Piattaforma Regionale

IOService
*Utilizzo della Piattaforma Regionale IOService ed integrazione*

Sommario

[1. Revisioni 2](#_Toc86155860)

[2. Scopo del documento 2](#_Toc86155861)

[3. Piattaforma Regionale IOService 3](#_Toc86155862)

[3.1 Caricamento Messaggi da CSV 5](#_Toc86155863)

[3.2 Storico messaggi 6](#_Toc86155864)

[4. Integrazione Web Service 7](#_Toc86155865)

[4.1 Invio dei Messaggi 9](#_Toc86155866)

[4.2 Recupero Stato Messaggio 10](#_Toc86155867)

[4.3 Aggiornamento Messaggio 11](#_Toc86155868)

[4.4 Cancellazione Messaggio 12](#_Toc86155869)

[4.5 Esempi di integrazione e modalità di test 13](#_Toc86155870)

[Appence A – Codici servizio per i comuni gestiti da Regione Marche 17](#_Toc86155871)

[Appendice B – Come impostare il separatore dei file csv in Windows 18](#_Toc86155872)

## Revisioni

|  |  |  |  |
| --- | --- | --- | --- |
| **Versione** | **Stato** | **Data** | **Descrizione** |
| 0.1 | BOZZA | 06/09/2021 | Prima edizione in consultazione pubblica |
| 0.2 | BOZZA | 14/09/2021 | Viene chiarito il valore del codice servizio interno per i servizi di Pagamento MPAY |
| 1.0 | OBSOLETO | 08/10/2021 | Prima edizione rilasciata |
| 1.1 | DEFINITIVO | 26/10/2021 | Aggiunte appendici A e B |
| 1.2 | DEFINITIVO | 14/12/2021 | Correzioni (indicate in giallo) |

## Scopo del documento

Il documento presenta la piattaforma regionale **IOService** e descrive le relative modalità di integrazione.

La piattaforma **IOService** si occupa di gestire e inoltrare i messaggi all’ **App IO**.

**IOService** è in grado di ricevere messaggi da inoltrare ad IO tramite diversi connettori disponibili (Excel, WebServices, DataBase, ecc.) fornendo un livello di astrazione e semplificazione delle interfacce di programmazione remota (API) esposte dalla piattaforma nazionale. **IOService** è anche in grado di gestire, memorizzare e schedulare l'invio di messaggi verso gli utenti registrati per ogni servizio.

Il funzionamento della piattaforma IOService è così schematizzabile:



Il documento presenta:

* un’introduzione all’accesso e all’utilizzo della Piattaforma Regionale **IOService** (piattaforma web)
* un’introduzione generale sulle modalità di integrazione, dettagliando successivamente le casistiche dei Web Service esposti
* alcune informazioni per eseguire dei test di pre-esercizio e verificarne il funzionamento

## Piattaforma Regionale IOService

La piattaforma regionale ***IOService*** gestisce un archivio di servizi esposti dagli enti locali sull'App IO. Ad ogni servizio sono associati tutti i dettagli necessari per la comunicazione con il backend della piattaforma nazionale. L'ente registrato e abilitato nella piattaforma nazionale di [[IO Italia]](https://developer.io.italia.it/) può beneficiare del sistema di IOService per l'**invio semplificato dei messaggi**.

|  |
| --- |
| La Piattaforma Regionale ***IOService*** è disponibile all’indirizzo [**https://ioservice.regione.marche.it**](https://ioservice.regione.marche.it)I **Web Services** sono disponibili all’indirizzo pubblico [**https://iows.regione.marche.it**](https://iows.regione.marche.it) |



Per accedere, cliccare sul tasto “**Accedi**” in alto a destra. Dopo la login, l’utente potrà prendere visione:



1. **Menù Ente** - delle informazioni di registrazione dell’Ente (o degli Enti di cui è delegato, nel caso di fornitore delegato)



1. **Menù Servizio** - dei servizi registrati dall’Ente (o dagli Enti di cui è delegato, nel caso di fornitore delegato)



1. **Menù Messaggio** - dei messaggi inviati dall’Ente (o dagli Enti di cui è delegato, nel caso di fornitore delegato e dei messaggi inviati come fornitore tecnologico) e dello stato di invio e consegna



Per lo stato **Inviato**:

