

DashboardServer

Reperimento dati complessivi nel db:

1. Elenco degli utenti
2. Elenco delle fonti
3. Elenco dei post
4. Elenco dei topics
5. Elenco delle coppie (Post, Topic)
6. Elenco delle coppie (Utente, Fonte)

Queries:

7. Ricerca utente, inserimento, aggiornamento e cancellazione di un utente mediante ID
8. Ricerca topic per ID
9. Ricerca fonte per ID
10. Ricerca Post per ID
11. Ricerca utenti per E-Mail, cancellazione, aggiornamento di un utente mediante E-Mail
12. Ricerca Topic per stringa
13. Ricerca fonte per nome della pagina
14. Ricerca fonte per nome
15. Ricerca dei post contenenti un dato topic (id post – id topic)
16. Ricerca dei topic contenuti in un dato post (id post – id topic)
17. Numero dei post relativi ad una fonte in un intervallo di date
18. Numero dei post positivi relativi ad una fonte in un intervallo di date
19. Numero dei post negativi relativi ad una fonte in un intervallo di date
20. Numero dei post relativi ad una fonte e un dato topic in un intervallo di date
21. Numero dei post positivi relativi ad una fonte e ad un dato topic in un intervallo di date
22. Numero dei post negativi relativi ad una fonte e ad un dato topic in un intervallo di date
23. Elenco dei link ai post relativi ad una fonte
24. Elenco dei link ai post positivi relativi ad una fonte
25. Elenco dei link ai post negativi relativi ad una fonte
26. Elenco dei link ai post relativi ad una fonte per un dato topic
27. Elenco dei link ai post positivi relativi ad una fonte per un dato topic
28. Elenco dei link ai post negativi relativi ad una fonte per un dato topic
29. Elenco delle pagine delle fonti relative ad dato topic
30. Elenco ID delle Fonti accessibili da un utente
31. Aggiungi permesso di lettura di una data fonte per utente
32. Elimina permesso di lettura di una data fonte per utente
33. Disattivazione di un utente
34. Modifica password di un utente
35. Aggiunta di un utente

Base URL: http://localhost:8080/DashboardServer/api

REPERIMENTO DATI COMPLESSIVI NEL DB

1. Elenco degli utenti

Fornisce l'elenco degli utenti presenti nel db con tutti i relativi campi.

Percorso	/Utenti	
Parametri	Nessuno	
Metodi	GET	
Media_Type	APPLICATION_JSON	
Attributi del risultato e tipo	ID	int
	Nome	String
	Cognome	String
	Email	String
	Password	String
	Data_nascita	String
	Citta	String
	Creato_da	int
	Data_ins	String
	Admin	Boolean
Esempio di risultato:	<pre>[{"ID":<int>,{ "Nome":<String>,{ "Cognome":<String>,{ "Email":<String>,{ "Password":<String>,{ "Data_nascita":<String>,{ "Citta":<String>,{ "Creato_da":<int>,{ "Data_ins":<String>,{ "Admin":<boolean>}}}}}}]]</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error":{"text": stringa d'errore}}</pre>	

2.Elenco delle fonti

Fornisce l'elenco delle fonti presenti nel db con tutti i relativi campi.

Percorso	/Sorgenti
Parametri	Nessuno
Metodi	GET

Media_Type	APPLICATION_JSON	
Attributi del risultato e tipo	ID	int
	Nome	String
	Pagina	String
	Link	String
	Tipo	String
	Autore	int
	Icona	String
Esempio di risultato:	<pre>[{"ID":"<int>","Nome":"<String>","Pagina":"<String>","Link":"<String>","Tipo":"<String>","Autore":"<int>","Icona":"<String>"},{}]</pre> oppure stringa JSON d'errore <pre>{"error":{"text": stringa d'errore}}</pre>	

3.Elenco dei Post

Fornisce l'elenco dei post presenti nel db con tutti i relativi campi.

Percorso	/Post	
Parametri	Nessuno	
Metodi	GET	
Media_Type	APPLICATION_JSON	
Attributi del risultato e tipo	ID	int
	Link	String
	Data	String
	Polarity	String enum('-1','+1','0')
	IDFonte	int
	Text	String
Esempio di risultato:	<pre>[{"ID":"<int>","Link":"<String>","Data":"<String>","Polarity":"<String>","ID_Fonte":"<int>","Text":"<String>"},{}]</pre> oppure stringa JSON d'errore <pre>{"error":{"text": stringa d'errore}}</pre>	

4.Elenco dei topics

Fornisce l'elenco dei topics presenti nel db con tutti i relativi campi.

