

KNOX-179: Simple way to introduce new Provider

This design is intended to address the following issues:

[KNOX-179@jira](#): Simple way to introduce new provider/servlet filters into the chains

The basic issues is that the development and delivery of code is currently required to add a new provider into Knox.

This results in a barrier to entry for integrating new providers with Knox.

In particular the integration of an existing servlet filter into Knox should not require code development.

The design below is intended to introduce a very simple configuration based mechanism to accomplish this.

Naturally this will only cover very simple use-cases.

More advanced requirements will still require the development of a ProviderDeploymentContributor.

Topology

```
<topology>
  <gateway>
    <provider>
      <role>authentication</role>
      <name>generic</name>
      <enabled>true</enabled>
      <param>
        <name>filterClassName</name>
        <value>org.opensource.filters.ExistingFilter</value>
      </param>
    </provider>
    ...
  </gateway>
  ...
</topology>
```

DefaultProviderDeploymentContributor

```
public class DefaultProviderDeploymentContributor {

    String getRole() {
        return "**"; // The "*" will require special handling in the framework.
    }

    String getName() {
        return "generic";
    }

    void initializeContribution( DeploymentContext context ) {
        // NoOp
    }

    void contributeProvider( DeploymentContext context, Provider provider ) {
        // NoOp
    }

    void contributeFilter(
        DeploymentContext context,
        Provider provider,
        Service service,
        ResourceDescriptor resource,
        List<FilterParamDescriptor> params ) {
        resource.addFilter()
            .name( getName() )
            .role( provider.getRole() )
            .impl( provider.getParams().get( "filterClassName" ) )
            .params( params );
    }

    void finalizeContribution( DeploymentContext context ) {
        // NoOp
    }
}
```