



**EBook Gratuito**

# APPENDIMENTO

## google-analytics-api

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#google-  
analytics-  
api

# Sommario

Di.....	1
<b>Capitolo 1: Iniziare con google-analytics-api.....</b>	<b>2</b>
Osservazioni.....	2
Versioni.....	4
Examples.....	4
Accesso alle API di Google Analytics.....	4
introduzione.....	5
Hello World Reporting API - Rest.....	7
<b>Capitolo 2: API di reporting (Analytics v4).....</b>	<b>8</b>
Osservazioni.....	8
Examples.....	8
Esempio di report singolo con OAuth2 C #.....	8
Esempio di rapporto singolo Rest.....	9
<b>Capitolo 3: API in tempo reale.....</b>	<b>11</b>
Sintassi.....	11
Parametri.....	11
Osservazioni.....	12
Examples.....	13
Esempio PHP.....	13
<b>Capitolo 4: Autenticazione.....</b>	<b>16</b>
Sintassi.....	16
Parametri.....	16
Osservazioni.....	16
Examples.....	18
OAuth2 C #.....	18
Servizio Account Authentcation Vb.net.....	19
<b>Capitolo 5: Metadata api.....</b>	<b>21</b>
Sintassi.....	21
Parametri.....	21
Osservazioni.....	21

Examples.....	21
Esempio di riposo.....	21
Esempio di Java.....	22
Esempio PHP.....	23
Esempio di Python.....	24
C # esempio.....	25
<b>Capitolo 6: Risposte di errore.....</b>	<b>27</b>
Sintassi.....	27
Parametri.....	27
Osservazioni.....	28
Examples.....	28
400 invalidParameter.....	28
<b>Titoli di coda.....</b>	<b>30</b>

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [google-analytics-api](#)

It is an unofficial and free google-analytics-api ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official google-analytics-api.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Capitolo 1: Iniziare con google-analytics-api

## Osservazioni

Le [API di Google Analytics](#) ti consentono di accedere ai dati all'interno di Google Analytics. Non deve essere confuso con il [protocollo di misurazione](#) utilizzato per l'inserimento di dati in [Google Analytics](#).

L'API di Google Analytics è suddivisa in parti serval.

### API di reporting di Google Analytics

1. L' [API di reporting principale di Google Analytics v3](#) consente di accedere alla maggior parte dei dati del report in Google Analytics. Con l'API Core Reporting puoi:
  - Crea dashboard personalizzati per visualizzare i dati di Google Analytics.
  - Risparmia tempo automatizzando complesse attività di reporting.
  - Integra i tuoi dati di Google Analytics con altre applicazioni aziendali.
2. L' [API di reporting di Google Analytics V4](#) è il metodo programmatico più avanzato per accedere ai dati dei rapporti in Google Analytics. Con l'API di reporting di Google Analytics, puoi:
  - Crea dashboard personalizzati per visualizzare i dati di Google Analytics.
  - Automatizza attività di reporting complesse per risparmiare tempo.
  - Integra i tuoi dati di Google Analytics con altre applicazioni aziendali.
2. L' [API Real Time Reporting](#) consente di richiedere dati in tempo reale, ad esempio attività in tempo reale su una vista, per un utente autenticato. Puoi utilizzare l'API di Real Time Reporting per:
  - Mostra i visualizzatori attivi di una pagina e crea un senso di urgenza per gli utenti che guardano un articolo con inventario finito.
  - Visualizza i contenuti più popolari come le prime 10 pagine attive.
  - Crea e visualizza una dashboard in tempo reale.

L'API Real Time Reporting, in versione beta limitata, è disponibile solo per l'anteprima dello sviluppatore. [Registrati](#) per accedere all'API.

3. L' [API di reporting canalizzazioni multicanale](#) consente di richiedere dati di canalizzazioni multicanale per un utente autenticato. I dati derivano dai dati del percorso di conversione, che mostra le interazioni dell'utente con varie sorgenti di traffico su più sessioni prima della conversione. Ciò ti consente di analizzare in che modo più canali di marketing influenzano le conversioni nel tempo. Per ulteriori dettagli su quali dati sono disponibili, leggi [Informazioni sulle canalizzazioni multicanale](#) e [Informazioni sui dati di canalizzazione multicanale](#). Con l'API Reporting canalizzazioni multicanale puoi:

- Crea rapporti personalizzati utilizzando i dati delle canalizzazioni multicanale. Ad esempio, è possibile utilizzare i dati dei percorsi di conversione principali per generare report su attributi come la posizione relativa delle interazioni in un percorso di conversione.
- Integra i dati delle canalizzazioni multicanale con i dati aziendali. Ad esempio, è possibile correlare i dati di conversione online con dati di vendita offline e dati sui costi dei media per ottenere un'immagine più completa del ROI di marketing.
- Mostra canalizzazioni multicanale in nuovi ambienti. Ad esempio, puoi creare visualizzazioni e altre presentazioni dei dati che comunicano il valore dei diversi canali di marketing nel generare conversioni.

## Helpers

1. L' [API dei metadati](#) restituisce l'elenco e gli attributi delle colonne (ossia dimensioni e metriche) esposti nelle API di reporting di Google Analytics. Gli attributi restituiti includono nome dell'interfaccia utente, descrizione, supporto per segmenti e altro ancora. Puoi utilizzare l'API dei metadati per:

- Scopri automaticamente nuove colonne.
- Accedi a tutti gli attributi di dimensioni e metriche per le API di reporting di Google Analytics.

Nota: restituisce solo i metadati per l'API di reporting principale e l'API di reporting. Metadati non in tempo reale.

2. L' [API di incorporamento di Google Analytics](#) è una libreria JavaScript che consente di creare e incorporare facilmente un dashboard su un sito Web di terzi in pochi minuti. Ti offre una serie di componenti collegabili che possono lavorare insieme per costruire strumenti complessi, rendendoli allo stesso tempo semplici e potenti.

