

What's New in Groovy 2.3

Craig Atkinson

About Me

- Craig Atkinson
- Principal Consultant at Object Partners
- craigatk1
- atk.craig@gmail.com

Topics

- Traits
- Markup Builder
- @Sortable
- @Builder
- Faster JSON

Traits

- Replace 'mixins' in Groovy
 - Mixins in metaclass
 - Traits in bytecode
- Support multiple inheritance
- Can have fields, methods, private variables

Trait Example

```
trait AddressTrait {  
    String street  
    String city  
    String state  
    String zipCode  
  
    String getFullAddress() {  
        "${street}, ${city}, ${state} ${zipCode}"  
    }  
}
```

Using Traits

```
class Person implements AddressTrait, NameTrait {  
    String occupation  
}
```

```
def 'should get full address'() {  
    when:  
        String fullAddress = new Person(street: '123 Main St',  
city: 'St. Paul', state: 'MN', zipCode: '55113').  
fullAddress  
  
    then:  
        assert fullAddress == '123 Main St, St. Paul, MN 55113'  
}
```

Markup Builder

- DSL for creating HTML & XML
- Render templates based on model

Markup Builder Example

```
html {  
  head {  
    title(pageTitle)  
  }  
  body {  
    div {  
      h1(pageHeading)  
    }  
  }  
}
```


Rendered Template

```
Map model = [pageTitle: "Markup Page",  
              pageHeading: "My First Markup Page"]
```

```
<html>  
  <head>  
    <title>Markup Page</title>  
  </head><body>  
    <div>  
      <h1>My First Markup Page</h1>  
    </div>  
  </body>  
</html>
```

@Sortable AST Transformation

- Generates comparison methods based on fields
 - Implements Comparable interface and adds compareTo method
- Comparison fields and order are configurable
 - Either include specific fields or exclude others

@Sortable Example

```
import groovy.transform.Sortable

@Sortable(includes = ['lastName', 'firstName', 'age'])
class SortablePerson {
    String firstName
    String lastName
    Integer age
}
```

@Builder

- AST Transformation that generates builder-style methods based on class fields
 - Change state then return themselves to support method chaining

Simple @Builder

```
@Builder
class Address {
    String street
    String city
    String state
    String zip
}
```

```
Address address = Address.builder()
    .street('123 Main St')
    .city('Minneapolis')
    .state('MN')
    .zip('55403')
    .build()
```

External @Builder

```
@Builder(builderStrategy=ExternalStrategy,  
forClass=Address)  
class AddressBuilder { }
```

```
Address address = new AddressBuilder()  
    .street('123 Main St')  
    .city('Minneapolis')  
    .state('MN')  
    .zip('55403')  
    .build()
```

Faster JSON

- Order of magnitude faster than previous Groovy JSON serialization/de-serialization
- Comparable to Java libraries (Jackson, GSON)

Code Examples

<https://github.com/craigatk/groovy-2.3>