

ItemOverview

Questa API consente di verificare rapidamente la presenza di fatture ricevute, inviate e scartate per una o più aziende a partire da un dato timestamp.

Il comportamento è identico alla vecchia `companyOverview`, è stata aggiornata la logica relativa alla paginazione (adeguandola al `continuationToken`) e la nomenclatura dei campi per renderla coerente con `itemId`. In query string è possibile indicare la dimensione della pagina attraverso il parametro `size` (valore massimo ammesso 1000) è consigliato usare valori più bassi come 100 e sfruttare il meccanismo del `continuation token` per recuperare le pagine successive. questo permette migliori performance e riduce il rischio di rallentare il db.

I valori di *`lastTimestampActive`* e *`lastTimestampPassive`* nel payload di richiesta devono essere espressi in millisecondi e non possono essere antecedenti al 01/01/2019 nè successive al timestamp corrente.

Funzionamento

Aggiornamento fatture passive

Quando TSDigital assegna una fattura ricevuta dal Sistema di Interscambio viene prodotto un evento che, **quando verrà processato**, aggiornerà le informazioni del campo *`lastTimestampPassive`* con il timestamp di assegnazione della fattura.

Esempio:

- TSDigital assegna la fattura all'azienda con codice fiscale ABCDEF e `itemId` `adcf9209-fe9b-4b37-9b68-8238da8fd6bb` alle 15:30 del 15/04/2020 (UTC), corrispondente al timestamp `1586964600000`
- Viene prodotto l'evento
- TSDigital processa l'evento precedente alle 17:00 del 15/04/2020 (UTC), corrispondente al timestamp `1587081600000`
- Il valore di *`lastTimestampPassive`* per l'`itemId` `adcf9209-fe9b-4b37-9b68-8238da8fd6bb` sarà dunque `1586964600000` (orario di assegnazione della fattura) e non `1587081600000` (orario in cui è stato processato l'evento)

Aggiornamento fatture attive/scartate

Vale quanto detto in precedenza, con l'unica differenza che l'evento di aggiornamento dell'`ItemOverview` viene prodotto a partire da un cambio stato della fattura e produrrà un cambiamento per i valori di *`lastTimestampActive`* e *`lastTimestampRejected`*.

Funzionamento delle API

Per semplicità prendiamo in esame un payload di richiesta relativo ad un solo item così definito:

```
{
  "items": [
    {
      "itemId": "ABCDEF",
      "lastTimestampActive": 1583143700735, // 2 Marzo 2020, 10:08:20 UTC
      "lastTimestampPassive": 1583244701735 // 3 Marzo 2020, 14:11:51 UTC
    }
  ]
}
```

L'itemId *ABCDEF* su database contiene queste informazioni:

```
{
  "itemId": "ABCDEF",
  "taxId": "XYZ",
  "lastTimestampActive": 1586159941109, // 6 Aprile 2020, 07:59:01 UTC
  "lastTimestampPassive": 1583251200000, // 3 Marzo 2020, 16:00:00 UTC
  "lastTimestampRejected": 1584378112720, // 16 Marzo 2020, 17:01:52 UTC
  [...]
}
```

I flag saranno calcolati così:

```
boolean active = lastTimestampActive > lastTimestampActiveFromRequest;
boolean rejected = lastTimestampRejected > lastTimestampActiveFromRequest;
boolean passive = lastTimestampPassive > lastTimestampPassiveFromRequest;

// Utilizzando i dati di esempio:

// 6 Aprile 2020, 07:59:01 UTC > 2 Marzo 2020, 10:08:20 UTC => true
boolean active = 1586159941109 > 1583143700735

// 16 Marzo 2020, 17:01:52 UTC > 2 Marzo 2020, 10:08:20 UTC => true
boolean rejected = 1584378112720 > 1583143700735

// 3 Marzo 2020, 16:00:00 UTC > 3 Marzo 2020, 14:11:51 UTC => true
boolean passive = 1583251200000 > 1583244701735
```

e produrranno quindi il seguente payload di risposta:

```
[...]
{
  "active": true,
  "passive": true,
  "rejected": true,
  "lastTimestampActive": 1586159941109, // 6 Aprile 2020, 07:59:01 UTC
  "lastTimestampPassive": 1583251200000, // 3 Marzo 2020, 16:00:00 UTC
  "lastTimestampRejected": 1584378112720, // 16 Marzo 2020, 17:01:52 UTC
  "itemId": "ABCDEF",
  "taxId": "XYZ"
}
[...]
```

Utilizzo

Ad esclusione della prima volta in cui si utilizza l'API per garantire il corretto comportamento è **fondamentale** utilizzare come valori di *lastTimestampActive* e *lastTimestampPassive* in request quelli ottenuti dalla response precedente di company overview.

Effettuare una prima chiamata all'API utilizzando un valore valido a piacere per i campi *lastTimestampActive* e *lastTimestampPassive*, ad esempio 1577836800000 (1 Gennaio 2020, 00:00:00 UTC).

Ottenuta la prima response è opportuno memorizzare i valori di *lastTimestampActive* e *lastTimestampPassive* che verranno utilizzati per la successiva chiamata di company Overview.

