



# Dialogue En Web

APPLICAZIONE BASATA SU WEB SERVICE

Federico D'Ignazio | Applicazioni Service Oriented | 10/09/17

## Commissione

L'applicazione è stata commissionata da IRAS COTIS per la gestione e la visualizzazione di alcuni segnaposto che fanno parte del progetto Dialogue en Route.

Lo scopo dell'applicazione era di potersi interfacciare con un database, per poter aggiungere o rimuovere dati, e di poter visualizzare su mappa i risultati.

Secondo le richieste, non è necessario poter far modifiche alle singole risorse, ma solo inserimenti e rimozioni.

## Servizio web

Essendo un servizio web concentrato su due risorse, è stato fatto secondo la convenzione REST. Si è fatto riferimento al RFC7231 per attenersi ai requisiti dovuti. In particolare, è stato usato [questo capitolo](#) per l'implementazione delle varie chiamate.

Delle operazioni CRUD, seguendo le richieste notiamo che le operazioni di Update non sono richieste.

Per questo, ci sono tre tipi di chiamate che sono stati implementati:

### GET:

Viene utilizzato per richiedere risorse specifiche, cercate in base al nome, all'Id, o anche senza un criterio, offrendole tutte quante.

Il risultato è sempre il JSON dell'oggetto, accompagnato da un codice di risposta 404 nel caso in cui la risorsa non venga trovata, piuttosto che da un esito 200 in caso contrario.

### POST:

Viene utilizzato per creare nuove risorse, comunicando tutti i dati necessari. Siccome il servizio è molto semplice, e non è richiesto il controllo di eventuali risorse doppie, la chiamata ai metodi POST (se ha buon esito) torna un esito 201 con la locazione della risorsa.

L'esito 400 dovuto ad una richiesta mal formata, viene automaticamente fornito dai Framework utilizzati.

Il RFC sottolinea che sia da preferire POST piuttosto che PUT per la creazione di singole risorse, in quei casi dove la creazione viene eseguita su un URI generico e non specifico. Esempio:

Voglio creare un oggetto?	Meglio usare POST su \Object\Create
Voglio creare un oggetto con id 5?	Meglio usare PUT su \Object\5

### DELETE:

Viene utilizzato per rimuovere risorse, in base all'id della risorsa stessa. Siccome la cancellazione sul server è sincrona, non verrà mai risposto con un esito 202 (Accept). Siccome il RFC lascia libertà tra l'esito 200 (Ok) e 204 (No Content), si è preferito usare il secondo in quanto permette di non aggiungere informazioni nella risposta. Provare a cancellare una risorsa inesistente, risulterà in un esito 404.

## PICCOLO CENNO – RESTFUL WAY TO SEND COMMANDS:

REST è orientato alle risorse. Questo può portare un po' di confusione nel momento in cui si vuole eseguire un comando astratto come "Ripristinami il Database".

Ho trovato interessante [questo Q&A](#) su stack-overflow. Per praticità, la chiamata è stata fatta inizialmente il metodo POST, ma è stata poi rimossa per rispettare lo stile architetturale.

### NOTA:

Non ho specificato tutti gli esiti possibili, ce ne sono molti altri che vengono aggiunti automaticamente (come il 405 – method not allowed) senza doverlo specificare a mano.

URL	METODO	PARAMETRI	CHIAMATA
/stazione	GET	String:nomeStazione	getStazioneByName()
/stazione /{stazioneId}	GET	Int:stazioneId	getStazioneById()
/stazione /{stazioneId}	DELETE	Int:stazioneId	eraseStazioneById()
/stazione	POST	Tutti i dati per definire una Stazione	addStazione()
/stazione/all	GET	Nessuno	getStazioni()
/stazione/info	GET	String:nomeStazione	getStazioneInfo()
/percorso	GET	String:name	getPercorsoByName()
/percorso /{percorsoId}	DELETE	Int:stazioneId	erasePercorsoById()
/percorso	POST	Tutti i dati per definire un Percorso	addPercorso()
/percorso/all	GET	Nessuno	getPercorsi()

### SPIEGAZIONE DELLE CHIAMATE:

#### getStazioneByName()

Prendendo in input il nome della stazione, restituisce il JSON della stazione corrispondente. In caso di più stazioni con lo stesso nome, casistica possibile, viene restituita la prima. Se non ci fosse una stazione corrispondente, viene restituito un JSON vuoto.

