

Serialization

How to turn objects into cereal

Serialization

Must be:

- Portable
- Persistent
- Resurrectable

Serializable

- Interface
- Very simple
- Nearly automatic serialization

```
private class MySerializableClass implements Serializable {  
    private static final long serialVersionUID = {generated GUID};  
}
```

Serializable

- Read using `ObjectInputStream`
- Write using `ObjectOutputStream`
- These streams can be used to read/write to files, networks, or anything else that has a stream.
- You can customize the format by overriding `readObject(...)` and `writeObject(...)`

Serializable

- Easy, but has some drawbacks.
- Does not maintain easily between versions.
- Can break simply by renaming or reordering variables (especially on inner classes).
- Have to override `readObject(...)` and `writeObject(...)` to explicitly handle versioning. Make sure to change `serialVersionUID`.

JSON

- Java Script Object Notation
- Very popular for web and mobile platforms.
- Encodes objects as dictionaries enclosed in curly braces.
- Encodes arrays as lists of values or objects enclosed in square brackets.
- Values are stored as name-value pairs

JSON

```
[1, 2, 3, 4, 5]
```

```
[ { name: "bob", age: 42 }, { name: "jane", age: 40 } ]
```

```
{ username: "meowmers", password: "canHazCheezburgur", lastLogin: "/Date(1198908717056)/" }
```

```
{ favoriteNumbers: [1, 42, 3.14159, -2],  
  favoriteMovie: {  
    title: "My Little Pony – The quest to be more pink",  
    rating: 5,  
    releaseDate: "/Date(34567890456)/"}  
}
```

org.json

JSONObject

JSONArray

JSONTokener

```
String json = SOME JSON STRING
```

```
JSONObject object = (JSONObject) new  
    JSONTokener(json).nextValue();
```

```
JSONArray array = (JSONArray) new  
    JSONTokener(json).nextValue();
```