## API di configurazione

1. L' [API di gestione di Analytics](#) consente l'accesso programmatico ai dati di configurazione di Google Analytics. Puoi creare applicazioni per gestire in modo più efficiente account Analytics grandi o complessi. Le grandi aziende con molte proprietà possono automatizzare l'impostazione dell'account. Anche se stai creando un'applicazione di reporting, l'API di gestione fornisce gli strumenti per navigare nel tuo account. Puoi utilizzare l'API di gestione di Google Analytics per:

- Elenca tutte le informazioni su Account, Proprietà e Visualizza per un utente.
- Gestisci proprietà, viste e obiettivi.
- Gestisci le autorizzazioni degli utenti per una gerarchia di account.
- Recupera un ID vista da utilizzare con l'API di reporting principale.
- Determina quali obiettivi sono attivi e accedi ai loro nomi configurati.
- Gestisci i collegamenti tra le proprietà di Analytics e gli account AdWords.
- Gestisci segmenti di pubblico di remarketing.

Le operazioni di scrittura nell'API di gestione (ad esempio creazione, aggiornamento,

eliminazione, patch) per le proprietà Web, View (profilo) e Goal sono attualmente disponibili come anteprima per sviluppatori in **versione beta limitata** . Se sei interessato a utilizzare queste funzionalità, [richiedi l'accesso alla versione beta](#) .

2. L' [API di provisioning](#) può essere utilizzata per creare nuovi account Google Analytics e abilitare Google Analytics per i tuoi clienti su larga scala. È destinato a fornitori di servizi qualificati e partner di grandi dimensioni. Ad esempio, è possibile utilizzare l'API di provisioning come parte di un nuovo processo di onboarding degli utenti per creare un nuovo account Google Analytics per un client e quindi utilizzare ulteriori risorse API di gestione per configurare a livello di programmazione l'account e collegarlo ad AdWords. Tutto questo può essere automatizzato e avviato dal tuo amministratore o dall'interfaccia di reporting.

**L'API di provisioning è disponibile solo su invito. Al momento non accettiamo nuovi progetti.**

## Versioni

Al momento sono disponibili tre versioni dell'API di Google Analytics.

[Google Analytics V2](#) Versione 2.4 dell'API di reporting principale è un'API di solo tipo XML compatibile principalmente con la versione 2.3. **(Legacy)**

[Google Analytics V3](#) che include l'API Core Reporting, l'API di gestione, l'API MetaData e un numero di altre API. Queste API restituiscono Json

[Google Analytics V4](#) che attualmente include solo l'API di reporting ed è considerato il modo più avanzato per recuperare i dati dei rapporti da Google Analytics.

## Examples

### Accesso alle API di Google Analytics

Puoi accedere tecnicamente alle API di Google Analytics utilizzando qualsiasi linguaggio di programmazione in grado di gestire una richiesta [HTTP Post](#) o [HTTP Get](#) .

Detto questo, Google ha anche creato una serie di librerie client standard ufficiali per aiutarti in questo. L'utilizzo di una libreria client standard per il linguaggio di programmazione scelto può essere molto più semplice rispetto alla codifica da zero.

### Librerie client OFFICAL con supporto dell'API di Google Analytics:

1. Libreria client PHP API di Google - [GitHub](#)
2. API di Google. Libreria client Net - [GitHub NuGet](#)
3. Libreria client Python per le API di Google - [GitHub](#)
4. Libreria client Java API di Google - [collegamento](#)
5. Libreria Objective-C API di Google - [GitHub](#)

Ci sono più librerie [qui](#) .

## introduzione

Le [API di Google Analytics](#) ti consentono di accedere ai dati all'interno di Google Analytics. Non deve essere confuso con il [protocollo di misurazione](#) utilizzato per l'inserimento di dati in [Google Analytics](#).

L'API di Google Analytics è suddivisa in parti serval.

### API di reporting di Google Analytics

1. L' [API di reporting principale di Google Analytics v3](#) consente di accedere alla maggior parte dei dati del report in Google Analytics. Con l'API Core Reporting puoi:

- Crea dashboard personalizzati per visualizzare i dati di Google Analytics.
- Risparmia tempo automatizzando complesse attività di reporting.
- Integra i tuoi dati di Google Analytics con altre applicazioni aziendali.

2. L' [API di reporting di Google Analytics V4](#) è il metodo programmatico più avanzato per accedere ai dati dei rapporti in Google Analytics. Con l'API di reporting di Google Analytics, puoi:

- Crea dashboard personalizzati per visualizzare i dati di Google Analytics.
- Automatizza attività di reporting complesse per risparmiare tempo.
- Integra i tuoi dati di Google Analytics con altre applicazioni aziendali.

2. L' [API Real Time Reporting](#) consente di richiedere dati in tempo reale, ad esempio attività in tempo reale su una vista, per un utente autenticato. Puoi utilizzare l'API di Real Time Reporting per:

- Mostra i visualizzatori attivi di una pagina e crea un senso di urgenza per gli utenti che guardano un articolo con inventario finito.
- Visualizza i contenuti più popolari come le prime 10 pagine attive.
- Crea e visualizza una dashboard in tempo reale.

L'API Real Time Reporting, in versione beta limitata, è disponibile solo per l'anteprima dello sviluppatore. [Registrati](#) per accedere all'API.

3. L' [API di reporting canalizzazioni multicanale](#) consente di richiedere dati di canalizzazioni multicanale per un utente autenticato. I dati derivano dai dati del percorso di conversione, che mostra le interazioni dell'utente con varie sorgenti di traffico su più sessioni prima della conversione. Ciò ti consente di analizzare in che modo più canali di marketing influenzano le conversioni nel tempo. Per ulteriori dettagli su quali dati sono disponibili, leggi [Informazioni sulle canalizzazioni multicanale](#) e [Informazioni sui dati di canalizzazione multicanale](#). Con l'API Reporting canalizzazioni multicanale puoi:

- Crea rapporti personalizzati utilizzando i dati delle canalizzazioni multicanale. Ad esempio, è possibile utilizzare i dati dei percorsi di conversione principali per generare report su attributi come la posizione relativa delle interazioni in un percorso di conversione.

- Integra i dati delle canalizzazioni multicanale con i dati aziendali. Ad esempio, è possibile correlare i dati di conversione online con dati di vendita offline e dati sui costi dei media per ottenere un'immagine più completa del ROI di marketing.
- Mostra canalizzazioni multicanale in nuovi ambienti. Ad esempio, puoi creare visualizzazioni e altre presentazioni dei dati che comunicano il valore dei diversi canali di marketing nel generare conversioni.

