MicroProfile JWT

This cheat sheet covers the basics of MicroProfile JWT specification uses.

REQUIRING MP-JWT ACCESS CONTROL

`org.eclipse.microprofile.jwt.LoginConfig`
provides the same information as the `web.xml` login-config element

```
@LoginConfig(authMethod = "MP-JWT", realmName =
"TEST-MP-JWT")
@ApplicationPath("/")
public class TCKApplication extends Application {
}
```

Integration with CDI

Injection of the `JsonWebToken` (injected as `@RequestScoped`)

```
@Inject
private JsonWebToken callerPrincipal;
```

Injection of the `Claim` (injected as `@Dependent`)

```
@Inject
@Claim("iss")
private ClaimValue<String> issuer;
```

Common Security Annotations for the Java Platform

`@RolesAllowed`, `@PermitAll`, `@DenyAll`
roles are mapped to the MP-JWT "groups" claim

```
@Path("/endpoint")
@DenyAll
@RequestScoped
public class RolesEndpoint {
    @GET
    @Path("/echo")
    @RolesAllowed("Echoer")
    public String echoInput() {
        // ...
    }

    @GET
    @Path("/echo2")
    @RolesAllowed("NoSuchUser")
    public String echoInput2()) {
        // ...
    }
}
```

CONFIGURATION OF THE ISSUER PUBLIC KEY

Using the MicroProfile Config values

`mp.jwt.verify.publickey`
the Public Key text itself supplied as a string

`mp.jwt.verify.publickey.location`
external or internal location of the Public Key to be specified, the value may be a relative path or a URL

`mp.jwt.verify.issuer`
the expected value of the iss claim

Author Martin Stefanko
Senior Software Engineer
Middleware Runtimes Sustaining Engineering Team