Percorso	/Topics	
Parametri	Nessuno	
Metodi	GET	
Media_Type	APPLICATION_JSON	
Attributi del risultato e tipo[{"ID":"1","Topic":"primotopic"}]	ID	int
	Topic	String
Esempio di risultato:	[{"ID":"<int>","Topic":"<String>"}] oppure stringa JSON d'errore {"error":{"text": stringa d'errore }} [{"ID":"1","Topic":"primotopic"}]	

5.Elenco delle coppie (Post, Topic)

Fornisce l'elenco delle associazioni tra post e topic;
ciascun post e ciascun topic è identificato dal proprio id;

Percorso	/Contenuto	
Parametri	Nessuno	
Metodi	GET	
Media_Type	APPLICATION_JSON	
Attributi del risultato e tipo	IDPost	int
	IDTopic	Int
Esempio di risultato:	[{"IDPost":"<int>","IDTopic":"<int>"}] oppure stringa JSON d'errore {"error":{"text": stringa d'errore }} [{"ID":"1","Topic":"primotopic"}]	

6. Elenco delle coppie (Utente, Fonte)

Fornisce l'elenco delle associazioni tra utente e fonte;
ciascun utente e ciascuna fonte è identificato dal proprio id;

Percorso	/Accesso	
Parametri	Nessuno	
Metodi	GET	
Media_Type	APPLICATION_JSON	
Attributi del risultato e tipo	IDUtente	int
	IDSorgente	Int
Esempio di risultato:	<pre>[{"IDUtente": "<int>", "IDSorgente": "<int>"}]</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error": {"text": stringa d'errore}}</pre> <pre>[{"ID": "1", "Topic": "primotopic"}]</pre>	

QUERIES

7.Ricerca utente per ID

Fornisce le informazioni relative all'utente cercato mediante ID e permette la cancellazione, l'inserimento, l'aggiornamento di un utente già presente nel db.

Percorso	/Utenti/id/id={ID}	
Parametri	{ID} : int	
Metodi	GET, PUT, POST, DELETE	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	ID	int
	Nome	String
	Cognome	String
	Email	String
	Password	String
	Data_nascita	String
	Citta	String
	Creato da	int

	Data_ins	String
	Admin	boolean
Esempio di risultato:	<pre>{ "ID":<int>, {"Nome":<String>, {"Cognome":<String>, {"Email":<String>, {"Password":<String>, {"Data_nascita":<String>, {"Citta":<String>, {"Creato_da":<int>, {"Data_ins":<String>, {"Admin":<boolean>}}</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error":{"text": stringa d'errore}}</pre>	

PUT		
Metodo	updateUtenti(ID , payload):void	{ID} :int ID dell'utente di cui si vogliono aggiornare le informazioni
		payload :String stringa in formato JSON contenente le info aggiornate sull'utente

POST		
Metodo	addUtenti(payload):String	payload :String stringa in formato JSON contenente le info sull'utente da inserire nel db
Esempio di risultato	<p>Stringa JSON con le info sull'utente inserito:</p> <pre>{ "ID":<int>, {"Nome":<String>, {"Cognome":<String>, {"Email":<String>, {"Password":<String>, {"Data_nascita":<String>, {"Citta":<String>, {"Creato_da":<int>, {"Data_ins":<String>, {"Admin":<boolean>}}</pre> <p>oppure stringa JSON d'errore {"error": {"text": stringa d'errore}}</p>	

DELETE		
Metodo	deleteUtenti(ID):void	{ID} :int ID dell'utente da eliminare

8.Ricerca topic per ID

Fornisce le informazioni relative al topic cercato mediante ID e permette la cancellazione, l'inserimento, l'aggiornamento di un topic già presente nel db.

