MicroProfile Config

MicroProfile Config API can be used to retrieve configuration information from different sources.

When we generated our Maven project we got an empty `microprofile-config.properties` file created in the `src/main/resources` package. MicroProfile Config files are created in this package as default. We will now add new properties and make our application read the properties from our config file. Open `microprofile-config.properties` file and add the following.

```
username=root
password=secret
microprofile.apis={"config", "cdi", "jax-rs", "json-p", "fault tolerance", "healt
check", "jwt auth", "metrics", "openapi", "opentracing","rest client"}
```

Now open the `BookStoreEndpoint.java` and make it look like this:

```java
@ApplicationScoped
@Path("/books")
public class BookStoreEndpoint {

    @Inject
    @ConfigProperty(name="username", defaultValue="admin")
    private String username;

    @GET
    @Path("mp-config")
    @Produces(MediaType.APPLICATION_JSON)
    public Response mpConfig() {
        Map<String, Object> configProperties = new HashMap<>();

        configProperties.put("username", username);

        return Response.ok(configProperties).build();
    }
}
```

`@Inject @ConfigProperty` is used for injecting a single configuration property. We could also use `@Inject Config` and use `getValue` to retrieve configuration properties like this:

```java
@ApplicationScoped
@Path("/books")
public class BookStoreEndpoint {

    @Inject
    @ConfigProperty(name="username", defaultValue="admin")
    private String username;

    @Inject
    Config config;

    @GET
    @Path("mp-config")
    @Produces(MediaType.APPLICATION_JSON)
    public Response mpConfig() {
        Map<String, Object> configProperties = new HashMap<>();

        configProperties.put("username", username);
        configProperties.put("password", config.getValue("password", String.class));
        configProperties.put("microprofile-apis", config.getValue("microprofile.apis",
String[].class));

        return Response.ok(configProperties).build();
    }
}
```

Start the `BookStoreClient` service if not already started and type the following in a new terminal window:

```
curl -i -H "Accept: application/json" -H "Content-Type: application/json" -X GET
http://localhost:8081/restapi/books/mp-config
```

Output

```
{"password":"secret","microprofile-apis":["{\"config\"","\"cdi\"","\"jax-
rs\"","\"json-p\"","\"fault tolerance\"","\"healt check\"","\"jwt
auth\"","\"metrics\"","\"openapi\"","\"opentracing\"","\"rest
client\"}"],"username":"root"}%
```

Summary

In this chapter, we learned how to use MicroProfile Config API.