

GWT & CDI

Heiko Braun <hbraun@redhat.com>

Topics

- CDI & GWT
- Use Cases & Requirements
- Relevant Concepts
- Likeminded Technologies
- Demo

GWT & CDI

Google Web Toolkit

- Create Ajax applications from Java sources
- GWT SDK: Compiler, API, Tools
- Creates highly optimized, portable javascript code

Context and Dependency Injection

- CDI: JSR-299, JSR-330
- Reference implementation: Weld
 - Seam 3
- EE6 component model
- Portable extensions
 - shrink-size or extend runtime (framework)



Use Cases & Requirements

Use Cases

- Invoke CDI beans from GWT clients
 - RPC
 - Pub/Sub
 - Eventing
- Expose domain models to GWT clients
 - Serialization

Example: RPC bean

```
@Service
@ApplicationScoped
public class AccountManagerBean implements AccountManager
{

    @Remote
    public interface AccountManager
    {
        void createAccount(Account account);
        Account getAccount(String id);
        List<Account> getAllAccounts();
        void deleteAccount(String id);
    }
}
```

Example: RPC Client

```
AccountManager call = MessageBuilder.createCall(  
    new RemoteCallback<List<Account>> ()  
    {  
        public void callback(List<Account> accounts)  
            {...}  
    }, AccountManager.class  
);  
  
call.getAllAccounts();
```

Requirements

- “Seamless integration”
 - Verified concepts
 - Concise programming model
 - Reduction of boilerplate

Example: Event Bean

```
@ApplicationScoped
public class AccountService
{
    @Inject @Any
    Event<Outbound> event;

    public void watchActivity(
        @Observes @Inbound AccountActivity activity)
    {
        Fraud payload = new Fraud(System.currentTimeMillis());
        event.fire( new Outbound(payload) );
    }
}
```

Example: Event Client

```
public class FraudClient extends LayoutPanel {

    @Inject
    public Event event;

    public FraudClient() {

        Button button =
            new Button("Create activity",
                new ClickHandler() {
                    public void onClick(ClickEvent clickEvent)
                    {
                        event.fire(new AccountActivity());
                    }
                }
            );
    }

    public void processFraud(@Observes Fraud fraudEvent) {
        // react to fraud event
    }
}
```

Relevant Concepts

Beans, Qualifiers & Scopes

- Bean type: Client-visible type (class, interface)
- Qualifier: Distinguish multiple beans of the same type
 - @Named, @Asynchronous
- Scope: Determines instance lifecycle
 - @SessionScoped,
@ConversationScoped, @RequestScoped

Build & Run-time

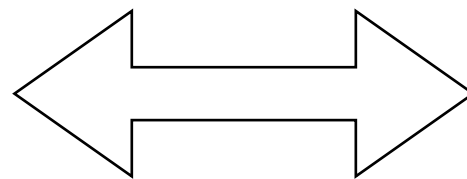
- Run-time

- Deployment
 - Same Origin Policy?
- Transactions, Security
 - Context propagation?
- Clustering, Failover
 - Stateful Ajax Client?

- Build-time

- GWT Compiler constraints
- Domain model & API?
- Hosted Mode
- CDI Bootstrap?
- Deferred Binding
- Boiler plate

Java



Javascript

Example: Conversation Scoped Bean

```
@ConversationScoped
@Service("wizard")
public class TextBufferWizard implements Serializable {

    @Inject
    private ErraiConversation conversation;

    @Command("first")
    public void first(Message message) {
        conversation.begin();
        append(message);
    }

    @Command("append")
    public void append(Message message) {
        buffer.append(message.get(String.class, "word"));
    }

    @Command("last")
    public void last(Message message) {
        append(message);
        conversation.end();

        // flush buffer
        buffer = new StringBuffer();
    }
}
```

Example: Client managed conversation (1)

```
@Inject
Conversation conversation;

public void provideWidget(ProvisioningCallback callback) {
    LayoutPanel panel = new LayoutPanel();

    panel.add(new Button("Begin conversation",
        new ClickHandler()
        {
            public void onClick(ClickEvent clickEvent)

                conversation.begin();

                MessageBuilder.createMessage()
                    .toSubject("wizard")
                    .command("first")
                    .with("text", "Hello ")
                    .done().sendNowWith(bus);

        }
    ));
};
```

Example: Client managed conversation (2)

```
panel.add(new Button("End conversation",
    new ClickHandler()
    {
        public void onClick(ClickEvent clickEvent) {
            MessageBuilder.createMessage()
                .toSubject("wizard")
                .command("last")
                .with("text", "world!")
                .done().sendNowWith(bus);

            if (!conversation.hasEnded())
                conversation.end();
        }
    })
);
```



Likeminded Technologies

Related Technologies

- JPA
 - Serialization
 - Detachment (lazy/eager loading)
 - Merging
- DataBinding
 - Widgets, Validation, Synchronization
- JMS
 - Ajax client as Topic Subscriber/Publisher



Project and Roadmap

Errai & Weld

- Errai acts as Umbrella (<http://jboss.org/errai>)
- Weld as Reference implementation
- Driven by both Errai and CDI Team
- Same runtimes Weld supports:
 - SE, Servlet, EE6, EC2:
“Start POC, grow big”

Questions?

- Project: <http://jboss.org/errai>
- Weld: <http://seamframework.org>
- Blog: <http://errai-blog.blogspot.com/>
- IRC: [irc.freenode.net#errai](irc:freenode.net#errai)