Riprendendo l'esempio di prima ed ipotizzando che la situazione delle fatture sia rimasta invariata per l'itemId ABCDEF la nuova chiamata avrà come payload di request:

```
{
  "items": [
    {
      "itemId": "ABCDEF",
      "lastTimestampActive": 1586159941109, // 6 Aprile 2020, 07:59:01 UTC ottenuto dalla
      response precedente
      "lastTimestampPassive": 1583251200000, // 3 Marzo 2020, 16:00:00 UTC ottenuto dalla
      response precedente
    }
  ]
}
```

```
}
```

e come response

```
[...]
{
  "active": false,
  "passive": false,
  "rejected": false,
  "lastTimestampActive": 1586159941109, // 6 Aprile 2020, 07:59:01 UTC
  "lastTimestampPassive": 1583251200000, // 3 Marzo 2020, 16:00:00 UTC
  "lastTimestampRejected": 1584378112720, // 16 Marzo 2020, 17:01:52 UTC
  "itemId": "ABCDEF",
  "taxId": "XYZ"
}
[...]
```

Request/Response

Esempio di payload di richiesta con una post a <https://b2bread-api-test.agyo.io/api/v2/items/overview?size=1>

```
{
  "items": [
    {
      "itemId": "ABCDEF00G00H000L",
      "lastTimestampActive": 1583143700735,
      "lastTimestampPassive": 1583143700735
    },
    {
      "itemId": "6f51950a-c982-40b7-99a3-405dd1328684",
      "lastTimestampActive": 1583143700735,
      "lastTimestampPassive": 1583143700735
    }
  ]
}
```

Esempio di payload di risposta:

```
{
  "_embedded": {
```

```

    "itemOverviewList": [
      {
        "active": true,
        "passive": true,
        "rejected": true,
        "lastTimestampActive": 1586159941109,
        "lastTimestampPassive": 1583243470858,
        "lastTimestampRejected": 1584378112720,
        "itemId": "ABCDEF00G00H000L",
        "taxId": "ABCDEF00G00H000L"
      }
    ]
  },
  "_links": {
    "first": {
      "href": "https://b2bread-api-
test.agyo.io/api/v2/items/overview?size=1&sort=agyoCompanyId,desc"
    },
    "self": {
      "href": "https://b2bread-api-
test.agyo.io/api/v2/items/overview?size=1&sort=agyoCompanyId,desc"
    },
    "next": {
      "href": "https://b2bread-api-
test.agyo.io/api/v2/items/overview?continuationToken=33dbhiloNf1U8m9EPLLkKceN2NR0iLWgzLNh6mnbFLD5MDEXZ9jWsKRnn5622zJqRbP44Twq4oh_XrY02zsulG3tD05ipcarnPv6gjQk6xqnd0x_BShZwfqTmItxsAlz&size=1&sort=agyoCompanyId,desc"
    }
  },
  "page": {
    "size": 1,
    "hasNext": true,
    "continuationToken":
"33dbhiloNf1U8m9EPLLkKceN2NR0iLWgzLNh6mnbFLD5MDEXZ9jWsKRnn5622zJqRbP44Twq4oh_XrY02zsulG3tD05ipcarnPv6gjQk6xqnd0x_BShZwfqTmItxsAlz"
  }
}

```

Nel payload di richiesta sono presenti due item ed in queryString è stata richiesta la pagina di un elemento ci aspettiamo dunque la presenza di ulteriori pagine, come confermato dal payload di

risposta `(page.hasNext=true)`. Possiamo accedere alla pagina successiva utilizzando il link presente in `_links.next` (non costruire il link a mano ma utilizzare quello ottenuto in risposta). In questo caso effettueremo una post a `https://b2bread-api-test.agyo.io/api/v2/items/overview?continuationToken=33dbhiloNf1U8m9EPLLkKceN2NROiLWgzLNh6mnbFLD5MDEXZ9jWsKRnn5622zJqRbP44Twq4oh_XrYO2zsulG3tDO5ipcarnPv6gjQk6xqnd0x_BShZwfqTmltxsAlz&size=1&sort=agyoCompanyId,desc`

Payload di risposta:

```
{
  "_embedded": {
    "itemOverviewList": [
      {
        "active": true,
        "passive": false,
        "rejected": false,
        "lastTimestampActive": 1583243700735,
        "lastTimestampPassive": 1563982802310,
        "itemId": "6f51950a-c982-40b7-99a3-405dd1328684",
        "taxId": "03200510166"
      }
    ]
  },
  "_links": {
    "self": {
      "href": "https://b2bread-api-test.agyo.io/api/v2/items/overview?continuationToken=33dbhiloNf1U8m9EPLLkKceN2NROiLWgzLNh6mnbFLD5MDEXZ9jWsKRnn5622zJqRbP44Twq4oh_XrYO2zsulG3tDO5ipcarnPv6gjQk6xqnd0x_BShZwfqTmltxsAlz&size=1&sort=agyoCompanyId,desc"
    }
  },
  "page": {
    "size": 1,
    "hasNext": false,
    "continuationToken": null
  }
}
```

Dal payload di risposta notiamo che non ci sono ulteriori pagine `(page.hasNext=false)`.

Revision #9

Created 2020-03-27 11:16:51 UTC by Zittino

Updated 2022-09-29 15:44:44 UTC by Luca Zilembo