

JSON LIBRARIES

mJson

Την συγκεκριμένη βιβλιοθήκη δεν την πέτυχα πουθενά στο internet και ο λόγος είναι ότι είναι αρκετά μικρή. Εκεί που έψαχνα για βιβλίο περί json libraries στην java ώστε να βοηθηθώ πιο πολύ μέσω του βιβλίου και πως δουλεύουν οι βιβλιοθήκες που υπάρχουν, βρήκα το “Java XML and JSON” 1η έκδοση του Jeff Friesen καθώς η 2η έκδοση του βιβλίου έχει και την Jackson μέσα δεν μπόρεσα να το βρω. Σχετικά με το mJson είναι μια βιβλιοθήκη αρκετά user-friendly θα έλεγα, η οποία έχει μόνο μία κλάση με όλες τις μεθόδους μέσα σε αυτή προφανώς δεν είναι πάρα πολλές, υποστηρίζει όμως parsing από JSON objects σε Java objects και το ανάποδο καθώς έχει μεθόδους για δημιουργία Json objects ή Java objects, μέθοδο για να διαχειριστείς την ιεραρχία των Json objects και υποστήριξη για JSON Schema Draft 4. Καθώς λοιπόν δούλεψα λίγο με την βιβλιοθήκη αυτή ήταν αρκετά εύκολο να περάσω json δεδομένα σε java object και το ανάποδο μέσω String και στην συνέχεια μέσω αρχείου αλλά όταν έβαλα για είσοδο αρχείο με περισσότερα από ένα Json object διάβαζε μόνο το 1ο είτε το έβαζα σε λούπα είτε χρησιμοποιώντας 2 φορές την ίδια εντολή το αποτέλεσμα ήταν πάντα η πρώτη γραμμή του αρχείου να επιστρέφεται σε Java object. Όποτε μία λύση ήταν να περάσω το αρχείο μία φορά και κάθε γραμμή να την βάζω ως είσοδο στην μέθοδο που έφτιαχνε Java object, αυτό δούλεψε απλά δουλεύει μόνο στην περίπτωση που κάθε γραμμή περιέχει και ένα Json object πράγμα που γίνεται στα περισσότερα αρχεία json με αρκετά δεδομένα αλλά προφανώς δεν είναι λύση αυτό γιατί αν χαθεί η στοίχιση κατευθείαν χάνεται και η αποτελεσματικότητα του προγράμματος. Υπήρχαν όμως μέθοδοι που επέστρεφαν τις τιμές σαν integer, float, λίστα, hashmap κτλ ή έλεγχαν αν η τιμή που επέστρεψε είναι object, array, integer κτλ. Θεωρώ όμως ότι το πρόβλημα με την στοίχιση ήταν αρκετό ώστε να μην συνεχίσω με αυτή την βιβλιοθήκη όπως και το ότι ήταν σχετικά μικρή βιβλιοθήκη και παρακάτω θα έλειπαν και άλλες λειτουργίες που θα μπορούσαν να χρειαστούν.

Gson

Όσο αναφορά την Gson είναι μία small Java-based library για parsing και δημιουργία Json objects την οποία δημιούργησε η Google για δικιά της χρήση και μετά την έδωσε και δημόσια για χρήση. Η Gson υποστηρίζει απλές μεθόδους toJson() και fromJson() για να φτιάχνεις Java objects σε Json objects και vice versa, handle collections, generic types και nested classes, επίσης υποστηρίζει serialization/deserialization και αυτά μπορούν να γίνουν και custom. Το serialization είναι ο τρόπος να δημιουργήσουμε Json objects μέσω Java objects και το deserialization ο τρόπος να δημιουργήσουμε Java objects από Json objects. Ο χρήστης μπορεί να φτιάξει custom serializer/deserializer που του επιτρέπει να χει τον πλήρη έλεγχο της όλης επεξεργασίας των Json objects. Ενώ απλό παράδειγμα είναι ότι αν έχουμε Json objects με πεδία name, surname, age τότε ο χρήστης μπορεί να φτιάξει κλάση Person με πεδία αυτά ώστε όταν περνάει το Json object σε Java object να χρησιμοποιείται η κλάση Person και να παίρνει τις τιμές των πεδίων στα ανάλογα πεδία της κλάσης έτσι ώστε να είναι διαθέσιμες όποτε τις χρειαστεί ο χρήστης. Όλο αυτό βέβαια απαιτεί να γνωρίζεις ακριβώς την μορφή των δεδομένων που θα περαστούν ως εισόδοι στο σύστημα και δεν θα αλλάζουν μέσα στον χρόνο, κάτι το οποίο δεν μας εξυπηρετεί καθόλου στις λειτουργίες που θέλουμε να έχει το σύστημα, Οπότε το custom serialization/deserialization δεν ήταν και ένα τεράστιο πλεονέκτημα που πρόσφερε η Gson και Jackson όπως θα αναφερθεί παρακάτω. Αν ψάξουμε την Gson λίγο θα δούμε 4 πακέτα:

