Verbale 2013-12-18

Gruppo SteakHolders — Progetto MaaP

Informazioni sul documento

Redazione	Federico Poli
Verifica	Nicolò Tresoldi
Approvazione	Giacomo Fornari
Uso	Interno
Distribuzione	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	Gruppo SteakHolders

Descrizione

Questo documento riassumere l'incontro del 2013-12-18 tra il gruppo SteakHolders ed il proponente CoffeeStrap

Indice

1 Informazioni Generali

1.1 Informazioni incontro

- **Luogo:** Padova, via Luzzati, Aula LUF1;
- **Modalità:** colloquio tramite Skype;
- **Data:** 2013-12-18;
- **Ora:** 15:30;
- **Partecipanti del gruppo:** Gianluca Donato - Federico Poli - Luca De Franceschi;
- **Partecipanti esterni:** Alessandro Maccagnan - Mahesh Casiraghi.

1.2 Argomenti

Gli argomenti vertono principalmente sulla richiesta di chiarimenti riguardo il capitolato, per avere più informazioni utili allo scopo di comprendere meglio le richieste del proponente Coffee-Strap.

2 Domande e Risposte riassunte

- **Si era parlato di permettere la registrazione di un utente non attraverso l'admin ma attraverso l'applicazione web, giusto? È un requisito opzionale?**
  Sì, questo per il prodotto non è essenziale. Per il servizio è chiaramente importante. Sì, è opzionale.
- **L'admin, nella sua pagina di amministrazione, può modificare le credenziali di un utente? Può quindi modificare la sua password e l'email?**
  Sì, ma non è fondamentale. L'importante è che vi sia un modo per recuperare la sua password. Se può farlo anche l'utente admin, meglio, ma non è fondamentale.
- **L'admin eventualmente può declassare sé stesso?**
  No, non può declassare sé stesso. Per questo io farei una sorta di super-admin, che è scritto fisicamente in un file di configurazione o qualcosa del genere, o una soluzione che trovate intelligente. Chiaramente se io declasso me stesso dopo sono fuori dal sistema e l'unico modo per entrare è modificare il database. Quindi diventa complicato.
- **Quindi ci sarà una gerarchia a tre, con un super admin, admin e utenti? L'admin non può modificare le credenziali di un altro admin ma può farlo solo il super admin?**
  Sì, giusto.
- **È necessario memorizzare altri attributi per gli utenti, oltre all'email, alla password e al tipo di account (admin o utente)?**
  No, non è necessario.
- **Dal DSL, è possibile modificare la visualizzazione della tabella utenti di amministrazione?**
  Non è essenziale nemmeno questo. Se possibile sì, assolutamente. Immagino che voi userete lo stesso stratagemma, le stesse librerie utilizzate per visualizzare le altre tabelle delle collection. È comodo che io programmatore possa dire che il campo password vada a sinistra piuttosto che a destra. Modificare il CSS per me non è interessante. Però, in generale, può essere carino che si possa cambiare il tema. Non è assolutamente richiesto ma è carino, pensateci.
- **Se un documento ha un altro documento come attributo bisogna visualizzare un link alla show-page del documento innestato? Oppure devo mettere un attributo del documento innestato?**
  Il campo innestato potrebbe avere un insieme grosso di campi, che potrebbe non essere comodo visualizzare nella pagina. Una prima soluzione è di permettere allo sviluppatore di poter dire "questo campo innestato lo voglio cliccabile e apribile in un'altra pagina".
- **Se un documento ha un array di documenti come attributo bisogna visualizzare un link alla index-page del campo innestato?**
  Un array di documenti non è una collection, non ha una index-page. È simile ma non è la stessa cosa. Potrebbe esserci anche un'array di interi o di stringhe, che non sono di documenti. La questione di un array di documenti è simpatica da gestire, perché non c'è nel database relazionale e quindi è diverso il discorso. Si potrebbe aprire un pop-up o un'altra pagina per la visualizzazione dell'array, ma fate attenzione che un array di documenti non è una collection.

