Purpose

Marine-ID is an integrated registration, authentication and accounting infrastructure for marine data users. The functions provided are:

1. User self-registration
2. Authentication and Single Sign On
3. Accounting

This document provide guidelines to integrate this functionalities in your web applications

Integrate Authentication with MARINE ID, principles

CAS presentation

Integrate authentication consist to implement a CAS SSO and manage the CAS API

You have to understand how this service works. For that, you can read the Wikipedia entry https://en.wikipedia.org/wiki/Central_Authentication_Service

The URL for authentication service is: https://users.marine-id.org/login

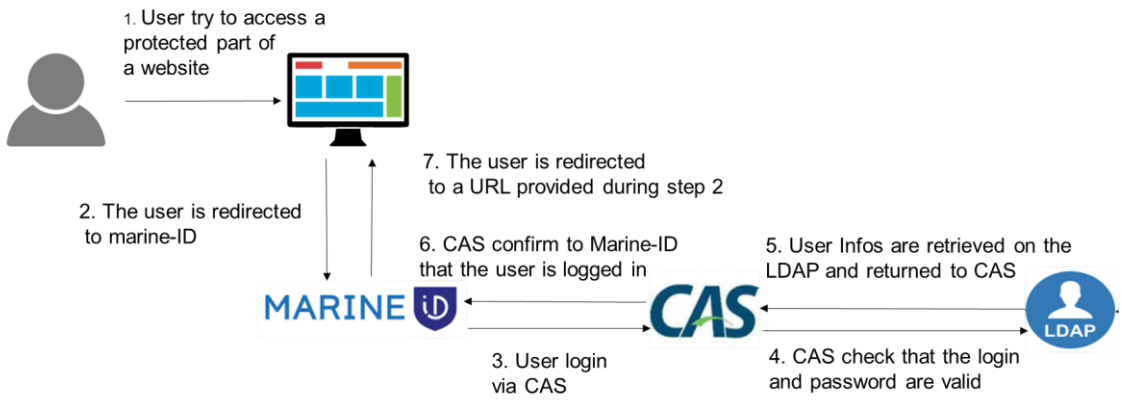
The CAS API allows service providers to rely on Marine-ID, to authenticate the users connecting the service and get some personal attributes (first name, email address, affiliation…).

For example:

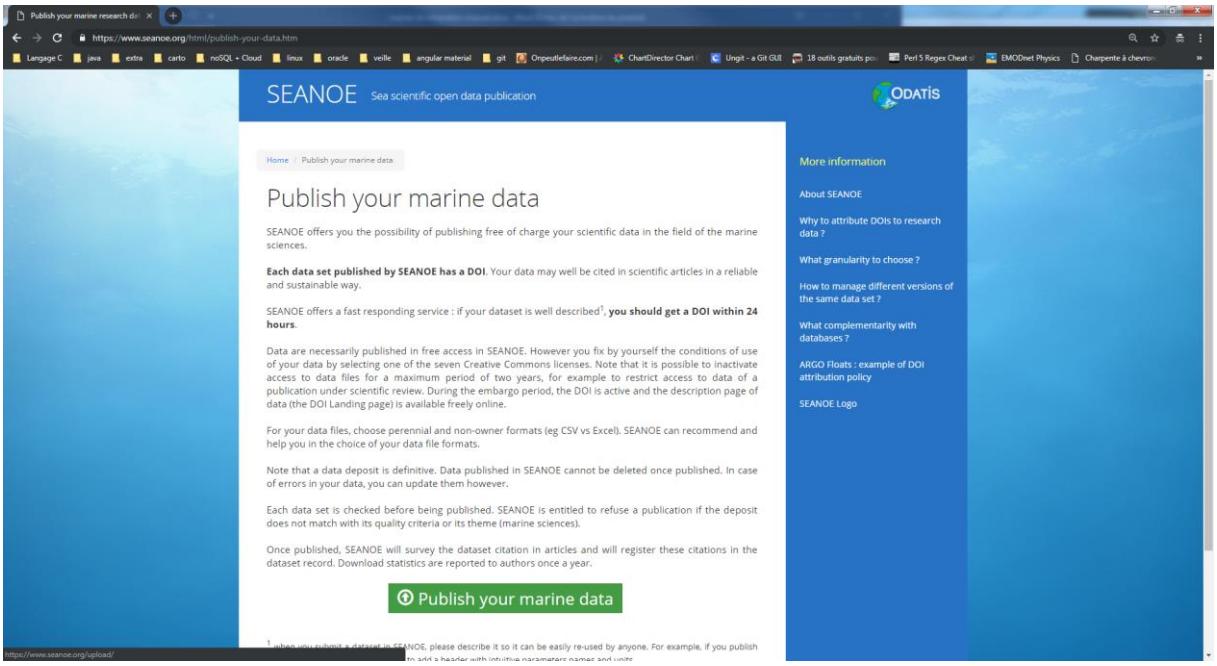
- with phpCAS client : https://wiki.jasig.org/display/casc/phpcas+examples
- with java : https://wiki.jasig.org/display/CASC/Saml11TicketValidationFilter+Example

