Immagine che contiene disegnando

Descrizione generata automaticamente

**QUESTA PAGINA È LASCIATA**

**INTENZIONALMENTE BIANCA**

|  |
| --- |
| STATO DEL DOCUMENTO |

|  |  |  |  |
| --- | --- | --- | --- |
| TITOLO DEL DOCUMENTO | | | |
| Ermes | | | |
| CLASSIFICAZIONE DEL DOCUMENTO | | | |
| USO INTERNO | | | |
| EDIZ. | REV. | DATA | AGGIORNAMENTO |
| 1 | 0 | 24/09/2020 | Prima stesura |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| STATO DI AGGIORNAMENTO | | | |
| PAR | EDIZ. | REVISIONE | MOTIVO DELL'AGGIORNAMENTO |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |
| --- | --- |
| NUMERO TOTALE PAGINE: | 15 |

|  |
| --- |
| LISTA DI DISTRIBUZIONE |

INTERNA A:

ESTERNA A:

# Sommario

[Sommario 5](#_Toc52543748)

[Indice delle Figure 6](#_Toc52543749)

[1. Scopo del documento 7](#_Toc52543750)

[1.1 Documenti Allegati 7](#_Toc52543751)

[1.2 Glossario 7](#_Toc52543752)

[2. Dominio Applicativo 8](#_Toc52543753)

[2.1 Etimologia del Nome 8](#_Toc52543754)

[3. Interfacce di Interoperabilità 8](#_Toc52543755)

[3.1 Authorization 8](#_Toc52543756)

[3.2 PostStatus 9](#_Toc52543757)

[3.3 PostImage 9](#_Toc52543758)

[3.4 PostVideo 10](#_Toc52543759)

[3.5 Authorization 10](#_Toc52543760)

[3.6 PostTweet 11](#_Toc52543761)

[3.7 PostImage 11](#_Toc52543762)

[3.8 PostVideo 12](#_Toc52543763)

[3.9 SendMessage 13](#_Toc52543764)

[3.10 SendPhoto 13](#_Toc52543765)

[3.11 SendVideo 14](#_Toc52543766)

**Indice delle Tabelle**

[Tabella 1 – Documenti Allegati 7](#_Toc52543735)

[Tabella 2 – Glossario 7](#_Toc52543736)

[Tabella 3 – Interfaccia Authorization 9](#_Toc52543737)

[Tabella 4 – Interfaccia PostStatus 9](#_Toc52543738)

[Tabella 5 – Interfaccia PostImage 10](#_Toc52543739)

[Tabella 6 – Interfaccia PostVideo 10](#_Toc52543740)

[Tabella 7 – Interfaccia Authorization 11](#_Toc52543741)

[Tabella 8 – Interfaccia PostTweet 11](#_Toc52543742)

[Tabella 9 – Interfaccia PostImage 12](#_Toc52543743)

[Tabella 10 – Interfaccia PostVideo 12](#_Toc52543744)

[Tabella 11 – Interfaccia SendMessage 13](#_Toc52543745)

[Tabella 12 – Interfaccia SendPhoto 14](#_Toc52543746)

[Tabella 13 – Interfaccia SendVideo 14](#_Toc52543747)

# Indice delle Figure

**Non è stata trovata alcuna voce dell'indice delle figure.**

# Scopo del documento

Questo documento ha una caratterizzazione prettamente tecnica ed ha lo scopo di descrivere il dominio applicativo e le interfacce di interoperabilità del microservizio Ermes.

## Documenti Allegati

|  |  |
| --- | --- |
| Riferimento e titolo del documento | Descrizione |
| Ermes.yaml | File YAML di definizione delle interfacce |

Tabella 1 – Documenti Allegati

## Glossario

|  |  |
| --- | --- |
| Abbreviazione | Significato |
| REST | Representational State Transfer |
| YAML | YAML Ain’t Markup Language |
| API | Application Programming Interface |
| HTTP | Hypertext Transfer Protocol |
| JSON | JavaScript Object Notation |

Tabella 2 – Glossario

# Dominio Applicativo

Ermes è un microservizio per l’integrazione con i Social Network che implementa, mediante l’uso di librerie specifiche per ognuno di essi, servizi che permettono l’interazione con i Social Network.

Attualmente Ermes fornisce servizi per l’integrazione con i seguenti Social Network: Facebook, Twitter e Telegram.

## Etimologia del Nome

Divinità della mitologia greca, il ruolo principale di Ermes era quello di messaggero degli dèi.

I suoi simboli sono il gallo e la tartaruga, mentre i tratti identificanti sono il suo borsellino, i suoi sandali e cappello alati e il bastone da messaggero, il caduceo.

Un compendio sulla mitologia greca lo descrive come una divinità mobile, inoltre i suoi sandali ne cancellano le impronte in modo che nessuno possa notare la sua presenza. Questi elementi sono vicini al concetto di microservizio. Difatti, così come per Ermes, di un microservizio riconosciamo la presenza, usufruiamo dei servizi che offre ma senza sapere dove effettivamente sia.