## Helpers

1. L' [API dei metadati](#) restituisce l'elenco e gli attributi delle colonne (ossia dimensioni e metriche) esposti nelle API di reporting di Google Analytics. Gli attributi restituiti includono nome dell'interfaccia utente, descrizione, supporto per segmenti e altro ancora. Puoi utilizzare l'API dei metadati per:

- Scopri automaticamente nuove colonne.
- Accedi a tutti gli attributi di dimensioni e metriche per le API di reporting di Google Analytics.

Nota: restituisce solo i metadati per l'API di reporting principale e l'API di reporting. Metadati non in tempo reale.

2. L' [API di incorporamento di Google Analytics](#) è una libreria JavaScript che consente di creare e incorporare facilmente un dashboard su un sito Web di terzi in pochi minuti. Ti offre una serie di componenti collegabili che possono lavorare insieme per costruire strumenti complessi, rendendoli allo stesso tempo semplici e potenti.

## API di configurazione

1. L' [API di gestione di Analytics](#) consente l'accesso programmatico ai dati di configurazione di Google Analytics. Puoi creare applicazioni per gestire in modo più efficiente account Analytics grandi o complessi. Le grandi aziende con molte proprietà possono automatizzare l'impostazione dell'account. Anche se stai creando un'applicazione di reporting, l'API di gestione fornisce gli strumenti per navigare nel tuo account. Puoi utilizzare l'API di gestione di Google Analytics per:

- Elenca tutte le informazioni su Account, Proprietà e Visualizza per un utente.
- Gestisci proprietà, viste e obiettivi.
- Gestisci le autorizzazioni degli utenti per una gerarchia di account.
- Recupera un ID vista da utilizzare con l'API di reporting principale.
- Determina quali obiettivi sono attivi e accedi ai loro nomi configurati.
- Gestisci i collegamenti tra le proprietà di Analytics e gli account AdWords.
- Gestisci segmenti di pubblico di remarketing.

Le operazioni di scrittura nell'API di gestione (ad esempio creazione, aggiornamento, eliminazione, patch) per le proprietà Web, View (profilo) e Goal sono attualmente disponibili come anteprima per sviluppatori in **versione beta limitata** . Se sei interessato a utilizzare queste funzionalità, [richiedi l'accesso alla versione beta](#) .

2. L' [API di provisioning](#) può essere utilizzata per creare nuovi account Google Analytics e

abilitare Google Analytics per i tuoi clienti su larga scala. È destinato a fornitori di servizi qualificati e partner di grandi dimensioni. Ad esempio, è possibile utilizzare l'API di provisioning come parte di un nuovo processo di onboarding degli utenti per creare un nuovo account Google Analytics per un client e quindi utilizzare ulteriori risorse API di gestione per configurare a livello di programmazione l'account e collegarlo ad AdWords. Tutto questo può essere automatizzato e avviato dal tuo amministratore o dall'interfaccia di reporting.

**L'API di provisioning è disponibile solo su invito. Al momento non accettiamo nuovi progetti.**

## Hello World Reporting API - Rest

```
POST https://analyticsreporting.googleapis.com/v4/reports:batchGet?access_token={Access token from auth request}
{
  "reportRequests": [
    {
      "viewId": "XXXX",
      "dateRanges": [
        {
          "startDate": "2015-06-15",
          "endDate": "2015-06-30"
        }
      ],
      "metrics": [
        {
          "expression": "ga:sessions"
        }
      ],
      "dimensions": [
        {
          "name": "ga:browser"
        }
      ]
    }
  ]
}
```

Leggi Iniziare con google-analytics-api online: <https://riptutorial.com/it/google-analytics-api/topic/7253/iniziare-con-google-analytics-api>

---

# Capitolo 2: API di reporting (Analytics v4)

## Osservazioni

L' [API di reporting di Google Analytics V4](#) è il metodo programmatico più avanzato per accedere ai dati dei rapporti in [Google Analytics](#) . Con l'API di reporting di Google Analytics, puoi:

- Crea dashboard personalizzati per visualizzare i dati di [Google Analytics](#) .
- Automatizza attività di reporting complesse per risparmiare tempo.
- Integra i tuoi dati di [Google Analytics](#) con altre applicazioni aziendali.

## Caratteristiche

[Google Analytics](#) è basato su una potente infrastruttura di reporting dei dati. L'API di reporting di Google Analytics V4 ti consente di accedere alla potenza della piattaforma di [Google Analytics](#) . L'API fornisce queste funzionalità chiave:

- Espressioni metriche: l'API consente di richiedere non solo metriche incorporate ma anche combinazioni di metriche espresse in operazioni matematiche. Ad esempio, è possibile utilizzare l'espressione `ga: goal1completions / ga: sessioni` per richiedere i completamenti obiettivo per numero di sessioni.
- Più intervalli di date: l'API ti consente in un'unica richiesta di ottenere dati in due intervalli di date.
- Coorti e valore a vita: l'API ha un ricco vocabolario per richiedere rapporti sui valori Cohort e Lifetime.
- Più segmenti: l'API ti consente di ottenere più segmenti in una singola richiesta.

## Examples

### Esempio di report singolo con OAuth2 C #

Questo esempio utilizza la libreria ufficiale di Google .net Client.

```
PM> Install-Package Google.Apis.AnalyticsReporting.v4
```

**Autorizzazione** Richiede uno dei seguenti ambiti OAuth:

- <https://www.googleapis.com/auth/analytics.readonly>
- <https://www.googleapis.com/auth/analytics>

### OAuth2

```
// These are the scopes of permissions you need.  
string[] scopes = new string[] { AnalyticsReportingService.Scope.AnalyticsReadonly }; //  
View your Google Analytics Data  
  
UserCredential credential;
```

```

using (var stream = new FileStream(clientSecretJson, FileMode.Open, FileAccess.Read))
{
    string credPath =
System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
    credPath = Path.Combine(credPath, ".credentials/",
System.Reflection.Assembly.GetExecutingAssembly().GetName().Name);

    // Requesting Authentication or loading previously stored authentication for userName
    credential =
GoogleWebAuthorizationBroker.AuthorizeAsync(GoogleClientSecrets.Load(stream).Secrets,
        scopes,
        userName,
        CancellationToken.None,
        new FileDataStore(credPath, true)).Result;
}

// Create Reporting API service.
var service = new AnalyticsReportingService(new BaseClientService.Initializer()
{
    HttpClientInitializer = credential,
    ApplicationName = string.Format("{0} Authentication",
System.Reflection.Assembly.GetExecutingAssembly().GetName().Name),
});

