

Periodici - API REST/JSON (v4.6)

- Convenzioni
 - Struttura della risposta
 - Stato dell'istanza di abbonamento
 - § Formato timestamp
- API fascicoli
 - find_issues
 - get_subscription_issues
- API anagrafiche
 - get_customer
 - create_customer
 - update_customer
 - request_customer_merge
 - authenticate_by_cod_abbo
 - get_id_customer_by_identity_uid
 - get_id_customer_by_cod_abbo
- API abbonamenti
 - get_customer_subscriptions
 - get_subscription
 - create_subscription
 - update_subscription_options
 - update_subscription_offering
 - pay_subscription
- API offerte e opzioni
 - get_offering
 - get_renewal_offering
 - get_option
- API fatture
 - get_customer_invoices
- API estrazione dati
 - find_subscriptions_by_action
 - § find_subscription_status_changes
- API per download file
 - get_invoice
 - get_payment_form
- Lookup - Codifiche statiche
 - id_tipo_anagrafica
 - id_magazine
 - id_payment_type
 - id_job
 - id_qualification
 - id_document_type
 - feature_tags



Convenzioni

Struttura della risposta

Al fine di mantenere il sistema di comunicazione unico e coerente su tutte le piattaforme, sarà necessario adottare uno standard di comunicazione definito dal seguente schema:

RICHIESTA	->	RISPOSTA
http POST	<-	JSON

La richiesta dovrà essere effettuata esclusivamente con HTTP POST ed entrambi i documenti JSON di richiesta e risposta saranno con charset UTF-8. Inoltre, il contenuto della richiesta dovrà avere una struttura definita dallo schema seguente:

status - Una stringa di due lettere rappresentate l'esito della comunicazione, i valori possibili sono: ER OK

error_code - Una stringa rappresentante un codice d'errore preso da una lista concordata tra le piattaforme

error_message - Eventuale messaggio d'errore

data - Un oggetto JSON rappresentante l'eventuale payload di risposta. Può anche essere NULL

Codici error_code:

0	NO_ERROR
1	INTERNAL_ERROR
2	EMPTY_PARAMETER
3	WRONG_PARAMETER_VALUE
4	WRONG_ACCESS_KEY
5	DATA_NOT_FOUND

Esempio:

```
{
  status: "OK",
  error_code: "0",
  error_message: "All went fine. ",
  data: { ... }
}
```

Sono indicati con § le funzioni o i valori cambiati rispetto alla versione precedente di queste API.

Stato dell'istanza di abbonamento

Lo stato dell'istanza di abbonamento si ottiene considerando alcuni valori di *get_subscription* o *get_customer_subscriptions*

In particolare, le istanze per cui **is_deferred_bill = true**, non hanno necessità di pagamento e vanno sempre considerate saldate e regolarizzate. Si tratta di abbonamenti pagati con fatturazione differita, quindi gestita dall'amministrazione.

Le istanze con **is_free_gift = true** sono omaggi dell'editore e vanno considerate saldate e regolarizzate.

Le istanze in cui **is_blocked = true** sono state disdettate e bloccate con urgenza. Quindi tutti i contenuti digitali abbinati dovrebbero risultare inaccessibili.

Tutti i casi in cui **is_paid = true** sono istanze che risultano saldate e regolarizzate.

Talvolta potrebbe verificarsi la situazione in cui un'abbonamento/istanza è saldato con **is_paid = true** ma una sua *option* non lo è. Ovvero, nell'array *options* **is_paid = false**

§ Formato timestamp

I dati indicati come 'timestamp' sono un intero di tipo **long**. Il numero fa riferimento all'orario interno del server dove risiedono le API.

API fascicoli

find_issues

Funzione che permette la ricerca dei fascicoli del periodico specificato con data nominale di uscita compresa nell'intervallo di date (estremi inclusi)

Endpoint: `http://base_url/find_issues`

Parametri POST:

access_key - chiave di riconoscimento

id_magazine - identificativo del periodico

dt_begin - data inizio ricerca

dt_end - data fine ricerca

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  issues: [
    {
      cm_issue: <string>, /* identificativo fascicolo */
      nominal_issue_date: "yyyy-mm-dd", /* data nominale, inizio
validità fascicolo */
      nominal_issue_end: "yyyy-mm-dd", /* data di fine validità
fascicolo */
      worth: <number>, /* numero di fascicoli aggregati: 1 fascicolo
singolo, 2 fascicolo doppio, 0 fascicolo speciale */
      description_number: <string>, /* numero/titolo del fascicolo */
      description_period: <string> /* descrizione del periodo di
uscita, esempio: "novembre-dicembre" */
      id_option: <string>, /* distribuito solo agli abbonati a questa
opzione */
    },
    ...
  ]
}
```

get_subscription_issues

Funzione che permette la ricerca dei fascicoli già pubblicati per l'istanza di abbonamento data

Endpoint: `http://base_url/get_subscription_issues`

Parametri POST:

access_key - chiave di riconoscimento

id_subscription - identificativo dell'istanza di abbonamento

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```

{
  issues: [
    {
      cm_issue: <string>, /* identificativo fascicolo */
      nominal_issue_date: "yyyy-mm-dd", /* data nominale, inizio
validità fascicolo */
      nominal_issue_end: "yyyy-mm-dd", /* data di fine validità
fascicolo */
      worth: <number>, /* numero di fascicoli aggregati: 1 fascicolo
singolo, 2 fascicolo doppio, 0 fascicolo speciale */
      description_number: <string>, /* numero/titolo del fascicolo */
      description_period: <string>, /* descrizione del periodo di
uscita, esempio: "novembre-dicembre" */
      id_option: <string>, /* distribuito solo agli abbonati a questa
opzione */
      media_app: <"true"|"false">, /* distribuito tramite app */
      media_paper: <"true"|"false">, /* distribuito su carta stampata */
    },
    ...
  ]
}

