

Flutter e Dart - Le basi

Flutter - Gestione dei dati

Luigi Durso

luigi.durso@si2001.it



SI2001

15 luglio 2022



Sommario

- 1 Lezione precedente
- 2 Persistenza
- 3 Persistenza locale

- 4 Dati in remoto
- 5 Esercitazione
- 6 Fine



Un po' di codice



Analizziamo l'elaborato precedente!



Persistenza dei dati

Quanti modi abbiamo per gestire i dati:



- Memoria del dispositivo,
- Dati remoti.



Persistenza locale

Sfruttare la memoria del dispositivo:

- Installazione database locale;
- Shared preferences;
- Utilizzo di librerie per la gestione del dato (hydrated BLoC)



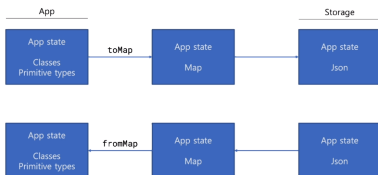
Hydrated BLoC

Utilizzare BLoC per la persistenza locale:

- Utilizzare l'estensione di BLoC: **hydrated bloc**;
- Memorizza automaticamente lo stato del bloc/cubit;



Flusso Hydrated BLoC



Come funziona hydrated BLoC?

- Gestione della memoria attraverso il formato JSON,
- Richiesta di override nello stato dei metodi toJson, fromJson.



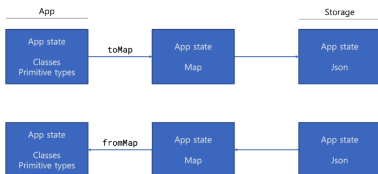
Richiesta di dati in remoto

Gestione di dati in remoto:

- Dati gestiti da altre applicazioni;
- Richiesta dei dati attraverso chiamate asincrone;



Architettura REST



Comunicare con delle REST API:

- Utilizzo del protocollo **http**;
- Gestione delle risorse attraverso degli **url**;
- Possiamo utilizzare il package **http**
▶ [Documentazione](#)
- In Dart lavoriamo con il codice asincrono attraverso i **Future**
▶ [Documentazione](#)



Migliorare la precedente esercitazione



Alcuni spunti:

- Aggiungere la persistenza al bloc/cubit di autenticazione
- Realizzare un backend ed utilizzare le chiamate remote con il pacchetto <http>

Grazie per l'attenzione!