* se colore ROSSO: il messaggio non è stato ancora inviato all’App IO
* se colore VERDE: il messaggio è stato inviato (con qualunque esito) all’App IO

Per la notifica di **Esito**:

* se colore ROSSO: il messaggio è stato inviato ma non è stato consegnato (segnalazione di errore)
* se colore GIALLO: il messaggio è stato inviato ma non è stato consegnato perché il cittadino non ha scaricato l’App IO
* se colore VERDE: il messaggio è stato correttamente inviato e consegnato al cittadino che lo ha ricevuto sulla propria App IO
1. **Menù Log** - dei log sulle procedure di integrazione con *IOService*



Il **log** avrà:

* colore ROSSO: se vi è stato un errore bloccante nelle procedure
* colore VERDE: se la procedura è andata a buon fine

## 3.1 Caricamento Messaggi da CSV

Per il solo profilo dell’**Ente Locale**, direttamente all’interno della piattaforma IOService, viene data la possibilità di caricare un file CSV con i messaggi da inviare all’app IO e quindi ai cittadini.

La funzionalità è disponibile alla voce di menu “Messaggio” e poi l’opzione “Scegli file”



Selezionare il file da importare e premere il tasto “Upload”. La procedura restituirà un avviso con i messaggi importati e/o eventualmente scartati



Il file CSV da importare deve essere così formato (separatore dei campi | vedere appendice per impostare il separatore corretto qualora si intenda usare Excel per la composizione del file)

|  |  |
| --- | --- |
| Nome campo | Descrizione |
| ID\_SERVIZIO | OBBLIGATORIOCodice del servizio dell’AppIO ottenuto dal backoffice IO o in alternativa codice servizio attribuito al servizio da IOService.Per i servizi pubblicati da Regione Marche specifici per i Comuni, il ID\_SERVIZIO **DEVE** essere uno dei valori indicati nell’Appence A – Codici servizio per i comuni  |
| OGGETTO\_MSG | OBBLIGATORIOoggetto del messaggio da inviare - tipo string [10 .. 120] |
| CORPO\_MSG | OBBLIGATORIO Corpo del messaggio in formato testo tipo string [80 .. 10000] |
| CODICE\_FISCALE | OBBLIGATORIOCodice fiscale del soggetto a cui inviare il messaggio - tipo string [16] |
| DATA\_SCADENZA\_MSG | OBBLIGATORIOData scadenza messaggio (DD/MM/YYYY) |
| IMPORTO | OPZIONALE, diventa OBBLIGATORIO nel caso la notifica sia relativa ad una scadenza di pagamento. Importo del pagamento espresso in centesimo di Euro - integer [1...9999999999 ].  |
| AVVISO\_PAGOPA | OPZIONALE, diventa OBBLIGATORIO nel caso la notifica sia relativa ad una scadenza di pagamento. Codice avviso pagoPA (IUV) |
| SCADENZA | OPZIONALE, diventa OBBLIGATORIO nel caso la notifica sia relativa ad una scadenza di pagamento. Indica se il pagamento può essere eseguito dopo la data di scadenza del messaggio (Y= si può pagare anche dopo, N= non si può pagare dopo) |
| EMAIL | OPZIONALE Indirizzo mail a cui inviare il messaggio nel caso in cui il soggetto non lo abbia impostato nel suo profilo su app IO |
| NRE | OPZIONALE Numero ricetta elettronica – tipo string 15 |
| IUP | OPZIONALE Identificativo Unico Prenotazione assegnato dal CUP per la prenotazione della prestazione – tipo string 15 |
| CF\_Medico | OPZIONALE Codice fiscale del medico prescrittore |

di seguito un esempio

|  |
| --- |
| ID\_SERVIZIO|OGGETTO\_MSG|CORPO\_MSG|CODICE\_FISCALE|DATA\_SCADENZA\_MSG|IMPORTO|AVVISO\_PAGOPA|SCADENZA|EMAIL|NRE|IUP|CF\_Medico|01AX01AX01AX01AX01AX|NUOVO AGGIORNAMENTO STATO PRATICA 001/2021/USR|SI COMUNICA CHE LA SUA PRATICA E' IN LAVORAZIONE|AAAAAA00A00A000A|30/04/2021|2520|A0001001|Y|prova@prova.it|P0001|I001|SCMSTT0011111 |

*Attenzione*: Il messaggio verrà messo in coda (accedere all’ultima pagina dell’elenco Messaggio) e inviato in modalità schedulata. Di conseguenza, inizialmente, il messaggio presenterà sia il campo Invio che Esito di colore rosso.



