API Specification

Identity Holder API

Endpoint	Header	Body/Query	Return Value
Presentation			

GET /v2/present	Authorization: Bearer {token}	`verifier_uri=http://localhost:8083`	`{`
			`"type":`
			`"CredentialType",`
			`requiredAttributes:`
			`["firstName",`
			`"lastName"]`
			`}`

POST /v2/present	Authorization: Bearer {token}	`{`
		`"verifier_uri":`
		`"http://localhost:80`
		`83",`
		`"credential_id":`
		`"ade1ee89-aa2f-4419-`
		`b120-0c8a513a01fc"`
		`}`
			`empty`

Issuance			

GET /v2/issuers	Authorization: Bearer {token}	`empty`	`{`
			`"issuers": [`
			`"did:example:123456"`
			`,"did:example:234567`
			`"`
			`]`
			`}`

GET /v2/issue	Authorization: Bearer {token}	`issuer_id=did:example:123456`	`{`
			`"types":`
			`["CredentialType1",`
			`"CredentialType2"],`
			`"oauth_servers":`
			`["http://issuer.com"`
			`]`
			`}`

POST /v2/issue	Authorization: Bearer {token}	`{`
		`"issuer_id":`
		`"did:example:123456"`
		`,`
		`"auth_code":`
		`"pj2093jht2",`
		`"redirect_uri":`
		`"http://wallet.com/c`
		`allback"`
		`}`
			`{`
			`"credential_id":`
			`"d9d9323a-70eb-4a9e-`
			`96b7-41f8c6fa72f2"`
			`}`

		}	
Managing users and credentials			

GET
/v2/credentials
	Authorization:
	Bearer {token}
		empty	{
			"credential_ids":
			["d04e2e7b-49b1-4c22
			-ab8a-e0ac4fc283f1",
			"7d61a3f9-c862-4a97-
			92fa-fb77ce91dc8e"]
			}

GET /v2/credential	Authorization:
	Bearer {token}
		credential_id=e8b089
		49-8369-48d2-9d6f-2a
		1c2f1d52ec
			{
			"issuer":
			"did:example:123456"
			,
			"type":
			["VerifiableCredenti
			ial",
			"CredentialType1"],
			"cryptosuite":
			"t11a-bookworms-bbs"
			,
			"credential": {
			"id":
			"did:example:34567",
			"firstName": "John",
			"lastName": "Smith"
			}
			}

DELETE
/v2/credential
	Authorization:
	Bearer {token}
		{
		"credential_id":
		"663f1be4-4911-4400-
		9a86-e71ecc4306bb"
		}
			empty

Authorization			

POST
/v2/auth/register
	empty	{
		"email":"email@email
		.com",
		"password":"hunter2"
		}
			{
			"token":"ef54dea5-4b
			6c-4447-b46f-4600edb
			c93a8"
			}

POST
/v2/auth/login
	empty	{
		"email":"email@email
		.com",
		"password":"hunter2"
		}
			{
			"token":
			"a07903b3-77be-482c-
			a3ca-a7cfec2a23ba"
			}

POST /v2/auth/logout	Authorization: Bearer {token}	empty	empty

Issuer API

Endpoint	Header	Body/Query	Return Value
Public endpoints			

GET /	empty	`empty`	`{`
			`did_uri:`
			`"did:example:12345",`
			`}`

DID
(did:example:12345)
.didDocument
.service[0]
.serviceEndpoint
	empty	`empty`	`{`
			`"credential_issuer":`
			`"did:example:12345",`
			`"credential_endpoint`
			`": "localhost:8082",`
			`"credential_configur`
			`ations_supported": {`
			`"DriverLicenceCreden`
			`tial": {`
			`"format": "ldp_vc"`
			`},`
			`"PhotoCardCredential`
			`": {`
			`"format": "ldp_vc"`
			`}`
			`}`
			`"name": "NSWGov"`
			`}`

GET
/v2/authorize
	empty	`response_type=code&c`
		`lient_id=email&redir`
		`ect_uri=http://walle`
		`t.com/callback&state`
		`=xyz&scope=DriverLic`
		`enceCredential`
			`HTTP 302`
			`Location:`
			`http://wallet.com/ca`
			`llback?code=maiowdjd`
			`pa&state=xyz`

POST
/v2/token
	empty	`{`
		`"grant_type":`
		`"authorization_code"`
		`,`
		`"code":`
		`"maiowdjdpa",`
		`"redirect_uri":`
		`"http://wallet.com/c`
		`allback",`
		`"client_id": "email"`
		`}`
			`{`
			`"access_token":`
			`"doiwahiio1239u",`
			`"token_type":`
			`"bearer"`
			`}`

POST	Authorization:	`{`	`{`

| /v2/credential | Bearer {access_token} | "format": "ldp_vc" } | "credential": {credential} } |

Private endpoints			
POST /v2/register (add user)	empty	{ "email": "user@email.com", "password": "hunter2" }	empty
POST /v2/info (add known info to user)	empty	{ "email": "user@email.com", "info": { "firstName": "John", "lastName": "Smith", "anyKey": "anyValue" } }	empty
POST /v2/format	empty	{ type: "DriverLicenceCredential", "attributes": ["firstName", "lastName", "dob"] }	empty
GET /v2/credentials	empty	empty	[{credential}, {credential}, ...] // up to 10 most recent issued credentials
POST /v2/name	empty	{ name: "newName" }	empty

