

Introduction to Qi4j



Rickard Öberg

Object Oriented Programming

Domain model

But where is it?

Value Objects

XML descriptors

SQL queries

But where is it?

Data Transfer Objects

Session Facades

Value Objects

XML descriptors

SQL queries

But where is it?

Data Transfer Objects

Session Facades

It's gotta be in here somewhere...

Let's abstract it!

Code generation

Domain Specific Languages XML Schemas

Let's abstract it!

Model Driven Architecture

Code generation

Domain Specific Languages XML Schemas

Containers

Let's abstract it!

Interceptors

Model Driven Architecture

Code generation

Domain Specific Languages XML Schemas

Pointcuts

Weaving

Containers

Let's abstract it!

Interceptors Byte code manipulation

Aspects

Model Driven Architecture

Code generation

Domain Specific Languages XML Schemas

Pointcuts

Weaving Containers

Let's abstract it!

Interceptors Byte code manipulation

Aspects

Model Driven Architecture

But where's the domain model?

Type less

Typeless

JavaScript

Python

Typeless

Scripting

Ruby

JavaScript

Testing

Python

Code conventions

Typeless

Specifications

Scripting

Ruby

JavaScript

Testing

Python

Code conventions

Typeless

Specifications

Ruby on Rails

Scripting

GRails

Ruby

JavaScript

Testing

Python

Code conventions

Contracts?

Typeless

Specifications

Reuse?

Ruby on Rails

Scripting

Ruby

GRails

Refactoring?

JavaScript

Testing

Python

Code conventions

Contracts?

Typeless

Specifications

Reuse?

Ruby on Rails

Scripting

Ruby

GRails

Refactoring?

I want my domain model goddamnit...

What do we need?

What do we need?

Domain Driven Design

What do we need?

Domain Driven Design OOP

What do we need?

UbiquitousLanguage

Domain Driven Design

OOP

What do we need?

Reuse

UbiquitousLanguage

Domain Driven Design

OOP

What do we need?

Reuse

Mixins

UbiquitousLanguage

Domain Driven Design

OOP

What do we need?

Reuse Mixins

UbiquitousLanguage Design Patterns

Domain Driven Design OOP

What do we need?

Reuse

Mixins

Contexts

UbiquitousLanguage

Design Patterns

Domain Driven Design

OOP

What do we need?

Server
Reuse Mixins Contexts
UbiquitousLanguage Design Patterns
Domain Driven Design OOP

What do we need?

Server Client
Reuse Mixins Contexts
UbiquitousLanguage Design Patterns
Domain Driven Design OOP

What do we need?

Server Client
Reuse Mixins Contexts Security
Persistence
UbiquitousLanguage Design Patterns
Domain Driven Design OOP

What do we need?

Server Client
Reuse Mixins Contexts Security
Multidomain Persistence
UbiquitousLanguage Design Patterns
Domain Driven Design OOP

What do we need?

Code control Reuse Mixins Server Client Security
Contexts
UbiquitousLanguage Design Patterns Multidomain Persistence
Domain Driven Design OOP

What do we need?

Strong typing Server Client
Code control Reuse Mixins Contexts Security
UbiquitousLanguage Design Patterns Multidomain Persistence
Domain Driven Design OOP

What do we need?

Refactoring

Strong typing

Server

Client

Security

Code control

Reuse

Mixins

Contexts

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

What do we need?

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

Domain Driven Design

OOP

What do we need?

Interceptors

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

What do we need?

Aspects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

Domain Driven Design

OOP

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Composition

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Composition

XML

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Security

Code control

Reuse

Mixins

Contexts

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Composition

DSL

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Security

Code control

Reuse

Mixins

Contexts

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Composition
Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Interfaces Annotations

