Matt Benson

Member, Apache Software Foundation
VP Apache BVal
PMC Member, Apache Commons
PMC Member, Apache Ant
Committer, Apache MyFaces
Mentor/Committer, Apache Deltaspike
Member, Apache TAC

mbenson@apache.org

Commons Sandbox [weaver]

Java bytecode post-processing

What does [weaver] provide?

- Java SPI
- Java API
- Apache Maven plugin
- Apache Ant task

Motivation/PoC

[weaver] currently provides a single SPI implementation module, 'privilizer'

Naive

```
public String foo() {
  return System.getProperty("foo");
}
```

SecurityManager-aware

```
public String foo() {
  return AccessController.doPrivileged(
    new PrivilegedAction<String>() {
      public String run() {
        return System.getProperty("foo");
      }
    );
}
```

Performant

```
public String foo() {
  if (System.getSecurityManager() != null) {
    return AccessController.doPrivileged(
      new PrivilegedAction<String>() {
        public String run() {
          return System.getProperty("foo");
        }
      );
  } else {
    return System.getProperty("foo");
  }
}
```

Privilizer: A better way (?)

```
public String foo() {
  return innerFoo();
}

@Privileged
private String innerFoo() {
  return System.getProperty("foo");
}
```

Yields:

```
private String innerFoo() {
  if (System.getSecurityManager() != null) {
    return AccessController.doPrivileged(new __privileged_foo_ACTION(this));
  } else {
    return __privileged_foo();
  }
}

private String __privileged_innerFoo() {
  return System.getProperty("foo");
}

private static class __privileged_innerFoo_ACTION implements PrivilegedAction<String> {
  private final MyClass __privileged_owner;
  private __privileged_foo_ACTION(MyClass __privileged_owner) {
  public String run() {
    return owner.__privileged_innerFoo();
  }
}
```

Drawback:

Every class must duplicate code to which privileges should be granted!

DRY

```
public class Utils {
  public static String foo() {
    if (System.getSecurityManager() != null) {
      return AccessController.doPrivileged(
        new PrivilegedAction<String>() {
          public String run() {
            return System.getProperty("foo");
          }
        );
    } else {
      return System.getProperty("foo");
    }
  }
}
```

DRY

```
public class Utils {
  public static String foo() {
    if (System.getSecurityManager() != null) {
      return AccessController.doPrivileged(
        new PrivilegedAction<String>() {
          public String run() {
            return System.getProperty("foo");
          }
        );
    } else {
      return System.getProperty("foo");
    }
  }
}
```

UNSAFE

With "blueprint" privileges

```
@Privilizing(CallTo(Utils.class))
public class MyClass {
  public String foo() {
    return Utils.foo();
  }
}

public class Utils {
  public static String foo() {
    return System.getProperty("foo");
  }
}
```

Yields:

```
public class MyClass {
  public String foo() {
    return _my_package_Utils$foo();
  }
  private String my_package_Utils$foo() {
    if (System.getSecurityManager() != null) {
      return AccessController.doPrivileged(
        new __privileged_my_package_Utils$foo_ACTION(this));
    } else {
      return __privileged_my_package_Utils$foo();
    }
  }
  private String __privileged_my_package_Utils$foo() {
    return System.getProperty("foo");
  }
  private static class __privileged_innerFoo_ACTION
    implements  PrivilegedAction<String> {
    private final MyClass __privileged_owner;
    private __privileged_foo_ACTION(MyClass __privileged_owner) {
    public String run() {
      return owner.__privileged_my_package_Utils$foo();
    }
  }
}
```

Future

- Release 1.0
- New Weaver implementations
- Explore the possibility of invoking Weavers at runtime
- Your contributions :)

Join us!

mailto:dev@commons.apache.org

http://svn.apache.org/repos/asf/commons/sandbox/weaver/trunk