is missing, sometimes the system may not be able to correctly associate any property to the given address.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*endpoint	enum ALLOWED: elenco_immobili, prospetto_catastale, ricerca_persona, ricerca_nazionale, indirizzo	Type of service requested

REQUEST BODY - application/json

ONE OF

OPTION 1{

callback

ONE OF

OPTION 1{

method enum

DEFAULT:JSON

ALLOWED:POST, JSON

Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.

field string

DEFAULT:data

This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.

url* string

A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum

timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.

retry integer between 0 and 5

DEFAULT:0

The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

headers {

A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.

key string

}

custom {

This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your application.

}

}

OPTION 2{

method enum ALLOWED:POST, JSON

Indicate through which method you want to receive data with the callback, by default it is POST

field string Name of the field containing the callback return data

url string Url where the callback call will be made

data {

Object to be completed at your convenience with additional data that you can use back with the callback

}

}

tipo_catasto*

enum

ALLOWED:T, F, TF

- **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where supported by the endpoint)

provincia*

string

Province code (2 letters).

comune*

string

Municipality name in uppercase.

sezione

string

Required only if the municipality exposes sections.

sezione_urbana

string

Never required; include only if available.

foglio*

ONE OF

prop0

string

prop1

integer

particella*

ONE OF

prop0

string

prop1

integer

}

OPTION 2{

callback

ONE OF

OPTION 1{

method enum

DEFAULT:JSON

ALLOWED:POST, JSON

Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.

field	string	DEFAULT: data
		This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.
url*	string	A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.
retry	integer	between 0 and 5 DEFAULT: 0
		The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.
headers {		
A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.		
	key string	
	}	
	custom {	
		This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your application.
	}	
	}	
OPTION 2{		
method	enum	ALLOWED: POST, JSON
		Indicate through which method you want to receive data with the callback, by default it is POST
field	string	Name of the field containing the callback return data
url	string	Url where the callback call will be made
data {		
		Object to be completed at your convenience with additional data that you can use back with the callback
	}	
	}	
tipo_catasto*	enum	ALLOWED: T, F, TF
		- **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where supported by the endpoint)
provincia*	string	Province code (2 letters).
comune*	string	Municipality name in uppercase.
sezione	string	Required only if the municipality exposes sections.
sezione_urbana	string	Never required; optional.
foglio*		
	ONE OF	
prop0	string	
prop1	integer	
particella*		
	ONE OF	
prop0	string	
prop1	integer	
subalterno		
	ONE OF	
prop0	string	
prop1	integer	

```

}
OPTION 3{
  callback
    ONE OF
    OPTION 1{
      method enum      DEFAULT:JSON
                        ALLOWED:POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                        Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                        value pair named data (customizable using the field parameter) within the request body. If the method is set
                        to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json.
                        The raw JSON data will be directly inserted into the request body without any encoding.

      field  string     DEFAULT:data
                        This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it
                        determines the name of the parameter that will be sent in the POST request.

      url*   string     A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring
                        authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum
                        timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is
                        received.

      retry  integer    between 0 and 5
                        DEFAULT:0
                        The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is
                        reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time
                        interval is often introduced between attempts to avoid overloading the server. This interval may increase
                        exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is
                        configured to retry the request only in case of specific status codes, typically those indicating temporary
                        errors (e.g., 502, 503) or network issues.

      headers {
        A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various
        purposes, including authentication, content negotiation, and custom metadata.
        key string
      }
      custom {
        This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within
        the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
        application.
      }
    }
  }
  OPTION 2{
    method enum      ALLOWED:POST, JSON
                    Indicate through which method you want to receive data with the callback, by default it is POST

    field  string     Name of the field containing the callback return data
    url    string     Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
  cf_piva*          string
  tipo_catasto*     enum      ALLOWED:T, F, TF
  provincia*        string
}
OPTION 4{
  callback
    ONE OF
    OPTION 1{
      method enum      DEFAULT:JSON
                        ALLOWED:POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                        Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                        value pair named data (customizable using the field parameter) within the request body. If the method is set
                        to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json.
                        The raw JSON data will be directly inserted into the request body without any encoding.

      field  string     DEFAULT:data

```

This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.

url* string A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.

retry integer **between 0 and 5**
DEFAULT:0

The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

headers {
A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.
 key string
}

custom {
This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your application.
}

OPTION 2{
 method enum **ALLOWED:POST, JSON**
 Indicate through which method you want to receive data with the callback, by default it is POST
 field string Name of the field containing the callback return data
 url string Url where the callback call will be made
 data {
 Object to be completed at your convenience with additional data that you can use back with the callback
 }
}

cf_piva* string
tipo_catasto* enum **ALLOWED:T, F, TF**

OPTION 5{
 callback
 ONE OF
 OPTION 1{
 method enum **DEFAULT:JSON**
 ALLOWED:POST, JSON
 Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.
 field string **DEFAULT:data**
 This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.
 url* string A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.
 retry integer **between 0 and 5**
 DEFAULT:0
 The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.
 headers {
 A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.
 }

```

    key string
  }
  custom {
    This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within
    the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
    application.
  }
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST
  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
id_indirizzo*      string
dal_civico          string
al_civico           string
}

```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```

{
  data {
    data {
      id          string
      endpoint    enum    ALLOWED:elenco_immobili, prospetto_catastale,
                           ricerca_persona_giuridica,
                           ricerca_persona_fisica, ricerca_nazionale_pg,
                           ricerca_nazionale_pf, indirizzo
      stato       string
      callback
        ONE OF
        OPTION 1{
          method enum    DEFAULT:JSON
                        ALLOWED:POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated
                        with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be
                        encoded into a key-value pair named data (customizable using the field parameter) within the
                        request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the
                        Content-Type header set to application/json. The raw JSON data will be directly inserted into the
                        request body without any encoding.
          field  string  DEFAULT:data
                        This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as
                        a string, it determines the name of the parameter that will be sent in the POST request.
          url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                        requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer
                        auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times
                        until a 200 status code is received.
          retry  integer  between 0 and 5
                        DEFAULT:0
                        The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                        attempts is reached without a successful status code (e.g., 200 OK), the request is considered a
                        final failure. A time interval is often introduced between attempts to avoid overloading the server.
                        This interval may increase exponentially with each attempt to reduce the impact on an overloaded
                        server. The 'retry' parameter is configured to retry the request only in case of specific status codes,
                        typically those indicating temporary errors (e.g., 502, 503) or network issues.
        }
      }
    }
  }
}

```

```

headers {
  A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
  various purposes, including authentication, content negotiation, and custom metadata.

  key string
}
custom {
  This object can be populated with any additional data you wish to include in the callback response. This data will be
  accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant
  information that you need for your application.