Percorso	/Topics/search/id={ID}
Parametri	ID : int

Metodi	GET, PUT, POST, DELETE	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	ID	int
	Nome	String
	Cognome	String
	Email	String
	Password	String
	Data_nascita	String
	Citta	String
	Creato_da	int
	Data_ins	String
	Admin	boolean
Esempio di risultato:	<pre>{"ID":"<int>","Topic":"<String>"}</pre> oppure stringa JSON d'errore <pre>{"error":{"text": stringa d'errore}}</pre>	

PUT		
Metodo	updateUtenti({ID} , payload):void	{ID} :int ID dell'utente di cui si vogliono aggiornare le informazioni
		payload :String stringa in formato JSON contenente le info aggiornate sull'utente

POST		
Metodo	addUtenti(payload):String	payload :String stringa in formato JSON contenente le info sull'utente da inserire nel db
Esempio di risultato	Stringa JSON con le info sull'utente inserito: <pre> {"ID":<int>,"Nome":<String>,"Cognome":<String>,"Email":<String>,"Password":<String>,"Data_nascita":<String>,"Citta":<String>,"Creato_da":<int>,"Data_ins":<String>,"Admin":<boolean>} oppure stringa JSON d'errore {"error": {"text": stringa d'errore}} </pre>	

DELETE		
Metodo	deleteUtenti({ID}):void	{ID} :int ID dell'utente da eliminare

9. Ricerca fonte per ID

Fornisce le informazioni relative alla fonte cercata mediante ID e permette la cancellazione, l'inserimento, l'aggiornamento di una fonte già presente nel db.

Percorso	/Sorgenti/search/id/id={ID}	
Parametri	{ID} : int	
Metodi	GET, PUT, POST, DELETE	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	ID	int
	Nome	String
	Pagina	String
	Link	String
	Tipo	String
	Autore	int
	Icona	String
Esempio di risultato:	<pre>{"ID":"<int>", "Nome":"<String>", "Pagina":"<String>", "Link":"<String>", "Tipo":"<String>", "Autore":"<int>", "Icona":"<String>", }, }</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error":{"text": stringa d'errore}}</pre>	

PUT		
Metodo	updateSorgente({ID}, payload):void	{ID} :int ID della fonte di cui si vogliono aggiornare le informazioni
		payload :String stringa in formato JSON contenente le info aggiornate sulla fonte

POST		
Metodo	addSorgente(payload):String	payload :String stringa in formato JSON contenente le info sulla fonte da inserire nel db
Esempio di risultato	Stringa JSON con le info sulla fonte inserita: <pre>[{"ID":"<int>", "Nome":"<String>", "Pagina":"<String>", "Link":"<String>", "Tipo":"<String>", "Autore":"<int>", "Icona":"<String>", }]</pre> <p>oppure stringa JSON d'errore</p>	

	<code>{"error": {"text": stringa d'errore}}</code>
--	--

DELETE		
Metodo	<code>deleteSorgente({ID}):void</code>	<code>{ID}</code> :int ID della fonte da eliminare

10. Ricerca post per ID

Fornisce le informazioni relative al post cercato mediante ID e permette la cancellazione, l'inserimento, l'aggiornamento di un post già presente nel db.

Percorso	/Post/id={ID}	
Parametri	{ID} : int	
Metodi	GET, PUT, POST, DELETE	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	ID	int
	Link	String
	Data	String
	Polarity	String enum('-1','+1','0')
	ID_Fonte	int
	Text	String
Esempio di risultato:	<pre>{ "ID": "<int>", "Link": "<String>", "Data": "<String>", "Polarity": "<String>", "ID_Fonte": "<int>", "Text": "<String>", }</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error":{"text": stringa d'errore}}</pre>	

PUT		
Metodo	<code>updatePost({ID}, payload):void</code>	<code>{ID}</code> :int ID del post di cui si vogliono aggiornare le informazioni
		<code>payload</code> :String stringa in formato JSON contenente le info aggiornate sul post

POST		
Metodo	<code>addPost(payload):String</code>	<code>payload</code> :String stringa in formato JSON contenente le info sul post da inserire nel db
Esempio di risultato	<p>Stringa JSON con le info sul post inserito:</p> <pre>[{"ID": "<int>", "Link": "<String>", "Data": "<String>", "Polarity": "<String>", "ID_Fonte": "<int>", "Text": "<String>", }]</pre>	

	oppure stringa JSON d'errore <pre>{ "error": { "text": stringa d'errore } }</pre>
--	--

DELETE		
Metodo	deletePost({ID}):void	{ID} :int ID del post da eliminare

11. Ricerca utente per E-Mail

Fornisce le informazioni relative all'utente cercato mediante E-Mail e permette la cancellazione, l'inserimento, l'aggiornamento di un utente già presente nel db.