```

API anagrafiche

get_customer

Funzione per ottenere tutti i dati relativi all'anagrafica cliente indicata.

Endpoint: http://base_url/get_customer

Parametri POST:

access_key - chiave di riconoscimento

id_customer - identificativo dell'anagrafica cliente

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```

{
  id_customer: <string>, /* identificativo dell'anagrafica cliente (in
caso di merge, potrebbe essere diverso da quello fornito) */
  address_title: <string>, /* titolo appellativo */
  address_first_name: <string>, /* nome di battesimo (opzionale) */
  address_last_name_company: <string>, /* cognome o ragione sociale */
  address_co: <string>, /* presso (opzionale) */
  address_address: <string>, /* indirizzo */
  address_locality: <string>, /* localita */
  address_province: <string>, /* sigla provincia (opzionale se estero)
*/
  address_zip: <string>, /* CAP (opzionale se estero) */
  address_country_code: <string>, /* codice nazione ISO 3166 (IT, FR,

```

```

UK...) */
    billing_title: <string>, /* titolo appellativo (opzionale) */
    billing_first_name: <string>, /* nome di battesimo (opzionale) */
    billing_last_name_company: <string>, /* cognome o ragione sociale
(opzionale) */
    billing_co: <string>, /* fatturazione/presso (opzionale) */
    billing_address: <string>, /* fatturazione/indirizzo (opzionale) */
    billing_locality: <string>, /* fatturazione/localita (opzionale) */
    billing_province: <string>, /* sigla provincia (opzionale) */
    billing_zip: <string>, /* fatturazione/CAP (opzionale) */
    billing_country_code: <string>, /* fatturazione/codice nazione ISO
3166 (IT, FR, UK...) (opzionale) */
    sex: <"m"|"f"|"">, /* sesso (opzionale) */
    cod_fisc: <string>, /* codice fiscale */
    piva: <string>, /* partita iva (opzionale) */
    cod_destinatario: <string>, /* codice destinatario/intermediario
agenzia entrate */
    pa: <"true"|"false">, /* anagrafica di pubblica amministrazione */
    cuf: <string>, /* codice unico ufficio PA (opzionale) */
    phone_mobile: <string>, /* cellulare (opzionale) */
    phone_landline: <string>, /* telefono fisso (opzionale) */
    email_primary: <string>, /* email primaria (opzionale) */
    pec: <string>, /* email posta certificata (opzionale) */
    adottatario: <"true"|"false">, /* è un adottatario libri Giunti
Scuola (opzionale) */
    identity_uid: <string>, /* lo uid con cui questa utenza è
riconosciuta tramite il Giunti Identity Manager (opzionale) */
    id_job: <integer>, /* id professione (opzionale) */
    id_qualification: <integer>, /* id titolo di studio (opzionale) */
    birth_date: "yyyy-mm-dd", /* data di nascita (opzionale) */
    consent_tos: <"true"|"false">, /* consenso terms of service */
    consent_marketing: <"true"|"false">, /* consenso comunicazioni
marketing */
    consent_profiling: <"true"|"false">, /* consenso profilazione */
    consent_update_date: "yyyy-mm-dd", /* data ultimo aggiornamento del
consenso */
    modified_date: "yyyy-mm-dd" /* data di ultima modifica */
}

```

create_customer

Funzione per la creazione di nuove anagrafiche cliente immediatamente valide. E' immediatamente valida e sarà sottoposta a una approvazione successiva da parte degli operatori.

Endpoint: http://base_url/create_customer

Parametri POST:

access_key - chiave di riconoscimento

human_check - <"true"|"false"> se "true" o non presente, i dati dovranno essere approvati dagli operatori (opzionale, **default="true"**)

id_tipo_anagrafica - <pvt|azn|scu...> id tipo anagrafica: privato, azienda, scuola...

sex - <"m"|"f"> (opzionale) non specificare per persone giuridiche

address_title - [6] titolo appellativo (opzionale)

address_first_name - [25] nome di battesimo (opzionale)
address_last_name_company - [30] cognome o ragione sociale
address_co - [28] presso (opzionale)
address_address - [36] indirizzo
address_locality - [36] localita
address_province - [2] sigla provincia (opzionale se estero)
address_zip - [5] CAP (opzionale se estero)
address_country_code - [2] codice nazione ISO 3166
billing_title - [6] titolo appellativo (opzionale)
billing_first_name - [25] nome di battesimo (opzionale)
billing_last_name_company - [30] cognome o ragione sociale
billing_co - [28] fatturazione/presso (opzionale)
billing_address - [36] fatturazione/indirizzo (opzionale)
billing_locality - [36] fatturazione/localita (opzionale)
billing_province - [2] sigla provincia (opzionale)
billing_zip - [5] fatturazione/CAP (opzionale)
billing_country_code - [2] fatturazione/codice nazione ISO 3166 (opzionale)
cod_fisc - [16] codice fiscale
piva - [16] partita iva (opzionale)
cod_destinatario - [8] codice destinatario/intermediario agenzia entrate (opzionale)
pa - <"true"/>"false"> anagrafica di pubblica amministrazione (opzionale, default "false")
cuf - [16] codice unico ufficio PA (opzionale)
phone_mobile - [32] cellulare (opzionale)
phone_landline - [32] telefono fisso (opzionale)
email_primary - [64] email primaria (opzionale)
pec - [64] email posta certificata (opzionale)
adottatario - <"true"/>"false"> è un adottatario Giunti Scuola (opzionale)
identity_uid - [16] lo *uid* con cui questa utenza è riconosciuta tramite il Giunti Identity Manager (opzionale)
id_job - id professione (opzionale)
id_qualification - id titolo di studio (opzionale)
birth_date - "yyyy-mm-dd" data di nascita (opzionale)
consent_tos - <"true"/>"false"> consenso terms of service
consent_marketing - <"true"/>"false"> consenso comunicazioni marketing
consent_profiling - <"true"/>"false"> consenso profilazione

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  id_customer: <string>
}
```