```

## Richiesta di segnalazione

```

// Create the DateRange object.
DateRange June2015 = new DateRange() { StartDate = "2015-01-01", EndDate = "2015-06-30" };
DateRange June2016 = new DateRange() { StartDate = "2016-01-01", EndDate = "2016-06-30" };
List<DateRange> dateRanges = new List<DateRange>() { June2016, June2015 };

// Create the ReportRequest object.
// This should have a large number of rows
ReportRequest reportRequest = new ReportRequest
{
    ViewId = ConfigurationManager.AppSettings["GoogleAnalyticsViewId"],
    DateRanges = dateRanges,
    Dimensions = new List<Dimension>() { new Dimension() { Name = "ga:date" }, new Dimension()
{ Name = "ga:usertype" } },
    Metrics = new List<Metric>() { new Metric() { Expression= "ga:users" }, new Metric() {
Expression = "ga:sessions" } },
    PageSize = 1000,
};

List<ReportRequest> requests = new List<ReportRequest>();
requests.Add(reportRequest);

var getReport = new GetReportsRequest() { ReportRequests = requests };
var response = service.Reports.BatchGet(getReport).Execute();

```

## Esempio di rapporto singolo Rest

Le richieste API sono POST HTTP con il token di accesso collegato alla fine dell'estremità dell'API.

Autorizzazione Richiede uno dei seguenti ambiti OAuth:

- <https://www.googleapis.com/auth/analytics.readonly>
- <https://www.googleapis.com/auth/analytics>

**Nota quando** `ContentType = "application/Json"`; **i dati usa** `ContentType = "application/Json"`;

```
https://analyticsreporting.googleapis.com/v4/reports:batchGet?Access_token={from auth}
{
  "reportRequests": [
    {
      "viewId": "XXXX",
      "dateRanges": [
        {
          "startDate": "2015-06-15",
          "endDate": "2015-06-30"
        }
      ],
      "metrics": [
        {
          "expression": "ga:sessions"
        }
      ],
      "dimensions": [
        {
          "name": "ga:browser"
        }
      ]
    }
  ]
}
```

Leggi API di reporting (Analytics v4) online: <https://riptutorial.com/it/google-analytics-api/topic/7265/api-di-reporting--analytics-v4->

# Capitolo 3: API in tempo reale

## Sintassi

- OTTIENI [https://www.googleapis.com/analytics/v3/data/realtime?ids=\[Indagine vista analitica Id\]](https://www.googleapis.com/analytics/v3/data/realtime?ids=[Indagine vista analitica Id]) e metriche = [Metriche in tempo reale] e access\_token = [Access\_token\_from\_Authentication]
- OTTIENI [https://www.googleapis.com/analytics/v3/data/realtime?ids=\[Indagine vista analitica Id\]](https://www.googleapis.com/analytics/v3/data/realtime?ids=[Indagine vista analitica Id]) e metriche = [Metriche in tempo reale] e dimensioni = [Metriche in tempo reale] e access\_token = [Access\_token\_from\_Authentication]

## Parametri

<b>(Richiesto)</b> Nome del parametro	Descrizione
ids	ID tabella univoco per il recupero dei dati di Analytics. L'ID della tabella è del formato ga: XXXX, dove XXXX è l'ID della vista Analytics (profilo).
metrica	Un elenco separato da virgole delle metriche di Analytics. Ad esempio, 'rt: activeUsers'. È necessario specificare almeno una metrica.
<b>(Opzionale)</b> Nome del parametro	Descrizione
dimensioni	Un elenco separato da virgole di dimensioni in tempo reale. Ad esempio, 'rt: medium, rt: city'.
filtri	Un elenco separato da virgole di filtri dimensione o metrica da applicare ai dati in tempo reale.
max-risultati	Il numero massimo di voci da includere in questo feed.
ordinare	Un elenco separato da virgole di dimensioni o metriche che determinano l'ordinamento per i dati in tempo reale.
<b>(Standard)</b> Nome del parametro	Descrizione
richiama	Nome della funzione di callback JavaScript che gestisce la risposta. Utilizzato nelle richieste JSON-P JavaScript.
prettyPrint	Restituisce la risposta in un formato leggibile dall'uomo se è vero. Valore

(Richiesto) Nome del parametro	Descrizione
	predefinito: true. Quando ciò è falso, può ridurre la dimensione del payload della risposta, il che potrebbe portare a prestazioni migliori in alcuni ambienti.
quotaUser	Consente di imporre quote per utente da un'applicazione lato server anche nei casi in cui l'indirizzo IP dell'utente è sconosciuto. Ciò può verificarsi, ad esempio, con applicazioni che eseguono lavori cron su App Engine per conto di un utente. È possibile scegliere qualsiasi stringa arbitraria che identifica univocamente un utente, ma è limitata a 40 caratteri. Consente di ignorare l'utente se entrambi vengono forniti. Ulteriori informazioni sulla limitazione dell'utilizzo.
userip	Consente di imporre quote per utente quando si chiama l'API da un'applicazione lato server. Ulteriori informazioni sulla limitazione dell'utilizzo.

## Osservazioni

**L'API Real Time Reporting, in versione beta limitata, è disponibile solo per l'anteprima dello sviluppatore. [Registrati](#) per accedere all'API.**

L' [API Real Time Reporting](#) consente di richiedere dati in tempo reale, ad esempio attività in tempo reale sulla proprietà, per un utente autenticato.

Puoi utilizzare l'API di Real Time Reporting per:

- Mostra i visualizzatori attivi di una pagina e crea un senso di urgenza per gli utenti che guardano un articolo con inventario finito.
- Visualizza i contenuti più popolari come le prime 10 pagine attive.
- Crea e visualizza una dashboard in tempo reale.

## Autorizzazione

Le chiamate all'API in tempo reale di Google Analytics richiedono l'autorizzazione con almeno uno dei seguenti ambiti (maggiori informazioni su autenticazione e autorizzazione).