Percorso	/Utenti/id/id={Email}	
Parametri	{Email} :String	
Metodi	GET, PUT, POST, DELETE	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	ID	int
	Nome	String
	Cognome	String
	Email	String
	Password	String
	Data_nascita	String
	Citta	String
	Creato_da	int
	Data_ins	String
	Admin	boolean
Esempio di risultato:	<pre>{ "ID":<int>, {"Nome":<String>, {"Cognome":<String>, {"Email":<String>, {"Password":<String>, {"Data_nascita":<String>, {"Citta":<String>, {"Creato_da":<int>, {"Data_ins":<String>, {"Admin":<boolean>}</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error":{"text": stringa d'errore}}</pre>	

PUT		
Metodo	updateUtenti({Email} , payload):void	{Email} :String e-Mail dell'utente di cui si vogliono aggiornare le

		informazioni
		payload :String stringa in formato JSON contenente le info aggiornate sull'utente

DELETE		
Metodo	deleteUtenti({Email}):void	{Email} :String E-Mail dell'utente da eliminare

12. Ricerca topic per stringa

Fornisce le informazioni relative al topic cercato mediante stringa e permette la cancellazione, l'inserimento, l'aggiornamento di un topic già presente nel db.

Percorso	/Topics/search/Topic={topic}	
Parametri	{topic} :String	
Metodi	GET, PUT, DELETE	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	ID	int
	Topic	String
Esempio di risultato:	<pre>{"ID":"1","Topic":"primotopic"}</pre> oppure stringa JSON d'errore <pre>{"error":{"text": stringa d'errore}}</pre>	

PUT		
Metodo	updateTopics({topic} , payload):void	{topic} :String stringa identificante il topic da aggiornare
		payload :String stringa in formato JSON contenente le info aggiornate sul topic

DELETE		
Metodo	deleteUtenti({topic}):void	{topic} :String topic da eliminare

13. Ricerca fonte per nome della pagina

Fornisce le informazioni relative alla fonte individuata mediante il nome di una specifica pagina

Percorso	/Sorgenti/search/pagina/pagina={Pagina}	
Parametri	{Pagina} :String	
Metodi	GET	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	ID	int
	Nome	String
	Pagina	String
	Tipo	String
	Autore	int
	Icona	String
Esempio di risultato:	<pre>[{"ID":"<int>", "Nome":"<String>", "Pagina":"<String>", "Link":"<String>", "Tipo":"<String>", "Autore":"<int>", "Icona":"<String>", }]</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error":{"text": stringa d'errore}}</pre>	

14. Ricerca fonte per nome

Fornisce l'elenco di tutte le specifiche fonti relative ad una stesso nome (che rappresenta la super-fonte)

Percorso	/Sorgenti/search/nome/nome={Nome}	
Parametri	{Nome} :String	
Metodi	GET	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	ID	int
	Nome	String
	Pagina	String
	Link	String
	Tipo	String
	Autore	int

	Icona	String
Esempio di risultato:	<pre>[{"ID":"<int>","Nome":"<String>","Pagina":"<String>","Link":"<String>","Tipo":"<String>","Autore":"<int>","Icona":"<String>"}]</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error":{"text": stringa d'errore}}</pre>	

15. Ricerca dei post contenenti un dato topic (id post – id topic)

Fornisce la coppia costituita dall'id del post e dall'id del topic cercato solo se è presente nel db.

Percorso	/Contenuto/idtopic={IDTopic}	
Parametri	{IDTopic} :int	
Metodi	GET	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	IDPost	int
	IDTopic	int
Esempio di risultato:	<pre>[{"IDPost":"<int>","IDTopic":"<int>"}]</pre> oppure stringa JSON d'errore <pre>{"error":{"text": stringa d'errore}}</pre>	

PUT		
Metodo	updateContenuto({IDTopic}, payload):void	{IDtopic} :int ID del topic di cui si vogliono aggiornare le informazioni nella relazione tra post e topic
		payload :String stringa in formato JSON contenente le info aggiornate sulla coppia (idpost, idtopic) rappresentante la relazione tra post e topic