Composition

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Less typing

Interfaces Annotations

Composition

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Scripting

Less typing

Interfaces Annotations

Composition

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Less typing

Interfaces Annotations

Composition

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Security

Code control

Reuse

Mixins

Contexts

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Less typing

Generic mixins

Interfaces Annotations

Composition

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Security

Code control

Reuse

Mixins

Contexts

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Less typing

Generic mixins

Interfaces Annotations

Dependency Injection

Composition

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Security

Code control

Reuse

Mixins

Contexts

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Less typing

Generic mixins @Inject

Interfaces Annotations

Dependency Injection

Composition

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Security

Code control

Reuse

Mixins

Contexts

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Less typing

Generic mixins

@Service

Interfaces Annotations

Dependency Injection

Composition

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Less typing

Generic mixins

@Service

Interfaces Annotations

Dependency Injection

Composition

@Entity

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Less typing

Generic mixins

@Service

Interfaces Annotations

Dependency Injection

Composition

@Entity

@ThisAs

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Less typing

Generic mixins

@Service

Interfaces Annotations

Dependency Injection

Composition

@Entity

@ThisAs

Scope

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Conventions Scripting

Infrastructure

Less typing

Generic mixins

@Service

Interfaces Annotations

Dependency Injection

Composition

@Entity

@ThisAs

Scope

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing

Server

Client

Code control

Reuse

Mixins

Contexts

Security

Multidomain

Persistence

UbiquitousLanguage

Design Patterns

OOP

Domain Driven Design

Persistence

Conventions Scripting

Infrastructure

Less typing

Generic mixins

@Service

Interfaces Annotations

Dependency Injection

Composition

@Entity @ThisAs Scope

Just Java

What do we need?

Constraints, Assertions, and SideEffects

Refactoring

IDE support Strong typing Server Client

Code control Reuse Mixins Contexts Security

Multidomain Persistence

UbiquitousLanguage Design Patterns

Domain Driven Design OOP

Persistence OSGi
Conventions Scripting Infrastructure
Less typing
Generic mixins @Service
Interfaces Annotations Dependency Injection
Composition @Entity @ThisAs Scope
Just Java
What do we need?
Constraints, Assertions, and SideEffects
Refactoring
IDE support Strong typing Server Client
Code control Reuse Mixins Contexts Security
UbiquitousLanguage Design Patterns Multidomain Persistence
Domain Driven Design OOP

Persistence OSGi
Conventions Scripting Grids Infrastructure
Less typing
Generic mixins @Service
Interfaces Annotations Dependency Injection
Composition @Entity @ThisAs Scope
Just Java
What do we need?
Constraints, Assertions, and SideEffects
Refactoring
IDE support Strong typing Server Client
Code control Reuse Mixins Contexts Security
UbiquitousLanguage Design Patterns Multidomain Persistence
Domain Driven Design OOP

Persistence OSGi
Conventions Scripting Grids Infrastructure
Less typing SBA
Generic mixins @Service
Interfaces Annotations Dependency Injection
Composition @Entity @ThisAs Scope
Just Java
What do we need?
Constraints, Assertions, and SideEffects
Refactoring
IDE support Strong typing Server Client
Code control Reuse Mixins Contexts Security
UbiquitousLanguage Design Patterns Multidomain Persistence
Domain Driven Design OOP

Persistence OSGi
Conventions Scripting
Less typing
Generic mixins
Interfaces Annotations
Composition
Just Java
Dependency Injection
@Service
@Entity @ThisAs
Scope
What do we need?
Constraints, Assertions, and SideEffects
Refactoring
IDE support Strong typing Server Client
Code control Reuse Mixins Contexts Security
UbiquitousLanguage Design Patterns Multidomain Persistence
Domain Driven Design OOP

Persistence OSGi
Grids Infrastructure RDF
Conventions Scripting
SBA Stream databases
Less typing
Generic mixins @Service
Interfaces Annotations
Composition @Entity @ThisAs Scope
Just Java
What do we need?
Constraints, Assertions, and SideEffects
Refactoring
IDE support Strong typing Server Client
Code control Reuse Mixins Contexts Security
UbiquitousLanguage Design Patterns Multidomain Persistence
Domain Driven Design OOP

I want it all

All the pieces are there

Scripting

Dependency Injection

All the pieces are there

Aspect Oriented Programming

Domain Driven Design

Scripting

Dependency Injection

What if we put
it all together?

Aspect Oriented Programming

Domain Driven Design



Composite Oriented Programming

Composite Oriented Programming

:: Best of AOP, DI, scripting and DDD

Terminology



Class

Terminology



The diagram consists of two stacked rounded rectangular boxes. The top box is yellow and contains the text 'Interceptor'. The bottom box is blue and contains the text 'Class'. Both boxes have a thin black border.

