



dstackgroup@gmail.com

Verbale Esterno 2019-02-23

Informazioni sul documento

Nome documento	Verbale Esterno 2019-02-23
Data approvazione	2019-02-23
Responsabile	Eleonora Signor
Redattore	Elton Stafa
Verificatore	Niccolò Vettorello
Stato	Approvato
Lista distribuzione	Gruppo <i>DStack</i> <i>Prof. Tullio Vardanega</i> <i>Prof. Riccardo Cardin</i> <i>Imola Informatica S.P.A.</i>
Uso	Esterno

Sommario

Il presente verbale riporta la comunicazione esterna effettuata dal gruppo *DStack* il giorno 2019-02-23 con *Imola Informatica S.P.A.*.

Indice

1	Informazioni sulla riunione	1
2	Ordine del giorno	1
3	Resoconto della riunione	2
3.1	Decisioni proposte	2
3.2	Tracciamento delle decisioni	3

1 Informazioni sulla riunione

- **Luogo della riunione:** Piattaforma Gmail per comunicazione esterna tra il gruppo *DStack* e *Imola Informatica S.P.A.*
- **Ora di inizio:** 11:00
- **Ora di fine:** 11.30
- **Segretario:** Elton Stafa
- **Partecipanti:** Davide Zanetti, il referente per *Imola Informatica S.P.A.*, e tutti i membri del gruppo DStack:
 - Federico Rispo;
 - Alberto Schiabel;
 - Eleonora Signor;
 - Harwinder Singh;
 - Elton Stafa;
 - Enrico Trinco;
 - Niccolò Vettorello.

2 Ordine del giorno

- Proposta e discussione dello stack tecnologico individuato per la progettazione del sistema Butterfly.

3 Resoconto della riunione

3.1 Decisioni proposte

Tramite email sono state proposte al signor Zanetti le decisioni progettuali prese nella riunione interna del 2019-02-15, riguardante lo stack tecnologico individuato al fine di realizzare il sistema Butterfly.

In particolare è stato proposto di:

- Utilizzare *Java_G*, alla versione 11, per la codifica dei *Producer_G* e dei *Consumer_G* e quindi dei microservizi che hanno un client *Apache Kafka_G* come dipendenza diretta.

Motivazione

Abbiamo deciso di utilizzare Java in quanto:

- Apache Kafka è sviluppato per la *JVM_G*;
- Ha la migliore community e documentazione per quanto concerne l'integrazione con Apache Kafka.
- Ha i risultati di benchmark migliori.

- Utilizzare *Avro_G* come sistema di serializzazione binaria tra Producer e Consumer.

Motivazione

Abbiamo deciso di utilizzare Avro in quanto:

- E' più compatto e veloce da serializzare/deserializzare rispetto a *JSON_G*;
- E' possibile definire uno schema descrittivo per i dati estendibile in futuro.
- Supporta tipi di dato primitivi (int, string, long, ecc.), complessi(enum, arrays, unions, optional), logici (dates, timestamp-millis, decimal), data record (nome e namespace).
- Supporta la documentazione embedded nello schema.

- Utilizzare *Spark_G* come *microframework_G HTTP_G* per i *Webhook_G* dei Producer.

Motivazione

Abbiamo deciso di utilizzare Spark in quanto:

- E' scritto interamente in JAVA;
- Consente di gestire delle azioni HTTP velocemente.

- Utilizzare *Node.js_G* per realizzare il server *REST_G* come intermediario tra il database e il *Gestore Personale_G*.

Motivazione

Abbiamo deciso di utilizzare Node.js in quanto:

- Permette al gruppo di imparare un nuovo linguaggio;
- Pratico ed efficace per servizi che espongono REST API.

- Utilizzare *Koa_G* come microframework HTTP per il Gestore Personale

Motivazione

Abbiamo deciso di utilizzare Koa in quanto:

- è pienamente integrato con Node.js;
- è molto leggero.
- Utilizzare *PostgreSQL*_G come *RDBMS*_G open-source per il Gestore Personale.

Motivazione

Abbiamo deciso di utilizzare PostgreSQL per via del fatto che:

- Supporta nativamente il tipo JSON;
- É corredato da una documentazione online di buona qualità;
- É caratterizzato da una community attiva e numerosa;
- É utilizzato stabilmente da aziende con milioni di utenti come Netflix, Spotify e Instagram;
- Trattandosi di un sistema relazionale, si adatta bene alla struttura del progetto.

La risposta da parte dell'azienda alla proposta è stata positiva.

3.2 Tracciamento delle decisioni

Codice Identificativo	Descrizione
VE-2019-02-23-1	Java 11 per la codifica di Producers e Consumers
VE-2019-02-23-2	AVRO per la serializzazione e deserializzazione dei messaggi
VE-2019-02-23-3	Spark come microframework HTTP per i webhooks
VE-2019-02-23-4	Node.js per lo codifica delle REST API per il Gestore Personale
VE-2019-02-23-5	Koa come microframework HTTP per il Gestore Personale
VE-2019-02-23-6	Postgres come RDBMS per il Gestore Personale

Tabella 1: Decisioni Riunione Esterna del 2019-02-23