POST		
Metodo	addContenuto(payload):String	payload :String stringa in formato JSON contenente la coppia id del post – id del topic
Esempio di risultato	Stringa JSON con le info sulla coppia inserita:	

	<pre>[{"IDPost": "<int>", "IDTopic": "<int>", }]</pre> <p>oppure stringa JSON d'errore</p> <pre>{"error": {"text": stringa d'errore}}</pre>
--	---

DELETE		
Metodo	deleteContenuto({IDTopic}):void	{IDTopic} :int ID del topic del quale si vogliono eliminare le relazioni con i post

16. Ricerca dei topic contenuti in un dato post (id post – id topic)

Fornisce la coppia costituita dall'id del post e dall'id del topic cercato solo se è presente nel db.

Percorso	/Contenuto/idpost={IDPost}	
Parametri	{IDPost} :int	
Metodi	GET	
Media_Type	APPLICATION_JSON	
GET		
Attributi del risultato e tipo	IDPost	int
	IDTopic	int
Esempio di risultato:	[{"IDPost": "<int>", "IDTopic": "<int>", }] oppure stringa JSON d'errore { "error": { "text": stringa d'errore } }	

PUT		
Metodo	updateContenuto({IDPost} , payload):void	{IDPost} :int ID del post di cui si vogliono aggiornare le informazioni nella relazione tra post e topic
		payload :String stringa in formato JSON contenente le info aggiornate sulla coppia (idpost, idtopic) rappresentante la relazione tra post e topic

POST

Metodo	addContenuto(payload):String	payload :String stringa in formato JSON contenente la coppia id del post – id del topic
Esempio di risultato	Stringa JSON con le info sulla coppia inserita: <pre>[{"IDPost": "<int>", "IDTopic": "<int>",}]</pre> oppure stringa JSON d'errore <pre>{"error": {"text": stringa d'errore}}</pre>	

DELETE		
Metodo	deleteContenuto({IDPost}):void	{IDPost} :int ID del post del quale si vogliono eliminare le relazioni con i post

17. Numero dei post relativi ad una fonte in un intervallo di date

Fornisce il numero di post di una fonte in un intervallo di date

Percorso	/Sorgenti/grafico/general/idfonte={ID_Fonte}/data1={Data1}/data2={Data2}
Parametri	{ID_Fonte} :int {Data1} :String formato AAAA-MM-GG {Data2} :String formato AAAA-MM-GG, posteriore alla prima
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<int> oppure 0

18. Numero dei post positivi relativi ad una fonte in un intervallo di date

Fornisce il numero di post con polarità positiva di una fonte in un intervallo di date

Percorso	/polarity/positive/general/idfonte={ID_Fonte}/data1={Data1}/data2={Data2}
Parametri	{ID_Fonte} :int {Data1} :String formato AAAA-MM-GG {Data2} :String formato AAAA-MM-GG, posteriore alla prima

Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<int> oppure 0

19. Numero dei post negativi relativi ad una fonte in un intervallo di date

Fornisce il numero di post con polarità negativa di una fonte in un intervallo di date

Percorso	/polarity/negative/general/idfonte={ID_Fonte}/data1={Data1}/data2={Data2}
Parametri	{ID_Fonte} :int {Data1} :String formato AAAA-MM-GG {Data2} :String formato AAAA-MM-GG, posteriore alla prima
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<int> oppure 0

20. Numero dei post relativi ad una fonte e un dato topic in un intervallo di date

Fornisce il numero di post di una fonte relativi ad un dato topic in un intervallo di date

Percorso	/Sorgenti/grafico/general/idfonte={ID_Fonte}/idtopic={ID_Topic}/data1={Data1}/data2={Data2}
Parametri	{ID_Fonte} :int {ID_Topic} : int {Data1} :String formato AAAA-MM-GG {Data2} :String formato AAAA-MM-GG, posteriore alla prima
Metodi	GET
Media_Type	TEXT_PLAIN
GET	

Esempio di risultato:	<int> oppure 0
-----------------------	-----------------------

21. Numero dei post positivi relativi ad una fonte e ad un dato topic in un intervallo di date

Fornisce il numero di post di una fonte con polarità positiva relativi ad un dato topic in un intervallo di date