Interceptor

Class

Terminology

A horizontal yellow rounded rectangle with a thin black border.

Advice

A horizontal blue rounded rectangle with a thin black border.

Class

Terminology

Constraint

Assertion

SideEffect

Class

Terminology

Constraint

Assertion

SideEffect

Class

Terminology

Constraint

Assertion

SideEffect

Mixin

Mixin

Mixin

Terminology

Constraint

Assertion

SideEffect

Mixin

Mixin

Mixin

Terminology

Composite

Constraint

Assertion

SideEffect

Mixin

Mixin

Mixin

```
public class HelloWorld
{
    String phrase;
    String name;

    public String getPhrase()
    {
        return phrase;
    }

    public void setPhrase( String phrase )
    {
        if( phrase == null ) {
            throw new IllegalArgumentException( "Phrase may not be null " );
        }
        this.phrase = phrase;
    }

    public String getName()
    {
        return name;
    }

    public void setName( String name )
    {
        if( name == null ) {
            throw new IllegalArgumentException( "Name may not be null " );
        }
        this.name = name;
    }

    public String say()
    {
        return phrase + " " + name;
    }
}
```

```
public interface HelloWorldComposite
    extends StandardComposite
{
    @Cached String say();

    void setPhrase( @NonEmptyString @Contains( "H" )String phrase )
        throws ValidationException;

    String getPhrase();

    void setName( @NotNull @Matches( "Universe|World" )String name )
        throws ValidationException;

    String getName();
}
```

```
function say()  
{  
    return phrase + " " + name;  
}
```

```
public interface StandardComposite
    extends InvocationCacheComposite, ConstraintsComposite, GenericMixinsComposite
{
}
```

```
@Mixins( InvocationCacheMixin.class )
@Assertions( ReturnCachedValueAssertion.class )
@SideEffects( { CacheInvocationResultSideEffect.class,
InvalidateCacheOnSettersSideEffect.class } )
public interface InvocationCacheComposite
    extends Composite
{
}
```

```

@AppliesTo( Cached.class )
public class ReturnCachedValueAssertion
    implements InvocationHandler
{
    @ThisAs private InvocationCache cache;
    @Invocation private Method method;
    @AssertionFor private InvocationHandler next;

    public Object invoke( Object proxy, Method method, Object[] args ) throws Throwable
    {
        // Try cache
        String cacheName = method.getName();
        if( args != null )
        {
            cacheName += Arrays.asList( args );
        }
        Object result = cache.getCachedValue( cacheName );
        if( result != null )
        {
            if( result == Void.TYPE )
            {
                return null;
            }
            else
            {
                return result;
            }
        }

        // No cached value found - call method
        return next.invoke( proxy, method, args );
    }
}

```

phrase.not.null=Phrase may not be null

phrase.contains=Phrase "{0}" must contain the string "{1}"

phrase.min.length=Length of phrase "{0}" is too short. It must be at least {1} characters

name.not.null=Name may not be null

name.matches=Name must be either Universe or World

Where are we?

Where are we?

:: www.qi4j.org

Where are we?

:: www.qi4j.org

:: Mailing list

Where are we?

- :: www.qi4j.org
- :: Mailing list
- :: Community plan

Where are we?

- :: www.qi4j.org
- :: Mailing list
- :: Community plan
 - :: OpenSource vs community vs business

Where are we?

- :: www.qi4j.org
- :: Mailing list
- :: Community plan
 - :: OpenSource vs community vs business
- :: Get involved!

Where are we?

Where are we?

:: Code in Subversion

Where are we?

- :: Code in Subversion

- :: Apache 2.0 license

Where are we?

- :: Code in Subversion
 - :: Apache 2.0 license
 - :: Framework

Where are we?

- :: Code in Subversion
 - :: Apache 2.0 license
 - :: Framework
 - :: Samples

Where are we?

- :: Code in Subversion
 - :: Apache 2.0 license
 - :: Framework
 - :: Samples
 - :: Tutorials

Where are we?

- :: Code in Subversion
 - :: Apache 2.0 license
 - :: Framework
 - :: Samples
 - :: Tutorials
 - :: Libraries

Questions?