



Oomph

# JSON parsing that isn't shit

Tom Adams  
BFPG July 2012

# The JSON

```
[
  {
    "c2dm_messages": 0,
    "messages": 10,
    "android_messages": 0,
    "start": "2011-09-22 00:00:00",
    "bb_messages": 0
  },
  {
    "c2dm_messages": 0,
    "messages": 5,
    "android_messages": 0,
    "start": "2011-09-22 01:00:00",
    "bb_messages": 0
  }
]
```



# The Shit

```
val defaultDate = "1976-10-08 15:30:00"
val f = DateTimeFormat.forPattern("y-M-d H:mm:ss")

def parse(json: JsValue): List[(Long, Int)] = json match {
  case JsArray(as) => {
    as.flatMap(_ match {
      case JsObject(o) =>
        val messages = o.find(_ match {
          case (JsString(k), _) if k == "messages" => true
          case _ => false
        })
        val date = o.find(_ match {
          case (JsString(k), _) if k == "start" => true
          case _ => false
        })
        date.map(_ match {
          case (_, JsString(v)) => f.parseDateTime(v).getMillis
          case _ => f.parseDateTime(defaultDate).getMillis
        }).flatMap(d => {
          messages.flatMap(_ match {
            case (_, JsNumber(v)) => Some((d, v.toInt))
            case _ => Some((d, 0))
          })
        }).flatMap(t => Some(List(t))).getOrElse Nil
      case _ => Nil
    })
  }
  case _ => Nil
}
```



# The Not Shit

```
val defaultDate = "1976-10-08 15:30:00"  
val f = DateTimeFormat.forPattern("y-M-d H:mm:ss")  
  
def parse(json: PossibleJson): List[(Long, Int)] =  
  json.arrayOrEmpty.map(j => {  
    val date = f.parseDateTime(j -| "start" stringOr defaultDate).getMillis  
    val messages = (j -| "messages" numberOr 0).toInt  
    (date, messages)  
  })
```



# For Comparison

```
default_date = "1976-10-08 15:30:00"
```

```
def parse(json)
  json.map do |j|
    date = Date.strptime(j["start"] || default_date).strftime("%s").to_i
    messages = j["messages"] || 0
    [date * 1000, messages]
  end
end
```



# Resources

- Argonaut: <https://github.com/ephox/argonaut>
- Source: <https://gist.github.com/9271a9f6f9d4bae1e1eb>