## 3.2 Storico messaggi

La piattaforma regionale **IOService** adotta politiche di minimizzazione dei dati al fine di garantire il rispetto del regolamento UE 679/2016 secondo le quali:

* **dopo 30 giorni** (dall’invio all’app IO del messaggio - Giorni Ritenzione Messaggio) le informazioni relative al destinatario (ovvero il codice fiscale) e il corpo del messaggio stesso vengono definitivamente cancellate dagli archivi di IOService.

*Di conseguenza dopo il tempo sopra indicato sarà possibile sapere che un messaggio è stato inviato/consegnato ma non è più possibile risalire al destinatario dello stesso o al suo contenuto. Rimangono possibili esclusivamente verifiche tecniche, ma i dati mantenuti persistenti non hanno più il significato di dati personali, poiché è definitivamente perduto il legame con il soggetto destinatario del messaggio.*

* **dopo 365 giorni** (dall’invio all’app IO del messaggio - Giorni Ritenzione Notifica) anche tutte le informazioni anonime collegate al messaggio stesso vengono definitivamente eliminate dalla base dati di IOService

*Non sono più possibili nemmeno verifiche tecniche, i dati sono fisicamente definitivamente eliminati.*

## Integrazione Web Service

La piattaforma **IOService** mette a disposizione 4 servizi di integrazione tramite Web Service (REST):



I quattro servizi possono essere chiamati previa **Basic Authentication**, definendo come:

* *username* il codice fiscale dell’ente
* *password* la concatenazione dell'AuthIOWS e il CodiceServizio separata dal carattere ':'.

Il **AuthIOWS** verrà fornito all’Ente che si vuole integrare con la piattaforma IOService.

Il **CodiceServizio** può essere alternativamente:

1. per i servizi attivati sul BackOffice IO da soggetti diversi dalla Regione Marche è il “Codice Servizio” rilasciato dal backoffice di IO Italia al URL <https://developer.io.italia.it/> al soggetto delegato dall’ente fruitore del servizio;
2. per i servizi gestiti direttamente dalla Regione Marche è il “Codice Servizio Interno” rilasciato dalla piattaforma IOService. Per i servizi destinati ai comuni utilizzare la tabella in Appence A – Codici servizio per i comuni gestiti da Regione Marche

|  |
| --- |
| *Esempio:* Se l’ente “*Comune di Test Web Service*” con codice fiscale 01234567890 vuole inviare messaggi sull’App IO dei propri cittadini e ha registrato un servizio sul backoffice di IO con codice A001A001A001A001A001A001, la Basic Authentication sarà così composta:* *username*: 01234567890
* *password*: WMWVXCOMIPCQMPEYPDKKHHHAFCJDNT:A001A001A001A001A001A001

dove la prima parte della password (AuthIOWS) verrà fornita da IOService dopo la registrazione dell’ente in piattaforma, mentre la seconda parte è del backoffice dell’App IO (NB: codice servizio e non api key)In maniera equivalente, se è stato comunicato un “Codice Servizio Interno” (es. 01, 02, …) questo sarà utilizzato nella password, come segue:* *username*: 01234567890
* *password*: WMWVXCOMIPCQMPEYPDKKHHHAFCJDNT:01
 |
| *Attenzione*: nella URL del servizio, la componente *{codiceFiscale}* deve essere sostituita con il codice fiscale del soggetto (cittadino) a cui è stato inviato il messaggio mentre *{IdMessaggio}* è il codice identificativo del messaggio in *IOService*. |
| *Attenzione*: Per i servizi di Pagamento MPAY, il Codice Servizio Interno **DEVE** essere il “Progressivo Macro Area per Ente Creditore” (es. 01 TRIBUTI, 02 CORPO DI POLIZIA LOCALE, …, 15 VARIE) come da Tassonomia dei servizi di incasso pubblicata al URL <https://www.pagopa.gov.it/it/pubbliche-amministrazioni/documentazione/> |

L’infrastruttura messa a disposizione dalla Regione Marche è direttamente l’infrastruttura di **produzione**, che invia giornalmente i messaggi ai codici fiscali destinatari.