Scopo	accesso garantito
<a href="https://www.googleapis.com/auth/analytics">https://www.googleapis.com/auth/analytics</a>	Accesso completo agli account Google Analytics fino al livello dell'accesso degli utenti autenticati.
<a href="https://www.googleapis.com/auth/analytics.readonly">https://www.googleapis.com/auth/analytics.readonly</a>	Accesso in sola lettura agli account Google Analytics degli utenti

Scopo	accesso garantito
	autenticati

## Examples

### Esempio PHP

Utilizza la [libreria client PHP](#)

```
/**
 * 1.Create and Execute a Real Time Report
 * An application can request real-time data by calling the get method on the Analytics
 service object.
 * The method requires an ids parameter which specifies from which view (profile) to retrieve
 data.
 * For example, the following code requests real-time data for view (profile) ID 56789.
 */
$optParams = array(
    'dimensions' => 'rt:medium');

try {
    $results = $analytics->data_realtime->get(
        'ga:56789',
        'rt:activeUsers',
        $optParams);
    // Success.
} catch (apiServiceException $e) {
    // Handle API service exceptions.
    $error = $e->getMessage();
}

/**
 * 2. Print out the Real-Time Data
 * The components of the report can be printed out as follows:
 */

function printRealtimeReport($results) {
    printReportInfo($results);
    printQueryInfo($results);
    printProfileInfo($results);
    printColumnHeaders($results);
    printDataTable($results);
    printTotalsForAllResults($results);
}

function printDataTable(&$results) {
    if (count($results->getRows()) > 0) {
        $table .= '<table>';

        // Print headers.
        $table .= '<tr>';

        foreach ($results->getColumnHeaders() as $header) {
            $table .= '<th>' . $header->name . '</th>';
        }
        $table .= '</tr>';
    }
}
```

```

// Print table rows.
foreach ($results->getRows() as $row) {
    $stable .= '<tr>';
    foreach ($row as $cell) {
        $stable .= '<td>'
            . htmlspecialchars($cell, ENT_NOQUOTES)
            . '</td>';
    }
    $stable .= '</tr>';
}
$stable .= '</table>';

} else {
    $stable .= '<p>No Results Found.</p>';
}
print $stable;
}

function printColumnHeaders(&$results) {
    $html = '';
    $headers = $results->getColumnHeaders();

    foreach ($headers as $header) {
        $html .= <<<HTML
<pre>
Column Name      = {$header->getName()}
Column Type      = {$header->getColumnType()}
Column Data Type = {$header->getDataType()}
</pre>
HTML;
    }
    print $html;
}

function printQueryInfo(&$results) {
    $query = $results->getQuery();
    $html = <<<HTML
<pre>
Ids              = {$query->getIds()}
Metrics          = {$query->getMetrics()}
Dimensions       = {$query->getDimensions()}
Sort             = {$query->getSort()}
Filters          = {$query->getFilters()}
Max Results     = {$query->getMax_results()}
</pre>
HTML;

    print $html;
}

function printProfileInfo(&$results) {
    $profileInfo = $results->getProfileInfo();

    $html = <<<HTML
<pre>
Account ID          = {$profileInfo->getAccountId()}
Web Property ID     = {$profileInfo->getWebPropertyId()}
Internal Web Property ID = {$profileInfo->getInternalWebPropertyId()}
Profile ID          = {$profileInfo->getProfileId()}
Profile Name        = {$profileInfo->getProfileName()}

```

```

Table ID                = {$profileInfo->getTableId()}
</pre>
HTML;

    print $html;
}

function printReportInfo(&$results) {
    $html = <<<HTML
    <pre>
Kind                    = {$results->getKind()}
ID                      = {$results->getId()}
Self Link               = {$results->getSelfLink()}
Total Results           = {$results->getTotalResults()}
    </pre>
HTML;

    print $html;
}

function printTotalsForAllResults(&$results) {
    $totals = $results->getTotalsForAllResults();

    foreach ($totals as $metricName => $metricTotal) {
        $html .= "Metric Name = $metricName\n";
        $html .= "Metric Total = $metricTotal";
    }

    print $html;
}

```

La versione originale faceva fronte alla [documentazione](#) ufficiale

Leggi API in tempo reale online: <https://riptutorial.com/it/google-analytics-api/topic/7305/api-in-tempo-reale>

# Capitolo 4: Autenticazione

## Sintassi

- OTTIENI [https://accounts.google.com/o/oauth2/auth?client\\_id= {clientid}](https://accounts.google.com/o/oauth2/auth?client_id={clientid}) e `redirect_uri = urn:ietf:wg:oauth:2.0:oob` & `scope = {Scopes}` & `response_type = codice`
- POST <https://accounts.google.com/o/oauth2/token> `code = {Codice da sopra chiamata}` & `client_id = {ClientId}` & `client_secret = {ClientSecret}` & `redirect_uri = urn:ietf:wg:oauth:2.0:oob` & `grant_type = codice_autorizzazione`
- POST <https://accounts.google.com/o/oauth2/token> `client_id = {ClientId}` e `client_secret = {ClientSecret}` e `refresh_token = {Dalla chiamata precedente}` & `grant_type = refresh_token`

## Parametri

Parametro	Descrizione
Identificativo cliente	Dalla console per <a href="#">sviluppatori di Google</a> identifica il tuo progetto o la tua applicazione
segreto	Dalla console per <a href="#">sviluppatori di Google</a> identifica il tuo progetto o la tua applicazione
redirect_uri	Dalla posizione dello <a href="#">sviluppatore di Google</a> a cui deve essere restituita l'autenticazione. Nel caso di Native applications <code>urn:ietf:wg:oauth:2.0:oob</code> può essere usato per denotare localhost

## Osservazioni

Per accedere a qualsiasi API di Google devi identificarti come sviluppatore e identificare il tuo progetto. Lo facciamo creando un nuovo progetto sulla [console di Google Developers](#) .

Quando crei il tuo progetto, se vuoi accedere alle API di Google Analytics devi abilitare le API a cui intendi accedere.

- **API di reporting** : accesso **all'API di reporting** di Google Analytics v4.
- **API di Analytics** : accesso a tutto il resto.

Ora devi decidere come vorresti accedere ai dati.

Con Google Data esistono due tipi di dati pubblici e privati.

- i dati pubblici non sono di proprietà di un utente. L'API dei metadati è un'API pubblica che non è necessario accedere per accedere a tali dati.
- L'API Reporting contiene dati di Google Analytics di utenti privati, non puoi guardarli a meno che l'utente non ti abbia autorizzato ad accedervi.