}
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST

  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
}
parametri
ONE OF
OPTION 1{
  tipo_catasto*  enum    ALLOWED:T, F, TF
                  - **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where
                  supported by the endpoint)

  provincia*     string  Province code (2 letters, e.g. **RM**, **GE**).
  comune*        string  Municipality name (e.g. **ROMA**).
  sezione       string  Cadastral section. **Required only for municipalities that have sections** (see
                          `Territorio.sezioni`). Otherwise leave empty.
  sezione_urbana string  Urban section. **Never required**. Provide only if known; otherwise leave empty.
  foglio*
    ONE OF
    prop0
      string
    prop1
      integer
  particella*
    ONE OF
    prop0
      string
    prop1
      integer
}
OPTION 2{
  tipo_catasto*  enum    ALLOWED:T, F, TF
                  - **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where
                  supported by the endpoint)

  provincia*     string  Province code (2 letters).
  comune*        string  Municipality name in uppercase.
  sezione       string  Required only for municipalities with sections; otherwise empty.
  sezione_urbana string  Never required; optional.
  foglio*
    ONE OF
    prop0
      string
    prop1
      integer
  particella*

```

```

    ONE OF
    prop0
    string
    prop1
    integer
    subalterno
    ONE OF
    prop0
    string
    prop1
    integer
  }
  OPTION 3{
    cf_piva*      string  Italian tax code (**CF**) or VAT number (**P.IVA**) of the subject/company.
    tipo_catasto* enum    ALLOWED:T, F, TF
                    - **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where
                    supported by the endpoint)
    provincia*   string  Province code (2 letters). Required for local searches.
  }
  OPTION 4{
    cf_piva*      string  Italian tax code (CF) or VAT number (P.IVA).
    tipo_catasto* enum    ALLOWED:T, F, TF
                    - **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where
                    supported by the endpoint)
  }
  OPTION 5{
    id_indirizzo* string  Address identifier returned by the address search endpoint.
    dal_civico     string  From street number (inclusive). Optional.
    al_civico      string  To street number (inclusive). Optional.
  }
  risultato {
    Always null in POST responses.
  }
  esito          string
  timestamp      integer
  owner          string
}
success          boolean
message          string
error            integer
}
success          boolean
message          string
error            integer
}

```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 402: Payment Required

RESPONSE MODEL - application/json

STATUS CODE - 406: Not Acceptable

RESPONSE MODEL - application/json

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

4.2 GET /richiesta

Extracts all requests

This method allows you to see all the requests you have made

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data [{
    id          string
    endpoint    string
    stato       string
    timestamp   integer
    owner       string
  }]
  success      boolean
  message      string
  error        integer
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

4.3 GET /richiesta/{id}

Detail of the request

Allows you to see the detail of the request identified by the id

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	Request id

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    data {
      id          string
      endpoint    enum
    }
  }
}
```

ALLOWED:elenco_immobili, prospetto_catastale, ricerca_persona_giuridica, ricerca_persona_fisica, ricerca_nazionale_pg,

```

ricerca_nazionale_pf, indirizzo

stato          string
callback
  ONE OF
  OPTION 1 {
    method enum    DEFAULT:JSON
                  ALLOWED:POST, JSON
                  Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated
                  with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be
                  encoded into a key-value pair named data (customizable using the field parameter) within the
                  request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the
                  Content-Type header set to application/json. The raw JSON data will be directly inserted into the
                  request body without any encoding.

    field  string  DEFAULT:data
                  This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as
                  a string, it determines the name of the parameter that will be sent in the POST request.

    url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                  requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer
                  auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times
                  until a 200 status code is received.

    retry  integer  between 0 and 5
                  DEFAULT:0
                  The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                  attempts is reached without a successful status code (e.g., 200 OK), the request is considered a
                  final failure. A time interval is often introduced between attempts to avoid overloading the server.
                  This interval may increase exponentially with each attempt to reduce the impact on an overloaded
                  server. The 'retry' parameter is configured to retry the request only in case of specific status codes,
                  typically those indicating temporary errors (e.g., 502, 503) or network issues.

    headers {
      A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
      various purposes, including authentication, content negotiation, and custom metadata.

      key string
    }
    custom {
      This object can be populated with any additional data you wish to include in the callback response. This data will be
      accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant
      information that you need for your application.
    }
  }
}
OPTION 2 {
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST

  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
parametri
  ONE OF
  OPTION 1 {
    tipo_catasto* enum    ALLOWED:T, F, TF
                  - **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where
                  supported by the endpoint)

    provincia*   string  Province code (2 letters, e.g. **RM**, **GE**).
    comune*      string  Municipality name (e.g. **ROMA**).
    sezione     string  Cadastral section. **Required only for municipalities that have sections** (see
                        `Territorio.sezioni`). Otherwise leave empty.
    sezione_urbana string Urban section. **Never required**. Provide only if known; otherwise leave empty.
    foglio*
      ONE OF
      prop0
      string

```

```

        prop1
        integer
    particella*
        ONE OF
        prop0
        string
        prop1
        integer
    }
    OPTION 2{
        tipo_catasto*  enum    ALLOWED:T, F, TF
                        - **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where
                        supported by the endpoint)

        provincia*    string  Province code (2 letters).
        comune*       string  Municipality name in uppercase.
        sezione      string  Required only for municipalities with sections; otherwise empty.
        sezione_urbana string  Never required; optional.
        foglio*
            ONE OF
            prop0
            string
            prop1
            integer
        particella*
            ONE OF
            prop0
            string
            prop1
            integer
        subalterno
            ONE OF
            prop0
            string
            prop1
            integer
    }
    OPTION 3{
        cf_piva*       string  Italian tax code (**CF**) or VAT number (**P.IVA**) of the subject/company.
        tipo_catasto*  enum    ALLOWED:T, F, TF
                        - **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where
                        supported by the endpoint)

        provincia*    string  Province code (2 letters). Required for local searches.
    }
    OPTION 4{
        cf_piva*       string  Italian tax code (CF) or VAT number (P.IVA).
        tipo_catasto*  enum    ALLOWED:T, F, TF
                        - **T** = Terreni (land) - **F** = Fabbricati (buildings) - **TF** = Both domains (where
                        supported by the endpoint)
    }
    OPTION 5{
        id_indirizzo* string  Address identifier returned by the address search endpoint.
        dal_civico     string  From street number (inclusive). Optional.
        al_civico      string  To street number (inclusive). Optional.
    }
    risultato
    ONE OF
    OPTION 1{
        immobili [{