Service Provider API

Endpoint	Header	Body/Query	Return Value
Public endpoints			
GET /v2/request	empty	empty	{ "presentation_defini

			`tion": {presentation_defini tion}, "state": "oeih1129" }`
POST /v2/present	empty	`{ "presentation_submis sion": {presentation_submis son}, "vp_token": {presentation}, "state": "oeih1129" }`	empty
Private endpoints			

GET /v2/presentations	empty	empty	`[{presentation}, {presentation}, ...]`
			
			`// up to 10 most recent presentations`

POST /v2/trust	empty	`{ "id": "did:example:123" }`	empty
POST /v2/untrust	empty	`{ "id": "did:example:123" }`	empty
POST /v2/definition (edit required credential)	empty	`{ type: "DriverLicenceCreden tial", "requiredAttributes" : ["firstName", "lastName", "dob"] }`	empty

Data specifications

Format of presentation (Private endpoint)

```
{
"issuer": "did:example:123456",
"type": "CredentialType1",
"cryptosuite": "t11a-bookworms-bbs",
```

```
"credential": {
"id": "did:example:34567",
"firstName": "John",
"lastName": "Smith"
},
"status": "accepted" // OR "denied"
}
```

Format of credential (Private endpoint)

```
{
"issuer": "did:example:123456",
"type": "CredentialType1",
"cryptosuite": "t11a-bookworms-bbs",
"credential": {
"id": "did:example:34567",
"firstName": "John",
"lastName": "Smith"
}
}
```

Format of credentials

```
{
    // A list of URLs that can be used to clarify the meaning of keys/values in this JSON-LD
    // document. The first must be the URL https://www.w3.org/ns/credentials/v2
    "@context": [
        "https://www.w3.org/ns/credentials/v2"
    ],
    // An array of strings where the first is VerifiableCredential, and optionally more types
    // like "DriversLicenceCredential". Added types must be clarified in a context URL.
    "type": ["VerifiableCredential", "DriverLicenceCredential"],
    // A DID of the issuer
    "issuer": "did:example:12345",
    // The claims about the subject of the credential
    "credentialSubject": {
        // The DID of the subject
        "id": "did:example:23456",
        // Further claims
        "firstName": "John",
        "lastName": "Smith"
    },
    "proof": {
        // The string "DataIntegrityProof"
        "type": "DataIntegrityProof",
        // The string "t11a-bookworms-bbs"
        "cryptosuite": "t11a-bookworms-bbs",
        // A url where the issuer's BBS publickey can be retrieved
        // To ensure that the verifier trusts this, we should use the same DID included as the issuer
        "verificationMethod": "did:example:12345/key-1",
        // The string "assertionMethod"
        "proofPurpose": "assertionMethod",
        // A BBS signature produced by the issuer to verify the claims in the credential
        "proofValue": "{bbs-signature}"
    }
}
```

An example credential object

Format of presentations

```
{
    // A list of URLs that can be used to clarify the meaning of keys/values in this JSON-LD
    // document. The first must be the URL https://www.w3.org/ns/credentials/v2
    "@context": [
        "https://www.w3.org/ns/credentials/v2"
    ],
    // An array of strings where the first is VerifiablePresentation, and optionally more types.
    // Added types must be clarified in a context URL.
    "type": ["VerifiablePresentation"],
    // A list of credentials to be presented
    "verifiableCredential": [
        {
            "@context": [
                "https://www.w3.org/ns/credentials/v2"
            ],
            "type": ["VerifiableCredential", "DriverLicenceCredential"],
            "issuer": "did:example:12345",
            "credentialSubject": {
                "firstName": "John",
                "lastName": "Smith"
            },
            "proof": {
                "type": "DataIntegrityProof",
                "cryptosuite": "t11a-bookworms-bbs",
                "verificationMethod": "did:example:12345/key-1",
                "proofPurpose": "assertionMethod",
                // Selectively disclose certain aspects of the subject by using a BBS proof
                // which can be verified using the original BBS signature without revealing
                // untransmitted claims
                "proofValue": ["{bbs-disclosed-indexes}", "{bbs-proof}"]
            }
        }
    ]
{
```

An example presentation object

Format of presentation definitions

```
{
    // ID required
    "id": "32f54163-7166-48f1-93d8-f217bdb0653",
    "input_descriptors": [
        {
            // This should be a credential type
            "id": "DriverLicenceCredential",
            // This allows us to specify which fields we need (for selective disclosure)
            "constraints": {
                "fields": [
                    {
                        // A list of places (specified from the root of the credentialSubject object)
                        // we expect to find this desired field
                        "path": ["$.firstName"]
                    },
                    {
                        // A list of places (specified from the root of the credentialSubject object)
                        // we expect to find this desired field
                        "path": ["$.lastName"]
                    }
                ]
            }
        }
    ]
}
```

An example presentation definition object

Format of presentation submissions

```
{
    // ID required
    "id": "b9407204-49b7-4541-aa57-1c8efbb8775c",
    // Matching the ID of presentation request object
    "definition_id": "32f54163-7166-48f1-93d8-f217bdb0653"
    // For each input descriptor we need to "map" to the relevant presentation
    "descriptor_map": [
        {
            // Matching the ID of an input descriptor object
            "id": "DriverLicenceCredential",
            "format": "ldp_vc",
            // Path from root of presentation object where we can find the relevant presentation
            "path": "$.verfiableCredential[0]"
        }
    ]
}
```

An example presentation submission object