Se stai solo accedendo ai dati pubblici, tutto ciò che devi fare è creare una chiave API pubblica e sarai in grado di accedere all'API in questione. Se si accederà a dati utente privati, sarà necessario creare credenziali OAuth2 o credenziali dell'account di servizio.

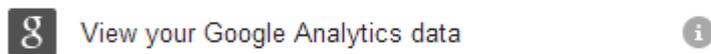
## Autorizzazione OAuth2

Per accedere ai dati degli utenti privati, è necessario disporre dell'autorizzazione del proprietario dei dati per accedervi. [OAuth2](#) ci consente di richiedere l'accesso da un utente.

Probabilmente hai già visto prima OAuth2.

### Google Analytics Windows ▾

This app would like to:



Google Analytics Windows and Google will use this information in accordance with their respective terms of service and privacy policies.

Cancel

Accept

L'applicazione "Google Analytics Windows" richiede l'accesso per visualizzare gli utenti "Dati di Google Analytics"

1. Google Analytics windows è il nome del progetto che è stato creato sulla Console per gli sviluppatori di Google.
2. Google Analytics Data è lo scopo delle autorizzazioni che abbiamo richiesto.

**Ambito** È necessario comunicare all'utente che cosa intendiamo fare l'API di analisi di Google ha due ambiti che è possibile utilizzare.

1. <https://www.googleapis.com/auth/analytics.readonly>
2. <https://www.googleapis.com/auth/analytics>

È meglio richiedere solo gli ambiti di cui hai bisogno. Se si leggeranno solo i dati degli utenti, sarà sufficiente richiedere l'ambito readonly.

## Account del servizio di autorizzazione

[Gli account di servizio](#) sono diversi in quanto sono pre-approvati. Se crei le credenziali dell'account di servizio, lo sviluppatore può prendere l'email dell'account di servizio e aggiungerla come utente sul tuo account Google Analytics **A livello di account**, questo concederà all'account di servizio l'accesso ai dati. Non è necessario aprire la finestra di autenticazione e richiedere l'accesso. L'account di servizio avrà accesso ai dati per tutto il tempo in cui è un utente nell'account Google Analytics.

## Conclusione

L'autenticazione è necessaria per accedere alla maggior parte dei dati esposti dall'API di Google Analytics.

**Non puoi utilizzare login / accesso e password del client per accedere a qualsiasi API di Google a partire da maggio 2015. È necessario utilizzare l'autenticazione aperta.**

\*\*

## Examples

### Oauth2 C #

Esempio utilizza la [libreria client Dotnet delle API di Google](#) .

PM> Install-Package Google.Apis.AnalyticsReporting.v4

```
/// <summary>
/// This method requests Authentcation from a user using Oauth2.
/// Credentials are stored in System.Environment.SpecialFolder.Personal
/// Documentation https://developers.google.com/accounts/docs/OAuth2
/// </summary>
/// <param name="clientSecretJson">Path to the client secret json file from Google
Developers console.</param>
/// <param name="userName">Identifying string for the user who is being
authenticated.</param>
/// <returns>DriveService used to make requests against the Drive API</returns>
public static AnalyticsReportingService AuthenticateOauth(string clientSecretJson, string
userName)
{
    try
    {
        if (string.IsNullOrEmpty(userName))
            throw new Exception("userName is required.");
        if (string.IsNullOrEmpty(clientSecretJson))
            throw new Exception("clientSecretJson is required.");
        if (!File.Exists(clientSecretJson))
            throw new Exception("clientSecretJson file does not exist.");

        // These are the scopes of permissions you need.
        string[] scopes = new string[] { AnalyticsReportingService.Scope.AnalyticsReadOnly
}; // View your Google Analytics Data

        UserCredential credential;
        using (var stream = new FileStream(clientSecretJson, FileMode.Open,
FileAccess.Read))
        {
            string credPath =
System.Environment.GetFolderPath(System.Environment.SpecialFolder.Personal);
            credPath = Path.Combine(credPath, ".credentials/",
System.Reflection.Assembly.GetExecutingAssembly().GetName().Name);

            // Requesting Authentication or loading previously stored authentication for
userName
```

```

        credential =
GoogleWebAuthorizationBroker.AuthorizeAsync(GoogleClientSecrets.Load(stream).Secrets,
                                            scopes,
                                            userName,

CancellationToken.None,

                                            new
FileDataStore(credPath, true)).Result;
    }

    // Create Drive API service.
    return new AnalyticsReportingService(new BaseClientService.Initializer()
    {
        HttpClientInitializer = credential,
        ApplicationName = string.Format("{0} Authentication",
System.Reflection.Assembly.GetExecutingAssembly().GetName().Name),
    });
}
catch (Exception ex)
{
    Console.WriteLine("Create Oauth2 DriveService failed" + ex.Message);
    throw new Exception("CreateOauth2DriveFailed", ex);
}
}

```

## Servizio Account Authentication Vb.net

Esempio utilizza il [pacchetto di installazione Google.Apis.AnalyticsReporting.v4](#)

```

Public Shared Function getServiceInitializer() As BaseClientService
    Dim serviceAccountCredentialFilePath = "Path to Json service account key file"
REM from Google Developers console
    Dim myKeyEMail = "XXX@developer.gserviceaccount.com" REM from Google Developers console
    Dim scope =
Google.Apis.AnalyticsReporting.v4.AnalyticsReportingService.Scope.AnalyticsReadonly

    Try

        Dim credential
        Using stream As New FileStream(serviceAccountCredentialFilePath, FileMode.Open,
FileAccess.Read)

            credential = GoogleCredential.FromStream(stream).CreateScoped(scope)

        End Using

        Dim initializer As New BaseClientService.Initializer()
        initializer.HttpClientInitializer = credential
        initializer.ApplicationName = "SRJCGMail"

        Dim service As New AnalyticsReportingService(initializer)
        Return service

    Catch ex As Exception
        Console.WriteLine(ex.Message)
        Return Nothing
    End Try

```

End Function

Leggi Autenticazione online: <https://riptutorial.com/it/google-analytics-api/topic/7268/autenticazione>

# Capitolo 5: Metadata api

## Sintassi

- HTTP OTTIENI [https://www.googleapis.com/analytics/v3/metadata/ {reportType} / columns?](https://www.googleapis.com/analytics/v3/metadata/{reportType}/columns?Key={APIKey})  
Key = {APIKey}
- HTTP GET [https://www.googleapis.com/analytics/v3/metadata/ {reportType} / columns?](https://www.googleapis.com/analytics/v3/metadata/{reportType}/columns?Access_token={Access_token})  
Access\_token = {Access\_token}

## Parametri

Nome del parametro	Descrizione
ReportType	Tipo di rapporto Valori ammessi: <b>ga</b> . Dove <b>ga</b> corrisponde alla <a href="#">Core Reporting API</a> .