#### getStazioneById()

Prendendo in input l'id della stazione, restituisce il JSON della stazione corrispondente. Se non ci fosse una stazione corrispondente, viene restituito un JSON vuoto.

`eraseStazioneById()`

Prendendo in input l'id della stazione, rimuove dal database la stazione corrispondente, restituendo un messaggio di successo/fallimento. Se non ci fosse una stazione corrispondente, viene restituito un messaggio di fallimento.

`addStazione()`

Aggiunge una stazione al database, restituisce un oggetto `ResponseEntity` per riportare le informazioni richieste e reindirizzare l'utente alla pagina voluta.

`getStazioni()`

Restituisce tutte le stazioni in formato JSON

`getStazioneInfo()`

Prendendo in input il nome della stazione, restituisce il JSON della stazione corrispondente, con informazioni aggiuntive come l'orario. In caso di più stazioni con lo stesso nome, casistica possibile, viene restituita la prima. Se non ci fosse una stazione corrispondente, viene restituito un JSON vuoto.

`getPercorsoByName()`

Prendendo in input il nome del percorso, restituisce il JSON del percorso corrispondente. In caso di più percorsi con lo stesso nome, casistica possibile, viene restituito il primo. Se non ci fosse un percorso corrispondente, viene restituito un JSON vuoto.

`erasePercorsoById()`

Prendendo in input l'id del percorso, rimuove dal database il percorso corrispondente, restituendo un messaggio di successo/fallimento. Se non ci fosse un percorso corrispondente, viene restituito un messaggio di fallimento.

`addPercorso()`

Aggiunge un percorso al database, restituisce un oggetto `ResponseEntity` per riportare le informazioni richieste e reindirizzare l'utente alla pagina voluta.

`getPercorsi()`

Restituisce tutti i percorsi in formato JSON

STRUTTURA DATI:

CENNI SULLA SCELTA DI HSQLDB:

L'applicazione contiene una classe dedicata interamente all'interazione con il database: questo significa che è possibile cambiare totalmente la sua struttura senza modificare il resto dell'applicazione.

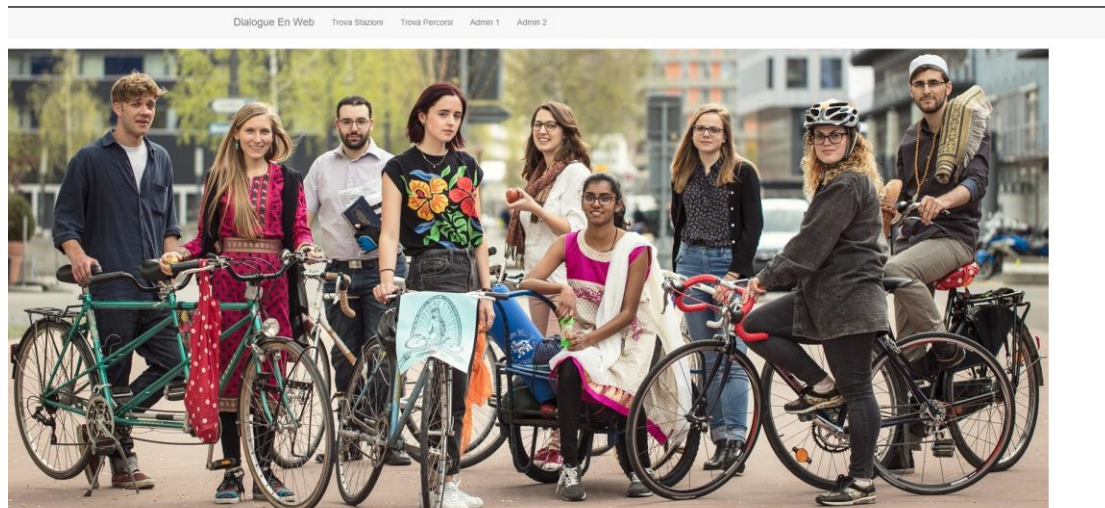
Il più grande vantaggio di HSQLDB è che può essere usato sia come server che come istanza interna ad un'applicazione:

esso permette, con pochi cambiamenti, di fare grandi modifiche sul database. Per adesso, che l'applicazione è molto piccola, è possibile salvare tutti i dati su un file apposito, ed in futuro si può adattare questa scelta in base alle eventuali necessità.

