Solid server – Proposed architecture v1.0.0 *(status: superseded)*

Ruben Verborgh – May 9, 2019

LDP and Access Control

HTTP

≪interface≫
HttpHandler

+ canHandle(HttpRequest) : Promise<boolean>
+ handle(HttpRequest, HttpResponse) : void

*

HttpServer

+ HttpServer(Array<HttpHandler>)
+ listen(port : int) : void
+ handle(HttpRequest, HttpResponse) : void

Authentication

≪interface≫
CredentialsExtractor

+ extractCredentials(HttpRequest) : Promise<Credentials>

creates

Credentials

LDP

1

LdpHandler

+ LdpHandler(OperationFactory)

1

TargetExtractor

creates

ResourceIdentifier

*

BodyParser

creates

Representation

Patch

1

PreferenceParser

creates

RepresentationPreferences

1

ResponseWriter

writes

HttpResponse

uses

Authorization

≪interface≫
Authorizer

+ hasPermissions(Credentials, ResourceIdentifier, PermissionSet) : Promise<boolean>

AclBasedAuthorizer

1

Operations

1

OperationFactory

+ OperationFactory(ResourceStore)
+ createOperation(method : string, ResourceIdentifier, RepresentationPreferences, . . .) : Operation

1

creates

≪interface≫
Operation

+ target : ResourceIdentifier
+ requestBody : Representation?
+ preferences : RepresentationPreferences
+ requiredPermissions : PermissionSet

+ execute() : Promise<ResponseDescription>

GetOperation

PostOperation

PutOperation

PatchOperation

Storage

1

≪interface≫
ResourceStore

+ getRepresentation(ResourceIdentifier, RepresentationPreferences) : Promise<Representation>
+ addResource(container : ResourceIdentifier, Representation) : Promise<ResourceIdentifier>
+ setRepresentation(ResourceIdentifier, Representation) : Promise<void>
+ deleteResource(ResourceIdentifier) : Promise<void>
+ modifyResource(ResourceIdentifier, Patch) : Promise<void>