Percorso	/polarity/negative/topic/idfonte={ID_Fonte}/data1={Data1}/data2={Data2}/topic={topic}
Parametri	{ID_Fonte} :int {Data1} :String formato AAAA-MM-GG {Data2} :String formato AAAA-MM-GG, posteriore alla prima {topic} :String
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<int> oppure 0

22. Numero dei post negativi relativi ad una fonte e ad un dato topic in un intervallo di date

Fornisce il numero di post di una fonte con polarità negativa relativi ad un dato topic in un intervallo di date

Percorso	/polarity/negative/topic/idfonte={ID_Fonte}/data1={Data1}/data2={Data2}/topic={topic}
Parametri	{ID_Fonte} :int {Data1} :String formato AAAA-MM-GG {Data2} :String formato AAAA-MM-GG, posteriore alla prima {topic} :String
Metodi	GET
Media_Type	TEXT_PLAIN
GET	

Esempio di risultato:	<int> oppure 0
-----------------------	------------------------------

23. Elenco dei link ai post relativi ad una fonte

Restituisce una stringa contenente i link dei post relativi ad una data fonte, separati da una virgola

Percorso	/links/general/all/idfonte={IDFonte}/limit={inf}-{sup}
Parametri	{IDFonte} :int {inf} :int, cardinalità della riga dalla quale iniziare a considerare il risultato {sup} :int cardinalità della riga alla quale terminare di considerare il risultato NOTA: {int} e {sup} definiscono la parte terminale della query: ovvero "...LIMIT {inf} , {sup}"
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<String>,<String>,<String> oppure '' stringa vuota

24. Elenco dei link ai post positivi relativi ad una fonte

Restituisce una stringa contenente i link dei post con polarità positiva relativi ad una data fonte, separati da una virgola

Percorso	/links/general/positive/idfonte={IDFonte}/limit={inf}-{sup}
Parametri	{IDFonte} :int {inf} :int, cardinalità della riga dalla quale iniziare a considerare il risultato {sup} :int cardinalità della riga alla quale terminare di considerare il risultato NOTA: {int} e {sup} definiscono la parte terminale della query: ovvero "...LIMIT {inf} , {sup}"
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<String>,<String>,<String>

	oppure '' stringa vuota
--	--------------------------------

25. Elenco dei link ai post negativi relativi ad una fonte

Restituisce una stringa contenente i link dei post con polarità negativa relativi ad una data fonte, separati da una virgola

Percorso	/links/general/negative/idfonte={IDFonte}/limit={inf}-{sup}
Parametri	{IDFonte} :int {inf} :int, cardinalità della riga dalla quale iniziare a considerare il risultato {sup} :int cardinalità della riga alla quale terminare di considerare il risultato NOTA: {int} e {sup} definiscono la parte terminale della query: ovvero "...LIMIT {inf} , {sup}"
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<String>,<String>,<String> oppure '' stringa vuota

26. Elenco dei link ai post relativi ad una fonte per un dato topic

Restituisce una stringa contenente i link dei post con polarità positiva associati ad un determinato topic e appartenenti ad una data fonte, separati da una virgola

Percorso	/links/topic/all/idfonte={IDFonte}/topic={topic}/limit={inf}-{sup}
Parametri	{IDFonte} :int {topic} :String {inf} :int, cardinalità della riga dalla quale iniziare a considerare il risultato {sup} :int cardinalità della riga alla quale terminare di considerare il risultato NOTA: {int} e {sup} definiscono la parte terminale della query: ovvero "...LIMIT {inf} , {sup}"
Metodi	GET

Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<String>,<String>,<String> oppure '' stringa vuota

27. Elenco dei link ai post positivi relativi ad una fonte per un dato topic

Restituisce una stringa contenente i link dei post con polarità positiva associati ad un determinato topic e appartenenti ad una data fonte, separati da una virgola

Percorso	/links/topic/positive/idfonte={IDFonte}/limit={inf}-{sup}
Parametri	{IDFonte} :int {topic} :String {inf} :int, cardinalità della riga dalla quale iniziare a considerare il risultato {sup} :int cardinalità della riga alla quale terminare di considerare il risultato NOTA: {int} e {sup} definiscono la parte terminale della query: ovvero "...LIMIT {inf} , {sup}"
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<String>,<String>,<String> oppure '' stringa vuota