```

Array of object:

```
sezione_urbana string
foglio
ONE OF
prop0
string
prop1
integer
particella
ONE OF
prop0
string
prop1
integer
subalterno integer
indirizzo string
sezione string
zona_censuaria string
categoria string
classe string
consistenza string
rendita string
partita string
id_immobile string
}]
}
OPTION 2{
  immobili [{
    Array of object:
      sezione_urbana string
      foglio
      ONE OF
      prop0
      string
      prop1
      integer
      particella
      ONE OF
      prop0
      string
      prop1
      integer
      subalterno integer
      qualita string
      classe string
      ha string hectare
      are string ara: 100 square meters
      ca string centiara: 1 square meter
      reddito_dominicale string
      reddito_agrario string
      partita string
      porzioni
      ONE OF
      0 {
        ::props {
          porzione string
          qualita string
```

```

        classe          string
        ha              string
        are             string
        ca              string
        reddito_dominicale string
        reddito_agrario string
    }
}
prop0
string
id_immobile          string
}]
}
OPTION 3{
    immobili [{
        Array of object:
        sezione_urbana string
        foglio
        ONE OF
        prop0
        string
        prop1
        integer
        particella
        ONE OF
        prop0
        string
        prop1
        integer
        subalterno      integer
        indirizzo        string
        sezione        string
        zona_censuaria   string
        categoria        string
        classe           string
        consistenza      string
        rendita          string
        partita          string
        id_immobile      string
        intestatari [{
            Array of object:
            denominazione string
            cf              string
            proprieta       string
            quota           string
        }]
    }]
}
OPTION 4{
    immobili [{
        Array of object:
        sezione_urbana string
        foglio
        ONE OF
        prop0
        string
        prop1

```

```

integer
particella
  ONE OF
  prop0
  string
  prop1
  integer
subalterno          integer
qualita             string
classe              string
ha                  string
are                 string
ca                  string
reddito_dominicale string
reddito_agrario     string
partita             string
porzioni
  ONE OF
  0 {
    ::props {
      porzione      string
      qualita        string
      classe         string
      ha             string
      are            string
      ca             string
      reddito_dominicale string
      reddito_agrario string
    }
  }
  prop0
  string
id_immobile          string
intestatari [{
  Array of object:
    denominazione string
    cf              string
    proprieta       string
    quota           string
  }]
}]
}
OPTION 5{
  soggetti [{
    Array of object:
      cognome      string
      nome          string
      data_nascita string
      luogo_nascita string
      sesso         string
      cf            string
      id_soggetto  string
      immobili [{
        Array of object:
          catasto      string
          titolarita   string
          ubicazione    string

```

```

        comune            string
        provincia         string
        codice_catastale  string
        sezione         string
        sezione_urbana  string
        foglio
        ONE OF
        prop0
        string
        prop1
        integer
    particella
        ONE OF
        prop0
        string
        prop1
        integer
    subalterno            integer
    classamento          string
    classe                string
    consistenza           string
    rendita               string
    id_immobile           string
    }
}
}
OPTION 6{
    soggetti [{
        Array of object:
        denominazione    string
        sede              string
        cf                string
        id_soggetto       string
        immobili [{
            Array of object:
            catasto        string
            titolarita     string
            ubicazione     string
            comune         string
            provincia      string
            codice_catastale string
            sezione      string
            sezione_urbana string
            foglio
            ONE OF
            prop0
            string
            prop1
            integer
        particella
            ONE OF
            prop0
            string
            prop1
            integer
        subalterno        integer
        classamento       string

```

```

        classe          string
        consistenza     string
        rendita          string
        id_immobile     string
    }
}
}
OPTION 7{
    soggetti [{
        Array of object:
        cognome          string
        nome              string
        data_nascita     string
        luogo_nascita    string
        sesso            string
        cf                string
        id_soggetto      string
        catasti [{
            Array of object:
            citta         string
            fabbricati    integer
            terreni       integer
            provincia     string
            comuni [{
                Array of object:
                comune     string
                fabbricati integer
                terreni    integer
            }]
        }]
    }]
}
OPTION 8{
    soggetti [{
        Array of object:
        denominazione    string
        sede              string
        cf                string
        id_soggetto      string
        catasti [{
            Array of object:
            citta         string
            fabbricati    integer
            terreni       integer
            provincia     string
            comuni [{
                Array of object:
                comune     string
                fabbricati integer
                terreni    integer
            }]
        }]
    }]
}
OPTION 9{
    immobili [{
        Array of object:

```



```

    sezione_urbana string
    foglio
    ONE OF
    prop0
    string
    prop1
    integer
    particella
    ONE OF
    prop0
    string
    prop1
    integer
    subalterno integer
    indirizzo string
    zona_censuaria string
    categoria string
    classe string
    consistenza string
    rendita string
    id_immobile string
  }]
}
esito string
timestamp integer
owner string
}
success boolean
message string
error integer
}
success boolean
message string
error integer
}

```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

5. TERRITORIO

Methods that allow consultation of provinces, municipalities, cadastral sections and Land Registry Offices (Conservatorie) of the Italian territory.

5.1 GET /territorio

List provinces

Returns all Italian provinces. If tipo_catasto is omitted, the buildings (F) dataset is used by default.

REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
tipo_catasto	enum ALLOWED: T, F	**F** = buildings (default), **T** = land.

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data [{
    provincia      string  Two-letter province code (e.g., **RM**, **GE**).
    nome_provincia string
    id             string  Internal identifier for the province entry.
  }]
  success          boolean
  message          string
  error            integer
}
```

5.2 GET /territorio/{provincia_or_nome_provincia_or_id}

Province detail

Returns the selected province with its municipalities. If tipo_catasto is omitted, the buildings (F) dataset is used by default.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*provincia_or_nome_provincia_or_id	string	Province code (**e.g., TR**), full province name (**e.g., TERNI**), or internal **id** .

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
tipo_catasto	enum ALLOWED: T, F	**F** = buildings (default), **T** = land.

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    provincia      string
    nome_provincia string
    comuni [{
      Array of object:
      comune      string
      sezioni      [string]
      codice_catastale string
    }]
    id            string
  }
  success        boolean
  message        string
  error          integer
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

5.3 GET /territorio/{provincia_or_nome_provincia_or_id}/{comune}

Municipality detail

Returns the selected municipality within the given province. If tipo_catasto is omitted, the buildings (F) dataset is used by default.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*provincia_or_nome_provincia_or_id	string	Province code (**e.g., TR**), full province name (**e.g., TERNI**), or internal **id**.
*comune	string	Municipality name returned by **Territorio**.

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
tipo_catasto	enum ALLOWED: T, F	**F** = buildings (default), **T** = land.

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    provincia      string
```

```

    nome_provincia      string
    comuni [{
      Array of object:
        comune          string
        sezioni          [string]
        codice_catastale string
      }]
    id                  string
  }
  success              boolean
  message              string
  error                integer
}

```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

5.4 GET /territorio/conservatorie

List of Land Registry Offices

Returns the list of all Italian Land Registry Offices (Conservatorie).