## Client

Il client si compone di 5 leggerissime pagine html basate sul framework di bootstrap, con 5 rispettivi file di javascript, i quali, usando chiamate AJAX (da jQuery), garantiscono un'interfaccia leggera, asincrona, e dal caricamento molto veloce.

LANDING PAGE:



Appena entrato, l'utente si ritrova nella landing page. Priva di testo, ma con il menu di navigazione ben in vista.

Il menù di navigazione ha quattro voci: due sono dedicate alla ricerca delle varie "stazioni" e dei "percorsi", mentre le altre due sono dedicate all'amministrazione delle risorse.

RICERCA STAZIONI:

Se l'utente desidera cercare delle stazioni, clicca sulla voce "Trova stazioni".

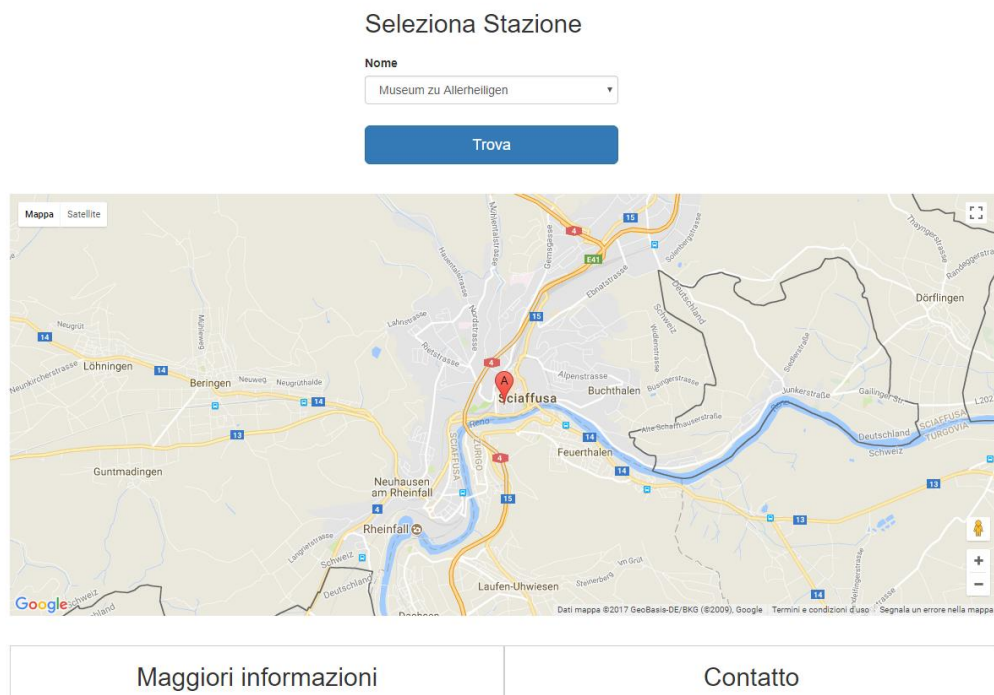
# Seleziona Stazione

Nome

Museum zu Allerheiligen

Trova

Attraverso questa droplist, l'utente può selezionare una delle stazioni disponibili nel database. Premendo “trova”, l'utente vede una schermata come questa:



Egli può quindi cliccare su “Maggiori informazioni” ed ottenere alcuni dettagli sul posto:



NOVA PERCORSI Admin 1 Admin 2

## Informazioni

Nome della stazione	Museum zu Allerheiligen
Descrizione	Museo delle religioni
Indirizzo	Klosterstrasse 16
Località	8200 Schaffhausen
Latitudine	47.6949329
Longitudine	8.6332932

Chiudi

Mentre se clicca su “Contatto”, appare un tipo diverso di informazioni:

Nome della stazione	Museum zu Allerheiligen
Telefono	052 633 07 77
Email	admin.allerheiligen@stsh.ch
Orario	11:00–17:00

Chiudi

Il tutto avviene sempre dentro dei piccoli Modal, finestre offerte dal Framework di bootstrap, che scompaiono se si clicca fuori da esse, o se l'utente finisce di interagirvi.

RICERCA PERCORSI:

Se invece l'utente desidera cercare delle stazioni, clicca sulla voce "Trova percorsi".

