NORRIS Framework

FLAMETECH Inc.

Verbale esterno 2015/03/31

Informazioni sul documento

Versione	1.0.0
Redazione	Merlo Gianluca
Verifica	Meneguzzo Francesco
Responsabile	Zanetti Davide
Uso	Esterno
Lista di distribuzione	**FlameTech Inc.** Prof. Vardanega Tullio

Descrizione

Verbale della riunione tra il gruppo **FlameTech Inc.** e il Proponente CoffeeStrap per il progetto Norris.

Stato	Modifica	Autore	Ruolo	Data	Versione
Approvato	Approvazione Documento	Zanetti Davide	Responsabile	2015/04/02	1.0.0
Verificato	Verifica Documento	Meneguzzo Francesco	Verificatore	2015/04/02	0.1.0
In Lavorazione	Stesura del contenuto	Merlo Gianluca	Progettista	2015/04/01	0.0.2
In Lavorazione	Inizio stesura scheletro documento	Merlo Gianluca	Progettista	2015/04/01	0.0.1

Indice

1 Informazioni sulla riunione

- **Data:** 2015/03/31
- **Ora:** 16.00
- **Luogo:** Lab P140
- **Partecipanti interni:**
  - Faggin Andrea
  - Merlo Gianluca
  - Sartor Michele
- **Partecipanti esterni:**
  - Maccagnan Alessandro

2 Domande e risposte

Di seguito sono presentate le domande poste al Proponente, con le relative risposte.

- **Per quanto riguarda il sistema di log degli errori, abbiamo pensato di scrivere gli errori su un file apposito. Può essere una buona idea o preferisce qualche altro metodo?**
  Non è una buona idea in quanto non si possono scrivere file su *Heroku*$_G$. Piuttosto è meglio trovare un altro modo per scrivere un file o fornirne uno, in ogni caso l'opzione migliore è console.log.

- **Come associamo un grafico al suo *URL*$_G$ specifico?**
  Una buona soluzione sarebbe una funzione automatica che dato un grafico gli associ e stampi un *URL*$_G$, e viceversa.

- **Per quanto riguarda la prima richiesta di pagina da parte di un *client*$_G$, è preferibile che utilizziamo una richiesta *HTTP*$_G$ o creiamo direttamente una connessione *WebSocket*$_G$?**
  Il problema principale è: posso essere sicuro di avere una connessione *WebSocket*$_G$ disponibile? Si, ma solo se uso versioni recenti dei browser.

- **La *dashboard*$_G$ per la visualizzazione dei grafici attivi sul *server*$_G$ deve essere configurabile? Se si, come?**
  Farebbe piacere. In tal caso anche salvare le preferenze potrebbe essere un problema a causa del *file system*$_G$.

- **Riguardo all'applicazione *Android*$_G$, la connessione *WebSocket*$_G$ che viene aperta deve restare tale? O solo finchè l'applicazione resta in primo piano?**
  La connessione deve rimanere attiva solo mentre l'applicazione è aperta e in primo piano.

- **Come immagina l'utilizzo delle *API*$_G$?**
  Mi immagino un utilizzo dell'sdk di Norris 'statico'. Scrivo il codice e lo eseguo. Norris aggiorna i grafici. Non mi immagino che dinamicamente io abbia la possibilità di aggiungere o rimuovere pagine. Per questo motivo non dovete implementare funzionalità che permettano la modifica dei grafici al di fuori dei metodi di update dei dati. In questo modo possiamo evitare molti problemi legati all'inconsistenza del codice scritto dallo sviluppatore e alla conseguente gestione.

- **Nell'applicazione *Android*$_G$ dobbiamo prevedere un sistema di login?**
  Sarebbe comodo che l'applicazione avesse un sistema di autenticazione in quanto CoffeeStrap non lo può gestire. Per usare dei cookie su *Android*$_G$ però è necessario saperne il nome, quindi il *server*$_G$ lo deve assegnare al momento della creazione.

3 Decisioni

A seguito di questo incontro con il Proponente vengono riportate le decisioni che sono state prese riguardo agli argomenti discussi:

- **Sistema di log degli errori:** si è deciso che i messaggi di errore verranno visualizzati tramite il metodo *console.log()* ;
- **Associazione *URL*$_G$ specifico al singolo grafico:** si è scelto di progettare un metodo all'interno dell'opportuna classe per svolgere questo compito automaticamente;
- **Gestione primo accesso dei *client*$_G$:** si è deciso di non utilizzare direttamente il protocollo *WebSocket*$_G$;
- **Configurazione *dashboard*$_G$:** si è deciso di progettare delle funzionalità di configurazione di alcuni aspetti della *dashboard*$_G$ e rendere questo aspetto un requisito opzionale per il prodotto;
- **Comunicazione con l'applicativo *Android*$_G$:** il Proponente ha deciso e specificato che la connessione *WebSocket*$_G$ deve rimanere aperta esclusivamente quando l'applicativo è in primo piano;
- ***API*$_G$ riguardanti le funzionalità di modifica:** si è deciso di non sviluppare i *casi d'uso*$_G$ 1.2 e figli, ad esclusione dei *casi d'uso*$_G$: 1.2.2.1, 1.2.4.1, 1.2.5.1, 1.2.7.1 che rappresentano i metodi di aggiornamento dei dati. Per maggiori dettagli si rimanda al documento *SpecificaTecnica_ver1.0.0*
- **Sistema di autenticazione nell'applicativo *Android*$_G$:** si è deciso di utilizzare i cookie *Android*$_G$ per la gestione di questo aspetto.