

Using VOSI Capabilities in Service Discovery

Patrick Dowler

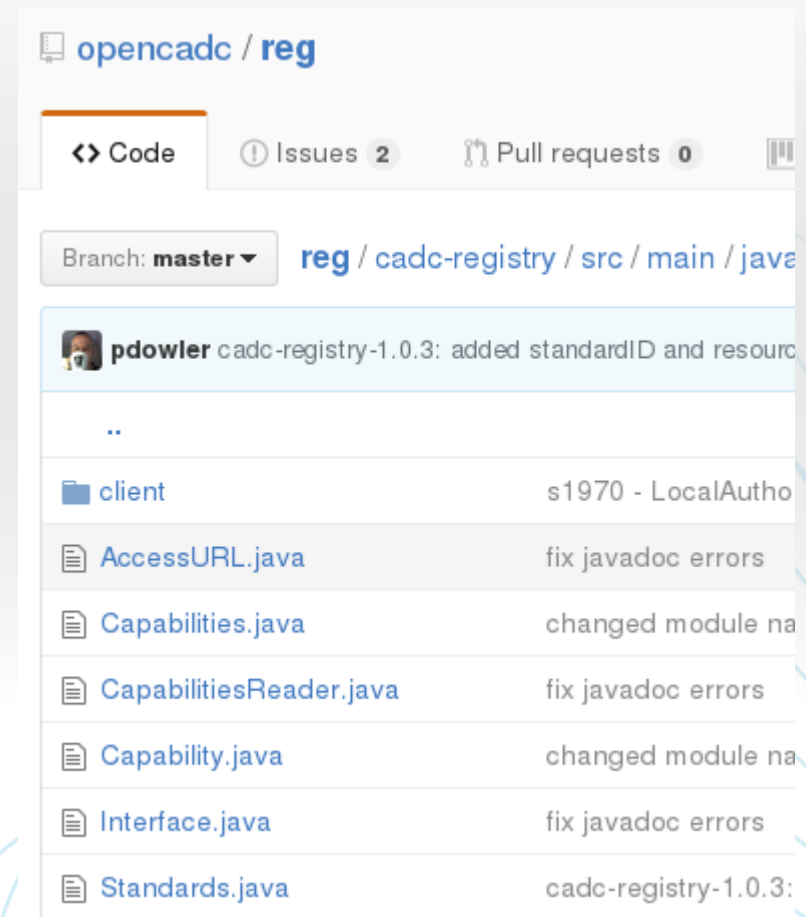
Canadian Astronomy Data Centre

2016-10-21



- capability standardID="<feature>"
 - interface
 - accessURL
 - securityMethod standardID="ivo://ivoa.net/sso#tls-with-certificate"
 - interface
 - accessURL
 - securityMethod standardID="ivo://ivoa.net/sso#BasicAA"
 - interface
 - accessURL
 - securityMethod standardID="ivo://ivoa.net/sso#cookie"
 - interface
 - securityMethod

- simple capabilities model
 - existing model works
- RegistryClient
 - cheat: does not query an IVOA registry service (because: speed)
 - reads /capabilities from service (because: definitive)
 - cache capabilities (because: speed)



The screenshot shows a GitHub repository for `opencadc/reg`. The current branch is `master`. The path is `reg / cadc-registry / src / main / java`. A commit by `pdowler` is shown with the message `cadc-registry-1.0.3: added standardID and resource`. Below the commit, a list of files is displayed:

File Name	Commit Message
<code>..</code>	
<code>client</code>	s1970 - LocalAutho
<code>AccessURL.java</code>	fix javadoc errors
<code>Capabilities.java</code>	changed module na
<code>CapabilitiesReader.java</code>	fix javadoc errors
<code>Capability.java</code>	changed module na
<code>Interface.java</code>	fix javadoc errors
<code>Standards.java</code>	cadc-registry-1.0.3:

Service Discovery - TODO

- have finally registered CANFAR VOSpace
resourceID: ivo://cadc.nrc.ca/vospace
 - with complete VOSI capabilities metadata
 - appears to have propagated to GAVO registry (at least)
- 10-15 other services: how much to add to registry records?
 - add complete capabilities metadata to registry
 - worry: consistency of deployment + VOSI cap vs. registry records
- add minimal capabilities metadata to registry?
 - assume applications will find and call /capabilities?