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```

{
  data [{
    Array of object:
      conservatoria string Land Registry Office name
      id            string Internal identifier
    }]
  success          boolean
  message          string
  error            integer
}

```

5.5 GET /territorio/conservatorie/{conservatoria_or_id}

Land Registry Office municipalities and cadastral sections

Returns the municipalities associated with a specific Italian Land Registry Office (Conservatoria). If the optional query parameter "comune" is provided, returns section for the selected municipality within the same Conservatoria.

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*conservatoria_or_id	string	Land Registry Office

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
comune	string	Municipality name. When provided, returns details for that municipality.

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data
    ONE OF
    0 {
      ::props {
        conservatoria      string
        comuni [{
          Array of object:
            comune          string
            codice_catastale string
          }]
      }
    }
    OPTION 1 {
      conservatoria      string
      comuni [{
        Array of object:
          comune          string
          catasto {
            terreni {
              sezioni [string]
            }
            fabbricati {
              sezioni [string]
            }
          }
        }
      }]
    }
  success      boolean
  message      string
  error        integer
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

6. VISURA CATASTALE

Methods that make it possible to obtain a Visura Catastale

6.1 GET /visura_catastale

Extracts all visure

This method allows you to see all the visure you have made

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data [{
    Array of object:
    id          string
    entita      enum      ALLOWED:immobile, soggetto
    stato       enum      ALLOWED:in_erogazione, evasa
    timestamp   integer
    owner       string
  }]
  success      boolean
  message      string
  error        integer
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

6.2 POST /visura_catastale

Create visura request

Create a request to obtain a Visura Catastale (property or subject).

- By property (entita: immobile)

- With id_immobile: the request is performed directly using the property identifier.

- Without id_immobile: the request is performed using the available cadastral data.

- By subject (entita: soggetto)

- With id_soggetto: provincia is optional. If provided, the visura is performed for that province; if omitted, the visura is performed for the first available province returned for that subject.

- Without id_soggetto: the request is performed using the available subject data.

- tipo_visura can be "ordinaria" or "storica".
- tipo_dettaglio can be "sintetica" or "analitica".

For subjects if tipo_dettaglio are omitted, the Visura Catastale will be "synthetic" by default.

For properties if the tipo_dettaglio is omitted, the Visura Catastale "storica" will be "analytical" by default, the Visura Catastale "ordinaria" is always complete.

Use tipo_catasto: T | F | TF.

The response returns a request id. Check status with GET /visura_catastale/{id} and download the PDF with GET /visura_catastale/{id}/documento when available.

REQUEST

REQUEST BODY - application/json

ONE OF

OPTION:1

ONE OF

OPTION 1 {

entita*	enum	ALLOWED:immobile
id_immobile*	string	
tipo_visura*	enum	ALLOWED:ordinaria, storica
tipo_dettaglio	enum	ALLOWED:sintetica, analitica
richiedente	string	
callback		

ONE OF

OPTION 1 {

method	enum	<p>DEFAULT:JSON</p> <p>ALLOWED:POST, JSON</p> <p>Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.</p>
field	string	<p>DEFAULT:data</p> <p>This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.</p>
url*	string	<p>A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.</p>
retry	integer	<p>between 0 and 5</p> <p>DEFAULT:0</p> <p>The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.</p>

headers {

A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.

```
key string
}
```

custom {

This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your application.

}

}

```

OPTION 2{
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST
  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
}
OPTION 2{
  entita*          enum          ALLOWED:immobile
  tipo_catasto*    enum          ALLOWED:T, F
  tipo_dettaglio   enum          ALLOWED:sintetica, analitica
  provincia*       string
  comune*          string
  sezione        string
  sezione_urbana string
  foglio*
  ONE OF
  prop0            string
  prop1            integer
  particella*
  ONE OF
  prop0            string
  prop1            integer
  subalterno
  ONE OF
  prop0            string
  prop1            integer
  tipo_visura*     enum          ALLOWED:ordinaria, storica
  richiedente      string
  callback
  ONE OF
  OPTION 1{
    method enum    DEFAULT:JSON
                  ALLOWED:POST, JSON
                  Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                  Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                  value pair named data (customizable using the field parameter) within the request body. If the method is set
                  to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json.
                  The raw JSON data will be directly inserted into the request body without any encoding.
    field  string  DEFAULT:data
                  This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it
                  determines the name of the parameter that will be sent in the POST request.
    url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring
                  authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum
                  timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is
                  received.
    retry  integer  between 0 and 5
                  DEFAULT:0
                  The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is
                  reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time
                  interval is often introduced between attempts to avoid overloading the server. This interval may increase

```


exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

headers {

A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.

key string
}

custom {

This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your application.

}

}

OPTION 2{

method enum ALLOWED:POST, JSON

Indicate through which method you want to receive data with the callback, by default it is POST

field string Name of the field containing the callback return data

url string Url where the callback call will be made

data {

Object to be completed at your convenience with additional data that you can use back with the callback

}

}

}

OPTION:2

ONE OF

OPTION 1{

entita* enum ALLOWED:soggetto

id_soggetto* string

tipo_catasto* enum
DEFAULT:TF
ALLOWED:T, F, TF

provincia string

tipo_visura* enum ALLOWED:ordinaria, storica

tipo_dettaglio enum ALLOWED:sintetica, analitica

richiedente string

callback

ONE OF

OPTION 1{

method enum
DEFAULT:JSON
ALLOWED:POST, JSON

Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.

field string
DEFAULT:data

This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.

url* string
A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.

retry integer between 0 and 5
DEFAULT:0

The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

headers {

A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.