Signing on workflow

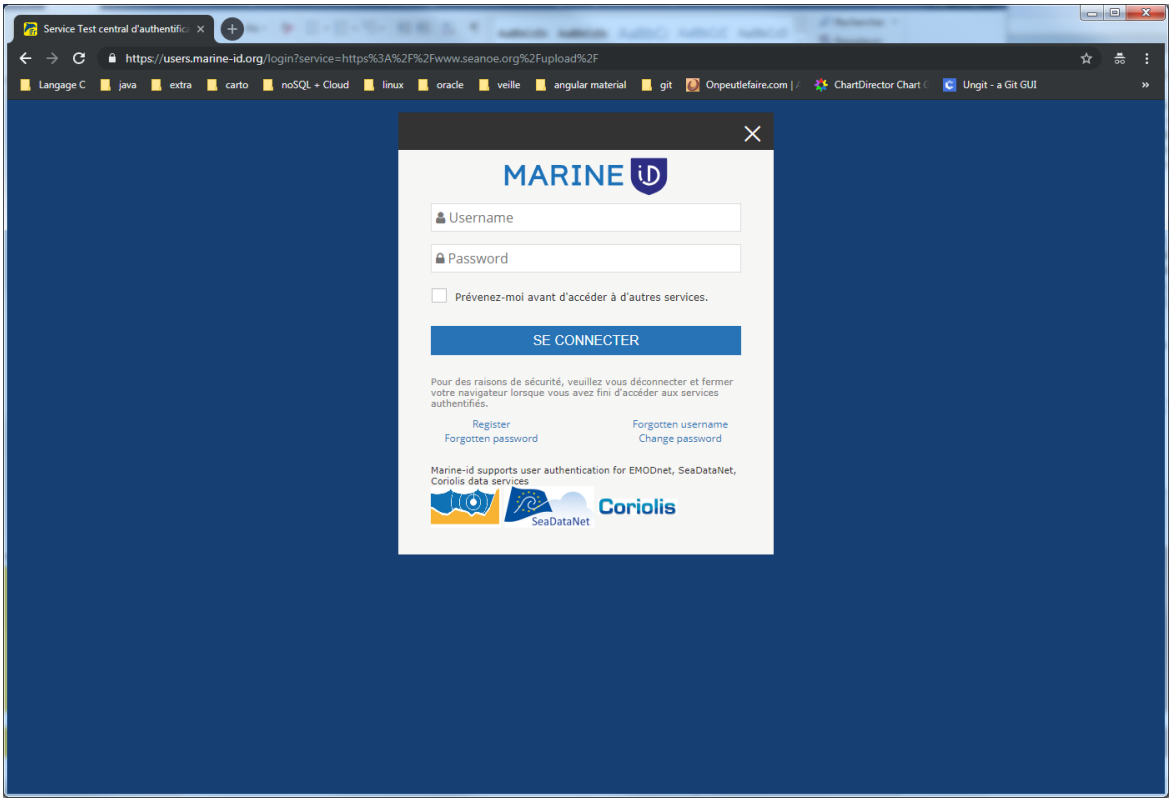
- Signing on



Exemple : Upload data in Seanoe



When you click publish your marine data, you are being redirected to the marine ID authentication page. Except if you already signed on with marine ID in your current session.