## Osservazioni

L' [API dei metadati](#) restituisce l'elenco e gli attributi delle colonne (ovvero dimensioni e metriche) esposti nelle [API di reporting di Google Analytics](#) (v2, v3 e v4). Gli attributi restituiti includono nome dell'interfaccia utente, descrizione, supporto per segmenti e altro ancora.

Puoi utilizzare l'API dei metadati per:

- Scopri automaticamente nuove colonne
- Accedi a tutti gli attributi di dimensioni e metriche per le API di reporting di Google Analytics

Questa è la stessa lista presente in [Dimensions & Metrics Explorer](#) .

Nota: le dimensioni e le metriche delle canalizzazioni in tempo reale e multicanale non sono attualmente disponibili.

## Examples

### Esempio di riposo

Le chiamate [all'API dei metadati](#) sono con HTTP Get:

### Utilizzo della chiave API pubblica

OTTIENI [https://www.googleapis.com/analytics/v3/metadata/ga/columns?key={YOUR\\_API\\_KEY}](https://www.googleapis.com/analytics/v3/metadata/ga/columns?key={YOUR_API_KEY})

## Utilizzo del token di accesso da OAuth2 o dall'autenticazione dell'account di servizio

OTTIENI

[https://www.googleapis.com/analytics/v3/metadata/ga/columns?access\\_token={Authenticated\\_Access\\_Token}](https://www.googleapis.com/analytics/v3/metadata/ga/columns?access_token={Authenticated_Access_Token})

## Esempio di Java

utilizza la [libreria client Java](#)

```
/**
 * 1. Execute a Metadata Request
 * An application can request columns data by calling the list method on the Analytics service
 object.
 * The method requires an reportType parameter that specifies the column data to retrieve.
 * For example, the following code requests columns for the ga report type.
 */

try {
    Columns results = getMetadata(analytics);
    // Success

} catch (GoogleJsonResponseException e) {
    // Catch API specific errors.
    handleApiError(e);
} catch (IOException e) {
    // Catch general parsing network errors.
    e.printStackTrace();
}

/**
 * 2. Print out the Columns data
 * The components of the result can be printed out as follows:
 */

private static Columns getMetadata(Analytics analytics) throws IOException {
    String reportType = "ga";
    return analytics.metadata()
        .columns()
        .list(reportType)
        .execute();
}

private static void printMetadataReport(Columns results) {
    System.out.println("Metadata Response Report");
    printReportInfo(results);
    printAttributes(results.getAttributeNames());
    printColumns(results.getItems());
}

private static void printReportInfo(Columns results) {
    System.out.println("### Metadata Report Info ###");
    System.out.println("Kind: " + results.getKind());
    System.out.println("Etag: " + results.getEtag());
    System.out.println("Total Results: " + results.getTotalResults());
    System.out.println();
}
```

```

}

private static void printAttributes(List<String> attributeNames) {
    System.out.println("## Attribute Names ##");
    for (String attribute : attributeNames) {
        System.out.println(attribute);
    }
}

private static void printColumns(List<Column> columns) {
    System.out.println("## Columns ##");

    for (Column column : columns) {
        System.out.println();
        System.out.println("Column ID: " + column.getId());
        System.out.println("Kind: " + column.getKind());

        Map<String, String> columnAttributes = column.getAttributes();

        for (Map.Entry<String, String> attribute: columnAttributes.entrySet()) {
            System.out.println(attribute.getKey() + ": " + attribute.getValue());
        }
    }
}
}

```

Nota: prima versione copiata da [Metadata.list](#)

## Esempio PHP

Utilizza la [libreria client PHP](#)

```

/**
 * 1. Execute a Metadata Request
 * An application can request columns data by calling the list method on the Analytics service
 object.
 * The method requires an reportType parameter that specifies the column data to retrieve.
 * For example, the following code requests columns for the ga report type.
 */

try {

    $results = $analytics->metadata_columns->listMetadataColumns('ga');
    // Success

} catch (apiServiceException $e) {
    // Handle API service exceptions.
    $error = $e->getMessage();
}

/**
 * 2. Print out the Columns data
 * The components of the result can be printed out as follows:
 */

function printMetadataReport($results) {
    print '<h1>Metadata Report</h1>';
}

```

```

printReportInfo($results);
printAttributes($results);
printColumns($results);
}

function printReportInfo(&$results) {
    $html = '<h2>Report Info</h2>';
    $html .= <<<HTML
<pre>
Kind                = {$results->getKind()}
Etag                = {$results->getEtag()}
Total Results      = {$results->getTotalResults()}
</pre>
HTML;
    print $html;
}

function printAttributes(&$results) {
    $html = '<h2>Attribute Names</h2><ul>';
    $attributes = $results->getAttributeNames();
    foreach ($attributes as $attribute) {
        $html .= '<li>'. $attribute . '</li>';
    }
    $html .= '</ul>';
    print $html;
}

function printColumns(&$results) {
    $columns = $results->getItems();
    if (count($columns) > 0) {
        $html = '<h2>Columns</h2>';
        foreach ($columns as $column) {
            $html .= '<h3>' . $column->getId() . '</h3>';
            $column_attributes = $column->getAttributes();
            foreach ($column_attributes as $name=>$value) {
                $html .= <<<HTML
<pre>
{$name}: {$value}
</pre>
HTML;
            }
        }
    } else {
        $html = '<p>No Results Found.</p>';
    }
    print $html;
}

```

Nota: versione originale copiata da [metadata.list](#)

## Esempio di Python

Utilizza la [libreria client Python](#)

```

# 1. Execute a Metadata Request
# An application can request columns data by calling the list method on the Analytics service

