

Progetto GeoAd

Documento di progetto – Backend

[github repository](#)

Tecnologie

Backend sviluppato in .NET: interfacciamento via Web API, gestione utenti con OAuth2.

Database: Postgresql con estensioni per query geolocalizzate. Supporto per push notification per Android.

Le foto sono memorizzate e trasmesse in Base64 per facilità di gestione (parsing nativo per javascript e Android) e evitare l'accesso autorizzato dei file. La scelta è stata inoltre presa per evitare problemi di autorizzazione di gestione dei file con il sito di hosting.

La soluzione

La soluzione del backend è formata da sette progetti:

- *GeoAdServer.Domain*: definisce i contratti, i dto, gli eventi e i model
- *GeoAdServer.Domain.ContractsImplementations*: contiene gli unit test delle implementazioni dei contratti dei repository
- *GeoAdServer.DataAccess*: contiene il singolo punto d'accesso per i repository
- *GeoAdServer.Postgresql*: contiene l'implementazione dei contratti dei repository sul database postgresql
- *GeoAdServer.EventsHandling*: progetto per la gestione delle push notification (per Android), mappatura della posizione dei dispositivi (utilizzata per push di aggiornamento dati) e per storage delle preferenze (luoghi preferiti e ignorati)
- *GeoAdServer.GeneralUtilities*: classi di log e utility varie
- *GeoAdServer.WebApi*: progetto asp.net web api per dashboard e rest api

Rest API

Le api sono descritte in modo schematic nel file [ApiDoc.txt](#) contenuto nel repository.

Accanto ad ogni url sono definite le action http supportate e il model di dati per chiamate POST e PUT.

Se un api necessita di autorizzazione con token OAuth2, è indicato con l'attributo "AUTH".

Dashboard

La soluzione è attualmente pubblicata a questo indirizzo <http://geoad.somee.com/>.

La pagina iniziale mostra i form di registrazione e login per l'accesso alla Dashboard.

L'accesso è pensato per i rappresentanti di attività commerciali, i quali possono utilizzare le seguenti funzionalità:

- Creazione nuova attività commerciale
- Gestione attività commerciali (modifica, cancellazione)
- Creazione nuove offerte per attività commerciali
- Gestione offerte per attività commerciali (modifica, cancellazione)
- Inserimento, cancellazione di foto per attività commerciali
- Cambio password

Un'attività commerciale è composta dalle seguenti proprietà:

- Nome
- Descrizione
- Categoria principale
- Categoria secondaria
- Posizione geografica

Le categorie principali sono fornite dal sistema e non possono essere modificate: le categorie secondarie, invece, possono essere aggiunte se mancanti solo se non esistono altre categorie con il nome aggiunto.

Ogni attività commerciale deve avere una categoria primaria, mentre la categoria secondaria può essere omessa: consigliamo comunque di impostare una categoria secondaria se possibile in quanto può facilitare la ricerca del luogo dall'applicazione.

Un'offerta è composta dalle seguenti proprietà:

- Nome
- Descrizione
- Data di scadenza

Le offerte verranno notificate alle app in prossimità dell'attività commerciale fino alla loro data di scadenza: vengono inoltre inviate ai dispositivi che hanno registrato l'attività commerciale come "Preferita", anche se non si trovano nelle vicinanze.

Dati dei luoghi e offerte

Le web API accettano i dati di posizione associati alla chiave di push notification e ritornano tutti i dati (luoghi e offerte) richiesti (l'app richiede le informazioni all'interno di un'area geografica dati i punti nordest e sudovest). Per aggiornare i dati mantenuti in memoria dalle applicazioni, vengono inviate le seguenti push notification corrispondenti ai rispettivi eventi:

- Location Created/Updated/Deleted
- Offer Created/Updated/Deleted

NB: solo "OfferCreated" viene visualizzata come Notification sull'app, tutte le altre vengono processate senza mostrare informazioni sulla UI.

Command flow (inserimento offerta)

- L'offerta viene inserita tramite dashboard o applicazione e inviata all'API
- Il backend effettua alcuni controlli di sicurezza e se non ci sono errori inserisce l'offerta nel database e inoltra un evento "OfferCreated" al gestore di eventi
- Il gestore di eventi processa la coda di eventi e quando giunge all'"OfferCreated" generato, recupera le chiavi di invio di push notification dei dispositivi vicini all'attività commerciale e quelle delle app che hanno selezionato l'attività come "Preferita" (vengono inoltre tolte le chiavi di quelle che hanno selezionato l'attività come "Ignorata")
- Vengono inviate le push notification
- L'app riceve la notifica, memorizza i dati nel database locale e mostra una Notification per avvisare l'utente, oltre ad aggiungere l'offerta nel widget (se visibile)

Migliorie

- Per ridurre la latenza, il database dovrebbe trovarsi nello stesso server della soluzione.
- Processare in asincrono anche le preferenze utente.

Bug noti

- Se un luogo viene eliminato, non arriva la push di evento ai dispositivi che erano in prossimità del luogo
- Se creo un'attività da portale e ho già effettuato il login sull'app, non visualizzo tale attività nella sezione "Le mie attività"
- L'autorizzazione tramite OAuth2 per come è stata implementata ora, non supporta multipli login per account, quindi se un utente loggato da applicazione entra nella dashboard (o viceversa) l'altro token verrà invalidato. Attualmente, sull'app viene richiesto il login mentre la dashboard reindirizza alla pagina di Registrazione/Login.