```

    key string
  }
  custom {
    This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within
    the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
    application.
  }
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST

  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
}
OPTION 2{
  entita*          enum          ALLOWED:soggetto
  cf_piva*         string
  tipo_catasto*    enum          ALLOWED:T, F, TF
  tipo_dettaglio  enum          ALLOWED:sintetica, analitica
  provincia*      string
  tipo_visura*    enum          ALLOWED:ordinaria, storica
  richiedente     string
  callback
  ONE OF
  OPTION 1{
    method enum    DEFAULT:JSON
                  ALLOWED:POST, JSON
                  Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                  Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                  value pair named data (customizable using the field parameter) within the request body. If the method is set
                  to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json.
                  The raw JSON data will be directly inserted into the request body without any encoding.

    field  string  DEFAULT:data
                  This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it
                  determines the name of the parameter that will be sent in the POST request.

    url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring
                  authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum
                  timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is
                  received.

    retry  integer  between 0 and 5
                  DEFAULT:0
                  The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is
                  reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time
                  interval is often introduced between attempts to avoid overloading the server. This interval may increase
                  exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is
                  configured to retry the request only in case of specific status codes, typically those indicating temporary
                  errors (e.g., 502, 503) or network issues.

  headers {
    A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various
    purposes, including authentication, content negotiation, and custom metadata.

    key string
  }
  custom {
    This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within
    the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
    application.
  }
}
}
OPTION 2{

```

```

method enum    ALLOWED:POST, JSON
                Indicate through which method you want to receive data with the callback, by default it is POST
field  string  Name of the field containing the callback return data
url    string  Url where the callback call will be made
data {
    Object to be completed at your convenience with additional data that you can use back with the callback
}
}
}

```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```

{
  data {
    id                string
    entita             enum                ALLOWED:immobile, soggetto
    stato             enum                ALLOWED:in_erogazione,
                                         in_lavorazione, evasa

    callback
      ONE OF
      OPTION 1{
        method enum    DEFAULT:JSON
                        ALLOWED:POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                        Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a
                        key-value pair named data (customizable using the field parameter) within the request body. If the method
                        is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/
                        json. The raw JSON data will be directly inserted into the request body without any encoding.

        field  string  DEFAULT:data
                        This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string,
                        it determines the name of the parameter that will be sent in the POST request.

        url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                        requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The
                        maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status
                        code is received.

        retry  integer  between 0 and 5
                        DEFAULT:0
                        The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts
                        is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time
                        interval is often introduced between attempts to avoid overloading the server. This interval may increase
                        exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is
                        configured to retry the request only in case of specific status codes, typically those indicating temporary
                        errors (e.g., 502, 503) or network issues.

        headers {
          A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various
          purposes, including authentication, content negotiation, and custom metadata.

          key string
        }
        custom {
          This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
          within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need
          for your application.
        }
      }
    }
  }
  OPTION 2{
    method enum    ALLOWED:POST, JSON
                Indicate through which method you want to receive data with the callback, by default it is POST

    field  string  Name of the field containing the callback return data

    url    string  Url where the callback call will be made
  }
}

```

```

    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
  parametri
  ONE OF
  OPTION 1{
    id_immobile      string
    tipo_catasto     enum      ALLOWED:T, F
    foglio
    ONE OF
    prop0            string
    prop1            integer
    particella
    ONE OF
    prop0            string
    prop1            integer
    subalterno       integer
    comune           string
    sezione         string
    sezione_urbana  string
    provincia         string
  }
  OPTION 2{
    id_soggetto      string
    tipo_catasto     enum      ALLOWED:T, F, TF
    cf_piva          string
    provincia         string
  }
  tipo_visura              enum      ALLOWED:ordinaria, storica
  richiedente              string
  documento                string
  esito                    string
  timestamp                integer
  owner                    string
}
success                    boolean
message                    string
error                      integer
}

```

STATUS CODE - 402: Payment Required

RESPONSE MODEL - application/json

STATUS CODE - 406: Not Acceptable

RESPONSE MODEL - application/json

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

6.3 GET /visura_catastale/{id}

Detail of the visura

Allows you to see the detail of the visura identified by the id

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	Visura id

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    id                string
    entita             enum      ALLOWED:immobile, soggetto
    stato             enum      ALLOWED:in_erosazione,
                                in_lavorazione, evasa

    callback
      ONE OF
      OPTION 1 {
        method enum      DEFAULT:JSON
                                ALLOWED:POST, JSON
                                Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with
                                the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded
                                into a key-value pair named data (customizable using the field parameter) within the request body. If the
                                method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to
                                application/json. The raw JSON data will be directly inserted into the request body without any
                                encoding.

        field string      DEFAULT:data
                                This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                                string, it determines the name of the parameter that will be sent in the POST request.

        url*  string      A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                                requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The
                                maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200
                                status code is received.

        retry integer      between 0 and 5
                                DEFAULT:0
                                The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                                attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                                failure. A time interval is often introduced between attempts to avoid overloading the server. This
                                interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                                The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                                those indicating temporary errors (e.g., 502, 503) or network issues.

        headers {
          A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
          various purposes, including authentication, content negotiation, and custom metadata.

          key string
        }
        custom {
          This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
          within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
          need for your application.
        }
      }
    }
  }
  OPTION 2 {
    method enum      ALLOWED:POST, JSON
                                Indicate through which method you want to receive data with the callback, by default it is POST

    field string      Name of the field containing the callback return data
```

```

    url      string  Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
}
parametri
  ONE OF
  OPTION 1{
    id_immobile      string
    tipo_catasto     enum    ALLOWED:T, F
    foglio
      ONE OF
      prop0
        string
      prop1
        integer
    particella
      ONE OF
      prop0
        string
      prop1
        integer
    subalterno      integer
    comune          string
    sezione        string
    sezione_urbana string
    provincia        string
  }
  OPTION 2{
    id_soggetto      string
    tipo_catasto     enum    ALLOWED:T, F, TF
    cf_piva          string
    provincia        string
  }
  tipo_visura              enum    ALLOWED:ordinaria, storica
  richiedente              string
  documento                string
  esito                    string
  timestamp                integer
  owner                    string
}
success                    boolean
message                    string
error                      integer
}

```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

6.4 GET /visura_catastale/{id}/documento

Download visura

This method allows you to directly download the document

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	Visura id

RESPONSE

STATUS CODE - 200: Content of the file

RESPONSE MODEL - application/PDF
string

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

STATUS CODE - 422: Unprocessable Entity

RESPONSE MODEL - application/json

7. VISURA IPOTECARIA

Methods to retrieve land registry notes (formalities) for subjects or properties, including Land Registry Office discovery, note listing, and note details.

7.1 POST /ipotecarie-ispezione_nazionale

National land registry inspection request

Starts a national land registry inspection for a subject identified by cf_piva. Fields richiedente and motivo can be provided to specify the requester and the reason.

Richiedente must be a valid tax code

REQUEST

REQUEST BODY - application/json