Di conseguenza, per procedere con la fase di test, si **raccomanda**:

* di verificare lo stato di attivazione del servizio nel backoffice dell’ App IO (<https://developer.io.italia.it/login>).



* di utilizzare solo i codici fiscali destinatari autorizzati (per i test)



*Per chiarire, questo CF (di test – “AAAAAA00A00A000A”) va inserito nel messaggio (la definizione è nel capitolo successivo), al campo* ***codice\_fiscale****: codice fiscale del soggetto a cui inviare il messaggio - tipo string. Utilizzando i “Codici fiscali destinatari autorizzati” (per i test) il messaggio arriverà sull’App IO del delegato che ha creato il Servizio sul BackOffice di [IO]*

* di aver inserito nella sezione degli IP, l’indirizzo di ***IOService*** comunicato da Regione Marche



## Invio dei Messaggi

Per l’invio dei messaggi (anche massivo) ad uno o più soggetti è possibile usare il seguente servizio. La chiamata deve essere fatta in POST. In caso di errore, la motivazione è restituita nel payload della response.

**POST url\_servizio/rest/Messages***(La url\_servizio sarà fornita in fase di configurazione, assieme alle chiavi per le autenticazione al servizio)*

|  |
| --- |
| **REQUEST**AUTHORIZATIONS: Basic Authentication (come definito nei capitoli precedenti)REQUEST BODY SCHEMA: application/json***codice\_fiscale\_fornitore****: codice fiscale di chi invia il messaggio (es. Ente Locale o suo fornitore tecnologico)****lista\_messaggi****: lista di messaggi da inviare, identificabili univocamente – tipo oggetto****progressivo\_messaggio****: progressivo messaggio - tipo integer [ 1 .. 9999999999 ]****codice\_fiscale****: codice fiscale del soggetto a cui inviare il messaggio - tipo string* ***contenuto****: è il messaggio da inviare - tipo oggetto** ***oggetto\_messaggio****: oggetto del messaggio da inviare - tipo string [ 10 .. 120 ]*
* ***corpo\_messaggio****: corpo del messaggio in formato testo o in formato Markdown – tipo string [ 80 .. 10000 ]*
* ***dati\_pagamento****: dati del pagamento da inviare - tipo oggetto*
	+ ***importo****: importo del pagamento in centesimo di Euro - integer [1...9999999999 ]*
	+ ***avviso\_pagoPA****: codice avviso pagoPA, Il formato è <aux digit (1n)>[<application code> (2n)]<codice IUV (15|17n)>, si vedano le specifiche pagoPA per ulteriori informazioni*
	+ ***scadenza\_pagamento****: indica se il pagamento può essere eseguito anche dopo la scadenza -tipo boolean*
* ***dati\_prescrizione: dati relativi alla prescrizione medica – tipo oggetto***
	+ ***numero\_ricetta\_elettronica****: numero ricetta elettronica – tipo string 15*
	+ ***identificativo\_univoco\_prescrizione****: Identificativo Unico Prenotazione assegnato dal CUP per la prenotazione della prestazione – tipo string 15*
	+ ***codice\_fiscale\_prescrittore****: codice fiscale del medico prescrittore*
* ***data\_scadenza:*** *data scadenza messaggio, string <UTCISODateFromString> (Timestamp) A date-time field in ISO-8601 format and UTC timezone.*
* ***email:*** *indirizzo mail a cui inviare il messaggio nel caso in cui il soggetto non lo abbia impostato nel suo profilo su app IO.*
 |
| **RESPONSE****201 Messages created** *(messaggio importato correttamente in IOService – restituisce Id del messaggio)*RESPONSE SCHEMA: application/json* Progressivomessaggio, progressivo messaggio della richiesta, integer
* Idmessaggio, identificativo del messaggio creato, integer

**401 Missing Authorization o Invalid Authorization o Missing Authentication o Invalid Authentication*** *Errore nella login o Utente non autorizzato*

**406 Service Disabled*** *Servizio non attivo*

**400 Invalid payload*** *BadRequest - il messaggio JSON non ha superato la validazione o il progressivo dei messaggi non univoco*