Possiamo, simbolicamente, pensare all’interazione con i Social Network come ad una serie di messaggi affidati ad Ermes che, adempiendo al suo compito di messaggero, recapiterà ai Social Network.

# Interfacce di Interoperabilità

Il microservizio espone verso l’esterno delle interfacce REST suddivise per funzionalità offerte per Social Network. Di seguito viene effettuata una descrizione di massima dei servizi esposti e per un dettaglio maggiore delle interfacce e delle obbligatorietà degli elementi dei messaggi si rimanda al file YAML allegato denominato Ermes.yaml.

## Authorization

Questo servizio rende possibile ottenere l’autorizzazione ad eseguire comportamenti in vece dell’utente specificando:

* la key dell’applicazione Facebook
* il secret dell’applicazione Facebook
* I permessi per cui si vuole ottenere l’autorizzazione

|  |  |
| --- | --- |
| **Dettaglio servizio Authorization** | |
| GET 🡪 /facebook/authorization?key&secret&permissions | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "accessToken": "string"  "applicationKey": "string"  "applicationName": "string"  "expires": "string"  "userId": "string"  }  } |

Tabella 3 – Interfaccia Authorization

## PostStatus

Questo servizio rende possibile la pubblicazione di un nuovo stato su una pagina Facebook specificando:

* Il token dell’utente in vece di cui si effettua la pubblicazione
* la key dell’applicazione Facebook
* il secret dell’applicazione Facebook
* la pagina Facebook su cui pubblicare lo stato
* il contenuto testuale dello stato

|  |  |
| --- | --- |
| **Dettaglio servizio PostStatus** | |
| POST 🡪 /facebook/postStatus?token&key&secret&permissions&page\_name&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 4 – Interfaccia PostStatus

Il parametro “token” è opzionale, in caso di sua assenza il servizio reperirà il token interrogando Facebook usando le informazioni in suo possesso.

## PostImage

Questo servizio rende possibile la pubblicazione di un nuovo stato con immagine allegata su una pagina Facebook specificando:

* Il token dell’utente in vece di cui si effettua la pubblicazione
* la key dell’applicazione Facebook
* il secret dell’applicazione Facebook
* la pagina Facebook su cui pubblicare lo stato
* l’URL pubblico dell’immagine da pubblicare
* l’eventuale descrizione da allegare all’immagine

|  |  |
| --- | --- |
| **Dettaglio servizio PostImage** | |
| POST 🡪 /facebook/postImage?token&key&secret&permissions&page\_name&image\_url&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 5 – Interfaccia PostImage

Il parametro “token” è opzionale, in caso di sua assenza il servizio reperirà il token interrogando Facebook usando le informazioni in suo possesso.

## PostVideo

Questo servizio rende possibile la pubblicazione di un nuovo stato con immagine allegata su una pagina Facebook specificando:

* Il token dell’utente in vece di cui si effettua la pubblicazione
* la key dell’applicazione Facebook
* il secret dell’applicazione Facebook
* la pagina Facebook su cui pubblicare lo stato
* l’URL pubblico del video da pubblicare
* l’eventuale descrizione da allegare al video

|  |  |
| --- | --- |
| **Dettaglio servizio PostVideo** | |
| POST 🡪 /facebook/postVideo?token&key&secret&permissions&page\_name&video\_url&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 6 – Interfaccia PostVideo

Il parametro “token” è opzionale, in caso di sua assenza il servizio reperirà il token interrogando Facebook usando le informazioni in suo possesso.

## Authorization

Questo servizio rende possibile ottenere l’autorizzazione ad eseguire comportamenti in vece dell’utente specificando:

* la key dell’applicazione Twitter
* il secret dell’applicazione Twitter

|  |  |
| --- | --- |
| **Dettaglio servizio Authorization** | |
| GET 🡪 /twitter/authorization?key&secret | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "accessToken": "string"  "accessTokenSecret": "string"  "applicationKey": "string"  "applicationSecret": "string"  "userId": "string"  }  } |

Tabella 7 – Interfaccia Authorization

## PostTweet

Questo servizio rende possibile la pubblicazione di un nuovo tweet su un profilo Twitter specificando:

* Il token dell’utente in vece di cui si effettua la pubblicazione
* Il secret del token
* l’id dell’utente
* la key dell’applicazione Twitter
* il secret dell’applicazione Twitter
* il contenuto testuale del tweet

|  |  |
| --- | --- |
| **Dettaglio servizio PostTweet** | |
| POST 🡪 /twitter/postTweet?token&token\_secret&user\_id&key&secret&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 8 – Interfaccia PostTweet

I parametri “token”, “token\_secret” e “user\_id” sono opzionali, in caso di loro assenza il servizio reperirà il token interrogando Twitter usando le informazioni in suo possesso.