## Selezione Percorso

Nome

Test

Trova

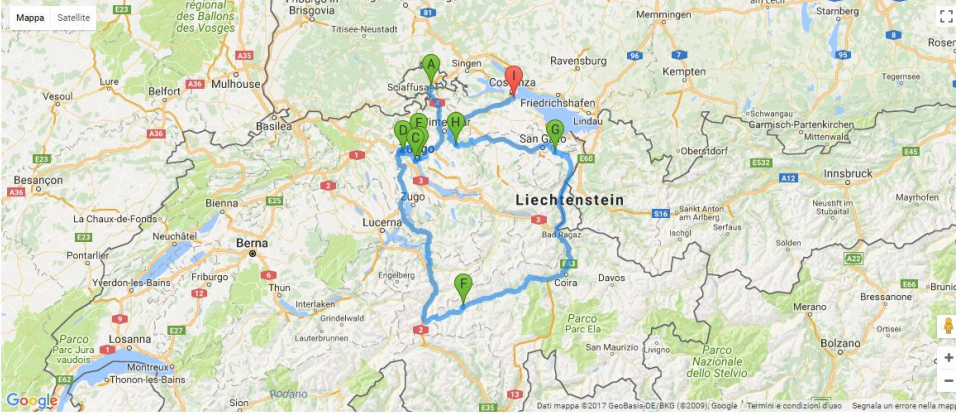
Come si può notare, l'interfaccia e la procedura rimangono invariate rispetto all'esempio visto prima: cambiano solo i dati utilizzati e la visualizzazione sulla mappa, che invece che essere composta da un singolo marker, ne avrà multipli:

Selezione Percorso

Nome

Test

Trova



Maggiori informazioni

Vedi stazioni collegate

Cliccando su "Maggiori informazioni" ...

## Informazioni

Nome del percorso	Test
Descrizione	Un percorso di test
Numero di stazioni	9
Identificativi delle stazioni	0,1,2,3,4,5,6,7,8

Chiudi

(Nota: Bootstrap non permette l'utilizzo di più Modal contemporaneamente, questo ha influito sullo sviluppo della pagina)

Infine, cliccando su “stazioni collegate”:




















Id	Nome stazione	Descrizione	Indirizzo	Telefono	Email
0	Museum zu Allerheiligen		Klosterstrasse 16 8200 Schaffhausen(47.6949329,8.6332932)	052 633 07 77	admin.allerheiligen@stsh.ch
1	Krishna-Gemeinschaft		Bergstrasse 54 8032 Zürich(47.3725963,8.5608217)	044 262 33 88	kgs@krishna.ch
2	Israelitische Cultusgemeinde Zürich		Lavaterstrasse 33 8002 Zürich(47.3631617,8.5302321)	044 283 22 22	
3	Dzemat Schlieren		Grabenstrasse 7 8952 Schlieren(47.3987441,8.4480122)	044 730 80 60	info@ibc-zh.ch
4	Shiva Tempel		Industriestrasse 34 8152 Glattbrugg(47.4308739,8.5539574)	044 371 02 42	
5	Benediktinerkloster Disentis		Via Claustra 1 7180 Disentis/Mustér(46.7064633,8.8528323)	081 929 69 00	abtei@kloster-disentis.ch
6	Kinderdorf Pestalozzi		Kinderdorfstrasse 20 9043 Trogen(47.4033,9.4578613)	071 343 73 73	info@pestalozzi.ch
7	Tibet Institute Rikon		Wildbergstrasse 10 8486 Zell(47.4365723,8.8004662)	052 383 17 29	info@tibet-institut.ch
8	Albanisch-Islamische Gemeinschaft Hëna e re		Romanshomerstrasse 16 8280 Kreuzlingen(47.6422454,9.1842958)	071 672 38 98	el.hikmeh@gmail.com

Chiudi

Passando il mouse sopra l'icona verde, appare in un piccolo pop-up la descrizione della stazione, se presente.