**500 InternalServerError*** *Eccezione generica non gestita*
 |

## Recupero Stato Messaggio

Per recuperare lo stato (di invio/consegna) di un Messaggio all’app IO (per il tramite di IOService) è possibile utilizzare il seguente servizio

**GET url\_servizio/rest/messages/{codice\_fiscale}/{id\_messaggio}**

*(La url\_servizio sarà fornita in fase di configurazione, assieme alle chiavi per le autenticazione al servizio)*

Dove:

* codice\_fiscale: codice fiscale del soggetto (cittadino) a cui è stato inviato il messaggio – tipo string
* id\_messaggio, identificativo del messaggio inviato – tipo string

|  |
| --- |
| **REQUEST**AUTHORIZATIONS: Basic Authentication (come definito nei capitoli precedenti)REQUEST BODY: nessuna informazione |
| **RESPONSE****200 Message found** *(messaggio individuato in IOService)*RESPONSE SCHEMA: application/json* **contenuto**: è il messaggio completo, come precedentemente definito con l’aggiunta delle informazioni
	+ response: Chiave di ACK dell’app IO o Errrore di invio all’app IO
	+ esito: esito dell’invio del messaggio all’app IO – bool (true / false)

**401 Missing Authorization o Invalid Authorization o Missing Authentication o Invalid Authentication*** *Errore nella login o Utente non autorizzato*

**403 Forbidden*** *Servizio non attivo*

**404 No message found for the provided ID.*** *Messaggio non trovato*

**406 Service Disabled*** *Servizio non attivo*

**500 InternalServerError*** *Eccezione generica non gestita*
 |

## Aggiornamento Messaggio

Per aggiornare un messaggio precedentemente inviato a *IOService* è possibile utilizzare il seguente servizio:

**PUT url\_servizio/rest/messages/{codice\_fiscale }/{id\_messaggio}**

 *(La url\_servizio sarà fornita in fase di configurazione, assieme alle chiavi per le autenticazione al servizio)*

Dove:

* codice\_fiscale: codice fiscale del soggetto (cittadino) a cui è stato inviato il messaggio – tipo string
* id\_messaggio, identificativo del messaggio inviato – tipo string

*L’aggiornamento sarà possibile solo se il messaggio non sia già stato inviato all’App IO in quanto non è, ad oggi, possibile rettificare un messaggio inviato all’App IO.*

|  |
| --- |
| **REQUEST**AUTHORIZATIONS: Basic Authentication (come definito nei capitoli precedenti)REQUEST BODY: * **codice\_fiscale**: codice fiscale del soggetto a cui inviare il messaggio - tipo string
* **contenuto**: è il messaggio da inviare - tipo oggetto (vedi relativo oggetto nel capitolo “Invio dei Messaggi”)
 |
| **RESPONSE****200 Message updated** *(messaggio correttamente aggiornato)*RESPONSE SCHEMA: application/json* Idmessaggio, identificativo del messaggio creato, integer

**401 Missing Authorization o Invalid Authorization o Missing Authentication o Invalid Authentication*** *Errore nella login o Utente non autorizzato*

**404 No message found for the provided ID.*** *Messaggio non trovato*

**403 Forbidden** * *Il Messaggio è già stato inviato, non è possibile l’aggiornamento*

**406 Service Disabled*** *Servizio non attivo*

**500 InternalServerError*** *Eccezione generica non gestita*
 |

## Cancellazione Messaggio

Per cancellare un messaggio è possibile utilizzare il seguente servizio:

**DELETE url\_servizio/rest/messages/{codice\_fiscale }/{id\_messaggio}**

 *(La url\_servizio sarà fornita in fase di configurazione, assieme alle chiavi per le autenticazione al servizio)*

Dove:

* codice\_fiscale: codice fiscale del soggetto (cittadino) a cui è stato inviato il messaggio – tipo string
* id\_messaggio, identificativo del messaggio inviato – tipo string

*La cancellazione sarà possibile solo se il messaggio non sia già stato inviato all’App IO in quanto non è, ad oggi, possibile cancellare un messaggio inviato all’App IO.*

|  |
| --- |
| **REQUEST**AUTHORIZATIONS: Basic Authentication (come definito nei capitoli precedenti)REQUEST BODY: nessuna informazione |
| **RESPONSE****200 Message deleted** *(messaggio cancellato in IOService)*RESPONSE SCHEMA: application/json* Idmessaggio, identificativo del messaggio creato, integer

