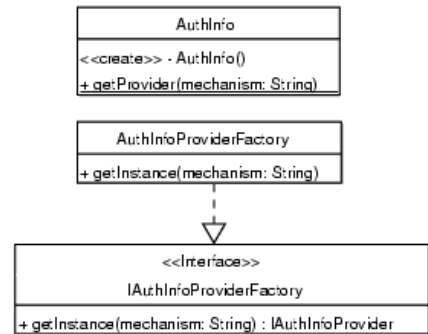


ClientFactory
<pre> + getNames(): Set - getNamesInternal(props: Map): String[] + getInstance(mechanism: String) + createSslClient(mechanisms: String[], authorisationID: String, protocol: String, serverName: String, props: Map, cbh: CallbackHandler) + getMechanismNames(props: Map): String[] - hasPolicy(propertyName: String, props: Map): boolean </pre>

ServerFactory
<pre> + getNames(): Set - getNamesInternal(props: Map): String[] + getInstance(mechanism: String) + createSslServer(mechanism: String, protocol: String, serverName: String, props: Map, cbh: CallbackHandler) + getMechanismNames(props: Map): String[] - hasPolicy(propertyName: String, props: Map): boolean </pre>

ClientMechanism
<pre> <<create>> # ClientMechanism(mechanism: String) # initMechanism(): void # resetMechanism(): void + evaluateChallenge(challenge: byte[]) : byte[] + hasInitialResponse(): boolean + isComplete(): boolean + unwrap(incoming: byte[], offset: int, len: int) : byte[] + wrap(outgoing: byte[], offset: int, len: int) : byte[] + getMechanismName(): String + getNegotiatedProperty(propName: String) : Object + dispose(): void + getAuthorizationID(): String # getNegotiatedQOP(): String # getNegotiatedStrength(): String # getNegotiatedServerAuth(): String # getNegotiatedMaxBuffer(): String # getNegotiatedRawSendSize(): String # getNegotiatedPolicyNoPlainText(): String # getNegotiatedPolicyNoActive(): String # getNegotiatedPolicyNoDictionary(): String # getNegotiatedPolicyNoAnonymous(): String # getNegotiatedPolicyForwardSecrecy(): String # getNegotiatedPolicyPassCredentials(): String # getPause(): String # engineUnwrap(incoming: byte[], offset: int, len: int) : byte[] # engineWrap(outgoing: byte[], offset: int, len: int) : byte[] + init(attributes: Map) : void + reset(): void </pre>



<<Interface>> IAuthInfoProvider
<pre> + activate(context: Map) : void + passivate(): void + contains(userName: String) : boolean + lookup(userID: Map) : Map + update(userCredentials: Map) : void + getConfiguration(mode: String) : Map </pre>

ServerMechanism
<pre> <<create>> # ServerMechanism(mechanism: String) # initMechanism(): void # resetMechanism(): void + evaluateResponse(response: byte[]) : byte[] + isComplete(): boolean + unwrap(incoming: byte[], offset: int, len: int) : byte[] + wrap(outgoing: byte[], offset: int, len: int) : byte[] + getMechanismName(): String + getAuthorizationID(): String + getNegotiatedProperty(propName: String) : Object + dispose(): void # getNegotiatedQOP(): String # getNegotiatedStrength(): String # getNegotiatedServerAuth(): String # getNegotiatedMaxBuffer(): String # getNegotiatedPolicyNoPlainText(): String # getNegotiatedPolicyNoActive(): String # getNegotiatedPolicyNoDictionary(): String # getNegotiatedPolicyNoAnonymous(): String # getNegotiatedPolicyForwardSecrecy(): String # getNegotiatedPolicyPassCredentials(): String # getNegotiatedRawSendSize(): String # getPause(): String # engineUnwrap(incoming: byte[], offset: int, len: int) : byte[] # engineWrap(outgoing: byte[], offset: int, len: int) : byte[] + init(attributes: Map) : void + reset(): void </pre>