update_customer

Funzione per la modifica di una anagrafica cliente esistente. E' immediatamente valida e sarà sottoposta a una approvazione successiva da parte degli operatori.

Endpoint: http://base_url/update_customer

Parametri POST:

access_key - chiave di riconoscimento
id_customer - identificativo dell'anagrafica da aggiornare
human_check - <"true"/>"false"> se "true" o non presente, i dati dovranno essere approvati dagli operatori (opzionale, **default="true"**)
id_tipo_anagrafica - <"pvt"/>"azn"/>"scu"...> id tipo anagrafica: privato, azienda, scuola...
sex - <"m"/>"f"> (opzionale) non specificare per persone giuridiche
address_title - [6] titolo appellativo (opzionale)
address_first_name - [25] nome di battesimo (opzionale)
address_last_name_company - [30] cognome o ragione sociale
address_co - [28] presso (opzionale)
address_address - [36] indirizzo
address_locality - [36] localita
address_province - [2] sigla provincia
address_zip - [5] CAP (opzionale se estero)
address_country_code - [2] codice nazione ISO 3166
billing_title - [6] titolo appellativo (opzionale)
billing_first_name - [25] nome di battesimo (opzionale)

billing_last_name_company - [30] cognome o ragione sociale
billing_co - [28] fatturazione/presso (opzionale)
billing_address - [36] fatturazione/indirizzo (opzionale)
billing_locality - [36] fatturazione/località (opzionale)
billing_province - [2] sigla provincia (opzionale)
billing_zip - [5] fatturazione/CAP (opzionale)
billing_country_code - [2] fatturazione/codice nazione ISO 3166 (opzionale)
cod_fisc - [16] codice fiscale (opzionale)
piva - [16] partita iva (opzionale)
cod_destinatario - [16] codice destinatario/intermediario agenzia entrate (opzionale)
pa - <"true"/>"false"> anagrafica di pubblica amministrazione (opzionale, default "false")
cuf - [16] codice unico ufficio PA (opzionale)
phone_mobile - [32] cellulare (opzionale)
phone_landline - [32] telefono fisso (opzionale)
email_primary - [64] email primaria (opzionale)
pec - [64] email posta certificata (opzionale)
adottatario - <"true"/>"false"> è un adottatario Giunti Scuola (opzionale)
identity_uid - [16] lo *uid* con cui questa utenza è riconosciuta tramite il Giunti Identity Manager (opzionale)
id_job - id professione (opzionale)
id_qualification - id titolo di studio (opzionale)
birth_date - "yyyy-mm-dd" data di nascita (opzionale)
consent_tos - <"true"/>"false">, //consenso terms of service
consent_marketing - <"true"/>"false">, //consenso comunicazioni marketing
consent_profiling - <"true"/>"false"> //consenso profilazione

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  id_customer: <string> /* se c'è stato un merge, potrebbe essere
diverso da quello fornito */
}
```

request_customer_merge

Funzione che permette di marcare come da abbinare due anagrafiche esistenti, in modo da unire sotto un solo cliente tutti gli abbonamenti e gli oggetti collegati. L'abbinamento non è automatico, sarà sottoposto al vaglio degli operatori e potrebbe necessitare di alcuni giorni prima di essere eventualmente approvato.

Endpoint: http://base_url/request_customer_merge

Parametri POST:

access_key - chiave di riconoscimento
id_customer - identificativo dell'anagrafica principale
id_customer_proposed - identificativo dell'anagrafica da fondere nella principale

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  request_enqueued: <"true" | "false">
}
```

authenticate_by_cod_abbo

Funzione che permette di ottenere l'identificativo dell'anagrafica di un cliente se c'è una corrispondenza tra la sigla della provincia e uno dei codici abbonamento di cui è beneficiario. Una mancata corrispondenza genererà un errore nella risposta.