From this page, you can sign on or register. At the end of the registration (or sign on if you already have an account), you are being redirected to your coming from URL.

In your application, you can access to the connected user's information's. Below, retrieve the user attributes in Java/JSP

```
AttributePrincipal principal = (AttributePrincipal)request.getUserPrincipal();
Map attributes = principal.getAttributes();

Iterator attributeNames = attributes.keySet().iterator();
for (; attributeNames.hasNext();) {
        String attributeName = (String) attributeNames.next();
      Object attributeValue = attributes.get(attributeName);
}
```

To implement this comportment, you have to modify your application following the next chapter explanations

Modify your J2EE web application

Principles

Note that depending on the context, "filter" or "servlet filter" may refer to the filter element in web.xml or to the underlying Java class that implements filter behavior. Here it refers to the filter definition that is added to web.xml. You will almost certainly need to define these 3 filters:

- CAS Authentication Filter
- CAS Validation Filter
- CAS HttpServletRequestWrapper Filter

There are other filters you may also want to declare (see the CAS Client documentation.) In addition, filter-mapping elements need to be added to web.xml to make sure that incoming requests are processed by the correct underlying filters. See the web.xml fragments below, as well as the CAS Client documentation for how to set up filters for the various CAS protocols.

You can read https://cuit.columbia.edu/cas-authentication/java

The Authentication Filter

You have to add a filter entry in your web.xml

```
<filter>
        <filter-name>CAS Authentication Filter</filter-name>
        <filter-class>org.jasig.cas.client.authentication.AuthenticationFil
ter</filter-class>
        <init-param>
            <param-name>casServerLoginUrl</param-name>
               <param-value>https://users.marine-id.org/login</param-value>
```

```
        </init-param>
        <init-param>
            <param-name>serverName</param-name>
            <param-value>https://your-hostname</param-value>
        </init-param>
</filter>
<filter-mapping>
    <filter-name>CAS Authentication Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

The Validation Filter

You have to add a filter and a filter-mapping entries in your web.xml

```
<filter>
 <filter-name>CAS Validation Filter</filter-name>
    <filter-class>org.jasig.cas.client.validation.Cas20ProxyReceivingTicketValidationFilter</filter-class>
    <init-param>
      <param-name>casServerUrlPrefix</param-name>
      <param-value>https://users.marine-id.org</param-value>
    </init-param>
    <init-param>
      <param-name>serverName</param-name>
      <param-value>https://your-hostname</param-value> <-- hostname of registered URLs
    </init-param>
</filter>
<filter-mapping>
    <filter-name>CAS Validation Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

The HttpServletRequest Wrapper Filter

You have to add a filter entry in your web.xml

```
<filter>
    <filter-name>CAS HttpServletRequest Wrapper Filter</filter-name>
    <filter-class>org.jasig.cas.client.util.HttpServletRequestWrapperFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>CAS HttpServletRequest Wrapper Filter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

Add marine ID log

Just add a new logger in your log configuration file

```
<logger name="org.jasig.cas.client" additivity="false">
      <level value="debug" />
      <appender-ref ref="[YOUR-APPENDER]" />
 </logger>
```

Required libraries

```
      cas-client-core-x.x.x.jar
      commons-logging-x.x.jar
      log4j-x.x.x.jar
      jstl.jar
      opensaml-x.x.jar
      xmlsec-x.x.x.jar
```