



Manuale Utente API

Todolist

Versione 1.0

ZEXTRAS

LISTA VERSIONI				
Versione	Data	Azione	Autore	Commento
1.0	13/09/2023	Redatto	VB	

REVISIONI		
Cap./Sez.	Motivo della revisione	Versione

Indice

1 Premessa	3
2 Note Generali	4
3 Funzioni	4
3.1 <i>Lettura singolo Todo</i>	4
3.2 <i>Inserimento singolo Todo</i>	6
3.3 <i>Lettura dei task legati a una voce todo</i>	7
3.4 <i>Inserimento singolo task</i>	8
3.4.1 Configurazione DB	10

1 Premessa

Il presente Manuale descrive le modalità di accesso ed utilizzo delle funzioni dell'api Todolist, facente parte del prodotto Zextras Suite.

L'applicazione, tramite chiamate a servizi dedicati, permette di gestire una todolist.

2 Note Generali

Per tutti gli URL descritti in questo documento l'indirizzo base sarà "<https://SRV-APP01:<porta>/openapi-jaxrs-server/rs>".

Se per recuperare quindi i dati di un singolo todo il path del modulo è "/todolist" e il path del metodo è "/todos", l'URL completo sarà "URL base + URL modulo + URL metodo", quindi "https://SRV-APP01:8080/openapi-jaxrs-server/rs/todolist/todos".

3 Funzioni

3.1 Lettura singolo Todo

Metodo: GET

Path: /todos

Parametri di input:

Nome	Descrizione	Tipo	Esempio	Obbligatorio
id	Identificativo univoco del singolo todo	Numerico	0	SI

Valori di output:

Nome	Descrizione	Tipo
id	Identificativo univoco del singolo todo	Numerico
name	Descrizione dell'attività	Stringa
creation	Timestamp di creazione	Stringa
numberOfTasks	Numero di sottoattività presenti	Numerico

Esempio di chiamata:

“<http://localhost:8080/openapi-jaxrs-server/todolist/todos/0>”

The screenshot shows a REST client interface with the following details:

- Request:** Method: GET, URL: `http://localhost:8080/openapi-jaxrs-server/todolist/todos/0`.
- Response:** Status: 200 OK, Time: 233 ms, Size: 223 B. The response body is displayed in JSON format:

```
{  "id": 0,  "name": "home",  "creation": "1665257912000",  "numberOfTasks": 4}
```

3.2 Inserimento singolo Todo

Metodo: PUT
Path: /todos

Parametri di input:

Nome	Descrizione	Tipo	Esempio	Obbligatorio
name	Nome da assegnare all'attività da inserire	Stringa	"palestra"	SI

Valori di output:

Messaggio di avvenuto inserimento o eventuale errore applicativo.

```
Todo palestra successfully added!
```

Esempio di chiamata:

"http://localhost:8080/openapi-jaxrs-server/todolist/todos"

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/openapi-jaxrs-server/todolist/todos?`
- Method:** PUT
- Body (JSON):**

```
{  \"name\": \"palestra\"}
```
- Response:** 200 OK, 5.09 s, 189 B. Message: `Todo palestra successfully added!`

3.3 Lettura dei task legati a una voce todo

Metodo: GET

Path: /todos

Parametri di input:

Nome	Descrizione	Tipo	Esempio	Obbligatorio
id	Identificativo univoco del singolo todo	Numerico	0	SI

Valori di output (lista di oggetti descritti dalla seguente tabella):

Nome	Descrizione	Tipo
taskId	Identificativo univoco del singolo task	Numerico
todoId	Identificativo univoco del singolo todo	Numerico
description	Descrizione della sottoattività	Stringa
creation	Timestamp di creazione	Stringa
isDone	Identifica se l'attività è stata completata o meno	Booleano

Esempio di chiamata:

"http://localhost:8080/openapi-jaxrs-server/todolist/tasks/0"

HTTP <http://localhost:8080/openapi-jaxrs-server/todolist/tasks/0> Save

GET <http://localhost:8080/openapi-jaxrs-server/todolist/tasks/0> Send

Params Auth Headers (6) Body Pre-req. Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
--	-----	-------	-----------

Body 200 OK 29 ms 536 B Save Response

Pretty Raw Preview Visualize JSON 🔍

```
1  [  
2  
3    "taskId": 0,  
4    "todoId": 0,  
5    "description": "cleaning",  
6    "creation": "1665257919000",  
7    "isDone": true  
8  ],  
9  
10   "taskId": 3,  
11   "todoId": 0,  
12   "description": "moving stuff",
```

3.4 Inserimento singolo task

Metodo: PUT

Path: /tasks

Parametri di input:

Nome	Descrizione	Tipo	Esempio	Obbligatorio
todoId	Nome da assegnare all'attività da inserire	Numerico	9	SI
description	Descrizione del singolo task	Stringa	"chest press"	SI



Valori di output:

Messaggio di avvenuto inserimento o eventuale errore applicativo.

```
Task chest press successfully added!
```

Esempio di chiamata:

"http://localhost:8080/openapi-jaxrs-server/todolist/tasks"

 **http://localhost:8080/openapi-jaxrs-server/todolist/tasks**  Save

PUT

http://localhost:8080/openapi-jaxrs-server/todolist/tasks


Send

Params Auth Headers (8) **Body** Pre-req. Tests Settings Cookies Beautify

raw

JSON

```
1 { "todoId" : "9",
2   "description" : "chest press" }
```

Body  200 OK 36 ms 192 B [Save Response](#)


Pretty



Raw

Preview

Visualize

JSON



```
1 Task chest press successfully added!
```


3.4.1 Configurazione DB

E' possibile impostare i parametri di connessione al Database nel file di configurazione "config.properties" contenuto all'interno del pacchetto "org.openapitools.resources".

Segue un esempio di configurazione:

```
db.url=jdbc:postgresql://localhost:5432/zextrasdb
db.user=postgres
db.password=password
```

E' possibile infine inizializzare il Database eseguendo lo script denominato "database_init.sql", il quale provvederà alla creazione delle tabelle necessarie al funzionamento del prodotto.