Endpoint: http://base_url/authenticate_by_cod_abbo

Parametri POST:

access_key - chiave di riconoscimento

cod_abbo - [6-7] identificativo di una sequenza di abbonamenti negli anni [7], in alternativa identificativo cliente [6]

address_province - [2] sigla provincia

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  id_customer: <string>
}
```

get_id_customer_by_identity_uid

Permette di ottenere l'identificativo dell'anagrafica di un cliente a partire dall'identificativo identity_uid del Giunti Identity Manager. Una mancata corrispondenza genererà un errore nella risposta.

Endpoint: `http://base_url/get_id_customer_by_identity_uid`

Parametri POST:

access_key - chiave di riconoscimento

identity_uid - [16] identificativo unico restituito da Giunti Identity Manager

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  id_customer: <string>
}
```

get_id_customer_by_cod_abbo

Permette di ottenere l'identificativo dell'anagrafica di un cliente a partire da uno dei codici abbonamento di cui è beneficiario. Una mancata corrispondenza genererà un errore nella risposta.

Endpoint: `http://base_url/get_id_customer_by_cod_abbo`

Parametri POST:

access_key - chiave di riconoscimento

cod_abbo - [7] identificativo di una sequenza di abbonamenti negli anni

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  id_customer: <string>
}
```

API abbonamenti

get_customer_subscriptions

Restituisce l'elenco degli abbonamenti del cliente. Sarà presente solamente l'ultima istanza di tutte quelle con lo stesso codice abbonamento. Nel caso in cui esistano istanze con inizio nel futuro saranno mostrate anche loro, oltre a quella in corso. L'elenco include sia gli abbonamenti di cui il cliente è beneficiario sia quelli per cui è solamente pagante (regalati ad altri).

Endpoint: `http://base_url/get_customer_subscriptions`

Parametri POST:

access_key - chiave di riconoscimento

id_customer - identificativo del periodico

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  subscriptions: [
    {
      cod_abbo: <string>, /* identificativo della sequenza di
abbonamenti negli anni */
      id_subscription: <string>, /* identificativo dell'istanza di
abbonamento */
      id_magazine: <string>, /* identificativo periodico */
      id_offering: <string>, /* identificativo del listino/tipo
abbonamento */
      id_renewal_offering: <string>, /* SE PRESENTE identificativo del
listino(tipo abbonamento) al rinnovo (opzionale) */
      id_customer_recipient: <string>, /* identificativo beneficiario */
      id_customer_payer: <string>, /* identificativo pagante SOLO SE
DIVERSO da beneficiario (opzionale) */
      options: [ /* opzioni abbinate all'istanza */
        {
          id_option: <string>, /* identificativo opzione */
          name: <string>, /* descrizione opzione */
          is_included: <"true"|"false">, /* se l'opzione è inclusa nel
prezzo */
          price: <number>, /* prezzo, se l'opzione non è inclusa
nell'offerta (opzionale) */
          is_paid: <"true"|"false">, /* se è presente un pagamento per
l'opzione */
        },
        ...
      ],
      quantity: <number>, /* numero di copie */
      cm_first_issue: <number>, /* primo fascicolo a cui ha diritto */
      cm_last_issue: <number>, /* ultimo fascicolo a cui ha diritto */
      subscription_begin: "yyyy-mm-dd", /* data nominale inizio istanza
abbonamento */
      subscription_end: "yyyy-mm-dd", /* data nominale fine istanza
abbonamento */
      is_paid: <"true"|"false">, /* se è presente un pagamento per
l'abbonamento base */
      is_deferred_bill: <"true"|"false">, /* se sarà pagato tramite
fatturazione differita */
      is_free_gift: <"true"|"false">, /* se è omaggio */
      is_blocked: <"true"|"false">, /* se è bloccato/annullato */
      is_purchase_proposal: <"true"|"false">, /* se è una proposta di
abbonamento creata da Giunti (opzionale) */
    }
  ]
}
```

```

        media_app: <"true"|"false">, /* se è distribuito tramite app */
        media_paper: <"true"|"false">, /* se è distribuito su carta
stampata */
        feature_tags: [ /* elenco dei tag dei servizi attivi per
l'istanza (opzionale) */
            <string>,
            ...
        ],
        price: <number>, /* prezzo dell'offerta */
        paid_amount: <number>, /* importo già pagato */
        cancellation_request_date: "yyyy-mm-dd", /* data disdetta */
        issues_total: <number>, /* numeri totali */
        issues_past: <number>, /* numeri già pubblicati o spediti */
        initial_gracing_end_date: "yyyy-mm-dd", /* fine della fase di
"gracing iniziale" */
        offering_stop_date: "yyyy-mm-dd", /* fine validità dell'offerta
in attesa di pagamento */
        charge_warning_date: "yyyy-mm-dd", /* data dell'avviso di
addebito su carta per l'istanza in corso */
        automatic_charge_date: "yyyy-mm-dd", /* data di addebito su carta
per l'istanza in corso */
        renewal_enabled_date: "yyyy-mm-dd", /* data da cui l'istanza è
rinnovabile */
        renewal_warning_date: "yyyy-mm-dd", /* data dell'avviso di
rinnovo con addebito su carta */
        automatic_renewal_date: "yyyy-mm-dd", /* data di rinnovo
automatico con addebito su carta */
        final_gracing_end_date: "yyyy-mm-dd", /* fine della fase di
"gracing finale" */
        modified_date: "yyyy-mm-dd" /* data di ultima modifica */
    },
    ...
]
}