```
{
  cf_piva*           string           Tax code (CF) or VAT number (PIVA) of the
                                subject.
  callback
    ONE OF
    OPTION 1 {
      method enum    DEFAULT:JSON
                        ALLOWED:POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                        Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                        value pair named data (customizable using the field parameter) within the request body. If the method is set to
                        'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw
                        JSON data will be directly inserted into the request body without any encoding.
      field  string   DEFAULT:data
                        This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it
                        determines the name of the parameter that will be sent in the POST request.
      url*   string   A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring
                        authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum
                        timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.
      retry  integer  between 0 and 5
                        DEFAULT:0
                        The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is
                        reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval
                        is often introduced between attempts to avoid overloading the server. This interval may increase exponentially
                        with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the
                        request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or
                        network issues.
    }
  headers {
    A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various
    purposes, including authentication, content negotiation, and custom metadata.
    key string
  }
  custom {
    This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the
    'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
    application.
  }
}
OPTION 2 {
  method enum    ALLOWED:POST, JSON
                        Indicate through which method you want to receive data with the callback, by default it is POST
  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
```



```

    }
  }
  richiedente      string      Codice Fiscale.
  motivo           string      Reason of the request.
}

```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```

{
  data {
    endpoint      string
    stato         string
    callback
    ONE OF
    OPTION 1{
      method enum    DEFAULT:JSON
                        ALLOWED:POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated
                        with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be
                        encoded into a key-value pair named data (customizable using the field parameter) within the
                        request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the
                        Content-Type header set to application/json. The raw JSON data will be directly inserted into the
                        request body without any encoding.

      field string   DEFAULT:data
                        This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                        string, it determines the name of the parameter that will be sent in the POST request.

      url*  string   A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                        requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'.
                        The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a
                        200 status code is received.

      retry integer  between 0 and 5
                        DEFAULT:0
                        The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                        attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                        failure. A time interval is often introduced between attempts to avoid overloading the server. This
                        interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                        The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                        those indicating temporary errors (e.g., 502, 503) or network issues.

      headers {
        A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
        various purposes, including authentication, content negotiation, and custom metadata.

        key string
      }
      custom {
        This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
        within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
        need for your application.

      }
    }
  }
  OPTION 2{
    method enum    ALLOWED:POST, JSON
                        Indicate through which method you want to receive data with the callback, by default it is POST

    field string   Name of the field containing the callback return data
    url  string    Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
}
parametri {
  Echo of request body.
}

```

```

    }
    risultato {
      Always null in POST responses.
    }
    esito string
    documento string
    richiedente string
    motivo string
    timestamp integer
    owner string
    id string
  }
  success boolean
  message string
  error integer
}

```

STATUS CODE - 406: Not Acceptable

RESPONSE MODEL - application/json

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

7.2 POST /ipotecarie-elenco-note

Land registry notes list request

Starts a request to retrieve the list of land registry entries (notes/formalities) for either a property or a subject, depending on the provided entita.

For buildings, the subaltern is not mandatory, if omitted the search is performed on the first subaltern of the searched building in the list.

REQUEST

REQUEST BODY - application/json

ONE OF

OPTION 1{

entita*	enum	ALLOWED:ispezione_immobile Request type.
comune*	string	Municipality name.
tipo_catasto*	string	Cadastral type (e.g. F for Fabbricati, T for Terreni).
conservatoria*	string	Land Registry Office identifier or name.
foglio*	integer	
particella*	integer	
subalterno	integer	
sezione	string	Optional cadastral section.
sezione_urbana	string	Optional urban section.

}

OPTION 2{

entita*	enum	ALLOWED:ispezione_soggetto Request type.
conservatoria*	string	Land Registry Office identifier or name.
cf_piva*	string	Tax code (CF) or VAT number (P.IVA) of the subject.

}

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    endpoint          string
    stato             string
    callback
    ONE OF
    OPTION 1{
      method enum    DEFAULT:JSON
                        ALLOWED:POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with
                        the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded
                        into a key-value pair named data (customizable using the field parameter) within the request body. If the
                        method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to
                        application/json. The raw JSON data will be directly inserted into the request body without any
                        encoding.
      field  string    DEFAULT:data
                        This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                        string, it determines the name of the parameter that will be sent in the POST request.
      url*   string    A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                        requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The
                        maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200
                        status code is received.
      retry  integer   between 0 and 5
                        DEFAULT:0
                        The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                        attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                        failure. A time interval is often introduced between attempts to avoid overloading the server. This
                        interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                        The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                        those indicating temporary errors (e.g., 502, 503) or network issues.

    headers {
      A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
      various purposes, including authentication, content negotiation, and custom metadata.
      key string
    }
    custom {
      This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
      within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
      need for your application.
    }
  }
  OPTION 2{
    method enum    ALLOWED:POST, JSON
                        Indicate through which method you want to receive data with the callback, by default it is POST
    field  string    Name of the field containing the callback return data
    url    string    Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
  parametri {
    Echo of request body.
  }
  risultato {
    Always null in POST responses.
  }
  esito          string
  documento      string
  richiedente    string
}
```

motivo	string
timestamp	integer
owner	string
id	string
}	
success	boolean
message	string
error	integer
}	

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

7.3 POST /ipotecarie-dettaglio-nota

Land registry note detail request

Starts a request to retrieve the detail of a land registry note (formalità), for either a subject (physical/legal) or a property, using the general or particular registry references.

REQUEST

REQUEST BODY - application/json

ONE OF

OPTION 1{

tipo_restrizione*	enum	ALLOWED: soggetto_fisico, soggetto_giuridico Restriction type for the inspection.
tipo_registro*	enum	ALLOWED: generale Registry type.
conservatoria*	string	Land Registry Office identifier or name.
anno*	integer	Year (4 digits).
registro_generale*	integer	General registry number.
cf_piva*	string	Tax code (CF) or VAT number (P.IVA) of the subject.
callback		

ONE OF

OPTION 1{

method	enum	DEFAULT: JSON ALLOWED: POST, JSON Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.
field	string	DEFAULT: data This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.
url*	string	A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.
retry	integer	between 0 and 5 DEFAULT: 0 The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

headers {

A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.

