Servlet API Library
# Table of Contents

- Screenshots ................................................................. 2
- How to configure/use ......................................................... 5
  - Classpath ................................................................. 5
  - Bootstrapping ............................................................ 5
- API & Implementation ....................................................... 6
- Known issues ................................................................. 7
- Dependencies ............................................................... 8
This module (isis-module-servletapi) provides access to various elements of the Servlet API, namely the ServletContext, the HttpServletRequest and the HttpServletResponse.

For each of these APIs a corresponding "provider" domain service exists; for example ServletContextProvider service provides access to the ServletContext.

This module is only available for Wicket viewer, not the REST API.
Screenshots

The module's functionality can be explored by running the quickstart with example usage using the org.incode.domainapp.example.app.modules.ExampleDomLibServletApiAppManifest.

A home page is displayed when the app is run:

The demo object has all of the various "provider" domain services injected into it. It uses the ServletContextProvider and the HttpServletRequestProvider services to simply show the servlet context name and user’s locale:
To demonstrate the use of the `HttpServletResponseProvider`, the demo object provides an "addHeader" action:

When invoked, this adds a HTTP header to the response:
How to configure/use

Classpath

Update your classpath by adding this dependency in your dom project's pom.xml:

```xml
<dependency>
    <groupId>org.isisaddons.module.servletapi</groupId>
    <artifactId>isis-module-servletapi-dom</artifactId>
</dependency>
```

Check for later releases by searching Maven Central Repo.

Bootstrapping

In the AppManifest, update its getModules() method, eg:

```java
@Override
public List<Class<?>> getModules() {
    return Arrays.asList(
        ...
        org.isisaddons.module.servletapi.ServletApiModule.class,
        ...
    );
}
```
The **ServletContextProvider** defines the following API:

```java
public class ServletContextProvider {
    public ServletContext getServletContext() { ... }
}
```

The **HttpServletRequestProvider** defines the following API:

```java
public class HttpServletRequestProvider {
    public HttpServletRequest getHttpServletRequest() { ... }
}
```

The **HttpServletResponseProvider** defines the following API:

```java
public class HttpServletResponseProvider {
    public HttpServletResponse getHttpServletResponse() { ... }
}
```

And finally, the **HttpSessionProvider** defines the following API:

```java
@DomainService(nature = NatureOfService.DOMAIN)
public class HttpSessionProvider {
    public Optional<HttpSession> getHttpSession() { ... }
    public <T> Optional<T> getAttribute(String key, Class<T> clazz) { ... } ①
    public <T> void setAttribute(String key, T value) { ... }
    public void removeAttribute(String key) { ... }
}
```

① obtains an attribute (previous set) by key, cast to the specified class

These actions are all programmatic and do not appear in the UI.
Known issues

None known at this time.
Dependencies

Maven can report modules dependencies using:

```
mvn dependency:list -o -pl modules/lib/servletapi/impl -D excludeTransitive=true
```

which, excluding Apache Isis itself, returns these compile/runtime dependencies:

```
org.apache.geronimo.specs:geronimo-servlet_3.0_spec:jar:1.0
org.apache.wicket:wicket-core:jar:7.8.0
```

The direct Wicket dependency is because the module implementation depends on Apache Isis, and in particular on Apache Isis' Wicket viewer. This means that it cannot be used within domain objects invoked from the REST API (Restful Objects viewer).

For further details on 3rd-party dependencies, see:

- [Apache Wicket](#)