```

get_subscription

Funzione per ottenere i dati di un'istanza di abbonamento

Endpoint: http://base_url/get_subscription

Parametri POST:

access_key - chiave di riconoscimento

id_subscription - identificativo dell'istanza di abbonamento

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```

{
    cod_abbo: <string>, /* identificativo della sequenza di abbonamenti
negli anni */

```

```

    id_subscription: <string>, /* identificativo dell'istanza di
abbonamento */
    id_magazine: <string>, /* identificativo periodico */
    id_offering: <string>, /* identificativo del listino/tipo abbonamento
*/
    id_renewal_offering: <string>, /* SE PRESENTE identificativo del
listino(tipo abbonamento) al rinnovo (opzionale) */
    id_customer_recipient: <string>, /* identificativo beneficiario */
    id_customer_payer: <string>, /* identificativo pagante SOLO SE
DIVERSO da beneficiario (opzionale) */
    options: [ /* opzioni abbinate all'istanza */
        {
            id_option: <string>, /* identificativo opzione */
            name: <string>, /* descrizione opzione */
            is_included: <"true"|"false">, /* se l'opzione è inclusa nel
prezzo */
            price: <number>, /* prezzo, se l'opzione non è inclusa
nell'offerta (opzionale) */
            is_paid: <"true"|"false">, /* se è presente un pagamento per
l'opzione */
        },
        ...
    ]
    quantity: <number>, /* numero di copie */
    cm_first_issue: <number>, /* primo fascicolo a cui ha diritto */
    cm_last_issue: <number>, /* ultimo fascicolo a cui ha diritto */
    subscription_begin: "yyyy-mm-dd", /* data nominale inizio istanza
abbonamento */
    subscription_end: "yyyy-mm-dd", /* data nominale fine istanza
abbonamento */
    is_paid: <"true"|"false">, /* se è presente un pagamento per
l'abbonamento base */
    is_deferred_bill: <"true"|"false">, /* se sarà pagato tramite
fatturazione differita */
    is_free_gift: <"true"|"false">, /* se è omaggio */
    is_blocked: <"true"|"false">, /* se è bloccato/annullato */
    is_purchase_proposal: <"true"|"false">, /* se è una proposta di
abbonamento creata da Giunti (opzionale) */
    media_app: <"true"|"false">, /* se è distribuito tramite app */
    media_paper: <"true"|"false">, /* se è distribuito su carta stampata
*/
    feature_tags: [ /* elenco dei tag dei servizi attivi per l'istanza
(opzionale) */
        <string>,
        ...
    ],
    price: <number>, /* prezzo dell'offerta */
    paid_amount: <number>, /* importo già pagato */
    cancellation_request_date: "yyyy-mm-dd", /* data disdetta */
    issues_total: <number>, /* numeri totali */

```

```

    issues_past: <number>, /* numeri già pubblicati o spediti */
    initial_gracing_end_date: "yyyy-mm-dd", /* fine della fase di
"gracing iniziale" */
    offering_stop_date: "yyyy-mm-dd", /* fine validità dell'offerta in
attesa di pagamento */
    charge_warning_date: "yyyy-mm-dd", /* data dell'avviso di addebito su
carta per l'istanza in corso */
    automatic_charge_date: "yyyy-mm-dd", /* data di addebito su carta per
l'istanza in corso */
    renewal_enabled_date: "yyyy-mm-dd", /* data da cui l'istanza è
rinnovabile */
    renewal_warning_date: "yyyy-mm-dd", /* data dell'avviso di rinnovo
con addebito su carta */
    automatic_renewal_date: "yyyy-mm-dd", /* data di rinnovo automatico
con addebito su carta */
    final_gracing_end_date: "yyyy-mm-dd", /* fine della fase di "gracing
finale" */
    modified_date: "yyyy-mm-dd" /* data di ultima modifica */
}

```

create_subscription

Funzione che permette la creazione di una nuova istanza di abbonamento

Nel caso in cui l'offerta identificata in `id_offering` non corrisponda allo stesso periodico indicato in `id_magazine`, il valore in `id_magazine` sarà ignorato e sarà scartato un eventuale `cod_abbo`. In altri termini, quando viene scelta un'offerta di un altro periodico, il risultato è un nuovo abbonamento all'altro periodico, con un nuovo codice abbonamento.

Endpoint: `http://base_url/create_subscription`

Parametri POST:

access_key - chiave di riconoscimento
cod_abbo - [8] identificativo della sequenza di abbonamenti negli anni, SE POPOLATO => RINNOVO (opzionale)
id_magazine - identificativo periodico
id_offering - identificativo del listino/tipo abbonamento
id_customer_recipient - identificativo beneficiario
id_customer_payer - identificativo pagante, popolato SOLO SE DIVERSO da beneficiario (opzionale)
options - [256] stringa con elenco identificativi separati da ";" (opzionale)
quantity - [4] numero di copie
cm_first_issue - primo fascicolo a cui ha diritto
cm_last_issue - ultimo fascicolo a cui ha diritto
id_payment_type - tipo pagamento (opzionale)
payment_amount - "#0.00" importo (opzionale)
payment_date - "yyyy-mm-dd" data pagamento (opzionale)
payment_trn - [128] identificativo transazione (opzionale)
payment_note - [32] note sul pagamento (opzionale)
invoice_row_annotation - [64] annotazione da aggiungere a ciascun articolo della fattura (opzionale)

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  cod_abbo: <string>, /* identificativo della sequenza di abbonamenti
negli anni */
  id_subscription: <string> /* identificativo dell'istanza di
abbonamento */
}
```

