

Quick start

Your first application

Dependencies

Just add in any Maven `pom.xml` - or gradle `build.gradle` the following dependency:

```
<dependency>
  <groupId>org.apache.meerowave</groupId>
  <artifactId>meerowave-core</artifactId>
  <version>${meerowave.version}</version>
</dependency>
```

If you intend to reuse our `cli` main you should also add:

```
<dependency>
  <groupId>commons-cli</groupId>
  <artifactId>commons-cli</artifactId>
  <version>1.4</version>
</dependency>
```

Finally, if you want to develop against `jakarta` namespace instead of `javax` one, you can use the all in one `jakarta` bundle:

```
<dependency>
  <groupId>org.apache.meerowave</groupId>
  <artifactId>meerowave-core</artifactId>
  <version>1.2.11</version> <!-- a least 1.2.11 is required -->
  <classifier>jakarta</classifier>
  <exclusions>
    <exclusion>
      <groupId>*</groupId>
      <artifactId>*</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

Runtime

Meerowave relies on JAX-RS and CDI so to start you just need to write a JAX-RS endpoint:

```
@Path("kitchen")
@ApplicationScoped
public class HelloKitchen {
    @GET
    public String getMenu() {
        return "good things";
    }
}
```

Then booting Meecrowave is as easy as launching - or reuse `org.apache.meecrowave.runner.Cli` provided main:

```
try (final Meecrowave meecrowave = new Meecrowave().bake()) {
    new Scanner(System.in).nextLine();
}
```

You should get some output containing:

```
[19:54:55.397][INFO][main][.meecrowave.cxf.CxfCdiAutoSetup] REST Application: / ->
org.apache.cxf.cdi.DefaultApplication
[19:54:55.399][INFO][main][.meecrowave.cxf.CxfCdiAutoSetup]           Service URI: /kitchen
-> org.app.HelloKitchen
[19:54:55.401][INFO][main][.meecrowave.cxf.CxfCdiAutoSetup]           GET
/kitchen/ ->           String getMenu()
```

And you can check it works doing:

```
curl http://localhost:8080/kitchen
```

You're in a hurry? Use groovy!



this feature is supported starting from version 0.3.0 only.

Create a file called `hello.groovy`:

```
@Grab('org.apache.meerowave:meerowave-core:0.3.0')

import org.apache.meerowave.Meerowave

import javax.ws.rs.GET
import javax.ws.rs.Path
import javax.enterprise.context.ApplicationScoped

@Path("hello")
@ApplicationScoped
class Hello {
    @GET
    hi() {
        "hi"
    }
}

new Meerowave().bake().await()
```

then

```
groovy hello.groovy
```

Finally you can test it:

```
curl http://localhost:8080/hello
```

And my war?

See [How to deploy a war](#) to see how to use meecrowave to deploy an existing war.

Sample

<https://github.com/apache/openwebbeans-meerowave-examples> contains ready to use examples using meecrowave.

You can also find more information on our [How To](#) page.