## AMMINISTRAZIONE STAZIONI:

L'interfaccia che si presenta all'utente per gestire le stazioni è la seguente (voce "Admin 1"):

Id	Nome stazione	Descrizione	Indirizzo	Località	Telefono	Email	Latitudine	Longitudine	
0	Museum zu Allerheiligen		Klosterstrasse 16	8200 Schaffhausen	052 633 07 77	admin.allerheiligen@stsh.ch	47.6949329	8.6332932	
1	Krishna-Gemeinschaft		Bergstrasse 54	8032 Zürich	044 262 33 88	kgs@krishna.ch	47.3725963	8.5608217	
2	Israelitische Cultusgemeinde Zürich		Lavaterstrasse 33	8002 Zürich	044 283 22 22		47.3631617	8.5302321	
3	Dzemat Schlieren		Grabenstrasse 7	8952 Schlieren	044 730 80 60	info@ibc-zh.ch	47.3987441	8.4480122	
4	Shiva Tempel		Industriestrasse 34	8152 Glattpburg	044 371 02 42		47.4308739	8.5539574	
5	Benediktinerkloster Disentis		Via Claustra 1	7180 Disentis/Mustér	081 929 69 00	abtei@kloster-disentis.ch	46.7064633	8.8528323	
6	Kinderdorf Pestalozzi		Kinderdorfstrasse 20	9043 Trogen	071 343 73 73	info@pestalozzi.ch	47.4033	9.4578613	
7	Tibet Institute Rikon		Wildbergstrasse 10	8486 Zell	052 383 17 29	info@tibet-institut.ch	47.4365723	8.8004662	
8	Albanisch-Islamische Gemeinschaft Héna e re		Romanshomerstrasse 16	8280 Kreuzlingen	071 672 38 98	el.hikmeh@gmail.com	47.6422454	9.1842958	
									

Dove, cliccando le icone blu con l'icona a forma di occhio, mi appare un modale che mi mostra maggiori informazioni:

## Informazioni

Moschea Albanese

Chiudi

Cliccando invece il cestino rosso, mando la richiesta per eliminare la risorsa.

Il pulsante blu con sopra l'icona a forma di "+" permette di aggiungere una stazione seguendo questo form/modal:

**Nome:**

Museo dei musei

Inserisci il nome della stazione

**Descrizione:**

Una bella stazione che si occupa di cultura

Inserisci una breve descrizione

**Indirizzo:**

Via dei pini 18

Inserisci la via ed il civico della stazione

**Località:**

Paperopoli 1234

Inserisci la città ed il CAP della stazione

**Telefono:**

1-(555)-555-5555

Inserisci un numero di telefono, il formato è a tua scelta

**Email:**

info@museodeimusei.pap

Inserisci l'indirizzo email di riferimento

**Latitudine:**

0

Inserisci la latitudine

**longitudine:**

0

Inserisci la longitudine

Submit

## AMMINISTRAZIONE PERCORSI:

Alla voce “Admin 2”, vi è l’amministrazione dei percorsi. L’interfaccia è simile alla precedente:

Id	Nome percorso	Descrizione	Stazioni collegate	
0	Moschee			
1	Templi Induisti			
2	Religioni a Zurigo			
3	Test			
				

E sono presenti le stesse funzioni: le informazioni aggiuntive premendo l’occhio

## Informazioni

Un percorso sui templi induisti nella svizzera tedesca

Chiudi

E l’aggiunta di percorsi:

**Nome:**

Inserisci il nome della stazione

**Descrizione:**

Inserisci una breve descrizione

**Stazioni:**

Inserisci le stazioni collegate

Submit

Chiudi

Infine, premendo il secondo occhio, quello nella colonna “Stazioni collegate”, si può vedere per esteso, nel dettaglio, la lista delle stazioni presenti nel percorso selezionato:

Id	Nome stazione	Descrizione	Indirizzo	Telefono	Email
1	Krishna-Gemeinschaft		Bergstrasse 54 8032 Zürich(47.3725963,8.5608217)	044 262 33 88	kgs@krishna.ch
4	Shiva Tempel		Industriestrasse 34 8152 Glattbrugg(47.4308739,8.5539574)	044 371 02 42	

Chiudi

## Conclusione

Il risultato soddisfa i requisiti del cliente, è veloce e pratico da utilizzare.

### FRAMEWORK UTILIZZATI:

- |                           |            |
|---------------------------|------------|
| • Spring (RestController) | – BACKEND  |
| • JAX RS                  | – BACKEND  |
| • Bootstrap               | – FRONTEND |

### LIBRERIE UTILIZZATE:

- |                                  |            |
|----------------------------------|------------|
| • Jsoup                          | – BACKEND  |
| • HSQLDB                         | – BACKEND  |
| • jQuery                         | – FRONTEND |
| • AJAX                           | – FRONTEND |
| • Google API (Direction and map) | – FRONTEND |
| • LoadingOverlay                 | – FRONTEND |