update_subscription_options

Funzione che permette l'aggiunta di una opzione, fornendo i dati del pagamento abbinato, se è stato effettuato.

Endpoint: http://base_url/update_subscription_options

Parametri POST:

access_key - chiave di riconoscimento
id_subscription - identificativo dell'istanza di abbonamento
options - [256] stringa con elenco identificativi separati da ";"
id_payment_type - tipo pagamento (opzionale)
payment_amount - "#0.00" importo (opzionale)
payment_date - "yyyy-mm-dd" data pagamento (opzionale)
payment_trn - [128] identificativo transazione (opzionale)
payment_note - [32] note sul pagamento (opzionale)
invoice_row_annotation - [64] annotazione da aggiungere a ciascun articolo della fattura (opzionale)

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  cod_abbo: <string>, /* identificativo della sequenza di abbonamenti
negli anni */
  id_subscription: <string> /* identificativo dell'istanza di
abbonamento */
}
```

update_subscription_offering

Funzione che permette il cambio di listino/tipo abbonamento. E' disponibile solamente per istanze di abbonamenti di riviste che seguono l'anno scolastico in condizioni ben precise:

- Istanza non pagata e non disdettata
- Istanza con inizio non antecedente ai 90 giorni
- I listini vecchio e nuovo devono essere della stessa durata, senza fatturazione differita e solo zona Italia

Endpoint: http://base_url/update_subscription_offering

Parametri POST:

access_key - chiave di riconoscimento
id_subscription - identificativo dell'istanza di abbonamento
id_offering - identificativo del nuovo listino/tipo abbonamento

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  cod_abbo: <string>, /* identificativo della sequenza di abbonamenti
negli anni */
  id_subscription: <string> /* identificativo dell'istanza di
abbonamento */
}
```

pay_subscription

Funzione che permette di saldare **completamente** un abbonamento (incluse opzioni già abbinate) comunicando un pagamento. Attenzione, comunicando un importo parziale l'abbonamento risulterà comunque saldato con quell'importo.

Endpoint: http://base_url/pay_subscription

Parametri POST:

access_key - chiave di riconoscimento

id_subscription - identificativo dell'istanza di abbonamento

id_payment_type - tipo pagamento

payment_amount - "#0.00" importo

payment_date - "yyyy-mm-dd" data pagamento

payment_trn - [128] identificativo transazione (opzionale)

payment_note - [32] note sul pagamento (opzionale)

invoice_row_annotation - [64] annotazione da aggiungere a ciascun articolo della fattura (opzionale)

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  cod_abbo: <string>, /* identificativo della sequenza di abbonamenti
negli anni */
  id_subscription: <string> /* identificativo dell'istanza di
abbonamento */
}
```

API offerte e opzioni

get_offering

Funzione per ottenere le caratteristiche di un listino (tipo abbonamento)

Endpoint: http://base_url/get_offering

Parametri POST:

access_key - chiave di riconoscimento

id_offering - identificativo del listino

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```

{
  id_offering: <string>, /* identificativo del listino/tipo abbonamento
*/
  id_magazine: <string>, /* identificativo periodico */
  name: <string> /* descrizione del listino/tipo abbonamento */
  price: <number>, /* prezzo listino (con le opzioni incluse) */
  included_issues_number: <number> /* durata in fascicoli */
  included_options: [ /* opzioni obbligatorie=incluse con questa
offerta */
    {
      id_option: <string>, /* identificativo opzione */
      name: <string> /* descrizione opzione */
    }
    ...
  ]
}

```

get_renewal_offering

Funzione che, in base al listino attuale, restituisce il listino al rinnovo disponibile nel giorno specificato.