**401 Missing Authorization o Invalid Authorization o Missing Authentication o Invalid Authentication*** *Errore nella login o Utente non autorizzato*

**404 No message found for the provided ID.*** *Messaggio non trovato*

**403 Forbidden** * *Il Messaggio è già stato inviato, non è possibile la cancellazione*

**406 Service Disabled*** *Servizio non attivo*

**500 InternalServerError*** *Eccezione generica non gestita*
 |

## Esempi di integrazione e modalità di test

Per effettuare i test di integrazione con i Web Service della Piattaforma Regionale *IOService* si possono utilizzare diversi strumenti disponibili gratuitamente. Uno tra i tanti è **Postman**.

Volendo quindi utilizzare **Postman** (o strumento equivalente) procedere nel seguente modo:

1. **Accesso a Postman e configurazione della Basic Auth**

Scaricare *Postman* e utilizzarlo senza login, premendo uno dei tasti sotto indicati



Scegliere quindi la seguente opzione



Per ogni servizio da chiamare, impostare la *Basic Autentication* come segue:



*Attenzione: nella URL sostituire* [*http://wsioservice.intra*](http://wsioservice.intra) *con l’indirizzo pubblico dei Web Service di IOService comunicati da Regione Marche.*

1. **Autenticazione WS e Invio del Messaggio (POST)**

Dopo aver configurato l’autenticazione (come descritto in precedenza), configurare il servizio POST per l’invio del messaggio

*POST: http://wsioservice.intra/rest/Messages*



Nella sezione BODY, selezionare



Nell’area sottostante inserire il messaggio da inviare, come esempio seguente, poi premere il tasto “Send”

|  |
| --- |
| {  "codice\_fiscale\_fornitore": "01122334455",  "lista\_messaggi": [    {      "progressivo\_messaggio": 1,      "codice\_fiscale": "SCMGLC85P07A462E",      "contenuto": {        "oggetto\_messaggio": "Prova Invio Messaggio Numero 1 della lista",        "corpo\_messaggio": "Gentile Mario Rossi si invia il primo messagggio della lista. Questo messaggio contiene le informazioni per il pagamento",        "dati\_pagamento": {          "importo": 100,          "avviso\_pagoPA": "4545454545454545454545",          "scadenza\_pagamento": **true**        },        "data\_scadenza": "2021-06-29T01:21:51.2160000Z",        "email": "prova@prova.it"      }    },    {      "progressivo\_messaggio": 2,      "codice\_fiscale": "SCMGLC85P07A462E",      "contenuto": {        "oggetto\_messaggio": "Prova Invio Messaggio Numero 2 della lista",        "corpo\_messaggio": "Gentile Mario Rossi si invia il secondo messagggio della lista.  Questo messaggio contiene le informazioni delle prescrizioni mediche",        "dati\_prescrizione": {          "numero\_ricetta\_elettronica": "A001B002",          "identificativo\_univoco\_prescrizione": "C001D002",          "codice\_fiscale\_prescrittore": "VNTSDR61P47A462K"        },        "data\_scadenza": "2021-06-29T01:21:51.2160000Z",        "email": "sample string 4"      }    }  ]} |

La *Response* fornirà, per ogni messaggio, l’Identificativo di rintracciabilità (IdMessaggio) in *IOService* (formato JSON)



1. **Accesso alla Piattaforma e Verifica inserimento su Piattaforma Regionale**

Accedendo alla piattaforma IOService, l’utente può verificare il corretto caricamento del messaggio (nella sezione **Messaggi**)



1. **Autenticazione WS e Modifica Messaggio**

Dopo aver configurato l’autenticazione (come descritto in precedenza), configurare il servizio PUT per la modifica di un messaggio precedentemente inviato

*PUT:* [http://wsioservice.intra/rest/Messages/{codice\_fiscale\_cittadino}/{*IdMessaggio*}](http://wsioservice.intra/rest/Messages/%7Bcodice_fiscale_cittadino%7D/%7BIdMessaggio%7D)