```

```

object.
# The method requires an reportType parameter that specifies the column data to retrieve.
# For example, the following code requests columns for the ga report type.

try:
    results = service.metadata().columns().list(reportType='ga').execute()

except TypeError, error:
    # Handle errors in constructing a query.
    print ('There was an error in constructing your query : %s' % error)

except HttpError, error:
    # Handle API errors.
    print ('Arg, there was an API error : %s : %s' %
          (error.resp.status, error._get_reason()))

# 2. Print out the Columns data
# The components of the result can be printed out as follows:

def print_metadata_report(results):
    print 'Metadata Response Report'
    print_report_info(results)
    print_attributes(results.get('attributeNames'))
    print_columns(results)

def print_report_info(columns):
    print "Metadata Report Info"
    if columns:
        print 'Kind           = %s' % columns.get('kind')
        print 'Etag           = %s' % columns.get('etag')
        print 'Total Results = %s' % columns.get('totalResults')

def print_attributes(attributes):
    if attributes:
        print 'Attribute Names:'
        for attribute in attributes:
            print attribute

def print_columns(columns_data):
    if columns_data:
        print 'Columns:'

        columns = columns_data.get('items', [])

        for column in columns:
            print
            print '%15s = %35s' % ('Column ID', column.get('id'))
            print '%15s = %35s' % ('Kind', column.get('kind'))

            column_attributes = column.get('attributes', [])

            for name, value in column_attributes.iteritems():
                print '%15s = %35s' % (name, value)

```

Nota: versione originale copiata da [metadata.list](https://riptutorial.com/it/home)

## C # esempio

Utilizza la [libreria client .Net](#)

PM> Install-Package Google.Apis.Analytics.v3

```
var metadataService = new AnalyticsMetaDataService(new BaseClientService.Initializer()
    {
        ApiKey = {Public API KEY},
        ApplicationName = "Metadata api",
    });

var result = Service.Metadata.Columns.List("ga").Execute();
```

Leggi Metadata api online: <https://riptutorial.com/it/google-analytics-api/topic/7297/metadata-api>

---

# Capitolo 6: Risposte di errore

## Sintassi

- 400 invalidParameter Indica che un parametro di richiesta ha un valore non valido.
- 400 badRequest Indica che la query non è valida. Ad esempio, mancava l'ID genitore o la combinazione di dimensioni o metriche richieste non era valida.
- 403 insufficientPermissions Indica che l'utente non dispone di autorizzazioni sufficienti per l'entità specificata nella query.
- 403 dailyLimitExceeded Indica che l'utente ha superato la quota giornaliera (per progetto o per visualizzazione (profilo)).
- 403 usageLimits.userRateLimitExceededUnreg Indica che l'applicazione deve essere registrata nella Console API di Google.
- 403 userRateLimitExceeded Indica che il limite di velocità dell'utente è stato superato. Il limite massimo di velocità è 10 qps per indirizzo IP.
- 403 rateLimitExceeded Indica che i limiti globali o globali della velocità del progetto sono stati superati. Riprova usando il back-off esponenziale. È necessario rallentare la velocità con cui si inviano le richieste.
- 403 quota superata Indica che sono state raggiunte le 10 richieste simultanee per vista (profilo) nell'API di reporting principale.
- 429 RESOURCE\_EXHAUSTED AnalyticsDefaultGroupCLIENT\_PROJECT-1d Indica che le Richieste al giorno per quota di progetto sono state esaurite.
- 429 RESOURCE\_EXHAUSTED AnalyticsDefaultGroupCLIENT\_PROJECT-100s Indica che le Richieste per 100 secondi per quota di progetto sono state esaurite.
- 429 RESOURCE\_EXHAUSTED AnalyticsDefaultGroupUSER-100s Indica che le richieste per 100 secondi per utente per quota di progetto sono state esaurite.
- 429 RESOURCE\_EXHAUSTED DiscoveryGroupCLIENT\_PROJECT-100s Indica che le richieste di rilevamento per quota di 100 secondi sono state esaurite.
- 500 internalServerError Si è verificato un errore del server interno imprevisto. Non riprovare più volte questa query.
- 503 backendError Server ha restituito un errore. Non riprovare più volte questa query.

## Parametri

Nome	Descrizione
dominio	Posizione dell'errore ex: globale
ragionare	Motivo dell'errore
Messaggio	Un messaggio che spiega l'errore e una possibile soluzione.
LocationType	Il tipo di posizione dell'errore ex: paramater
Posizione	La posizione effettiva dell'errore

## Osservazioni

Se una richiesta ha esito positivo, l'API restituisce un codice di stato HTTP 200 insieme ai dati richiesti nel corpo della risposta.

Se si verifica un errore con una richiesta, l'API restituisce un codice di stato HTTP e un motivo nella risposta in base al tipo di errore. Inoltre, il corpo della risposta contiene una descrizione dettagliata di ciò che ha causato l'errore.

## Examples

### 400 invalidParameter

È importante leggere la risposta all'errore restituita dal server dell'API di Google Analytics. In molti casi possono dirti esattamente cosa c'è che non va.

```
400 invalidParameter

{
  "error": {
    "errors": [
      {
        "domain": "global",
        "reason": "invalidParameter",
        "message": "Invalid value '-1' for max-results. Value must be within the range: [1, 1000]",
        "locationType": "parameter",
        "location": "max-results"
      }
    ],
    "code": 400,
    "message": "Invalid value '-1' for max-results. Value must be within the range: [1, 1000]"
  }
}
```

In questo caso il messaggio:

"messaggio": "Valore non valido '-1' per risultati massimi. Il valore deve essere compreso nell'intervallo: [1, 1000]",

Ci sta dicendo che abbiamo inviato un valore -1 per il parametro max-results che non è valido, possiamo solo inviare un valore da 1-1000.

Leggi Risposte di errore online: <https://riptutorial.com/it/google-analytics-api/topic/7376/risposte-di-errore>

## Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con google-analytics-api	<a href="#">Community</a> , <a href="#">DalmTo</a> , <a href="#">K48</a> , <a href="#">MarkeD</a>
2	API di reporting (Analytics v4)	<a href="#">DalmTo</a>
3	API in tempo reale	<a href="#">DalmTo</a>
4	Autenticazione	<a href="#">DalmTo</a> , <a href="#">Matt</a>
5	Metadata api	<a href="#">DalmTo</a>
6	Risposte di errore	<a href="#">DalmTo</a>