```

    key string
  }
  custom {
    This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within
    the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
    application.
  }
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST

  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
richiedente      string      Requester tax code (Codice Fiscale).
motivo           string      Reason of the request.
}
OPTION 2{
  tipo_restrizione*      enum      ALLOWED:soggetto_fisico,
                                soggetto_giuridico

  tipo_registro*         enum      ALLOWED:particolare
  conservatoria*         string
  anno*                  integer    Year (4 digits).
  tipo_nota*             enum      ALLOWED:trascrizioni, iscrizioni,
                                annotazioni, privilegi_agrari,
                                privilegi_speciali,
                                privilegi_minerali
                                Note type (category).

  registro_particolare*  integer    Particular registry number.
  cf_piva*               string
  callback
  ONE OF
  OPTION 1{
    method enum    DEFAULT:JSON
                  ALLOWED:POST, JSON
                  Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                  Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                  value pair named data (customizable using the field parameter) within the request body. If the method is set
                  to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json.
                  The raw JSON data will be directly inserted into the request body without any encoding.

    field  string  DEFAULT:data
                  This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it
                  determines the name of the parameter that will be sent in the POST request.

    url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring
                  authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum
                  timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is
                  received.

    retry  integer  between 0 and 5
                  DEFAULT:0
                  The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is
                  reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time
                  interval is often introduced between attempts to avoid overloading the server. This interval may increase
                  exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is
                  configured to retry the request only in case of specific status codes, typically those indicating temporary
                  errors (e.g., 502, 503) or network issues.

  headers {
    A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various
    purposes, including authentication, content negotiation, and custom metadata.
    key string

```

```

    }
    custom {
      This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within
      the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
      application.
    }
  }
  OPTION 2{
    method enum    ALLOWED:POST, JSON
                                Indicate through which method you want to receive data with the callback, by default it is POST

    field  string  Name of the field containing the callback return data
    url    string  Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
  richiedente string
  motivo      string
}
OPTION 3{
  tipo_restrizione* enum    ALLOWED:immobile
  tipo_registro*   enum    ALLOWED:generale
  conservatoria*   string
  anno*            integer  Year (4 digits).
  registro_generale* integer
  comune*          string   Municipality identifier or name.
  tipo_catasto*    string
  foglio*          integer
  particella*      integer
  subalterno       integer
  sezione        string
  sezione_urbana string
  callback
  ONE OF
  OPTION 1{
    method enum    DEFAULT:JSON
                                ALLOWED:POST, JSON
                                Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the
                                Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-
                                value pair named data (customizable using the field parameter) within the request body. If the method is set
                                to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json.
                                The raw JSON data will be directly inserted into the request body without any encoding.

    field  string  DEFAULT:data
                                This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it
                                determines the name of the parameter that will be sent in the POST request.

    url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring
                                authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum
                                timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is
                                received.

    retry  integer between 0 and 5
                                DEFAULT:0
                                The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is
                                reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time
                                interval is often introduced between attempts to avoid overloading the server. This interval may increase
                                exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is
                                configured to retry the request only in case of specific status codes, typically those indicating temporary
                                errors (e.g., 502, 503) or network issues.

    headers {
      A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various
      purposes, including authentication, content negotiation, and custom metadata.
      key string
    }
  }
}

```

custom {

This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your application.

}

}

OPTION 2{

method enum **ALLOWED:POST, JSON**

Indicate through which method you want to receive data with the callback, by default it is POST

field string Name of the field containing the callback return data

url string Url where the callback call will be made

data {

Object to be completed at your convenience with additional data that you can use back with the callback

}

}

richiedente string

motivo string

}

OPTION 4{

tipo_restrizione* enum **ALLOWED:immobile**

tipo_registro* enum **ALLOWED:particolare**

conservatoria* string

anno* integer

tipo_nota* enum **ALLOWED:trascrizioni, iscrizioni, annotazioni, privilegi_agrari, privilegi_speciali, privilegi_minerali**

registro_particolare* integer

comune* string

tipo_catasto* string

foglio* integer

particella* integer

subalterno integer

sezione string

sezione_urbana string

callback

ONE OF

OPTION 1{

method enum **DEFAULT:JSON**

ALLOWED:POST, JSON

Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded into a key-value pair named data (customizable using the field parameter) within the request body. If the method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to application/json. The raw JSON data will be directly inserted into the request body without any encoding.

field string **DEFAULT:data**

This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a string, it determines the name of the parameter that will be sent in the POST request.

url* string A valid URL capable of handling POST requests must be entered into this parameter. For endpoints requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200 status code is received.

retry integer **between 0 and 5**

DEFAULT:0

The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final failure. A time interval is often introduced between attempts to avoid overloading the server. This interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

headers {

A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various

purposes, including authentication, content negotiation, and custom metadata.

```
key string
}
custom {
  This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within
  the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your
  application.
}
}
OPTION 2{
  method enum    ALLOWED:POST, JSON
                  Indicate through which method you want to receive data with the callback, by default it is POST
  field  string  Name of the field containing the callback return data
  url    string  Url where the callback call will be made
  data {
    Object to be completed at your convenience with additional data that you can use back with the callback
  }
}
}
richiedente      string
motivo           string
}
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  data {
    endpoint      string
    stato         string
    callback
  }
  ONE OF
  OPTION 1{
    method enum    DEFAULT:JSON
                  ALLOWED:POST, JSON
                  Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with
                  the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded
                  into a key-value pair named data (customizable using the field parameter) within the request body. If the
                  method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to
                  application/json. The raw JSON data will be directly inserted into the request body without any
                  encoding.
    field  string  DEFAULT:data
                  This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                  string, it determines the name of the parameter that will be sent in the POST request.
    url*   string  A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                  requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The
                  maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200
                  status code is received.
    retry  integer  between 0 and 5
                  DEFAULT:0
                  The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                  attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                  failure. A time interval is often introduced between attempts to avoid overloading the server. This
                  interval may increase exponentially with each attempt to reduce the impact on an overloaded server.
                  The 'retry' parameter is configured to retry the request only in case of specific status codes, typically
                  those indicating temporary errors (e.g., 502, 503) or network issues.
  }
  headers {
    A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for
    various purposes, including authentication, content negotiation, and custom metadata.
    key string
  }
}
```