28. Elenco dei link ai post negativi relativi ad una fonte per un dato topic

Restituisce una stringa contenente i link dei post con polarità negativa associati ad un determinato topic e appartenenti ad una data fonte, separati da una virgola

Percorso	/links/topic/negative/idfonte={IDFonte}/limit={inf}-{sup}
Parametri	{IDFonte} :int {topic} :String {inf} :int, cardinalità della riga dalla quale iniziare a considerare il risultato {sup} :int cardinalità della riga alla quale terminare di considerare il

	risultato NOTA: {int} e {sup} definiscono la parte terminale della query: ovvero "...LIMIT {inf} , {sup}
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<String>,<String>,<String> oppure ' ' stringa vuota

29.Elenco delle pagine delle fonti relative ad dato topic

Restituisce una stringa contenente i nomi delle pagine di una data fonte in cui vi sono dei post relativi ad un dato topic e tali nomi sono separati da una virgola.

Percorso	/Sorgenti/search/bobbles/nomefonte={fonte}/topic={topic}
Parametri	{nomefonte} :String {topic} :String
Metodi	GET
Media_Type	TEXT_PLAIN
GET	
Esempio di risultato:	<String>,<String>,<String> oppure ' ' stringa vuota

30.Elenco degli id delle fonti accessibili da un dato utente

Restituisce una stringa contenente gli id di tutte le fonti accessibili da un utente specifico, ogni id è separato dal successivo mediante una virgola

Percorso	/Accesso/search/utente={IDUtente}
Parametri	{IDUtente} :int
Metodi	GET
Media_Type	TEXT_PLAIN

GET	
Esempio di risultato:	<pre>"<int>,<int>,<int>"</pre> <p>oppure</p> <pre>' ' stringa vuota</pre>

31. Aggiungi permesso di lettura di una data fonte per utente

Da ad un utente identificato mediante id il permesso di visualizzare le statistiche e tutti i dati correlati ad una specifica fonte identificata anch'essa mediante id.

Percorso	/Accesso/mod/add/utente={IDUtente}/fonte={IDFonte}
Parametri	<pre>{IDUtente} :int</pre> <pre>{IDFonte} :int</pre>
Metodi	PUT
Media_Type	TEXT_PLAIN
PUT	
Esempio di risultato:	<pre>"true"</pre> <p>oppure</p> <pre>"false"</pre>

32. Elimina permesso di lettura di una data fonte per utente

Revoca ad un utente identificato mediante id il permesso di visualizzare le statistiche e tutti i dati correlati ad una specifica fonte identificata anch'essa mediante id.

Percorso	/Accesso/mod/add/utente={IDUtente}/fonte={IDFonte}
Parametri	<pre>{IDUtente} :int</pre> <pre>{IDFonte} :int</pre>
Metodi	PUT
Media_Type	TEXT_PLAIN
PUT	
Esempio di risultato:	<pre>"true"</pre>

	oppure "false"
--	-----------------------

33.Disattiva utente

Disattiva l'account di un dato utente, individuato mediante email, ma non ne rimuove i dati dal database.

Percorso	/Utenti/disattiva/email={Email}
Parametri	{Email} :String
Metodi	PUT
Media_Type	TEXT_PLAIN
PUT	
Esempio di risultato:	"true" oppure "false"

34.Modifica password di un utente

Modifica la password di un utente identificato mediante email

Percorso	/Utenti/modificaPwd/email={Email}/pwd={Pwd}
Parametri	{Email} :String {Pwd} :String
Metodi	PUT
Media_Type	TEXT_PLAIN
PUT	
Esempio di risultato:	"true" oppure "false"

35. Aggiunta di un utente

Aggiunge un utente nel database mediante una stringa in formato JSON contenente tutte le informazioni necessarie

Percorso	/Utenti/add/utenteJSON={payload}
Parametri	{payload} :String
Esempio di payload	<pre>{"ID": "0", {"Nome": "Mario", {"Cognome": "Rossi", {"Email": "MarioRossi@esempio.it", {"Password": "md5pass", {"Data_nascita": "1960-04-10", {"Citta": "frosinone", {"Creato_da": "1", {"Data_ins": "2013-04-18", {"Admin": "false"}}</pre>
Metodi	POST
Media_Type	APPLICATION_JSON
POST	
Esempio di risultato:	<pre>"" oppure stringa JSON d'errore {"error": {"text": stringa d'errore}}</pre>