Firebase è un tool che manipola JSON in real-time e ha un tool di visualizzazione interno che è fatto molto bene. Quindi potete guardare come visualizza gli array e prendere spunto da lì. Io direi di fare una piccola anteprima dell'array e mettere il link per visualizzarlo in un'altra pagina, che sarà un array-index simile ad una collection-index.

- **Nel primo esempio del capitolato a pagina 14, perché il populate deve essere messo nell'index e non nel column?**
  Ognuno di questi documenti può avere un campo dati con un array dei contatti, contenente gli object-id dei documenti. Di questi id io voglio fare l'analogo della join, voglio vedere i campi che ci sono dentro ai documenti riferiti. La populate al posto di ogni object-id nei contatti mette il documento che ha quell'id.
- **È lo sviluppatore a dover sapere a priori che un documento ha certi campi innestati, su cui poi fare la populate?**
  Sì, assolutamente.
- **L' object-id è semplicemente un intero o è un tipo di dato particolare?**
  È un tipo di dato particolare offerto da Mongoose o dal driver di Mongo. È alfanumerico ma è configurato un certo modo. Ci si può memorizzare dentro la data di creazione, per esempio.
- **Nel momento in cui lo sviluppatore vuole registrare una collection nel DSL, per specificare quale collection è sufficiente dare il nome della collection? Oppure ci sono altri modi? Magari con una query di MongoDB?**
  Il nome della collection è univoco nel database.
- **Nel capitolato, a pagina 14, il secondo esempio definisce l'index passando come parametro una query. Questa è una query interna alla collection di cui si visualizza la index-page?**
  Io l'ho immaginato così: questa query sta all'interno di una certa collection, users, per esempio. Io di quella collection non voglio vedere tutti gli users, ma solo gli users che rispettano la query. Quindi il database interpreta la query, ed è mia responsabilità scrivere quella query in maniera corretta, e l'esecutore della query restituisce un insieme ristretto di documenti.
- **Riferendoci al punto 5 dei requisiti opzionali, su quali dati si possono proporre gli indici da creare?**
  Mongo ha una funzione explain-query che dice come sta facendo internamente la query Mongo. Fra le varie cose dice se sta usando un indice, e questo può essere un punto di partenza per proporre in maniera proattiva quali index creare per andare più veloce su quali query. Un'analisi del database è molto più complicata.
- **È l'admin a decidere quali indici creare?**
  Sì, ma è una cosa delicata. Magari gli si può dire "Guarda che se crei questo indice vai più veloce", proponendo il codice da eseguire per creare l'indice. Poi, io sviluppatore faccio il copia e incolla. Così è già più che sufficiente, perché se fate voi la creazione dell'indice andate incontro a tutta una serie di problemi. Per esempio, se il database in analisi è enorme creare quell'indice blocca tutto il database, ed è una responsabilità che è meglio non prendersi. La comodità per l'admin è avere il comando da eseguire già pronto, gli basta fare il copia incolla nella shell di Mongo.
- **Che ne dice di una pagina con la lista degli attributi, in cui l'admin può selezionare con una check-box su quali creare un indice?**

È una cosa che si può fare, attento che ci sono anche indici composti e indici temporali. È fattibile ma con un grosso avviso che dice all'admin "Guarda che potresti bloccare il database anche per un ora". Oppure ci sono dei modi per creare indici in background, ma non ne sono informato in dettaglio.

- **La creazione di un nuovo documento è possibile farla?**
  È possibile farla, ma bisogna stare attenti. Sicuramente si deve poter permettere la modifica di documenti esistenti, anche se non c'è la business logic. Anche la creazione del documento si può fare, ma è poco utile perché si troverebbe con un documento vuoto a cui aggiungere parametri. Quindi, o lo crei con tutti i parametri di default, e però ti serve lo schema di Mongoose, o ha poco senso. La creazione non è fondamentale.