Endpoint: http://base_url/get_renewal_offering

Parametri POST:

access_key - chiave di riconoscimento

id_offering - identificativo del listino

renewal_date - data in cui dovrà iniziare l'istanza dell'abbonamento rinnovato

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```

{
  id_offering: <string>, /* identificativo del listino/tipo abbonamento
originale */
  id_magazine: <string>, /* identificativo periodico */
  id_renewal_offering: <string>, /* SE DEFINITO identificativo del
listino/tipo abbonamento da assegnare al rinnovo (opzionale) */
}

```

get_option

Funzione che restituisce i dati riguardanti un'opzione

Endpoint: http://base_url/get_option

Parametri POST:

access_key - chiave di riconoscimento

id_option - identificativo dell'opzione

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  id_option: <string>, /* identificativo dell'opzione */
  id_magazine: <string>, /* identificativo periodico */
  name: <string> /* descrizione dell'opzione */
  price: <number>, /* prezzo dell'opzione se non inclusa in un'offerta
*/
}
```

API fatture

get_customer_invoices

Funzione che restituisce l'elenco delle fatture emesse per il cliente.

Endpoint: `http://base_url/get_customer_invoices`

Parametri POST:

access_key - chiave di riconoscimento

id_customer - identificativo del cliente

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  invoices: [
    {
      id_invoice: <string>, /* identificativo fattura */
      date: "yyyy-mm-dd", /* data fattura */
      total_amount: <string>, /* importo nel formato "#0.00" */
      id_document_type: <string>, /* FAT se fattura, NDC se nota di
credito/rimborso */
      is_file_available: <"true"|"false">, /* se la stampa PDF è
disponibile */
    },
    ...
  ]
}
```

API estrazione dati

find_subscriptions_by_action

Funzione che restituisce un elenco di istanze per cui, nell'intervallo di date fornito (estremi inclusi), deve essere effettuata l'azione specificata in **action**. L'intervallo non può superare un mese.

L'interrogazione è paginata, un risultato vuoto significa che si è raggiunta l'ultima pagina.

Endpoint: `http://base_url/find_subscriptions_by_action`

Parametri POST:

access_key - chiave di riconoscimento

id_magazine - identificativo del periodico

action - azione rispetto a cui estraggo la lista, valori possibili:

"CHARGE_WARNING" - istanze che, nell'intervallo di date, devono ricevere l'avviso di imminente addebito

"CHARGE" - istanze da saldare con addebito nell'intervallo di date

"RENEWAL_WARNING" - istanze che, nell'intervallo di date, devono ricevere l'avviso di imminente rinnovo

"RENEWAL" - istanze da rinnovare con addebito nell'intervallo di date

dt_begin - data iniziale dell'intervallo di ricerca

dt_end - data finale (inclusa) dell'intervallo di ricerca

page - pagina dei risultati (parte da 0)

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  subscriptions: [
    {
      cod_abbo: <string>, /* identificativo della sequenza di
abbonamenti negli anni */
      id_subscription: <string>, /* identificativo istanza abbonamento
*/
      id_magazine: <string>, /* identificativo periodico */
      id_offering: <string>, /* identificativo del listino(tipo
abbonamento) */
      id_renewal_offering: <string>, /* SE PRESENTE identificativo del
listino(tipo abbonamento) al rinnovo */
      id_customer_recipient: <string>, /* identificativo beneficiario */
      email_recipient: <string>, /* email del beneficiario */
      id_customer_payer: <string>, /* identificativo pagante SOLO SE
DIVERSO da beneficiario */
      email_payer: <string>, /* email del pagante SE PRESENTE */
      subscription_begin: "yyyy-mm-dd", /* data nominale inizio istanza
abbonamento */
      subscription_end: "yyyy-mm-dd" /* data nominale fine istanza
abbonamento */
    },
    ...
  ]
}
```

§ find_subscription_status_changes

Restituisce l'elenco di istanze che a partire dal **begin_timestamp** hanno subito cambiamenti in termini di pagamento, blocco o disdetta.

Lo **id_payment_type** restituito è popolato solo in caso di pagamento e solo nel caso in cui il pagamento non sia tramite fattura. Nel caso di pagamento con fattura differita **is_paid** è true e **id_payment_type** è vuoto.

L'interrogazione è paginata, un risultato vuoto significa che si è raggiunta l'ultima pagina.

Ciascun oggetto restituito riporta il **modified_timestamp**, cioè il timestamp di ultima modifica. Memorizzando il valore più grande dopo ciascuna interrogazione, si avrà a disposizione un **begin_timestamp** da usare come parametro di input per le interrogazioni successive.

Endpoint: http://base_url/find_subscription_status_changes

Parametri POST:

access_key - chiave di riconoscimento

begin_timestamp - timestamp iniziale, l'intervallo di ricerca sarà il tempo incluso dal timestamp fino all'istante attuale

page - pagina dei risultati (parte da 0)