- com.google.gson: Προσφέρει access στην Gson, την βασική κλάση για να δουλέψει κανείς με την Gson.
- com.google.gson.annotations: Προσφέρει annotation τύπους για χρήση με την Gson.
- com.google.gson.reflect: Προσφέρει μια utility κλάση για να διαχειρίζεται ο χρήστης πληροφορίες από generic type.
- com.google.gson.stream: Προσφέρει utility κλάσεις για reading/writing σε Json-encoded τιμές.

Δουλεύοντας με την Gson ήταν και εύκολο να μεταφέρω Json objects σε Java objects και το αντίθετο είτε από String είτε από αρχείο. Επιστρέφοντας τα Json objects σε HashMap<String, Object> χρησιμοποιώντας TypeToken ήταν αρκετά χρήσιμο στο να βλέπω τα πεδία και τις τιμές των Json objects είτε ήταν απλά key: value είτε ήταν εμφωλευμένα ή array. Βέβαια στην συνέχεια όταν αποφάσισα να δω πως συμπεριφέρεται η βιβλιοθήκη σε αρχεία που είχαν είσοδο πολλαπλά Json objects συνάντησα κάποιες δυσκολίες καθώς πέταγε errors. Αρχικά πέταγε exception σχετικά με το ότι δεν είναι lenient που σημαίνει ότι κάποιο Json object παραβιάζε έναν από τους structure rules και έτσι πέταγε exception, έψαξα λίγο και βρήκα ότι το lenient by default της βιβλιοθήκης είναι false πράγμα που σημαίνει ότι δεν δέχεται Json objects που

παραβιάζουν κάποιον από τους structure rules, έτσι βάζοντας το true λύθηκε το προβληματάκι. Όσο αναφορά τους structure rules είναι:

- Streams that start with the non-execute prefix, ")]}\n".
- Streams that include multiple top-level values. With strict parsing, each stream must contain exactly one top-level value.
- Top-level values of any type. With strict parsing, the top-level value must be an object or an array.
- Numbers may be NaNs or infinities.
- End of line comments starting with // or # and ending with a newline character.
- C-style comments starting with /* and ending with */. Such comments may not be nested.
- Names that are unquoted or 'single quoted'.
- Strings that are unquoted or 'single quoted'.
- Array elements separated by ; instead of ,.
- Unnecessary array separators. These are interpreted as if null was the omitted value.
- Names and values separated by = or => instead of :.
- Name/value pairs separated by ; instead of ,.

Στην συνέχεια βάζοντας το αντικείμενο της κλάσης JsonReader σε επανάληψη ώστε να διαβάζει όλα τα αντικείμενα Json objects που χει μέσα το αρχείο πέταγε error σχετικά με το ότι θα έπρεπε να διαβάζει begin_object token και όχι end_document, ψάχνοντας λοιπόν λίγο τι παίζει με το error αυτό βρήκα ότι μπορώ να χρησιμοποιήσω JsonTokens begin_object, end_object κτλ οπότε βάζοντας στην επανάληψη να διαβάζει μέχρι να βρει END_DOCUMENT token αποφεύγει και αυτό το error, μέσα στην επανάληψη έβαζα να τα επιστρέφει σε HashMap<String, Object> ώστε να παίρνω key και value από τα πεδία του Json object. Χειροκίνητα λοιπόν όποτε είχε array για παράδειγμα μπορούσα να χειριστώ τις τιμές του ή να εμφανίσω οποιά θέλω, ακόμα και να τις αλλάξω ή να αλλάξω τα ονόματα των πεδίων γενικά στο αντικείμενο η βιβλιοθήκη μου έδινε την δυνατότητα. Γενικά ήταν αρκετά user-friendly και εύχρηστη βιβλιοθήκη και ένα από τα μεγαλύτερα πλεονεκτήματα που έχει είναι ότι υπάρχει αρκετή βοήθεια στο internet σχετικά με την συγκεκριμένη βιβλιοθήκη.

Jackson