1. Utilities
   1. Reading & Writing JSON

Beberapa Perintah yang ada di libgdx untuk menggunakan fungsi JSON

1. JsonWriter: membuat style API untuk mengeluarkan dengan menggunakan JSON.
2. JsonReader: parsing JSON dan membangun sebuah DOM dengan objek dari JsonValue
3. JsonValue: Berupa sebuah JsonObject,array,string,float,long,Boolean,atau null.
4. Json: menulis atau membaca semaunya object graph dengan menggunakan JsonReader & JsonWriter

1. Contoh dari JsonWriter (class untuk ngambil data dan ngirim data)

public class Person {

private String name;

private int age;

private ArrayList numbers;

}

public class PhoneNumber {

private String name;

private String number;

}

Object graph

Person person = new Person();

person.setName("Nate");

person.setAge(31);

ArrayList numbers = new ArrayList();

numbers.add(new PhoneNumber("Home", "206-555-1234"));

numbers.add(new PhoneNumber("Work", "425-555-4321"));

person.setNumbers(numbers);

code serialize

Json json = new Json();

json.setOutputType(OutputType.json);

json.setElementType(Person.class, "numbers", PhoneNumber.class);

System.out.println(json.prettyPrint(person));

{

"numbers": [

{

"number": "206-555-1234",

"name": "Home"

},

{

"number": "425-555-4321",

"name": "Work"

}

],

"name": "Nate",

"age": 31

}

1. Reading Object

Untuk read data menggunakan json kita hanya perlu mendeserialize data tersebut. dengan menggunakan parsing data dari fromjson untuk mengambil sebuah object dari

data yang kita punya. Dan di parsing ke tojson lalu setelah json sudah menjadi sebuah maps,array dan value maka jsonreader bisa digunakan.

Class yang bisa digunakan dijson adalah

1. POJOs
2. OrderedMap
3. Array
4. String
5. Float
6. Boolean
   1. Reading & Writing XML

Sama seperti JSON XML mempunyai 2 fungsi yaitu XMLReader(code) untuk membaca file yang bertype XML ke sebuah DOM . dan XMLWriter(code) digunakan untuk menggunakan API dan mengeluarkan data XML.

* 1. Collections

1. List
2. Array(code)

Sebuah objek yang dapat disesuaikan, diurutkan atau tidak berurutan. Ini sering menggantikan ArrayList. Ini menyediakan akses langsung ke array dukungan, yang bisa berupa tipe tertentu daripada hanya Object [].

1. Primitive List
2. IntArray(code)
3. FloatArray(code)
4. BooleanArray(code)
5. CharArray(code)
6. LongArray(code)
7. ByteArray(code)
8. Specialized Array
9. SnapshotArray(code)

Ini identik dengan Array kecuali menjamin bahwa entri array yang diberikan oleh begin (), antara indeks 0 dan ukuran pada saat mulai dipanggil, tidak akan diubah sampai akhir () dipanggil. Ini bisa digunakan untuk menghindari modifikasi bersamaan. Hal ini membutuhkan iterasi untuk dilakukan dengan cara tertentu:

1. DelayedRemovalArray(code)

Ini identik dengan Array kecuali setiap kepindahan yang dilakukan setelah start () disebut antri dan hanya terjadi sekali akhir () dipanggil.

1. PooledLinkedList(code)
2. SortedIntList(code)
3. Maps
4. ObjectMap(code)

Untuk mengidentifikasi sebuat objek yang ada di sebuah map.

1. Primitive Maps
2. IntMap()
3. LongMap()
4. ObjectInMap()
5. ObjectFloatMap()
6. Specilize Maps
7. OrderedMap()
8. IdentityMap()
9. ArrayMap()
10. Sets
11. ObjectSets()  
    Persis seperti ObjectMap, kecuali hanya kunci yang tersimpan. Tidak ada nilai yang disimpan untuk setiap tombol.
12. Primitive Sets
13. IntSet()
14. BinaryHeap  
    Bisa berupa min-heap atau max-heap.
    1. Reflection  
       Untuk memanfaatkan refleksi dengan cara cross-platform, LibGDX menyediakan bungkus kecil di sekitar Java’s Reflection API. Pembungkus terdiri dari dua kelas yang berisi metode statis yang akan Anda gunakan untuk melakukan operasi refleksi:

ArrayReflection = fungsi array yg diambil dari API java.lang.reflect.array

ClassReflection = fungsi class yg diambil dari java.lang.class

1. GWT

Untuk menggunakan fungsi GWT ini tidak sama dengan cara menggunakan reflection pada umumnya. Jika ingin menggunakan GWT dan mengimplementasikan refleksi maka harus mengdeclare class mana yang ingin di refleksi. Saat mengkompilasi proyek HTML, LibGDX mengambil informasi itu dan menghasilkan cache refleksi yang berisi informasi tentang dan menyediakan akses ke konstruktor, bidang, dan metode kelas yang ditentukan. LibGDX kemudian menggunakan cache refleksi ini untuk mengimplementasikan refleksi api.

* 1. Jnigen

jnigen adalah perpustakaan kecil yang bisa digunakan dengan atau tanpa libgdx yang memungkinkan kode C / C ++ untuk ditulis sejajar dengan kode sumber Java. Hal ini meningkatkan lokalitas kode yang dimiliki bersama secara konseptual (metode kelas asli Jawa dan penerapan sebenarnya) dan membuat refactoring jauh lebih mudah dibandingkan dengan alur kerja JNI biasa.