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```

{
  subscriptions: [
    {
      cod_abbo: <string>, /* identificativo della sequenza di
abbonamenti negli anni */
      id_subscription: <string>, /* identificativo istanza abbonamento
*/
      id_magazine: <string>, /* identificativo periodico */
      id_customer_recipient: <string>, /* identificativo beneficiario */
      id_customer_payer: <string>, /* identificativo pagante SOLO SE
DIVERSO da beneficiario */
      modified_timestamp: <timestamp> /* timestamp dell'ultima modifica
di questo oggetto*/
      is_blocked: <"true"|"false">, /* se è bloccato/annullato */
      is_cancelled: <"true"|"false">, /* se è disdettato */
      is_paid: <"true"|"false">, /* se è pagato */
      id_payment_type: <id_payment_type>, /*tipo pagamento, SE PAGATO*/
    },
    ...
  ]
}

```

API per download file

get_invoice

Funzione per ottenere il file PDF relativo a una fattura (o nota di credito). Il gestionale conserva le fatture solo per un tempo limitato.

Endpoint: http://base_url/get_invoice

Parametri POST:

access_key - chiave di riconoscimento

id_invoice - identificativo dalla fattura: numero fattura

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```

{
  id_invoice: <string>, //identificativo della fattura
  mime_type: <string>, //mime-type del valore in 'content_base64'
  file_name: <string>, //nome del file fattura
  content_base64: <string> //il contenuto binario del file codificato
in base64
}

```

get_payment_form

Funzione per ottenere il file PDF del bollettino da pagare, nel caso in cui l'abbonamento non sia ancora stato saldato.

Endpoint: http://base_url/get_payment_form

Parametri POST:

access_key - chiave di riconoscimento

id_subscription - identificativo dell'istanza di abbonamento

La risposta sarà un oggetto JSON aderente allo standard descritto precedentemente con payload così definito:

```
{
  id_subscription: <string>, //identificativo dell'istanza di
abbonamento
  mime_type: <string>, //mime-type del valore in 'content_base64'
  file_name: <string>, //nome del file bollettino
  content_base64: <string> //il contenuto binario del file codificato
in base64
}
```

Lookup - Codifiche statiche

id_tipo_anagrafica

PVT	Privato
AGE	Agente
AZN	Azienda
LIB	Libreria
SCU	Scuola
BIB	Biblioteca

id_magazine

A	La Vita Scolastica
B	Psicologia Contemporanea
C	Età Evolutiva
D	Psicologia e Scuola
M	Scuola dell'infanzia
N	Nidi D'infanzia
Q	Art e Dossier
W	Archeologia Viva
H	io+

id_payment_type

BOL	Bollettino
CCR	Carta di credito
ASS	Assegno

BNF	Bonifico
CDO	Carta docente
PYP	Paypal

id_job

- 1 - Altro
- 2 - Architetto
- 3 - Artigiano
- 4 - Artista
- 5 - Avvocato
- 6 - Bibliotecario
- 7 - Casalinga
- 8 - Commercialista
- 9 - Commerciante
- 10 - Counselor
- 11 - Dirigente quadro funzionario
- 12 - Dirigente scolastico/preside
- 13 - Disoccupato/a
- 14 - Educatore
- 15 - Geometra
- 16 - Impiegato
- 17 - Imprenditore
- 18 - Ingegnere
- 19 - Insegnante di sostegno
- 20 - Insegnante scuola dell'infanzia
- 21 - Insegnante scuola primaria
- 22 - Insegnante scuola secondaria 1°grado
- 23 - Insegnante scuola secondaria 2°grado
- 24 - Docente universitario
- 25 - Libero professionista
- 26 - Logopedista
- 27 - Medico
- 29 - Neuropsichiatra
- 30 - Operaio
- 32 - Pensionato
- 33 - Psicologo
- 34 - Sacerdote/religiosi
- 35 - Studente scuola secondaria 2°grado
- 36 - Studente universitario psicologia
- 37 - Studente universitario scienze della formazione
- 38 - Studente universitario materie letterarie
- 39 - Studente universitario materie storiche archeologiche
- 40 - Studente universitario materie artistiche
- 41 - Studente universitario altre facoltà
- 42 - Insegnante (non specifica)
- 45 - Agente/Rappresentante
- 46 - Agronomo

47 - Archeologo
48 - Assistente Sociale
49 - Conservatore o Restauratore
50 - Giornalista
51 - Infermiere professionale
52 - Psicoterapeuta
53 - Ricercatore
54 - Studente (non specifica)
55 - Veterinario
56 - Formatore

id_qualification

1	Licenza elementare
2	Licenza media
3	Diploma
4	Laurea
5	Laurea con master
6	Dottorato/ricercatore

id_document_type

FAT	Fattura
NDC	Nota di credito (rimborso)

feature_tags

Elenco dei valori attualmente validi per l'array "feature_tags". In vecchie istanze possono comparire altri valori non più in uso e non inclusi in questo elenco.

AREAEXTRA	Ha accesso all'area extra / contenuti su sito
WEBTV	Ha accesso alla Web TV
TANDEM1A	Istanza primaria acquistata in abbinamento con una istanza di [A] La vita scolastica
TANDEM1D	Istanza primaria acquistata in abbinamento con una istanza di [D] Psicologia e scuola
TANDEM1M	Istanza primaria acquistata in abbinamento con una istanza di [M] Scuola dell'infanzia
TANDEM1N	Istanza primaria acquistata in abbinamento con una istanza di [N] Nidi d'infanzia
TANDEM2	Istanza generata dall'acquisto di una istanza TANDEM1*
MEMBERSHIP	Include l'accesso ai servizi io+