Orale OOP

cecca.francesco

January 2022

1 /revision statistics/statistic type

1.1 @Getmapping

1.1.1 ResponseEntity<Object>

E' un tipo pensato per rappresentare l'intera risposta HTTP e lo uso di fatti come tipo per tutte le richieste HTTP.
Un esempio lo troviamo nella parte delle statistiche:

```
ResponseEntity<Object> output = null;
  // faccio tutte le statistiche e inserisco i risultati in "data"
output = new ResponseEntity<>(data, HttpStatus.OK);
return output;
```

1.1.2 @PathVariable

Serve a recuperare un valore presente nell'url e di assegnarlo ad un determinato paramentro

```
@GetMapping("/users/{userId}")
public String getUser(@PathVariable("userId") String id){
  // ...
}
```

Il placeholder userId, consente di recuperare il valore presente nell'URL e di assegnarlo al parametro stringa id.

1.1.3 Optional<String>

Oggetto contenitore che può contenere o meno un valore diverso da null.

1.1.4 @RequestParam

Permette di associare un parametro della richiesta HTTP ad un corrispondente parametro dell'handler method.

```
@PostMapping("/users")
public String newUser(@RequestParam("name") String userName){
  // ...
}
```

In questo esempio ho associato il parametro userName del metodo al parametro name inviato nel payload della POST.
Questa associazione fa sì che la variabile userName riceva come argomento il valore del corrispondente parametro della richiesta HTTP.

1.2 Check e Set dei parametri

1.2.1 getError

@SuppressWarnings("unchecked") Si tratta di un'annotazione per sopprimere gli avvisi di compilazione relativi a operazioni generiche non controllate (non eccezioni), ad esempio i cast.
Implica essenzialmente che il programmatore non desiderava essere informato di questi di cui è già a conoscenza durante la compilazione di un particolare bit di codice.

info.keySet().retainAll(stringProperties) Info, è fatta in questo modo

```
{
   "info":
   {
      "path": "\uni",
      "mode": "id",
      "recursive": true
   }
}
```

RetainAll deve eliminare tutte le key in "info" che sono inutili.
Dato che info.keySet() restituisce una lista di chiavi, li confronto congli elementi di stringProperties.
Se in keyList c'è un elemento non presente in stringProperties, esso viene eliminato da "info".

1.3 Connessione - Ottenimento della lista di revisioni

```
Vector<Revision> revisions = null;
try {
  revisions = fileService.getRevisionList(httpsReq.rootCall(parameters));
    }
catch (NullPointerException e) {
  return new ResponseEntity<>(e.getMessage(), HttpStatus.BAD_REQUEST);
       }
```

1.3.1 rootCall

Classe HttpURLConnection Un URLConnection con supporto per funzionalità specifiche di HTTP.
Ogni istanza httpURLConnection viene utilizzata per effettuare una singola richiesta, ma la connessione di rete sottostante al server HTTP può essere condivisa in modo trasparente da altre istanze.
La chiamata ai metodi close() su InputStream o OutputStream di un oggetto HttpURLConnection dopo una richiesta può liberare risorse di rete associate a questa istanza, ma non ha alcun effetto su alcuna connessione persistente condivisa.

byte[] Passo attraverso il DataOutputStream il body per poter utilizzare l'API messa a disposizione da Dropbox

```
try( OutputStream os = openConnection.getOutputStream())
 {
   byte[] input = info.getBytes("utf-8");
   os.write(input, 0, input.length);
 }
```

Questa è una struttura dati consigliata dal docente e messa su Learn (il significato è sconosciuto a molti).

1.4 Filter

1.4.1 FilterImpl

toUnsignedLong E' un metodo che converte l'argomento Integer in un long, senza valutare il segno. In una conversione senza segno in un long, i 32 bit di ordine alto del long sono zero e i 32 bit di ordine basso sono uguali ai bit dell'argomento integer.

1.4.2 RevisionFilter

Lambda Expression Introdotto in Java 8, separa i parametri (lato sinistro) dall'implementazione (lato destro).
Una sua implementazione è la seguente:

```
private Predicate<Revision> belowThreshold() {
    return p -> (p.getSize() <= minSize);
}
```

Che equivale a scrivere:

```
Object method (Object p) {
   return (p.getSize() <= minSize) //quindi un boolean
}
```

Predicate E' un interfaccia funzionale (con un sol metodo) e può essere usato come target per una "Lambda Expression".
Di conseguenza, il tipo restituito da belowThreshold è una Lambda Expression (boolean).