```

    custom {
      This object can be populated with any additional data you wish to include in the callback response. This data will be accessible
      within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you
      need for your application.
    }
  }
  OPTION 2{
    method enum    ALLOWED:POST, JSON
                                Indicate through which method you want to receive data with the callback, by default it is POST
    field  string  Name of the field containing the callback return data
    url    string  Url where the callback call will be made
    data {
      Object to be completed at your convenience with additional data that you can use back with the callback
    }
  }
  parametri {
    Echo of request body.
  }
  risultato {
    Always null in POST responses.
  }
  esito string
  documento string
  richiedente string
  motivo string
  timestamp integer
  owner string
  id string
}
success boolean
message string
error integer
}

```

STATUS CODE - 428: Precondition Required

RESPONSE MODEL - application/json

7.4 GET /ipotecarie

Extracts all mortgage inspections

This method allows you to see all the mortgage inspection requests you have made.

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```

{
  data [{
    Array of object:
    id string
    endpoint enum    ALLOWED:ispezione_nazionale, ispezione_elenco_note,
                    ispezione_dettaglio_nota
  }
}

```

```

    stato      enum      The service that generated the request.
                        ALLOWED: evasa, in_lavorazione, in_erogazione
                        Current processing status.
    timestamp  integer    Unix timestamp.
    owner      string
  }}
  success      boolean
  message      string
  error        integer
}
```

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

7.5 GET /ipotecarie/{id}

Detail of a mortgage inspection request

Allows you to see the detail of the mortgage inspection request identified by the id, including result and document (if available).

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	Request id

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```

{
  data {
    endpoint      string
    stato         string
    callback
    ONE OF
    OPTION 1 {
      method      enum      DEFAULT: JSON
                        ALLOWED: POST, JSON
                        Specify the callback method. When set to 'POST', a standard HTTP POST request will be initiated with
                        the Content-Type header set to application/x-www-form-urlencoded. The JSON data will be encoded
                        into a key-value pair named data (customizable using the field parameter) within the request body. If the
                        method is set to 'JSON', a direct HTTP POST request will be made with the Content-Type header set to
                        application/json. The raw JSON data will be directly inserted into the request body without any
                        encoding.
      field      string      DEFAULT: data
                        This parameter works in conjunction with the 'method' parameter set to 'POST'. When provided as a
                        string, it determines the name of the parameter that will be sent in the POST request.
      url*      string      A valid URL capable of handling POST requests must be entered into this parameter. For endpoints
                        requiring authentication, the 'headers' parameter can be used to specify 'basic auth' or 'bearer auth'. The
                        maximum timeout is 30 seconds. If 'retry' is set, the system will retry up to 'retry' times until a 200
                        status code is received.
      retry      integer      between 0 and 5
                        DEFAULT: 0
                        The maximum number of attempts is defined by the 'retry' parameter. If the maximum number of
                        attempts is reached without a successful status code (e.g., 200 OK), the request is considered a final
                        failure. A time interval is often introduced between attempts to avoid overloading the server. This
```

interval may increase exponentially with each attempt to reduce the impact on an overloaded server. The 'retry' parameter is configured to retry the request only in case of specific status codes, typically those indicating temporary errors (e.g., 502, 503) or network issues.

headers {

A collection of HTTP headers to be sent with the request. Each header is defined as a key-value pair. Headers can be used for various purposes, including authentication, content negotiation, and custom metadata.

key string

}

custom {

This object can be populated with any additional data you wish to include in the callback response. This data will be accessible within the 'custom' parameter of the callback function. You can use this to store and retrieve any relevant information that you need for your application.

}

}

OPTION 2{

method enum ALLOWED:POST, JSON

Indicate through which method you want to receive data with the callback, by default it is POST

field string Name of the field containing the callback return data

url string Url where the callback call will be made

data {

Object to be completed at your convenience with additional data that you can use back with the callback

}

}

parametri {

Parameters used to start the request (varies by entita).

}

risultato

ONE OF

OPTION:1

ONE OF

OPTION 1{

risultato {

soggetti [{

Array of object:

cognome string

nome string

data_nascita string

Sesso string

luogo_nascita string

cf string

conservatorie [{

Array of object:

conservatoria string

trascrizioni string

iscrizioni string

annotazioni string

}]

}]

}

}

OPTION 2{

risultato {

soggetti [{

Array of object:

denominazione string

sede string

cf string

conservatorie [{

Array of object:

```

        conservatoria string
        trascrizioni  string
        iscrizioni    string
        annotazioni   string
    }
}
}
}
OPTION:2
ONE OF
OPTION 1{
    risultato {
        immobili [{
            Array of object:
            identificativo_definitivo {
                sezione      string
                sezione_urbana string
                foglio          string
                particella      string
                subalterno      string
            }
            identificativo_provvisorio {
                tipo_denuncia  string
                numero         string
                anno           string
            }
            note [{
                Array of object:
                tipo_nota*      string
                data_nota*      string
                registro_particolare* string
                registro_generale* string
                pubblico_ufficiale* string
                repertorio*     string
                data_repertorio* string
                tipo_atto*      string
                citta*          string
                provincia*      string
                documenti_correlati* {
                }
            }
        }
    }
}
}
OPTION 2{
    risultato {
        soggetti [{
            Array of object:
            cognome      string
            nome         string
            data_nascita  string
            luogo_nascita string
            sesso        string
            cf            string
            note [{
                Array of object:
                tipo_nota*      string

```

```

        data_nota*           string
        registro_particolare* string
        registro_generale*   string
        pubblico_ufficiale*  string
        repertorio*          string
        data_repertorio*     string
        tipo_atto*           string
        citta*               string
        provincia*           string
        documenti_correlati* {
    }
    }
    }
}
OPTION 3{
    risultato {
        soggetti [{
            Array of object:
            denominazione    string
            sede              string
            cf                string
            note [{
                Array of object:
                tipo_nota*    string
                data_nota*    string
                registro_particolare* string
                registro_generale* string
                pubblico_ufficiale* string
                repertorio*    string
                data_repertorio* string
                tipo_atto*     string
                citta*         string
                provincia*     string
                documenti_correlati* {
            }
        }
    }
}
}
OPTION 3{
    risultato {
        nota {
            tipo_nota    string
            data_nota    string
            registro_particolare string
            registro_generale  string
        }
    }
}
esito                string
documento            string
richiedente          string
motivo               string
timestamp            integer
owner                string
id                   string

```

```
    }
    success          boolean
    message          string
    error            integer
  }
```

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

7.6 GET /ipotecarie/{id}/documento

Download visura

This method allows you to directly download the document

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*id	string	Visura id

RESPONSE

STATUS CODE - 200: Content of the file

RESPONSE MODEL - application/PDF

string

STATUS CODE - 400: Bad Request

RESPONSE MODEL - application/json

STATUS CODE - 404: Not Found

RESPONSE MODEL - application/json

STATUS CODE - 422: Unprocessable Entity

RESPONSE MODEL - application/json