## PostImage

Questo servizio rende possibile la pubblicazione di un nuovo tweet con immagine allegata su un profilo Twitter specificando:

* Il token dell’utente in vece di cui si effettua la pubblicazione
* Il secret del token
* L’id dell’utente
* la key dell’applicazione Twitter
* il secret dell’applicazione Twitter
* l’URL pubblico dell’immagine da pubblicare
* l’eventuale descrizione da allegare all’immagine

|  |  |
| --- | --- |
| **Dettaglio servizio PostImage** | |
| POST 🡪 /twitter/postImage?token&token\_secret&user\_id&key&secret&image\_url&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 9 – Interfaccia PostImage

I parametri “token”, “token\_secret” e “user\_id” sono opzionali, in caso di loro assenza il servizio reperirà il token interrogando Twitter usando le informazioni in suo possesso.

## PostVideo

Questo servizio rende possibile la pubblicazione di un nuovo tweet con immagine allegata su un profilo Twitter specificando:

* Il token dell’utente in vece di cui si effettua la pubblicazione
* Il secret del token
* l’id dell’utente
* la key dell’applicazione Twitter
* il secret dell’applicazione Twitter
* l’URL pubblico del video da pubblicare
* l’eventuale descrizione da allegare al video

|  |  |
| --- | --- |
| **Dettaglio servizio PostVideo** | |
| POST 🡪 /twitter/postVideo?token&token\_secret&user\_id&key&secret&video\_url&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 10 – Interfaccia PostVideo

I parametri “token”, “token\_secret” e “user\_id” sono opzionali, in caso di loro assenza il servizio reperirà il token interrogando Twitter usando le informazioni in suo possesso.

## SendMessage

Questo servizio rende possibile l’invio di un nuovo messaggio su un gruppo o canale Telegram specificando:

* il token del bot Telegram
* l’id del canale o gruppo cui inviare il messaggio
* il contenuto testuale del messaggio

|  |  |
| --- | --- |
| **Dettaglio servizio SendMessage** | |
| POST 🡪 /telegram/sendMessage?token&chat\_id&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 11 – Interfaccia SendMessage

Il parametro “chat\_id” può essere fornito in tre modi diversi:

1. Fornendo l’URL del canale o gruppo recuperato da Telegram Web
   1. Disponibile per canali pubblici o privati e per gruppi pubblici o privati
2. Fornendo il link ad un messaggio recuperato dal canale o gruppo
   1. Possibile sia da Telegram Web sia da Telegram App
   2. Disponibile solo per canali pubblici o privati e per gruppi pubblici
   3. Questo metodo non può essere utilizzato per gruppi privati
3. Fornendo l’id del canale o gruppo recuperato dal link d’invito oppure dall’URL del canale o gruppo recuperato da Telegram Web
   1. Disponibile solo per canali pubblici e per gruppi pubblici

## SendPhoto

Questo servizio rende possibile l’invio di un nuovo messaggio con foto allegata su un gruppo o canale Telegram specificando:

* il token del bot Telegram
* l’id del canale o gruppo cui inviare il messaggio
* l’URL pubblico della foto da pubblicare
* l’eventuale contenuto testuale del messaggio

|  |  |
| --- | --- |
| **Dettaglio servizio SendPhoto** | |
| POST 🡪 /telegram/sendPhoto?token&chat\_id&image\_url&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 12 – Interfaccia SendPhoto

Il parametro “chat\_id” può essere fornito in tre modi diversi:

1. Fornendo l’URL del canale o gruppo recuperato da Telegram Web
   1. Disponibile per canali pubblici o privati e per gruppi pubblici o privati
2. Fornendo il link ad un messaggio recuperato dal canale o gruppo
   1. Possibile sia da Telegram Web sia da Telegram App
   2. Disponibile solo per canali pubblici o privati e per gruppi pubblici
   3. Questo metodo non può essere utilizzato per gruppi privati
3. Fornendo l’id del canale o gruppo recuperato dal link d’invito oppure dall’URL del canale o gruppo recuperato da Telegram Web
   1. Disponibile solo per canali pubblici e per gruppi pubblici

## SendVideo

Questo servizio rende possibile l’invio di un nuovo messaggio con video allegato su un gruppo o canale Telegram specificando:

* il token del bot Telegram
* l’id del canale o gruppo cui inviare il messaggio
* l’URL pubblico del video da pubblicare
* l’eventuale contenuto testuale del messaggio

|  |  |
| --- | --- |
| **Dettaglio servizio SendVideo** | |
| POST 🡪 /telegram/sendVideo?token&chat\_id&video\_url&text | |
| Esempio Request | Esempio Response |
|  | {  "code": 200,  "success": true,  "message": "string",  "data": {  "link": "string"  }  } |

Tabella 13 – Interfaccia SendVideo

Il parametro “chat\_id” può essere fornito in tre modi diversi:

1. Fornendo l’URL del canale o gruppo recuperato da Telegram Web
   1. Disponibile per canali pubblici o privati e per gruppi pubblici o privati
2. Fornendo il link ad un messaggio recuperato dal canale o gruppo
   1. Possibile sia da Telegram Web sia da Telegram App
   2. Disponibile solo per canali pubblici o privati e per gruppi pubblici
   3. Questo metodo non può essere utilizzato per gruppi privati
3. Fornendo l’id del canale o gruppo recuperato dal link d’invito oppure dall’URL del canale o gruppo recuperato da Telegram Web
   1. Disponibile solo per canali pubblici e per gruppi pubblici