Configurare il BODY come punto 2, e nell’area sottostante inserire le informazioni del messaggio da rettificare, come esempio seguente, poi premere il tasto “Send”

|  |
| --- |
| {   "codice\_fiscale": "SCMGLC85P07A462E",      "contenuto": {        "oggetto\_messaggio": "Prova Invio Messaggio Numero 1 della lista MODIFICATO",        "corpo\_messaggio": "Gentile Mario Rossi si invia il primo messagggio della lista. Questo messaggio contiene le informazioni per il pagamento MODIFICATO",        "dati\_pagamento": {          "importo": 150,          "avviso\_pagoPA": "000010010000000110001",          "scadenza\_pagamento": **false**        },        "data\_scadenza": "2022-06-29T01:21:51.2160000Z",        "email": "prova@provamodificato.it"      }} |

1. **Autenticazione WS e Cancellazione Messaggio**

Dopo aver configurato l’autenticazione (come descritto in precedenza), configurare il servizio DELETE per la cancellazione di un messaggio precedentemente inviato

*DELETE: http://wsioservice.intra/rest/Messages/{codice\_fiscale\_cittadino}/{id\_messaggio}*

Non inserire nulla nel BODY e premere il tasto “Send”

1. **Accesso alla Piattaforma (giorno successivo) e verifica Invio ed Esito**

Accedendo alla Piattaforma è possibile prendere visione dell’esito dell’invio e della consegna



1. **Autenticazione WS e Recupero Stato Messaggio**

Dopo aver configurato l’autenticazione (come descritto in precedenza), configurare il servizio GET per ottenere lo stato (e tutte le informazioni) di un messaggio precedentemente inviato a IOService

*GET:* [http://wsioservice.intra/rest/Messages/{codice\_fiscale\_cittadino}/{id\_messaggio}](http://wsioservice.intra/rest/Messages/%7Bcodice_fiscale_cittadino%7D/%7Bid_messaggio%7D)



## Appence A – Codici servizio per i comuni gestiti da Regione Marche

La Regione Marche ha pubblicato nel backoffice AppIO per i comuni aderenti al Progetto DigiPalm i seguenti servizi di notifica, così identificati.

|  |  |
| --- | --- |
| **Codice** | **Descrizione** |
| 01 | TRIBUTI |
| 02 | CORPO DI POLIZIA LOCALE |
| 03 | RAGIONERIA GENERALE |
| 04 | DEMOGRAFICI ELETTORALI E STATISTICI |
| 05 | STAZIONE UNICA APPALTANTE, CONTRATTI E GESTIONE LAVORI PUBBLICI ED EDILIZIA PRIVATA |
| 06 | TUTELA AMBIENTALE, SANITÀ E IGIENE PUBBLICA |
| 07 | POLITICHE EDUCATIVE GIOVANILI E DEL LAVORO |
| 08 | CULTURA, RELIGIONI, PARI OPPORTUNITÀ, SPORT |
| 09 | SERVIZI ALLA PERSONA E SERVIZI SOCIALI |
| 10 | LOCAZIONE LOCALI E IMPIANTI |
| 11 | SVILUPPO ECONOMICO |
| 12 | SERVIZI IDRICI |
| 13 | AVVOCATURA CIVICA |
| 14 | PATRIMONIO |
| 15 | VARIE oppure Pagamento MPay |

Questi servizi corrispondono alle macroaree di attività dei comuni individuate alla tabella Tassonomia di pagamento pubblicata da PagoPA al url:

https://www.pagopa.gov.it/it/pubbliche-amministrazioni/documentazione/

## Appendice B – Come impostare il separatore dei file csv in Windows

Il carattere che Excel adotta per separare i campi quando si salva un file .xlsx in formato .csv è quello definito nelle impostazioni regionali del sistema operativo, che si solito corrisponde al carattere **;** salvo eccezioni.

Per modificare il separatore procedere nel seguente modo:

* Aprire il pannello di controllo di windows e individuare “Area geografica”, fare click sull’icona corrispondente, comparirà il form intitolato “Area geografica”



* Fare click su “Impostazioni aggiuntive”, comparirà il form intitolato “Personalizza formato”. Qui nel campo “Separatore di elenco” cancellare il simbolo **;** e inserire il simbolo **|.**

Quindi Fare click su “Applica”, poi su “OK”.

 

* Ricompare il form “Area geografica” fare ancora click su “Applica”, quindi su “OK”. Ora il separatore impostato è quello voluto.

