

















































































































































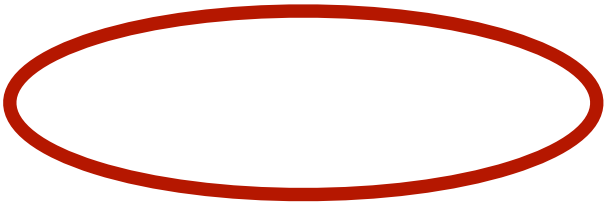




**Generics**

**typen assurance**

```
val json = """  
[  
  "name": "Parzival",  
  "password": "qwerty123"  
]  
""".trimIndent()
```



There's no such thing as  
`List<User>::class` because the type  
information is lost after compilation





public <T> readValue(String content, Class<T> valueType)

```
val clients: List<User> = mapPrereadValue(json, List<User>: class.java)
```

# Generics

## type erasure

```
public <T> T readValue(String content, Class<T> valueType)
```

```
val json = """  
    [{  
        "name": "Parzival",  
        "password": "qwerty123"  
    }]  
""".trimIndent()
```

There's no such thing as  
`List<User>::class` because the type  
information is lost after compilation

```
val clients: List<User> = mapper.readValue(json, List<User>::class.java)
```

# Generics

## type erasure

```
public <T> T readValue(String content, Class<T> valueType)
```

```
val json = """  
    [{  
        "name": "Parzival",  
        "password": "qwerty123"  
    }]  
""".trimIndent()
```

```
val clients: List<User> = mapper.readValue(json, List<User>::class.java)
```